LilyPond

Le système de gravure musicale

Morceaux choisis

L'équipe de développement de LilyPond

Ce document regroupe un certain nombre d'exemples, trucs et astuces pour LilyPond issus du LilyPond Snippet Repository (`http://lsr.di.unimi.it`) (LSR). Il est dans le domaine public.

Nous profitons de ces lignes pour adresser tous nos remerciements à Sebastiano Vigna, qui assure la maintenance du site LSR et de la base de données qu'il contient, ainsi qu'à l'université de Milan qui l'héberge.

Notez bien que cet ouvrage n'est pas une simple copie du LSR. D'une part, certains exemples proviennent directement des sources de LilyPond – plus précisément le répertoire `input/new` – et, d'autre part, ce qui provient du LSR est mis à jour à l'aide de `convert-ly` – le LSR repose sur une version stable de LilyPond, alors que ce manuel correspond à la version 2.23.2.

Les extraits sont regroupés par catégorie, correspondant au découpage du manuel de notation de LilyPond. Un même extrait peut se voir attribuer plusieurs catégories, et toutes les catégories du LSR ne sont pas forcément reprises.

Si vous consultez la version HTML de cet ouvrage, un clic sur le nom du fichier ou sur l'image vous affichera le code sous-jacent.

Pour connaître la place qu'occupe ce manuel dans la documentation, consultez la page Section "Manuels" dans *Informations générales*.

Si vous ne disposez pas de certains manuels, la documentation complète se trouve sur `https://lilypond.org/`.

Ce document a été placé dans le domaine public ; en France, les auteurs renoncent à tous leurs droits patrimoniaux.

Pour LilyPond version 2.23.2

Note : La génération automatique de ce document ne permet malheureusement pas l'affichage des rubriques dans la langue de Molière ; veuillez nous en excuser.

Table des matières

Pitches **2**
Un ambitus par voix 2
Ajout d'une indication d'octave pour une seule voix 2
Variante fine des notes profilées Aiken 3
Modification de la hauteur de hampes ligaturées 3
Ambitus après l'armure 4
Ambitus sur plusieurs voix 4
Ambitus 5
Profilage des notes selon leur degré dans la gamme 5
Changement automatique de l'orientation de hampe de la note médiane selon la mélodie 6
Modification du texte des marques d'octaviation 7
Réglage de l'affichage d'un ambitus 7
Modification de l'intervalle des lignes de la portée 8
Transposition arbitraire d'une clef 9
Coloration des notes selon leur hauteur 9
Création d'une séquence de notes de même hauteur 10
Création d'armures personnalisées 10
Impression forcée d'un bécarre avant une accidentelle 11
Impression forcée de la clef 11
Génération de notes aléatoires 12
Non répétition de l'altération après saut de ligne sur liaison de prolongation 12
Conservation de la taille lors d'un changement de clef 13
Exemple de musique « Makam » 13
Modification de la pente de l'extension d'octaviation 14
Armures inhabituelles 14
Easy play – chiffres en lieu et place des lettres 15
Orchestre, chœur et piano 16
Suppression des bécarres superflus 19
Suppression des bécarres superflus lors d'un changement de tonalité 20
Citation d'une autre voix et transposition 20
Séparation entre altérations annulées et nouvelle armure 21
Transposition et réduction du nombre d'altérations accidentelles 22
Exemple de makam turc 23
Affinage des propriétés d'une clef 24
Utilisation d'`autoChange` avec plus d'une voix 25

Rhythms **27**
Coexistence de ligatures et liaisons avec des rythmes comprenant des n-olets 27
Ajout de parties de batterie 27
Ajustement de l'espacement des notes d'ornement 28
Alignement des numéros de mesure 28
Brève alternative, avec deux barres verticales 29
Appoggiature avant une barre de mesure 29
Subdivision de ligatures automatiques 30
Changement automatique des durées 30
Définition de règles de ligature pour la partition 31
Ligature au moment d'un saut de ligne 32

Ligature et directions de hampe inversées ... 32
Modifier l'apparence d'un silence multimesure ... 33
Spécification du nombre de points d'augmentation d'une note ... 33
Changement de tempo sans indication sur la partition ... 33
Modifier l'apparence du chiffre de n-olet ... 34
Modification de la métrique au sein d'un fragment polymétrique à l'aide de `\scaleDurations` ... 34
Notation pour psalmodie ... 35
Métrique décomposée ... 35
Signes de direction, signes de sous-groupe ... 36
Adaptation de l'alignement des numéros de mesure ... 37
Contrôle de l'impression des crochets de n-olet ... 38
Création d'une indication métronomique sous forme d'étiquette ... 39
Dessin à main levée de liaisons de tenue ... 39
Impression de trémolo avec ligature flottante ... 39
Plusieurs triolets avec une seule commande `\tuplet` ... 40
Crochet rectiligne et débordement de ligature ... 41
Fixation arbitraire du numéro de repère de départ ... 42
Génération de crochets personnalisés ... 42
Rythmique et guitare ... 43
Indication personnalisée d'une polymétrie complexe ... 44
Recours à la propriété `'transparent` pour rendre des objets invisibles ... 45
Dessin d'une liaison d'articulation au trait discontinu ... 46
Contrôle manuel du positionnement des ligatures ... 47
Fusion de silences multimesures dans une partie polyphonique ... 47
Modification de la longueur d'un crochet de n-olet ... 48
Déplacement des notes pointées dans une polyphonie ... 48
Contrôle de la taille d'un silence multimesure ... 48
Ajout de texte à un silence multimesure ... 49
N-olets au chiffrage inhabituel ... 49
Dénombrer une unique mesure de silence ... 50
PartCombine et autoBeamOff ... 51
Exemple pour percussions ... 51
Saut de ligne au milieu d'un n-olet avec ligature ... 53
Positionnement des ligatures de notes d'ornement à la hauteur de celles des notes normales ... 53
Positionnement des notes d'ornement avec espace flottant ... 54
Positionnement des silences multimesures ... 54
Préservation de l'indication de n-olet lors d'un repère final ... 55
Imprimer les numéros de mesure à intervalle régulier ... 56
Impression du numéro des mesures tronquées ... 57
Inscrire le numéro de mesure dans un cadre ou un cercle ... 57
Impression du numéro de mesure selon `modulo-bar-number-visible` ... 58
Changement de la fréquence d'impression du numéro de mesure ... 58
Impression du métronome et des repères sous la portée ... 59
Impression de musique aux métriques différentes ... 59
Afficher le numéro de la première mesure ... 63
Redéfinition des réglages de mise en forme par défaut des notes d'ornement ... 63
Suppression des numéros de mesure d'une partition ... 64
Suppression de la partie interportée des barres de mesure d'un regroupement autre que `ChoirStaff` ... 64
Styles de silences ... 65
Annulation des règles de ligature par défaut ... 66
Barres rythmiques ... 67

Sauts de notes en mode paroles (2) 67
Sauts de notes en mode paroles 68
Moignons de hampe 68
Ligature à la pulsation 68
Subdivision des ligatures 69
Exemple pour tam-tam 70
Encadrement sur trois côtés 70
Métrique entre parenthèses – méthode 3 71
Métrique entre parenthèses 71
Affichage seulement du numérateur d'une métrique (au lieu d'une fraction) 71
Mise en forme des notes d'ornement 72
Personnalisation de la métrique 72
Crochets de style alternatif 73
Utilisation de hampe barrée pour une note normale 74
Liaison de tenue et arpège 74

Expressive marks 75

Coexistence de ligatures et liaisons avec des rythmes comprenant des n-olets 75
Mise entre parenthèses d'un signe d'interprétation ou d'une note d'un accord 75
Ajout de marques temporelles à un long glissando 75
Ajustement du galbe des chutes ou sauts 76
Alignement des bornes de soufflet relativement aux `NoteColumns` 77
Brève alternative, avec deux barres verticales 77
Liaison asymétrique 77
Signes de respiration 78
Soufflet de crescendo partiellement interrompu 78
Césure en forme de « voie de chemin de fer » avec point d'orgue 79
Texte centré sous un soufflet 80
Modifier la taille d'un \flageolet 81
Modification du texte et de l'extension de nuances textuelles 82
Modification de l'aspect des liaisons d'articulation 82
Modification de l'indicateur de respiration 82
Spécification du nombre de points d'augmentation d'une note 83
Combinaison de nuance et de texte 83
Glissando contemporain 83
Contrôle de la visibilité des extensions d'objet après saut de ligne 84
Contrôle de l'ordre vertical des articulations et ornements 84
Création d'un groupetto retardé 85
Arpège distribué sur plusieurs voix 85
Arpège distribué sur une partition pour piano 86
Arpège distribué pour un autre contexte que le piano 86
Création de doigtés sur deux caractères 87
Indications de nuance vraiment entre parenthèses 87
Création de repères simultanés 88
Création d'une liaison entre plusieurs voix 89
Création d'extensions de texte 89
Personnalisation des extenseurs de nuance postfix 90
Extensions de nuance postfix 91
Glissando par dessus un objet graphique 91
Stylisation des lignes de soufflet 92
Masquage de l'extension des nuances textuelles 92
Alignement de nuances personnalisées comme « sempre pp », « più f » ou « subito p » 92
Impression d'un double repère sur une même barre (méthode 1) 96

Impression d'un double repère sur une même barre (méthode 2) 97
Insertion d'une césure 98
Liaison « Laissez vibrer » 98
Terminaison de ligne en flèche 99
Dessin d'une liaison d'articulation au trait discontinu 99
Modification de la signification des raccourcis pour les signes d'articulation 100
Déplacement vertical des liaisons d'articulation 100
Déplacer les extrémités de soufflets 101
Positionnement des arpeggios 101
Positionnement d'une annotation à l'intérieur d'une liaison 102
Différents styles de soufflet 102
Impression de soufflets « al niente » 103
Impression du métronome et des repères sous la portée 103
Soufflets et barres de mesure 103
Ajustement de la longueur d'un soufflet 104
Impression d'une même articulation des deux côtés d'une note ou d'un accord 104
Slap ou pizzicato Bartok 105
Remplacement du signe de respiration par une coche 105
Utilisation d'un `arpeggioBracket` pour rendre les divisions plus évidentes 106
Accords et double liaison d'articulation 106
Utilisation de la propriété `whiteout` 107
Articulation baroque en forme de coche 107
Alignement vertical des nuances sur plusieurs notes 107

Repeats 109

Ajout du crochet de reprise à d'autres portées 109
Numéros de mesure centrés 109
Modification des barres de mesure par défaut 110
Trémolo et changement de portée 111
Impression de trémolo avec ligature flottante 111
Répétition en pourcent isolée 112
Compteur de mesures 112
Numérotation de groupes de mesures 113
Affichage du numéro de répétition en pourcent 114
Compteur de répétition en pourcent 114
Positionnement des segno et coda (avec saut de ligne) 115
Succession de reprises et style de barre par défaut 117
Diminution de la taille du crochet d'alternative 117
Crochet de reprise sous les chiffrages d'accord 118
Volta multiportée 118
Indication de reprise avec texte grâce à `repeatCommands` 119

Simultaneous notes 121

Ajout de voix pour éviter les collisions 121
Modification de la taille d'une note particulière d'un accord 121
Modification des indications de parties combinées 122
Clusters 122
Combinaison de deux parties sur une même portée 123
Impression d'accords complexes 124
Décalage horizontal forcé 124
Recours à la propriété `'transparent` pour rendre des objets invisibles 125
Déplacement des notes pointées dans une polyphonie 125
Suppression des avertissements de chevauchement 126
Deux `\partCombine` sur une même portée 126

Staff notation **129**
Un ambitus par voix 129
Ajout d'une portée supplémentaire après un saut de ligne 129
Ajout d'une portée supplémentaire 130
Ajout d'indicateurs à une portée dédoublée après un saut de ligne 131
Indication de l'instrument cité dans l'accompagnement d'une partition pour chœur 135
Ajout de marques temporelles à un long glissando 137
Numérotation des mesures et alternatives 137
Ambitus après l'armure 138
Numéros de mesure centrés 139
Modification des barres de mesure par défaut 140
Modification du nombre de lignes de la portée 140
Modification de la taille d'une portée 141
Impression de papier à musique 142
Création d'armures personnalisées 144
Création de doigtés sur deux caractères 144
Hampes interportées 144
Indicateur de regroupement et portée unique 145
Allongement d'une marque de trille (`TrillSpanner`) 146
Rappel du glissando à l'occasion d'une alternative 146
Liaison de prolongation aplatie 148
Adaptation de la largeur de mesure selon le `MetronomeMark` 150
Glissando par dessus un objet graphique 151
Impression d'un double repère sur une même barre (méthode 1) 151
Impression d'un double repère sur une même barre (méthode 2) 152
Incipit 153
Insertion d'un fragment dans un *markup* au-dessus de la portée 157
Impression de la corde aiguë d'un `TabStaff` en bas 158
Tablature en lettres 159
Saut de ligne et glissando 159
Empâtement de certaines lignes d'une portée 160
Compteur de mesures 160
Présentation à l'ancienne (barres de mesure entre les portées) 161
Modification de la pente de l'extension d'octaviation 161
Imbrications de regroupements de portées 162
Armures inhabituelles 163
Numérotation de groupes de mesures 163
Orchestre, chœur et piano 164
Impression des paroles dans la portée 168
Citation d'une autre voix et transposition 169
Citation d'une autre voix 169
Retrait de l'accolade à la première ligne d'une pièce pour piano 170
Masquage de la première ligne si elle est vide 171
Séparation visuelle entre les systèmes 173
Barre de mesure en encoche 175
Métrique entre parenthèses – méthode 3 175
Métrique entre parenthèses 175
Affinage des propriétés d'une clef 175
Deux `\partCombine` sur une même portée 177
Indication de regroupement de portées par un rectangle 179
Utilisation d'`autoChange` avec plus d'une voix 179
Utilisation de lignes de repères dans une partition « à la française » 180
Alignement vertical de `StaffGroup`s sans `SystemStartBar` 183
Crochet de reprise sous les chiffrages d'accord 191

Volta multiportée 192

Editorial annotations 193

Ajout de doigtés à une partition 193
Ajout de liens à des objets 193
Ajout de *markups* à une tablature 195
Impression des doigtés à l'intérieur de la portée 195
Numérotation des mesures et alternatives 196
Crochets d'analyse au-dessus de la portée 197
Crochet d'analyse avec texte 197
Profilage des notes selon leur degré dans la gamme 198
Blanchiment de lignes de portée avec la commande `\whiteout` 199
Modification de la taille d'une note particulière d'un accord 199
Modification de l'aspect des liaisons d'articulation 200
Coloration des notes selon leur hauteur 200
Contrôler la position des doigtés dans un accord 201
Création d'un groupetto retardé 202
Impression de papier à musique 202
Création de doigtés sur deux caractères 204
Direction par défaut des hampes de la ligne médiane 204
Taille de police différente pour `instrumentName` et `shortInstrumentName` 205
Encadrement d'objets 206
Encerclement de notes 207
Encercler divers objets 207
Intégration de PostScript dans un *markup* 207
Apparence du quadrillage temporel 208
Quadrillage temporel : mise en évidence du rythme et synchronisation des notes 209
Hammer on et pull off avec accords 210
Hammer on et pull off gérés par les voix 210
Hammer on et pull off 211
Impression d'un double repère sur une même barre (méthode 1) 211
Impression d'un double repère sur une même barre (méthode 2) 212
Empâtement de certaines lignes d'une portée 213
Marking notes of spoken parts with a cross on the stem (Sprechstimme) 213
Compteur de mesures 214
Numérotation de groupes de mesures 214
Positionnement précis des indications de doigté 215
Positionnement d'une annotation à l'intérieur d'une liaison 216
Impression de texte de droite à gauche 216
Ligne de prolongation pour numéro de corde 217
Utilisation de PostScript pour générer des têtes de note à l'allure particulière 217
Utilisation de la propriété `whiteout` 218

Text 219

Ajout de *markups* à une tablature 219
Ajout de la date du jour à une partition 219
Ajustement de l'espacement vertical des paroles 220
Alignement des noms d'instrument 221
Alignement des objets créés par la commande `\mark` 222
Alignement de la première syllabe d'un mélisme 223
Blanchiment de lignes de portée avec la commande `\whiteout` 223
Texte centré sous un soufflet 224
Modification du texte des marques d'octaviation 225

Changement des fontes textuelles par défaut ... 226
Combinaison de nuance et de texte ... 227
Combinaison de deux parties sur une même portée ... 227
Indications de nuance vraiment entre parenthèses ... 228
Création de repères simultanés ... 229
Création d'extensions de texte ... 230
Champs d'entête ... 230
Intégration de PostScript dans un *markup* ... 232
Mise en forme individuelle de syllabes ... 232
Séparation de syllables par une liaison ... 232
Alignement des syllabes ... 233
Liste de *markups* ... 233
Ajout de texte à un silence multimesure ... 235
De l'ubiquité des objets *markup* ... 236
Impression du numéro de version ... 237
Piano et paroles entre les portées ... 237
Changement de la fréquence d'impression du numéro de mesure ... 238
Indication d'un repère en fin de ligne ... 238
Impression des indications sur toutes les portées d'un système ... 239
Impression de texte de droite à gauche ... 239
Impression des paroles dans la portée ... 240
Bloc de texte indépendant sur deux colonnes ... 240
Ligne de prolongation pour numéro de corde ... 241
Encadrement sur trois côtés ... 242
UTF-8 ... 242
Ensemble vocal avec alignement des paroles selon le contexte ... 244
Indication de reprise avec texte grâce à `repeatCommands` ... 245

Vocal music ... 247

Un ambitus par voix ... 247
Ajout d'indicateurs à une portée dédoublée après un saut de ligne ... 247
Indication de l'instrument cité dans l'accompagnement d'une partition pour chœur ... 251
Ajustement de l'espacement vertical des paroles ... 253
Alignement de la première syllabe d'un mélisme ... 253
Ambitus après l'armure ... 254
Ambitus sur plusieurs voix ... 254
Ambitus ... 255
Exemples de notation ancienne – transcription moderne de musique grégorienne ... 256
Psalmodie anglicane ... 256
Agencement de paroles séparées sur une seule ligne ... 259
Changement de fontes des couplets ... 260
Notation pour psalmodie ... 261
Impression forcée de tirets entre syllabes ... 262
Mise en forme individuelle de syllabes ... 262
Séparation de syllables par une liaison ... 263
Modèle pour cantique ... 263
Alignement des syllabes ... 265
Marking notes of spoken parts with a cross on the stem (Sprechstimme) ... 265
Espacement des paroles selon les pratiques de la version 2.12 ... 266
Orchestre, chœur et piano ... 268
Piano, mélodie et paroles ... 272
Impression des paroles dans la portée ... 273
Modèle pour chœur SATB, sur quatre portées ... 274
Paroles, musique et accords ... 275

Paroles, musique, accords et diagrammes de fret 276
Portée unique et paroles 277
Sauts de notes en mode paroles (2) 278
Sauts de notes en mode paroles 278
Utilisation d'un `arpeggioBracket` pour rendre les divisions plus évidentes 278
Recours aux balises pour produire une partition ancienne et moderne à partir de la même source 279
Positionnement d'une ossia et des paroles 281
Centrage vertical de paroles communes 282
Ensemble vocal avec réduction pour piano 283
Ensemble vocal avec alignement des paroles selon le contexte 285
Ensemble vocal avec couplet et refrain 286
Ensemble vocal (simple) 288

Chords 291

Ajout d'une basse chiffrée au-dessus ou au-dessous des notes 291
Chiffrages et barres de mesure 291
Notation d'accords barrés pour guitare (avec extension textuelle) 292
Personnalisation du séparateur d'accords 293
Adoption de la dénomination germanique ou semi-germanique des accords 293
Emplacement des altération en basse continue 294
Modèles de chiffrage d'accords 295
Chiffrage d'un maj7 295
Noms d'accord alternatifs 295
Accords avec doigté espacé pour `FretBoards` et `TabVoice` 305
Clusters 306
Contrôler la position des doigtés dans un accord 306
Accord distribué et problème de hampe - solution 307
Impression d'accords complexes 308
Interruption manuelle des prolongations de certains chiffrages 308
Impression des accords si changement 309
Chanson simple 309
Paroles, musique et accords 310
Paroles, musique, accords et diagrammes de fret 310
Mélodie simple et accords 311
Prolongateur commun de basse figurée 312
Crochet de reprise sous les chiffrages d'accord 313

Keyboards 314

Symboles de chant pour accordéon 314
Symboles de registre pour accordéon 317
Modification du texte des indications de pédale 318
Clusters 318
Contrôler la position des doigtés dans un accord 319
Création d'une liaison entre plusieurs voix 319
Accord distribué et problème de hampe - solution 320
Trémolo et changement de portée 321
Affinage des indications de pédale 322
Indication d'un accord à cheval sur deux portées par un crochet d'arpège 322
Symboles de jazz 323
Liaison « Laissez vibrer » 329
Piano – cannevas simple 330
Piano et paroles entre les portées 330

Piano, mélodie et paroles 331
Retrait de l'accolade à la première ligne d'une pièce pour piano 332
Utilisation d'`autoChange` avec plus d'une voix 333
Ensemble vocal avec réduction pour piano 334

Percussion 337

Ajout de parties de batterie 337
Personnalisation de batterie pour les sorties imprimable et MIDI 338
Indication personnalisée d'une polymétrie complexe 340
Symboles de jazz 341
Baguettes pour percussion 347
Exemple pour percussions 350
Impression de musique aux métriques différentes 351
Exemple pour tam-tam 355

Fretted strings 356

Ajout de doigtés à une partition 356
Ajout de doigtés à des tablatures 356
Ajout de *markups* à une tablature 357
Impression des doigtés à l'intérieur de la portée 357
Barrés et diagrammes de fret automatiques 358
Notation d'accords barrés pour guitare (avec extension textuelle) 358
Orientation des diagrammes de fret 359
Glissando d'accords et tablature 360
Changement d'accord et diagramme de fret 361
Accords avec doigté espacé pour `FretBoards` et `TabVoice` 362
Contrôler la position des doigtés dans un accord 362
Personnalisation de diagrammes de fret 363
Personnalisation des diagrammes de fret 364
Création de diagrammes de fret prédéfinis pour d'autres instruments 366
Simulation d'un *hammer* en tablature 368
Doigtés, indications de corde, et doigtés main droite 368
Notation de flamenco 369
Construction et développement de diagrammes de fret 373
Diagrammes de fret alternatifs 379
Harmoniques et tablature 381
Indication d'un glissé de guitare 382
Rythmique et guitare 383
Hammer on et pull off avec accords 384
Hammer on et pull off gérés par les voix 384
Hammer on et pull off 385
Repositionnement d'un diagramme de fret 385
Symboles de jazz 386
Liaison « Laissez vibrer » 392
Impression de la corde aiguë d'un `TabStaff` en bas 393
Tablature en lettres 393
Harmoniques sur corde à vide en tablature 394
Positionnement des doigtés main droite 396
Polyphonie en mode tablature 397
Glissando et tablature 398
Hampes et ligatures en mode tablature 398
Ligne de prolongation pour numéro de corde 399

Unfretted strings ... 400
Création d'une liaison entre plusieurs voix ... 400
Harmoniques pointées ... 400
Slap ou pizzicato Bartok ... 401
Quatuor à cordes (conducteur) ... 401
Quatuor à cordes, avec parties séparées ... 402

Winds ... 406
Modification de la taille d'un diagramme pour bois ... 406
Symboles de doigtés pour instruments à vent ... 406
Coup de langue à la flûte ... 407
Ajout de texte à un diagramme de doigté ... 408
Doigtés pour flûte à bec ... 408
Liste des différents diagrammes de doigtés pour bois ... 409
Liste des diagrammes de doigtés pour bois ... 410

Ancient notation ... 413
Ajout d'une basse chiffrée au-dessus ou au-dessous des notes ... 413
Gravure de musique ancienne ... 413
Exemples de notation ancienne – transcription moderne de musique grégorienne ... 418
Métrique ancienne ... 419
Notation pour psalmodie ... 419
Guidons ... 420
Incipit ... 421
Présentation à l'ancienne (barres de mesure entre les portées) ... 425
Styles de silences ... 426
Recours aux balises pour produire une partition ancienne et moderne à partir de la même source ... 427
Articulation baroque en forme de coche ... 428

World music ... 430
Improvisation en musique arabe ... 430
Exemple de musique « Makam » ... 430
Armures inhabituelles ... 430
Impression de texte de droite à gauche ... 431
Exemple de makam turc ... 431

Contexts and engravers ... 433
Ajout d'une basse chiffrée au-dessus ou au-dessous des notes ... 433
Ajout d'une portée supplémentaire après un saut de ligne ... 433
Ajout d'une portée supplémentaire ... 434
Changement automatique de l'orientation de hampe de la note médiane selon la mélodie ... 435
Numéros de mesure centrés ... 436
Affectation d'un canal MIDI par voix ... 436
Modification de la métrique au sein d'un fragment polymétrique à l'aide de `\scaleDurations` ... 437
Notation pour psalmodie ... 438
Impression de papier à musique ... 439
Création d'armures personnalisées ... 441
Hampes interportées ... 441
Définition d'un graveur en Scheme : graveur d'ambitus ... 442

Affichage d'un système complet lorsqu'une seule portée est active 448
Les graveurs un par un 450
Présentation à l'ancienne (barres de mesure entre les portées) 456
Imbrications de regroupements de portées 456
Numérotation de groupes de mesures 457
Suppression des numéros de mesure d'une partition 458
Indication de regroupement de portées par un rectangle 458
Utilisation de lignes de repères dans une partition « à la française » 459
Ensemble vocal avec alignement des paroles selon le contexte 462
Ensemble vocal avec couplet et refrain 463

Tweaks and overrides 466

Ajout d'une indication d'octave pour une seule voix 466
Ajout de liens à des objets 466
Ajout de *markups* à une tablature 468
Ajout de marques temporelles à un long glissando 469
Ajustement de l'espacement des notes d'ornement 470
Ajustement de l'espacement vertical des paroles 470
Modification de la hauteur de hampes ligaturées 471
Numérotation des mesures et alternatives 471
Crochets d'analyse au-dessus de la portée 472
Crochet d'analyse avec texte 473
Liaison asymétrique 474
Césure en forme de « voie de chemin de fer » avec point d'orgue 474
Modification de la taille d'une note particulière d'un accord 474
Modification de l'épaisseur et de l'écartement des ligatures 475
Modifier l'apparence d'un silence multimesure 475
Modification des propriétés d'objets particuliers 476
Modification du texte et de l'extension de nuances textuelles 476
Changement des fontes textuelles par défaut 476
Modification de la taille d'une portée 478
Changement de tempo sans indication sur la partition 478
Modification du texte des indications de pédale 479
Contrôle de la visibilité des extensions d'objet après saut de ligne 479
Contrôle de l'ordre vertical des articulations et ornements 480
Contrôle de l'impression des crochets de n-olet 480
Création d'un groupetto retardé 481
Création d'armures personnalisées 482
Création de doigtés sur deux caractères 482
Création de repères simultanés 482
Création d'extensions de texte 483
Accord distribué et problème de hampe - solution 484
Hampes interportées 485
Guidons 486
Personnalisation de diagrammes de fret 487
Personnalisation des diagrammes de fret 488
Indicateur de regroupement et portée unique 489
Affichage de la généalogie d'un objet 490
Harmoniques pointées 492
Encadrement d'objets 492
Encercler divers objets 493
Personnalisation des extenseurs de nuance postfix 493
Extensions de nuance postfix 494
Allongement d'une marque de trille (`TrillSpanner`) 494

Rappel du glissando à l'occasion d'une alternative 495
Affinage des indications de pédale 496
Liaison de prolongation aplatie 497
Impression forcée d'un bécarre avant une accidentelle 499
Décalage horizontal forcé 499
Construction et développement de diagrammes de fret 500
Génération de crochets personnalisés 506
Glissando par dessus un objet graphique 508
Stylisation des lignes de soufflet 508
Alignement de nuances personnalisées comme « sempre pp », « più f » ou « subito p » 508
Repositionnement d'un diagramme de fret 512
Impression d'un double repère sur une même barre (méthode 1) 513
Impression d'un double repère sur une même barre (méthode 2) 514
Insertion d'une césure 515
Conservation de la taille lors d'un changement de clef 515
Terminaison de ligne en flèche 516
Recours à la propriété `'transparent` pour rendre des objets invisibles 516
Saut de ligne et glissando 517
Contrôle manuel du positionnement des ligatures 518
Présentation à l'ancienne (barres de mesure entre les portées) 518
Modification de la pente de l'extension d'octaviation 518
Déplacement des notes pointées dans une polyphonie 519
Déplacement vertical des liaisons d'articulation 519
Imbrications de regroupements de portées 520
Personnalisation de certains types d'articulation 521
Affichage du numéro de répétition en pourcent 523
Positionnement des arpeggios 523
Positionnement des silences multimesures 523
Positionnement d'une annotation à l'intérieur d'une liaison 524
Inscrire le numéro de mesure dans un cadre ou un cercle 525
Impression du métronome et des repères sous la portée 525
Impression du nom des notes avec ou sans indication d'octave 526
Espacement strictement proportionnel des notes 526
Retrait de l'accolade à la première ligne d'une pièce pour piano 527
Suppression de la partie interportée des barres de mesure d'un regroupement autre que `ChoirStaff` 528
Masquage de la première ligne si elle est vide 528
Styles de silences 530
Barres rythmiques 531
Séparation entre altérations annulées et nouvelle armure 532
Soufflets et barres de mesure 532
Séparation visuelle entre les systèmes 533
Impression d'une même articulation des deux côtés d'une note ou d'un accord 535
Ligne de prolongation pour numéro de corde 535
Suppression des avertissements de chevauchement 536
Métrique entre parenthèses – méthode 3 536
Métrique entre parenthèses 537
Affichage seulement du numérateur d'une métrique (au lieu d'une fraction) 537
Crochet de n-olet et changement de portée 537
Affinage des propriétés d'une clef 538
Mise en forme des notes d'ornement 540
Crochets de style alternatif 540
Utilisation de `ly:grob-object` pour accéder aux *grobs* avec `\tweak` 541

Utilisation de PostScript pour générer des têtes de note à l'allure particulière 542
Utilisation de `\tweak` pour retoucher des objets particuliers 543
Alignement vertical des nuances indications textuelles 543
Positionnement d'une ossia et des paroles 544
Prolongateur commun de basse figurée 545

Paper and layout 546
Alignement des noms d'instrument 546
Agencement de paroles séparées sur une seule ligne 547
Partitionnement d'un ouvrage 548
Modification de la taille d'une portée 552
Découpe de systèmes 553
Impression de papier à musique 556
Champs d'entête 557
Séparation visuelle entre les systèmes 558
Table des matières 561
Alignement vertical de `StaffGroup`s sans `SystemStartBar` 562

Titles 571
Ajout de la date du jour à une partition 571
Alignement des noms d'instrument 571
Champs d'entête 573
Impression du numéro de version 574

Spacing 575
Ajustement de l'espacement vertical des paroles 575
Impression des doigtés à l'intérieur de la portée 575
Référencement de page 576
Espacement strictement proportionnel des notes 578
Alignement vertical des nuances indications textuelles 579
Positionnement d'une ossia et des paroles 579

MIDI 581
Affectation d'un canal MIDI par voix 581
Changement de tempo sans indication sur la partition 582
Création de nuance particulière pour la sortie MIDI 582
Personnalisation de batterie pour les sorties imprimable et MIDI 583
Démonstration de `midiInstrument` 585
Réglage de l'égalisation par défaut des instruments MIDI 589

Templates 591
Exemples de notation ancienne – transcription moderne de musique grégorienne 591
Psalmodie anglicane 592
Modèle pour cantique 594
Symboles de jazz 596
Orchestre, chœur et piano 602
Piano – cannevas simple 606
Piano et paroles entre les portées 607
Piano, mélodie et paroles 608
Modèle pour chœur SATB, sur quatre portées 609
Score for diatonic accordion 610
Paroles, musique et accords 615

Paroles, musique, accords et diagrammes de fret 616
Mélodie simple et accords 617
Portée unique et paroles 617
Portée unique avec quelques notes 618
Quatuor à cordes (conducteur) 618
Quatuor à cordes, avec parties séparées 620
Ensemble vocal avec réduction pour piano 623
Ensemble vocal avec alignement des paroles selon le contexte 625
Ensemble vocal avec couplet et refrain 626
Ensemble vocal (simple) 628
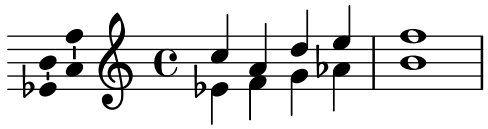
Pitches

Section “Pitches” dans *Manuel de notation*

Un ambitus par voix

L’`ambitus` peut être individualisé par voix. Il faut en pareil cas éviter qu’ils se chevauchent.

```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus.X-offset = #2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```

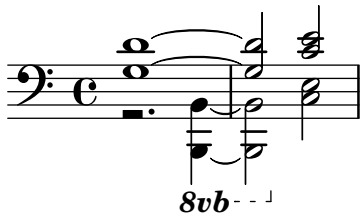


Ajout d’une indication d’octave pour une seule voix

Lorsque plusieurs voix cohabitent sur une même portée, déterminer l’octaviation d’une voix affectera la position des notes de toutes les voix, jusqu’à la fin du crochet d’octaviation. Si l’octaviation ne doit s’appliquer qu’à une seule voix, les positionnements du do central (propriété `middleCPosition`) et du crochet d’octaviation peuvent s’indiquer explicitement. Dans l’exemple suivant, le `middleCPosition` qui a normalement une valeur de 6 en clef de fa – soit six crans au-dessus de la ligne médiane – est déterminé à 7 crans au-dessus (une octave) pour la durée de l’*ottava*.

```
{
  \clef bass
  << { <g d'>1~ q2 <c' e'> }
  \\
    {
      r2.
      \set Staff.ottavation = #"8vb"
      \once \override Staff.OttavaBracket.direction = #DOWN
      \set Voice.middleCPosition = #(+ 6 7)
      <b,,, b,,>4 ~ |
      q2
      \unset Staff.ottavation
      \unset Voice.middleCPosition
```

```
      <c e>2
    }
  >>
}
```

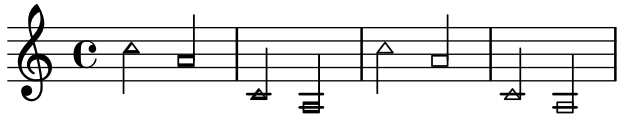


Variante fine des notes profilées Aiken

Les notes profilées Aiken, lorsqu'elles sont blanches, deviennent difficile à distinguer avec des tailles de portée réduites, notamment en présence de ligne supplémentaire. Perdant du blanc sur leur intérieur les fait alors ressembler à des noires.

```
\score {
  {
    \aikenHeads
    c''2 a' c' a

    % Switch to thin-variant noteheads
    \set shapeNoteStyles = ##(doThin reThin miThin
                              faThin sol laThin tiThin)
    c'' a' c' a
  }
}
% END EXAMPLE
```

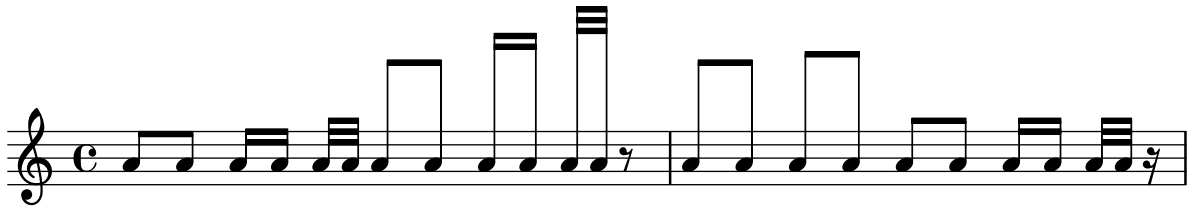


Modification de la hauteur de hampes ligaturées

La hauteur de hampe des notes ligaturées est gérée par la sous-propriété `beamed-lengths` des `details` de l'objet `Stem`. Lorsqu'elle ne comporte qu'une seule valeur, cette hauteur s'appliquera à toutes les hampes. En présence de plusieurs arguments, le premier affectera les croches, le second les doubles croches, et ainsi de suite. Le dernier argument s'appliquera aussi aux notes de plus courte durée. Les arguments peuvent être des valeurs décimales.

```
\relative c'' {
  \override Stem.details.beamed-lengths = #'(2)
  a8[ a] a16[ a] a32[ a]
  \override Stem.details.beamed-lengths = #'(8 10 12)
  a8[ a] a16[ a] a32[ a] r8
  \override Stem.details.beamed-lengths = #'(8)
  a8[ a]
  \override Stem.details.beamed-lengths = #'(8.5)
  a8[ a]
  \revert Stem.details.beamed-lengths
```
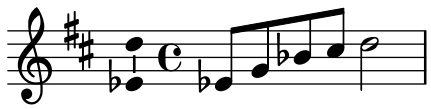
```
  a8[ a] a16[ a] a32[ a] r16
}
```



Ambitus après l'armure

L'ambitus se place par défaut à gauche de la clef. La fonction `\ambitusAfter` permet cependant de modifier ce positionnement ; sa syntaxe est `\ambitusAfter` *`grob-interface`* – voir Section "Graphical Object Interfaces" dans *Référence des propriétés internes* pour une liste des valeurs de *`grob-interface`* possibles. L'un des cas d'usage est d'insérer l'ambitus entre l'armure et la métrique.
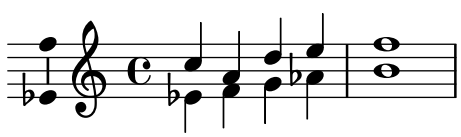
```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```



Ambitus sur plusieurs voix

Si plusieurs voix se trouvent sur une même portée, on peut attribuer le graveur `Ambitus_engraver` au contexte `Staff` afin d'obtenir l'ambitus sur toutes les voix cumulées, non d'une seule des voix actives.

```
\new Staff \with {
  \consists "Ambitus_engraver"
  }
<<
  \new Voice \relative c'' {
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```
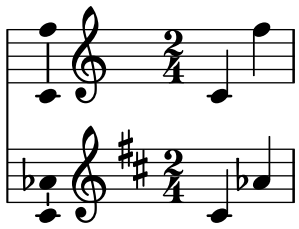


Ambitus

Un *ambitus* indique la tessiture, autrement dit les hauteurs extrêmes d'une voix.

Seules seront affichées les altérations non comprises dans l'armure. Les objets `AmbitusNoteHead` peuvent avoir des lignes supplémentaires.

```
\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

<<
  \new Staff {
    \relative c' {
      \time 2/4
      c4 f'
    }
  }
  \new Staff {
    \relative c' {
      \time  2/4
      \key d \major
      cis4 as'
    }
  }
>>
```



Profilage des notes selon leur degré dans la gamme

La propriété `shapeNoteStyles` permet d'affecter un profil particulier à chaque degré de la gamme – à partir de l'armure ou de la propriété `tonic`. Ses valeurs sont constituées d'une liste de symboles, qu'il s'agisse de formes géométriques (`triangle`, `cross` ou `xcircle`) ou basés sur la tradition des graveurs américains (avec quelques noms de note latins).

LilyPond dispose de deux raccourcis, `\aikenHeads` et `\sacredHarpHeads`, permettant de reproduire d'anciens recueils de chansons américaines.

L'exemple suivant montre plusieurs manières de profiler les têtes de note, ainsi que la capacité de transposer tout en respectant la fonction harmonique de chaque note dans la gamme.
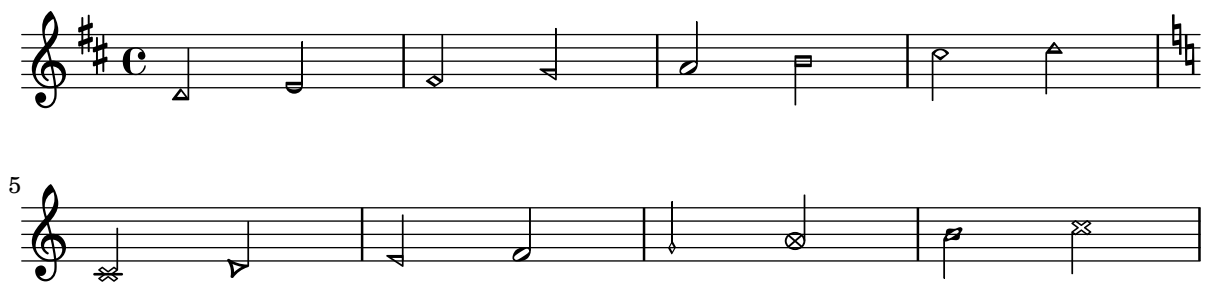
```
fragment = {
  \key c \major
  c2 d
  e2 f
  g2 a
  b2 c
```

```
}

\new Staff {
  \transpose c d
  \relative c' {
    \set shapeNoteStyles = ##(do re mi fa
                              #f la ti)
    \fragment
  }

  \break

  \relative c' {
    \set shapeNoteStyles = ##(cross triangle fa #f
                              mensural xcircle diamond)
    \fragment
  }
}
```
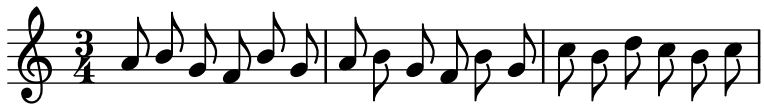


Changement automatique de l'orientation de hampe de la note médiane selon la mélodie

Afin de suivre la ligne mélodique, LilyPond peut inverser l'orientation de hampe de la note médiane, dès lors qu'aura été ajouté au contexte de voix le graveur `Melody_engraver`.

La propriété de contexte `suspendMelodyDecisions` permet, si besoin, de désactiver temporairement ce comportement.

```
\relative c'' {
  \time 3/4
  a8 b g f b g |
  \set suspendMelodyDecisions = ##t
  a  b g f b g |
  \unset suspendMelodyDecisions
  c  b d c b c |
}

\layout {
  \context {
    \Voice
    \consists "Melody_engraver"
    \autoBeamOff
  }
}
```

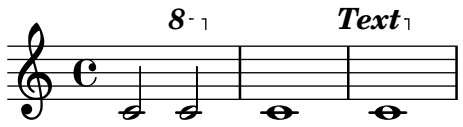


Modification du texte des marques d'octaviation

En interne, la fonction `\ottava` détermine les propriétés `ottavation` (par ex. en `"8va"` ou `"8vb"`) et `middleCPosition`. Vous pouvez modifier le texte d'une marque d'octaviation en définissant `ottavation` après avoir fait appel à `ottava`.

Un texte bref est particulièrement utile lorsque l'octaviation est courte.

```
{
  c'2
  \ottava #1
  \set Staff.ottavation = #"8"
  c''2
  \ottava #0
  c'1
  \ottava #1
  \set Staff.ottavation = #"Text"
  c''1
}
```



Réglage de l'affichage d'un ambitus

L'affichage d'un *ambitus* peut s'affiner pour répondre à vos préférences en matière d'esthétique.
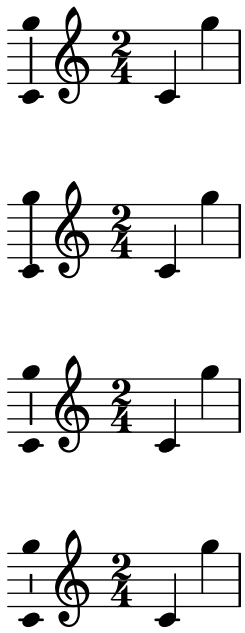
```
\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

\new Staff {
  \time 2/4
  % Default setting
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = #0
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine.gap = #1
  c'4 g''
}
```

```
\new Staff {
  \time 2/4
  \override AmbitusLine.gap = #1.5
  c'4 g''
}
```
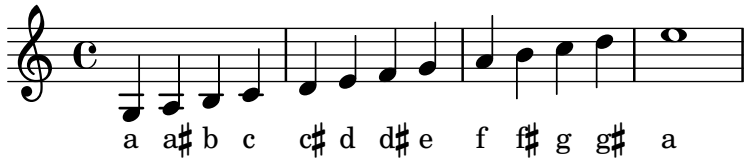


Modification de l'intervalle des lignes de la portée

`staffLineLayoutFunction` permet de changer le positionnement des notes. Dans cet exemple, la valeur qui lui est attribuée – `ly:pitch-semitones` – génère une gamme chromatique où l'écart entre chaque ligne et interligne de la portée est ramené au demi ton.

```
scale = \relative c' {
  a4 ais b c
  cis4 d dis e
  f4 fis g gis
  a1
}
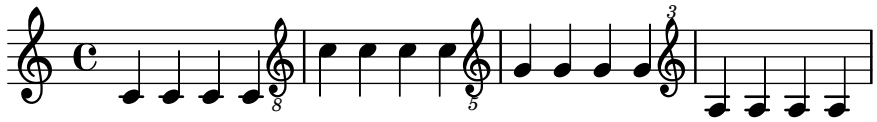
\new Staff \with {
  \remove "Accidental_engraver"
  staffLineLayoutFunction = #ly:pitch-semitones
}
{
  <<
    \scale
    \context NoteNames {
      \set printOctaveNames = ##f
      \scale
    }
  >>
}
```



Transposition arbitraire d'une clef

Les clefs peuvent être transposées d'un intervalle différent de l'octave.

```
\relative c' {
  \clef treble
  c4 c c c
  \clef "treble_8"
  c4 c c c
  \clef "treble_5"
  c4 c c c
  \clef "treble^3"
  c4 c c c
}
```



Coloration des notes selon leur hauteur

Les têtes de note peuvent adopter une couleur différente selon leur hauteur ou leur nom ; la fonction utilisée ici fait même la distinction entre enharmoniques.
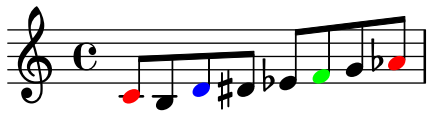
```
%Association list of pitches to colors.
#(define color-mapping
   (list
    (cons (ly:make-pitch 0 0 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 0 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 1 FLAT) (x11-color 'green))
    (cons (ly:make-pitch 0 2 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 2 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 3 FLAT) (x11-color 'red))
    (cons (ly:make-pitch 0 3 NATURAL) (x11-color 'green))
    (cons (ly:make-pitch 0 4 SHARP) (x11-color 'red))
    (cons (ly:make-pitch 0 5 NATURAL) (x11-color 'green))
    (cons (ly:make-pitch 0 5 FLAT) (x11-color 'red))
    (cons (ly:make-pitch 0 6 SHARP) (x11-color 'red))
    (cons (ly:make-pitch 0 1 NATURAL) (x11-color 'blue))
    (cons (ly:make-pitch 0 3 SHARP) (x11-color 'blue))
    (cons (ly:make-pitch 0 4 FLAT) (x11-color 'blue))
    (cons (ly:make-pitch 0 5 SHARP) (x11-color 'blue))
    (cons (ly:make-pitch 0 6 FLAT) (x11-color 'blue))))

%Compare pitch and alteration (not octave).
#(define (pitch-equals? p1 p2)
   (and
    (= (ly:pitch-alteration p1) (ly:pitch-alteration p2))
    (= (ly:pitch-notename p1) (ly:pitch-notename p2))))
```

```
#(define (pitch-to-color pitch)
   (let ((color (assoc pitch color-mapping pitch-equals?)))
     (if color
         (cdr color))))

#(define (color-notehead grob)
   (pitch-to-color
    (ly:event-property (event-cause grob) 'pitch)))

\score {
  \new Staff \relative c' {
    \override NoteHead.color = #color-notehead
    c8 b d dis ees f g aes
  }
}
```
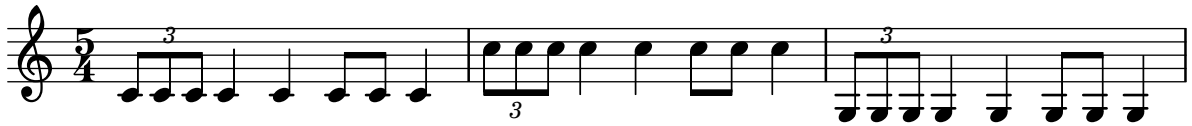


Création d'une séquence de notes de même hauteur

Lorsque la musique comporte à de nombreuses reprises une même séquence sur des hauteurs différentes, la fonction musicale ci-dessous peut s'avérer fort utile. Elle considère une note dont seule la hauteur est utilisée. Cet exemple reproduit le rythme utilisé tout au long de « Mars », l'une des pièces de l'œuvre de Gustav Holst « Les planètes ».

```
rhythm =
#(define-music-function (p) (ly:pitch?)
   "Make the rhythm in Mars (the Planets) at the given pitch"
  #{ \tuplet 3/2 { $p 8 8 8 } 4 4 8 8 4 #})

\new Staff {
  \time 5/4
  \rhythm c'
  \rhythm c''
  \rhythm g
}
```



Création d'armures personnalisées

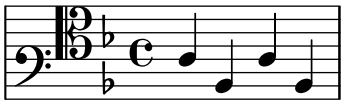
LilyPond accepte les armures personnalisées. Dans cet exemple, il s'agit d'une tonalité de ré mineur à laquelle on rajoute des bémols.

```
\new Staff \with {
  \override StaffSymbol.line-count = #8
  \override KeySignature.flat-positions = #'((-7 . 6))
  \override KeyCancellation.flat-positions = #'((-7 . 6))
  % presumably sharps are also printed in both octaves
```

```
  \override KeySignature.sharp-positions = #'((-6 . 7))
  \override KeyCancellation.sharp-positions = #'((-6 . 7))

  \override Clef.stencil = #
  (lambda (grob)(grob-interpret-markup grob
  #{ \markup\combine
    \musicglyph "clefs.C"
    \translate #'(-3 . -2)
    \musicglyph "clefs.F"
   #}))
    clefPosition = #3
    middleCPosition = #3
    middleCClefPosition = #3
}
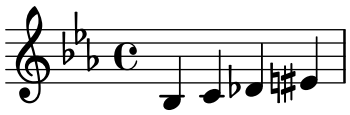
{
  \key d\minor
  f bes, f bes,
}
```



Impression forcée d'un bécarre avant une accidentelle

L'exemple suivant indique comment forcer l'impression d'un bécarre avant un changement d'altération.

```
\relative c' {
  \key es \major
  bes c des
  \tweak Accidental.restore-first ##t
  eis
}
```
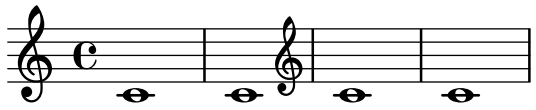


Impression forcée de la clef

Lorsqu'une clef a déjà été imprimée et qu'aucun changement n'est intervenu, LilyPond ignore toute répétition de la commande `\clef` puisque le signe est le même. Il est cependant possible de rappeler la clef à s'imprimer à l'aide de l'instruction `\set Staff.forceClef = ##t`.

```
\relative c' {
  \clef treble
  c1
  \clef treble
  c1
  \set Staff.forceClef = ##t
  c1
  \clef treble
```
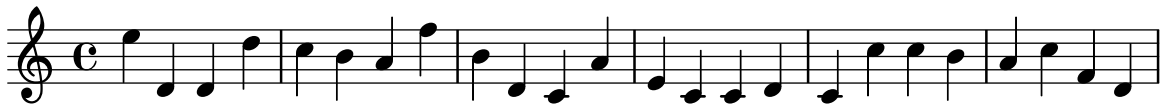
```
  c1
}
```



Génération de notes aléatoires

Le fragment de code Scheme ci-dessous génère aléatoirement 24 notes (ou autant que nécessaire), à partir de l'heure courante (ou un nombre quelconque donné en argument, afin d'obtenir toujours les mêmes notes aléatoires). Pour obtenir une autre série de notes, il suffit de changer ce nombre.

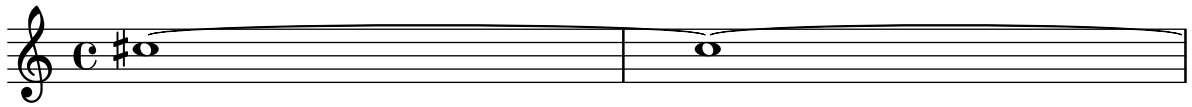
```
\score {
  {
    $(let ((random-state (seed->random-state (current-time))))
       (make-sequential-music
         (map (lambda (x)
                (let ((idx (random 12 random-state)))
                  (make-event-chord
                   (list
                    (make-music 'NoteEvent
                                'duration (ly:make-duration 2 0 1/1)
                                'pitch (ly:make-pitch
                                        (quotient idx 7)
                                        (remainder idx 7)
                                        0))))))
              (make-list 24))))
  }
}
```

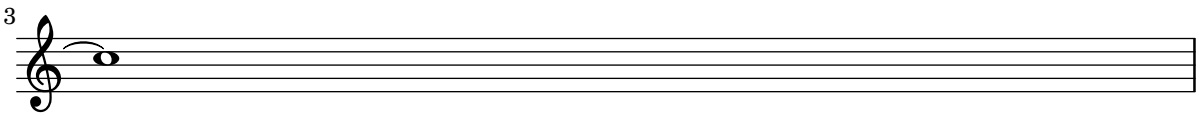


Non répétition de l'altération après saut de ligne sur liaison de prolongation

Cet exemple illustre comment, lorsqu'une note affublée d'une altération accidentelle est prolongée, ne pas répéter cette altération après un saut de ligne.

```
\relative c'' {
  \override Accidental.hide-tied-accidental-after-break = ##t
  cis1~ cis~
  \break
  cis
}
```
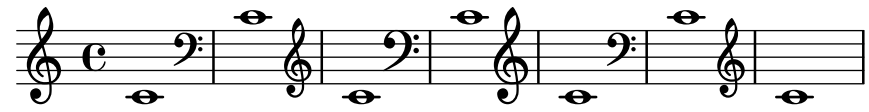




Conservation de la taille lors d'un changement de clef

Lorsqu'un changement de clef intervient, le nouveau signe s'imprime dans une taille inférieure. On peut y déroger avec `full-size-change`.

```
\relative c' {
  \clef "treble"
  c1
  \clef "bass"
  c1
  \clef "treble"
  c1
  \override Staff.Clef.full-size-change = ##t
  \clef "bass"
  c1
  \clef "treble"
  c1
  \revert Staff.Clef.full-size-change
  \clef "bass"
  c1
  \clef "treble"
  c1
}
```
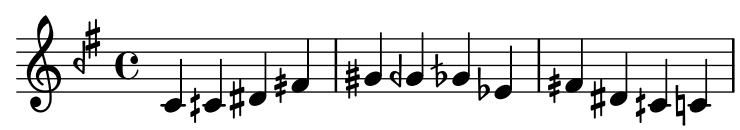


Exemple de musique « Makam »

Le « makam » est une forme de mélodie turque qui utilise des altérations d'un neuvième de ton.

Consultez le fichier d'initialisation `ly/makam.ly` pour plus de détails sur les hauteurs et altérations utilisées (voir le chapitre 4.6.3 - Autres sources d'information du manuel d'initiation pour le localiser).

```
% Initialize makam settings
\include "makam.ly"

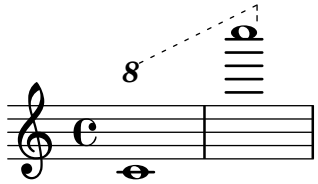
\relative c' {
  \set Staff.keyAlterations = #`((6 . ,(- KOMA)) (3 . ,BAKIYE))
  c4 cc db fk
  gbm4 gfc gfb efk
  fk4 db cc c
}
```



Modification de la pente de l'extension d'octaviation

Il est possible d'adapter la pente d'une indication d'octaviation.

```
\relative c'' {
  \override Staff.OttavaBracket.stencil = #ly:line-spanner::print
  \override Staff.OttavaBracket.bound-details =
    #`((left . ((Y . 0) ; Change the integer here
                (attach-dir . ,LEFT)
                (padding . 0)
                (stencil-align-dir-y . ,CENTER)))
       (right . ((Y . 5) ; Change the integer here
                 (padding . 0)
                 (attach-dir . ,RIGHT)
                 (text . ,(make-draw-dashed-line-markup
                           (cons 0 -1.2))))))
  \override Staff.OttavaBracket.left-bound-info =
     #ly:line-spanner::calc-left-bound-info-and-text
  \override Staff.OttavaBracket.right-bound-info =
     #ly:line-spanner::calc-right-bound-info
  \ottava #1
  c1
  c'''1
}
```



Armures inhabituelles

La commande `\key` détermine la propriété `keyAlterations` d'un contexte `Staff`. Des armures inhabituelles peuvent être spécifiées en modifiant directement cette propriété.

Il s'agit en l'occurrence de définir une liste :
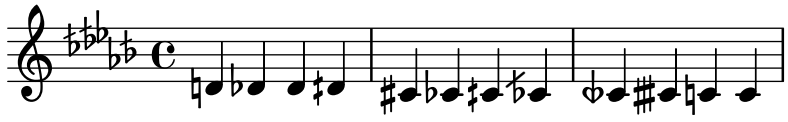
```
\set Staff.keyAlterations =
  #`(((octave . pas) . altération) ((octave . pas) . altération)
  @dots{})
```

dans laquelle, et pour chaque élément, `octave` spécifie l'octave (0 pour celle allant du do médium au si supérieur), `pas` la note dans cette octave (0 pour do et 6 pour si), et `altération` sera `,SHARP` ou `,FLAT` ou `,DOUBLE-SHARP`, etc. (attention à la virgule en préfixe).

Une formulation abrégée – `(pas . altération)` – signifie que l'altération de l'élément en question sera valide quelle que soit l'octave. En ce qui concerne les gammes microtonales dans lesquelles un « dièse » n'est pas d'un centième, `altération` se réfère à un deux-centième de ton entier.

```
\include "arabic.ly"
\relative do' {
  \set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
                                 (1 . ,SEMI-FLAT)
                                 (2 . ,FLAT)
                                 (5 . ,FLAT)
```

```
                                (6 . ,SEMI-FLAT))
%\set Staff.extraNatural = ##f
  re reb \dwn reb resd
  dod dob dosd \dwn dob |
  dobsb dodsd do do |
}
```



Easy play – chiffres en lieu et place des lettres

En mode « easy play », les têtes de note utilisent la propriété `note-names` attachée à l'objet `NoteHead` pour déterminer ce qui apparaîtra dans la tête. Intervenir sur cette propriété permet d'imprimer un chiffre correspondant au degré dans la gamme.

La création d'un graveur dédié permet de traiter toutes les notes.

```
#(define Ez_numbers_engraver
   (make-engraver
    (acknowledgers
     ((note-head-interface engraver grob source-engraver)
      (let* ((context (ly:translator-context engraver))
             (tonic-pitch (ly:context-property context 'tonic))
             (tonic-name (ly:pitch-notename tonic-pitch))
             (grob-pitch
              (ly:event-property (event-cause grob) 'pitch))
             (grob-name (ly:pitch-notename grob-pitch))
             (delta (modulo (- grob-name tonic-name) 7))
             (note-names
              (make-vector 7 (number->string (1+ delta)))))
        (ly:grob-set-property! grob 'note-names note-names))))))

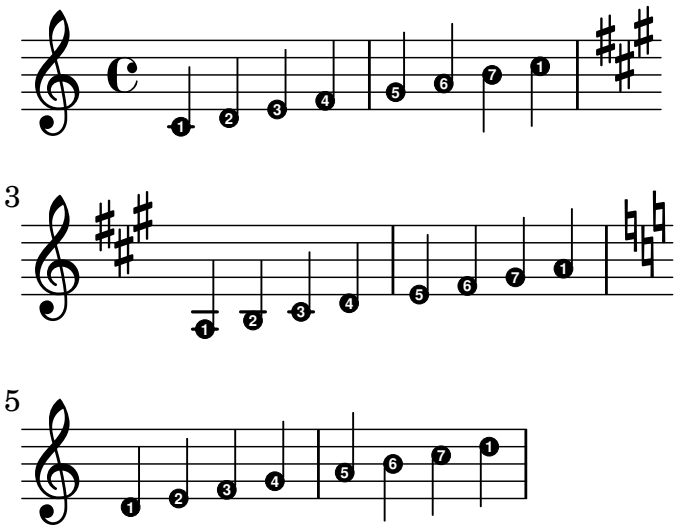
#(set-global-staff-size 26)

\layout {
  ragged-right = ##t
  \context {
    \Voice
    \consists \Ez_numbers_engraver
  }
}

\relative c' {
  \easyHeadsOn
  c4 d e f
  g4 a b c \break

  \key a \major
  a,4 b cis d
  e4 fis gis a \break
```

```
  \key d \dorian
  d,4 e f g
  a4 b c d
}
```



Orchestre, chœur et piano

Ce canevas illustre l'utilisation de contextes `StaffGroup` pour regrouper les instruments selon leur famille, imbriqués dans un `GrandStaff`, ainsi que le recours à la fonction `\transpose` pour les instruments transpositeurs. Dans tous les identificateurs, la musique est stockée en ut. Les notes peuvent tout aussi bien être saisies en ut ou dans la tonalité particulière de l'instrument avant d'être transposées puis affectées à une variable.

```
#(set-global-staff-size 17)
\paper {
  indent = 3.0\cm  % add space for instrumentName
  short-indent = 1.5\cm  % add less space for shortInstrumentName
}

fluteMusic = \relative c' { \key g \major g'1 b }

% Pitches as written on a manuscript for Clarinet in A
% are transposed to concert pitch.

clarinetMusic = \transpose c' a
  \relative c'' { \key bes \major bes1 d }

trumpetMusic = \relative c { \key g \major g''1 b }

% Key signature is often omitted for horns

hornMusic = \transpose c' f
  \relative c { d'1 fis }

percussionMusic = \relative c { \key g \major g1 b }

sopranoMusic = \relative c'' { \key g \major g'1 b }
```

```
sopranoLyrics = \lyricmode { Lyr -- ics }

altoIMusic = \relative c' { \key g \major g'1 b }

altoIIMusic = \relative c' { \key g \major g'1 b }

altoILyrics =  \sopranoLyrics

altoIILyrics = \lyricmode { Ah -- ah }

tenorMusic = \relative c' { \clef "treble_8" \key g \major g1 b }

tenorLyrics = \sopranoLyrics

pianoRHMusic = \relative c { \key g \major g''1 b }

pianoLHMusic = \relative c { \clef bass \key g \major g1 b }

violinIMusic = \relative c' { \key g \major g'1 b }

violinIIMusic = \relative c' { \key g \major g'1 b }

violaMusic = \relative c { \clef alto \key g \major g'1 b }

celloMusic = \relative c { \clef bass \key g \major g1 b }

bassMusic = \relative c { \clef "bass_8" \key g \major g,1 b }

\score {
  <<
    \new StaffGroup = "StaffGroup_woodwinds" <<
      \new Staff = "Staff_flute" \with { instrumentName = "Flute" }
      \fluteMusic

      \new Staff = "Staff_clarinet" \with {
        instrumentName = \markup { \concat { "Clarinet in B" \flat } }
      }

      % Declare that written Middle C in the music
      % to follow sounds a concert B flat, for
      % output using sounded pitches such as MIDI.
      %\transposition bes

      % Print music for a B-flat clarinet
      \transpose bes c' \clarinetMusic
    >>

    \new StaffGroup = "StaffGroup_brass" <<
      \new Staff = "Staff_hornI" \with { instrumentName = "Horn in F" }
       % \transposition f
        \transpose f c' \hornMusic
```

```
  \new Staff = "Staff_trumpet" \with { instrumentName = "Trumpet in  C" }
  \trumpetMusic

>>
\new RhythmicStaff = "RhythmicStaff_percussion"
\with { instrumentName = "Percussion" }
<<
  \percussionMusic
>>
\new PianoStaff \with { instrumentName = "Piano" }
<<
  \new Staff { \pianoRHMusic }
  \new Staff { \pianoLHMusic }
>>
\new ChoirStaff = "ChoirStaff_choir" <<
  \new Staff = "Staff_soprano" \with { instrumentName = "Soprano" }
  \new Voice = "soprano"
  \sopranoMusic

  \new Lyrics \lyricsto "soprano" { \sopranoLyrics }
  \new GrandStaff = "GrandStaff_altos"
  \with { \accepts Lyrics } <<
    \new Staff = "Staff_altoI"  \with { instrumentName = "Alto I" }
    \new Voice = "altoI"
    \altoIMusic

    \new Lyrics \lyricsto "altoI" { \altoILyrics }
    \new Staff = "Staff_altoII" \with { instrumentName = "Alto II" }
    \new Voice = "altoII"
    \altoIIMusic

    \new Lyrics \lyricsto "altoII" { \altoIILyrics }
  >>

  \new Staff = "Staff_tenor" \with { instrumentName = "Tenor" }
    \new Voice = "tenor"
    \tenorMusic

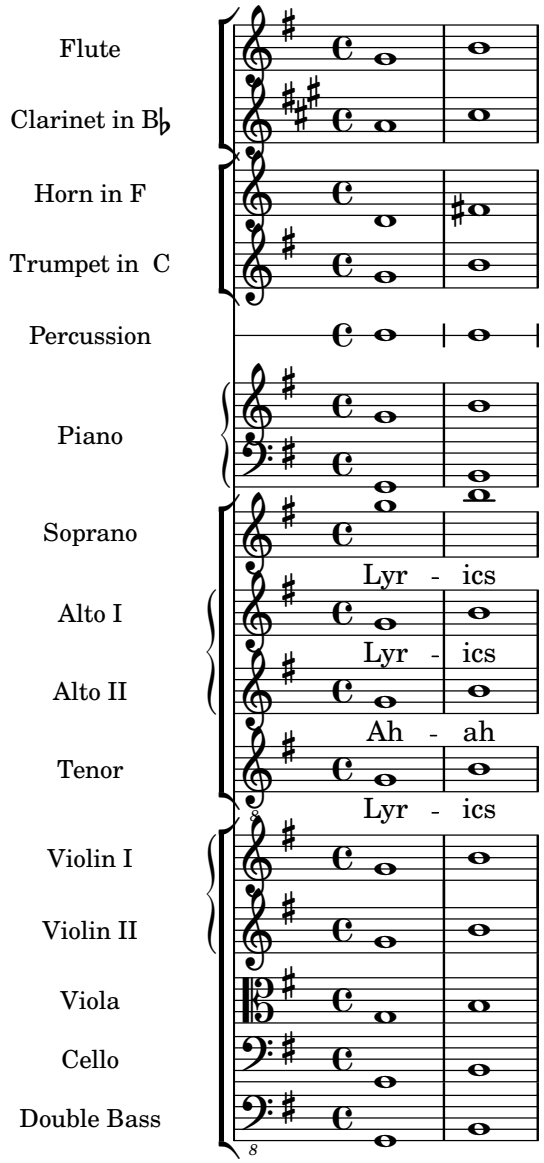
  \new Lyrics \lyricsto "tenor" { \tenorLyrics }
>>
\new StaffGroup = "StaffGroup_strings" <<
  \new GrandStaff = "GrandStaff_violins" <<
    \new Staff = "Staff_violinI" \with { instrumentName = "Violin I" }
    \violinIMusic

    \new Staff = "Staff_violinII" \with { instrumentName = "Violin II" }
    \violinIIMusic
  >>

  \new Staff = "Staff_viola" \with { instrumentName = "Viola" }
  \violaMusic
```

```
      \new Staff = "Staff_cello" \with { instrumentName = "Cello" }
      \celloMusic

      \new Staff = "Staff_bass" \with { instrumentName = "Double Bass" }
      \bassMusic
    >>
  >>
  \layout { }
}
```
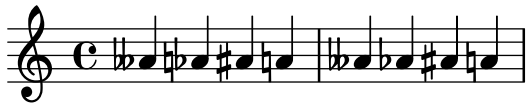


Suppression des bécarres superflus

En accord avec les règles traditionnelles de l'écriture musicale, on grave un bécarre avant un dièse ou un bémol si la note était auparavant affublée d'un double-dièse ou double-bémol. Pour adopter un comportement plus contemporain, la propriété `extraNatural` du contexte `Staff` doit se voir attribuer la valeur `##f` (faux).

```
\relative c'' {
```
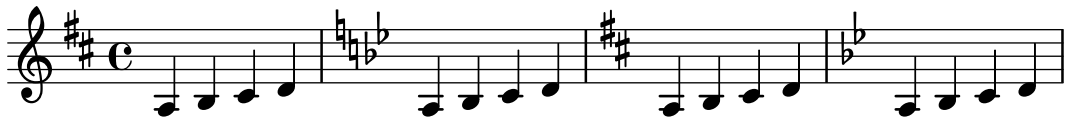
```
  aeses4 aes ais a
  \set Staff.extraNatural = ##f
  aeses4 aes ais a
}
```



Suppression des bécarres superflus lors d'un changement de tonalité

Après un changement de tonalité, un bécarre est imprimé pour annuler toute altération précédente. Ce comportement s'annule en désactivant la propriété `printKeyCancellation` du contexte `Staff`.

```
\relative c' {
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
  \set Staff.printKeyCancellation = ##f
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
}
```



Citation d'une autre voix et transposition

Les citations tiennent compte de la transposition, aussi bien celle de l'instrument d'origine que celle de la partie où elles interviennent. Dans l'exemple suivant, tous les instruments sont en tonalité de concert et seront repris par un instrument en fa. Le destinataire de la citation peut à son tour transposer à l'aide de la commande `\transpose`. En pareil cas, toutes les hauteurs seront transposées, y compris celles de la citation.
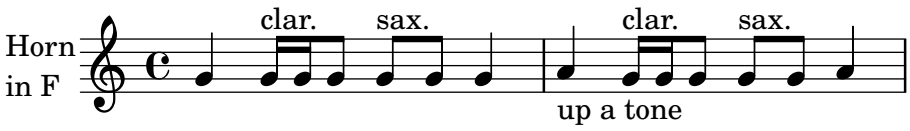
```
\addQuote clarinet {
  \transposition bes
  \repeat unfold 8 { d'16 d' d'8 }
}

\addQuote sax {
  \transposition es'
  \repeat unfold 16 { a8 }
}

quoteTest = {
  % french horn
  \transposition f
  g'4
```

```
  << \quoteDuring "clarinet" { \skip 4 } s4^"clar." >>
  << \quoteDuring "sax" { \skip 4 } s4^"sax." >>
  g'4
}

{
  \new Staff \with {
    instrumentName = \markup { \column { Horn "in F" } }
  }
  \quoteTest
  \transpose c' d' << \quoteTest s4_"up a tone" >>
}
```



Séparation entre altérations annulées et nouvelle armure

Les altérations supprimées lors d'un changement de tonalité sont par défaut accolées à la nouvelle armure. Ce comportement peut s'adapter grâce à la propriété `break-align-orders` de l'objet `BreakAlignment`.
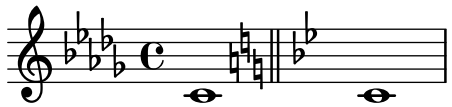
La valeur de `break-align-orders` est constituée d'un vecteur à trois composantes listant l'ordre des éléments à l'occasion d'une rupture. Seule la deuxième liste est ici modifiée, `key-cancellation` étant déplacé avant `staff-bar`. En ne modifiant que la deuxième liste, cette modification dans l'ordre d'apparition des éléments sera effective seulement en cours de système, et non en début ou fin de ligne.

```
\new Staff {
  \override Score.BreakAlignment.break-align-orders =
    ##((left-edge ambitus breathing-sign clef staff-bar
                  key-cancellation key-signature time-signature custos)

       (left-edge ambitus breathing-sign clef key-cancellation
                  staff-bar key-signature time-signature custos)

       (left-edge ambitus breathing-sign clef key-cancellation
                  key-signature staff-bar time-signature custos))

  \key des \major
  c'1
  \bar "||"
  \key bes \major
  c'1
}
```



Transposition et réduction du nombre d'altérations accidentelles

Cet exemple, grâce à un peu de code Scheme, donne la priorité aux enharmoniques afin de limiter le nombre d'altérations supplémentaires. La règle applicable est :

- Les altérations doubles sont supprimées
- Si dièse -> Do
- Mi dièse -> Fa
- Do bémol -> Si
- Fa bémol -> Mi

Cette façon de procéder aboutit à plus d'enharmoniques naturelles.

```
#(define (naturalize-pitch p)
   (let ((o (ly:pitch-octave p))
         (a (* 4 (ly:pitch-alteration p)))
         ;; alteration, a, in quarter tone steps,
         ;; for historical reasons
         (n (ly:pitch-notename p)))
     (cond
      ((and (> a 1) (or (eqv? n 6) (eqv? n 2)))
       (set! a (- a 2))
       (set! n (+ n 1)))
      ((and (< a -1) (or (eqv? n 0) (eqv? n 3)))
       (set! a (+ a 2))
       (set! n (- n 1))))
     (cond
      ((> a 2) (set! a (- a 4)) (set! n (+ n 1)))
      ((< a -2) (set! a (+ a 4)) (set! n (- n 1))))
     (if (< n 0) (begin (set! o (- o 1)) (set! n (+ n 7))))
     (if (> n 6) (begin (set! o (+ o 1)) (set! n (- n 7))))
     (ly:make-pitch o n (/ a 4))))
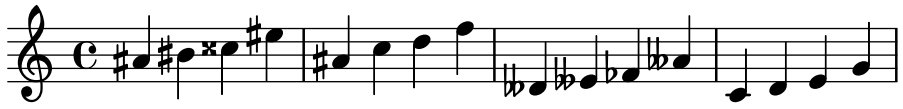
#(define (naturalize music)
   (let ((es (ly:music-property music 'elements))
         (e (ly:music-property music 'element))
         (p (ly:music-property music 'pitch)))
     (if (pair? es)
         (ly:music-set-property!
          music 'elements
          (map naturalize es)))
     (if (ly:music? e)
         (ly:music-set-property!
          music 'element
          (naturalize e)))
     (if (ly:pitch? p)
         (begin
           (set! p (naturalize-pitch p))
           (ly:music-set-property! music 'pitch p)))
     music))

naturalizeMusic =
#(define-music-function (m)
   (ly:music?)
```

```
   (naturalize m))

music = \relative c' { c4 d e g }

\score {
  \new Staff {
    \transpose c ais { \music }
    \naturalizeMusic \transpose c ais { \music }
    \transpose c deses { \music }
    \naturalizeMusic \transpose c deses { \music }
  }
  \layout { }
}
```



Exemple de makam turc

Ce canevas utilise le début d'un *saz semai* turc bien connu du répertoire aux fins d'illustrer certains éléments de la notation musicale turque.

```
% Initialize makam settings
\include "turkish-makam.ly"

\header {
    title = "Hüseyni Saz Semaisi"
    composer = "Lavtacı Andon"
}
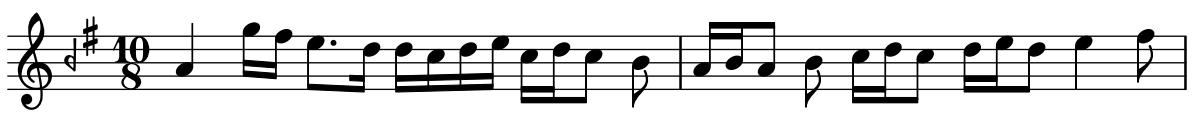
\relative {
  \set Staff.extraNatural = ##f
  \set Staff.autoBeaming = ##f
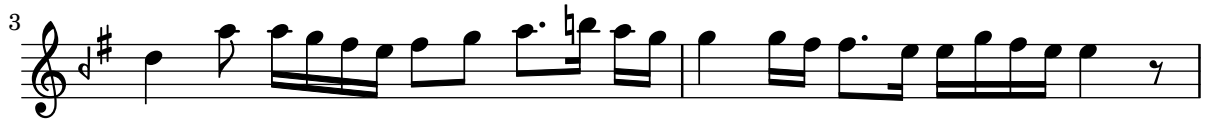
  \key a \huseyni
  \time 10/8

  a'4 g'16 [fb] e8. [d16] d [c d e] c [d c8] bfc |
  a16 [bfc a8] bfc c16 [d c8] d16 [e d8] e4 fb8 |
  d4 a'8 a16 [g fb e] fb8 [g] a8. [b16] a16 [g] |
  g4 g16 [fb] fb8. [e16] e [g fb e] e4 r8 |
}
```

Hüseyni Saz Semaisi

Lavtacı Andon





Affinage des propriétés d'une clef

Modifier le glyphe, la position de la clef ou son octaviation ne changeront pas la position des notes ; il faut pour y parvenir modifier aussi la position du do médium. La redéfinition préalable de `middleCClefPosition` permet de placer l'armure sur les bonnes lignes. Le positionnement est relatif à la ligne médiane, un nombre positif faisant monter, un nombre négatif abaissant.

Par exemple, la commande `\clef "treble_8"` équivaut à définir `clefGlyph`, `clefPosition` – qui contrôle la position verticale de la clef – `middleCPosition` et `clefOctavation`. Une nouvelle clef apparaîtra dès lors que l'une de ces propriétés, à l'exception de `middleCPosition`, aura été modifiée.

Les exemples qui suivent illustrent les différentes possibilités de définir ces propriétés manuellement. Sur la première ligne, la position relative des notes par rapport aux clefs est préservée, ce qui n'est pas le cas pour la deuxième ligne.

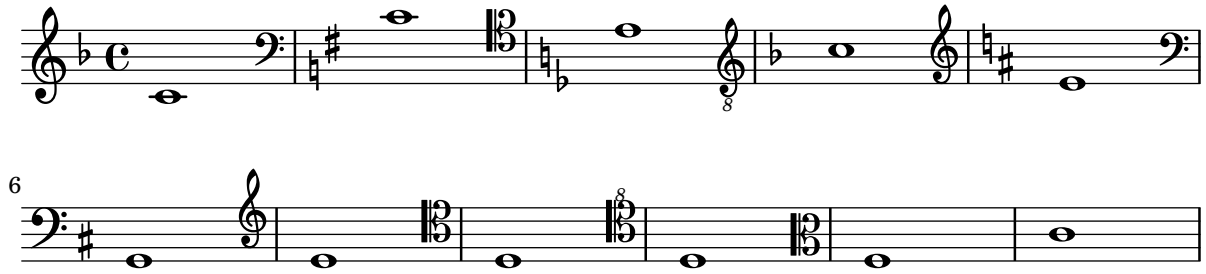
```
{
  % The default treble clef
  \key f \major
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  \set Staff.middleCPosition = #6
  \set Staff.middleCClefPosition = #6
  \key g \major
  c'1
  % The baritone clef
  \set Staff.clefGlyph = #"clefs.C"
  \set Staff.clefPosition = #4
  \set Staff.middleCPosition = #4
  \set Staff.middleCClefPosition = #4
  \key f \major
  c'1
  % The standard choral tenor clef
  \set Staff.clefGlyph = #"clefs.G"
  \set Staff.clefPosition = #-2
  \set Staff.clefTransposition = #-7
  \set Staff.middleCPosition = #1
  \set Staff.middleCClefPosition = #1
  \key f \major
  c'1
  % A non-standard clef
  \set Staff.clefPosition = #0
  \set Staff.clefTransposition = #0
  \set Staff.middleCPosition = #-4
  \set Staff.middleCClefPosition = #-4
  \key g \major
  c'1 \break

  % The following clef changes do not preserve
  % the normal relationship between notes, key signatures
  % and clefs:
```

```
  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  c'1
  \set Staff.clefGlyph = #"clefs.G"
  c'1
  \set Staff.clefGlyph = #"clefs.C"
  c'1
  \set Staff.clefTransposition = #7
  c'1
  \set Staff.clefTransposition = #0
  \set Staff.clefPosition = #0
  c'1

  % Return to the normal clef:

  \set Staff.middleCPosition = #0
  c'1
}
```



Utilisation d'autoChange avec plus d'une voix

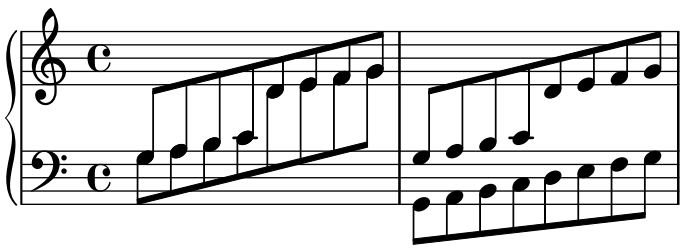
L'instruction `\autoChange` est opérationnelle y compris en présence de voix multiples.

```
\score
{
  \new PianoStaff
  <<
    \new Staff = "up" {
      <<
        \set Timing.beamExceptions = #'()
        \set Timing.beatStructure = #'(4)
        \new Voice {
          \voiceOne
          \autoChange
          \relative c' {
            g8 a b c d e f g
            g,8 a b c d e f g
          }
        }

        \new Voice {
          \voiceTwo
          \autoChange
```

```
          \relative c' {
            g8 a b c d e f g
            g,,8 a b c d e f g
          }
        }
      >>
    }

    \new Staff = "down" {
      \clef bass
    }
  >>
}
```



Rhythms

Section “Rhythms” dans *Manuel de notation*
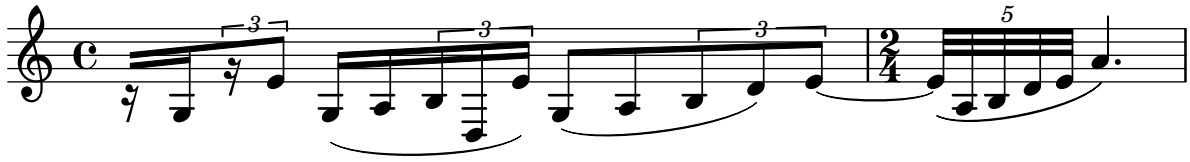
Coexistence de ligatures et liaisons avec des rythmes comprenant des n-olets

La syntaxe de LilyPond demande parfois un positionnement inhabituel des parenthèses, crochets, etc. qui peuvent s'entrelacer.

Par exemple, le crochet ouvrant une ligature manuelle se place **à la suite** de la note de départ et sa durée, non pas avant. De même, le crochet fermant se place à la fin de la ligature, y compris lorsque la dernière note se trouve incluse dans un n-olet.

Cet extrait illustre la manière de combiner ligatures manuelles, liaisons d'articulation, de prolongation ou de phrasé, avec des nolets bornées par des accolades.

```
{
  r16[ g16 \tuplet 3/2 { r16 e'8] }
  g16( a \tuplet 3/2 { b d e') }
  g8[( a \tuplet 3/2 { b d') e'] ~ }
  \time 2/4
  \tuplet 5/4 { e'32\( a b d' e' } a'4.\)
}
```



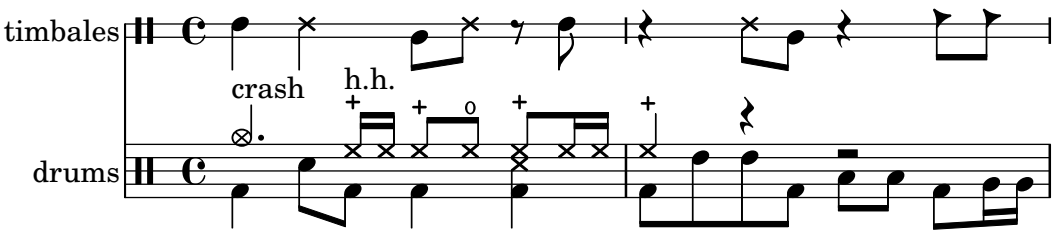
Ajout de parties de batterie

Grâce à la puissance des outils préconfigurés tels que la fonction `\drummode` et le contexte `DrumStaff`, la saisie de parties pour percussions est extrêmement simplifiée : chaque composant d'une batterie trouve sa place sur une portée dédiée (avec une clef particulière) et les têtes de note sont spécifiques à chaque élément. Il est également possible d'affecter un symbole particulier à chaque élément, tout comme de restreindre le nombre de lignes de la portée.

```
drh = \drummode {
        cymc4.^"crash" hhc16^"h.h." hh hhc8 hho hhc8 hh16 hh
        hhc4 r4 r2
      }
drl = \drummode {
        bd4 sn8 bd bd4 << bd ss >>
        bd8 tommh tommh bd toml toml bd tomfh16 tomfh
      }
timb = \drummode {
         timh4 ssh timl8 ssh r timh r4
         ssh8 timl r4 cb8 cb
       }

\score {
  <<
    \new DrumStaff \with {
      instrumentName = "timbales"
```

```
      drumStyleTable = #timbales-style
      \override StaffSymbol.line-count = #2
      \override BarLine.bar-extent = #'(-1 . 1)
    }
    <<
      \timb
    >>
    \new DrumStaff \with { instrumentName = "drums" }
    <<
      \new DrumVoice { \stemUp \drh }
      \new DrumVoice { \stemDown \drl }
    >>
  >>
  \layout { }
  \midi { \tempo 4 = 120 }
}
```
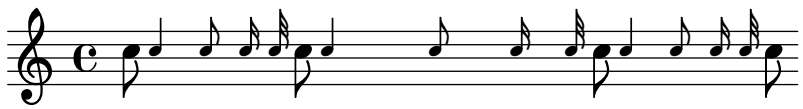


Ajustement de l'espacement des notes d'ornement

La propriété `spacing-increment` de `Score.GraceSpacing` permet d'ajuster l'espacement d'un groupe de notes d'ornement.

```
graceNotes = {
  \grace { c4 c8 c16 c32 }
  c8
}

\relative c'' {
  c8
  \graceNotes
  \override Score.GraceSpacing.spacing-increment = #2.0
  \graceNotes
  \revert Score.GraceSpacing.spacing-increment
  \graceNotes
}
```

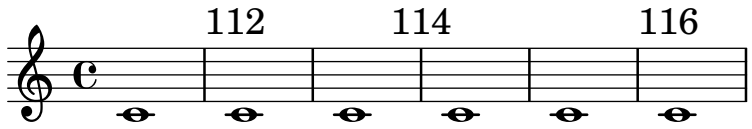


Alignement des numéros de mesure

Les numéros de mesure s'alignent en principe sur la droite de l'objet dont ils dépendent. C'est normalement le coin gauche de la portée ou, en cours de ligne, à gauche de la barre. Vous pouvez toutefois les centrer par rapport à la barre ou les afficher à droite de la barre.

```
\relative c' {
```

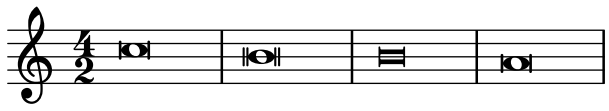
```
  \set Score.currentBarNumber = #111
  \override Score.BarNumber.break-visibility = #all-visible
  % Increase the size of the bar number by 2
  \override Score.BarNumber.font-size = #2
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
  c1 | c1
  % Center-align bar numbers
  \override Score.BarNumber.self-alignment-X = #CENTER
  c1 | c1
  % Left-align bar numbers
  \override Score.BarNumber.self-alignment-X = #LEFT
  c1 | c1
}
```



Brève alternative, avec deux barres verticales

Voici comment obtenir une brève – aussi appelée note carée – flanquée de deux barres verticales, au lieu d'une comme habituellement.
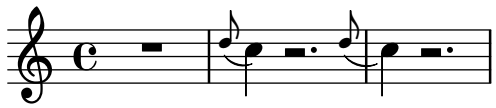
```
\relative c'' {
  \time 4/2
  c\breve |
  \override Staff.NoteHead.style = #'altdefault
  b\breve
  \override Staff.NoteHead.style = #'baroque
  b\breve
  \revert Staff.NoteHead.style
  a\breve
}
```



Appoggiature avant une barre de mesure

Par défaut, appogiatures et autres notes d'ornement sur le premier temps d'une mesure s'impriment après la barre. Elles peuvent toutefois précéder la barre grâce à l'insertion d'une barre invisible puis d'une visible, comme indiqué ici.

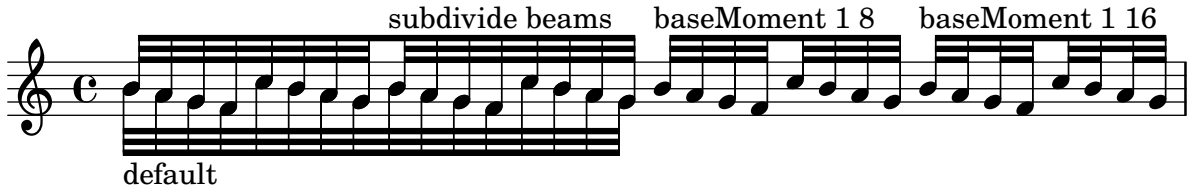
```
{
  R1
  %% default
  \appoggiatura d''8 c''4 r2.
  %% cheated
  \appoggiatura { \bar "" d''8 \bar "|" } c''4 r2.
}
```



Subdivision de ligatures automatiques

Dès lors que la propriété `subdivideBeams` aura été activée, une ligature sera subdivisée selon les préceptes de `beatLength`.

```
\new Staff {
  \relative c'' {
    <<
      {
        \voiceOne
        \set subdivideBeams = ##t
        b32[ a g f c' b a g
        b32^"subdivide beams" a g f c' b a g]
      }
      \new Voice {
        \voiceTwo
        b32_"default"[ a g f c' b a g
        b32 a g f c' b a g]
      }
    >>
    \oneVoice
    \set baseMoment = #(ly:make-moment 1/8)
    \set beatStructure = 2,2,2,2
    b32^"baseMoment 1 8"[ a g f c' b a g]
    \set baseMoment = #(ly:make-moment 1/16)
    \set beatStructure = 4,4,4,4
    b32^"baseMoment 1 16"[ a g f c' b a g]
  }
}
```



Changement automatique des durées

`shiftDurations` permet de modifier la longueur des notes d'un morceau.
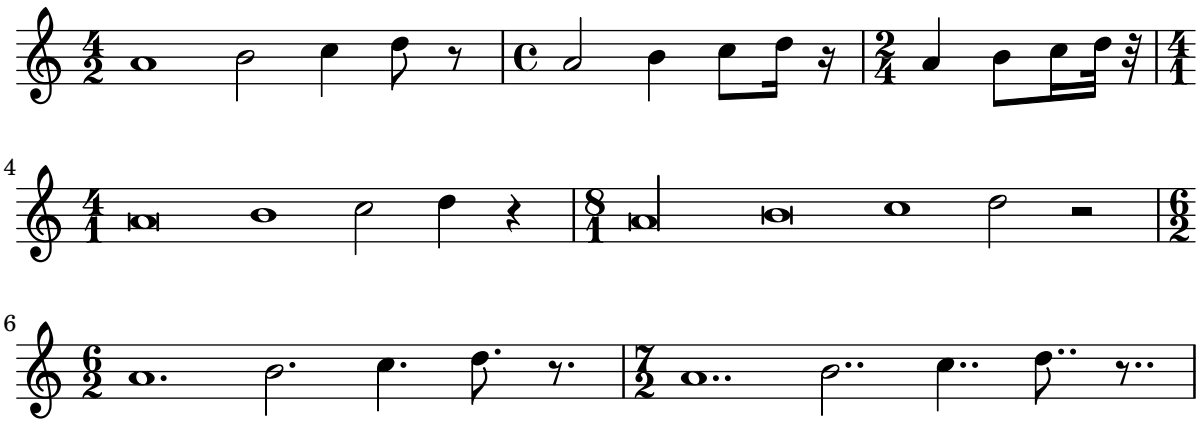
Cette instruction prend deux arguments : un coefficient d'échelonnement (une puissance de deux) et un nombre de points d'augmentation (entier positif).

```
\paper { indent = 0 }

music = \relative c'' { a1 b2 c4 d8 r }

\score {
  \new Voice {
    \time 4/2
    \music
    \time 4/4
```

```
    \shiftDurations #1 #0 { \music }
    \time 2/4
    \shiftDurations #2 #0 { \music }
    \time 4/1
    \shiftDurations #-1 #0 { \music }
    \time 8/1
    \shiftDurations #-2 #0 { \music }
    \time 6/2
    \shiftDurations #0 #1 { \music }
    \time 7/2
    \shiftDurations #0 #2 { \music }
  }
}
```
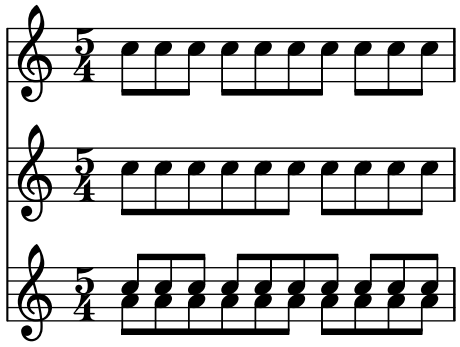


Définition de règles de ligature pour la partition

Les règles de ligature définies au niveau du contexte `Score` s'appliqueront à toutes les portées. Il est toutefois possible de moduler au niveau `Staff` ou `Voice` :

```
\relative c'' {
  \time 5/4
  % Set default beaming for all staves
  \set Score.baseMoment = #(ly:make-moment 1/8)
  \set Score.beatStructure = 3,4,3
  <<
    \new Staff {
      c8 c c c c c c c c c
    }
    \new Staff {
      % Modify beaming for just this staff
      \set Staff.beatStructure = 6,4
      c8 c c c c c c c c c
    }
    \new Staff {
      % Inherit beaming from Score context
      <<
        {
          \voiceOne
          c8 c c c c c c c c c
        }
```

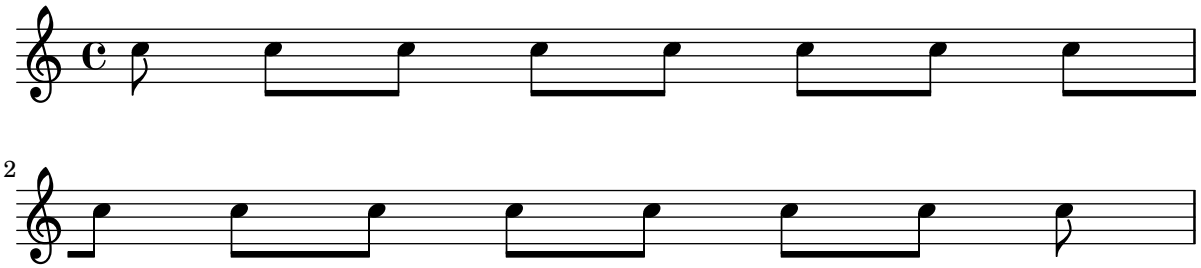
```
        % Modify beaming for this voice only
        \new Voice {
          \voiceTwo
          \set Voice.beatStructure = 6,4
          a8 a a a a a a a a a
        }
      >>
    }
  >>
}
```



Ligature au moment d'un saut de ligne

Il est normalement impensable qu'un saut de ligne tombe au milieu d'une ligature. LilyPond permet néanmoins de l'obtenir.
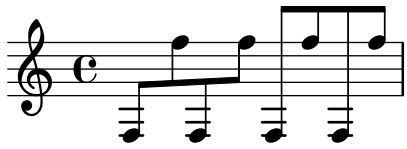
```
\relative c'' {
  \override Beam.breakable = ##t
  c8 c[ c] c[ c] c[ c] c[ \break
  c8] c[ c] c[ c] c[ c] c
}
```



Ligature et directions de hampe inversées

LilyPond insère automatiquement des ligatures coudées – certaines hampes vers le haut, d'autres vers le bas – lorsqu'il détecte un intervalle important entre des têtes de notes. Ce comportement peut être changé par l'intermédiaire de l'objet `auto-knee-gap` – défini par défaut à 5,5 espaces, plus la largeur et la pente de la ligature en question.

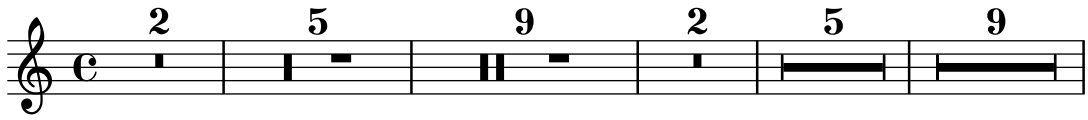
```
{
  f8 f''8 f8 f''8
  \override Beam.auto-knee-gap = #6
  f8 f''8 f8 f''8
}
```



Modifier l'apparence d'un silence multimesure

Dans le cas où ce silence dure moins de dix mesures, LilyPond imprime sur la portée des « silences d'église » – *Kirchenpause* en allemand – et qui sont une simple suite de rectangles. La propriété `expand-limit` permet d'obtenir un silence unique :
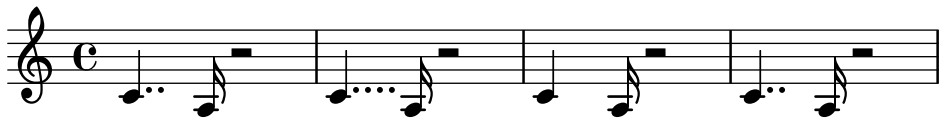
```
\relative c'' {
  \compressMMRests {
    R1*2 | R1*5 | R1*9
    \override MultiMeasureRest.expand-limit = #3
    R1*2 | R1*5 | R1*9
  }
}
```



Spécification du nombre de points d'augmentation d'une note

Le nombre de points d'augmentation affectés à une note en particulier peut se modifier indépendamment des points placés après la note.

```
\relative c' {
  c4.. a16 r2 |
  \override Dots.dot-count = #4
  c4.. a16 r2 |
  \override Dots.dot-count = #0
  c4.. a16 r2 |
  \revert Dots.dot-count
  c4.. a16 r2 |
}
```
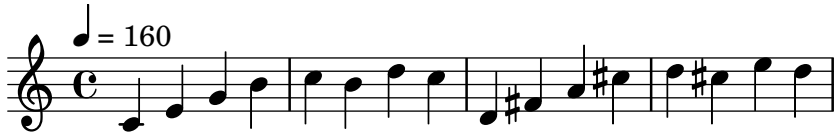


Changement de tempo sans indication sur la partition

Vous pouvez indiquer un changement de tempo pour le fichier MIDI sans pour autant l'imprimer. Il suffit alors de le rendre invisible aux musiciens.

```
\score {
  \new Staff \relative c' {
    \tempo 4 = 160
    c4 e g b
    c4 b d c
    \set Score.tempoHideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
```

```
  \layout { }
  \midi { }
}
```

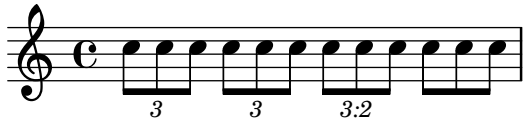


Modifier l'apparence du chiffre de n-olet

LilyPond imprime par défaut le numérateur de la fraction fournie en argument à la commande `\tuplet` du côté du crochet de n-olet.

Il est toutefois possible d'imprimer la fraction entière *num*:*den*, voire de ne rien imprimer du tout.

```
\relative c'' {
  \tuplet 3/2 { c8 c c }
  \tuplet 3/2 { c8 c c }
  \override TupletNumber.text = #tuplet-number::calc-fraction-text
  \tuplet 3/2 { c8 c c }
  \omit TupletNumber
  \tuplet 3/2 { c8 c c }
}
```



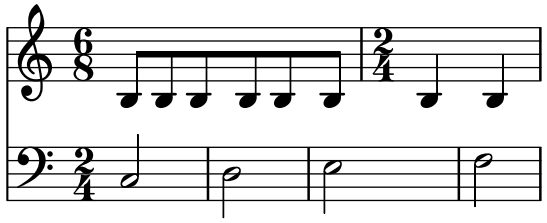
Modification de la métrique au sein d'un fragment polymétrique à l'aide de `\scaleDurations`

La propriété `measureLength`, conjointement avec `measurePosition`, détermine l'endroit où une barre de mesure est requise. L'utilisation d'un `\scaleDurations` a l'inconvénient que l'échelonnement des durées rend compliqués les changements de métrique. Il est nécessaire en pareil cas de définir manuellement `measureLength`, avec un appel à `ly:make-moment`. Le second argument devra être identique au deuxième argument de `\scaleDurations`.

```
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Default_bar_line_engraver"
  }
  \context {
    \Staff
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  }
}

<<
  \new Staff {
```

```
    \scaleDurations 8/5 {
      \time 6/8
      \set Timing.measureLength = #(ly:make-moment 6/5)
      b8 b b b b b
      \time 2/4
      \set Timing.measureLength = #(ly:make-moment 4/5)
      b4 b
    }
  }
  \new Staff {
    \clef bass
    \time 2/4
    c2 d e f
  }
>>
```

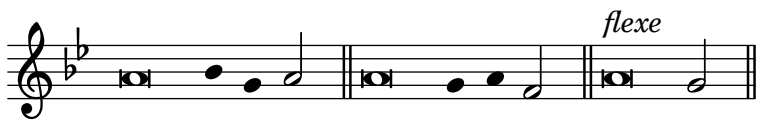


Notation pour psalmodie

Ce style de notation permet d'indiquer la mélodie d'une psalmodie lorsque les strophes sont de longueur inégale.

```
stemOff = \hide Staff.Stem
stemOn  = \undo \stemOff

\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \bar "||"
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \bar "||"
    \stemOff a'\breve^\markup { \italic flexe }
    \stemOn g'2 \bar "||"
  }
}
```
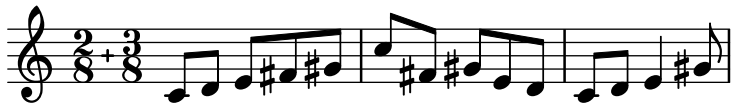


Métrique décomposée

Des métriques telles que « 5/8 » peuvent s'interpréter sous une forme décomposée – « 3/8 + 2/8 » par exemple – qui combine plusieurs métriques. LilyPond est capable de rendre ce type de

notation, plus facile à lire et à interpréter, en imprimant cette métrique composite et en adaptant les règles de ligature automatique en conséquence.

```
\relative c' {
  \compoundMeter #'((2 8) (3 8))
  c8 d e fis gis
  c8 fis, gis e d
  c8 d e4 gis8
}
```



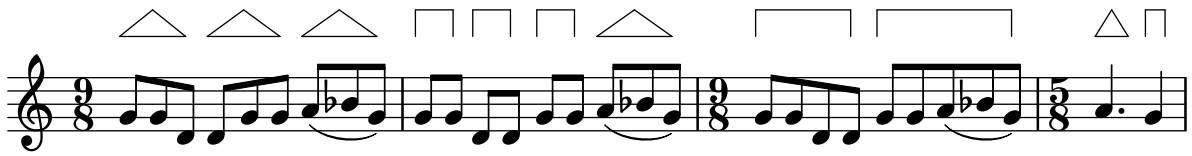
Signes de direction, signes de sous-groupe

Les règles de ligature par mesure sont gérées par la propriété de contexte `beatStructure`. Ses valeurs par défaut sont répertoriées, par métrique, dans le fichier `scm/time-signature-settings.scm`. Elles sont modifiables grâce à la commande `\set`.

La fonction Scheme `set-time-signature` permet quant à elle de définir à la fois la métrique et la pulsation. Celle-ci prend trois arguments : le nombre de pulsations, la durée de la pulsation et le regroupement des pulsations dans la mesure. `\time` et `set-time-signature` s'appliquent tous deux au contexte `Timing` ; ils ne redéfiniront donc pas les valeurs de `beatStructure` ou `baseMoment` lorsqu'elles sont modifiées dans un contexte de niveau inférieur comme `Voice` par exemple.

Si l'on fait appel au `Measure_grouping_engraver`, la fonction `set-time-signature` créera aussi des symboles `MeasureGrouping`. Ces symboles aident à la lecture des œuvres modernes à la rythmique complexe. Dans l'exemple qui suit, la mesure à 9/8 est divisée en 2, 2, 2 et 3, alors que la mesure à 5/8 répond aux règles par défaut contenues dans le fichier `scm/time-signature-settings.scm`.

```
\score {
  \new Voice \relative c'' {
    \time 9/8
    g8 g d d g g a( bes g) |
    \set Timing.beatStructure = 2,2,2,3
    g8 g d d g g a( bes g) |
    \time 4,5 9/8
    g8 g d d g g a( bes g) |
    \time 5/8
    a4. g4 |
  }
  \layout {
    \context {
      \Staff
      \consists "Measure_grouping_engraver"
    }
  }
}
```



Adaptation de l'alignement des numéros de mesure

L'alignement des numéros de mesure par la gauche peut générer des problèmes de chevauchement, notamment avec les crochets de regroupement de portées.

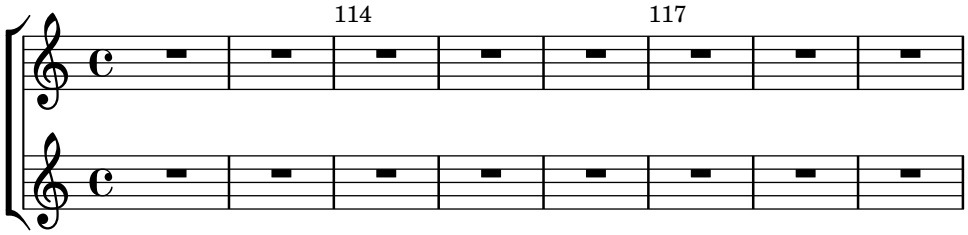
L'exemple ci-dessous offre une solution en alignant par la droite le numéro de la mesure suivant un saut de ligne.

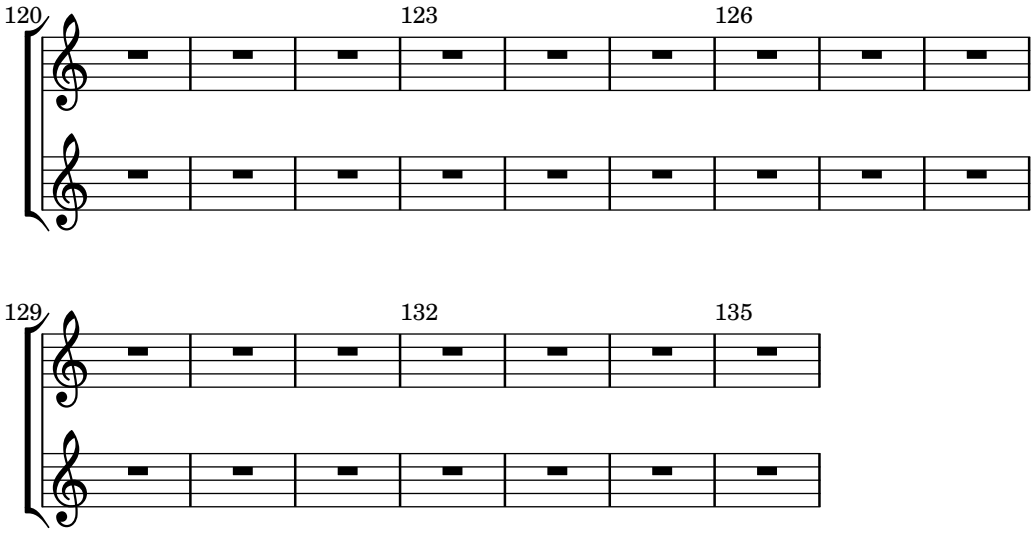
```
consistentlyLeftAlignedBarNumbers = {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \override Score.BarNumber.self-alignment-X =
    #(lambda (grob)
       (let ((break-dir (ly:item-break-dir grob)))
         (if (= break-dir RIGHT) RIGHT LEFT)))
}

\new ChoirStaff <<
  \new Staff {
    \relative c' {
      \set Score.barNumberVisibility = #(every-nth-bar-number-visible 3)
      \bar ""
      \consistentlyLeftAlignedBarNumbers

      \set Score.currentBarNumber = #112
      \repeat unfold 8 { R1 }
      \break
      \repeat unfold 9 { R1 }
      \break
      \repeat unfold 7 { R1 }
    }
  }
  \new Staff {
    \relative c' {
      \repeat unfold 24 { R1 }
    }
  }
>>

\layout {
  indent = #0
  ragged-right = ##t
  ragged-last = ##t
}
```





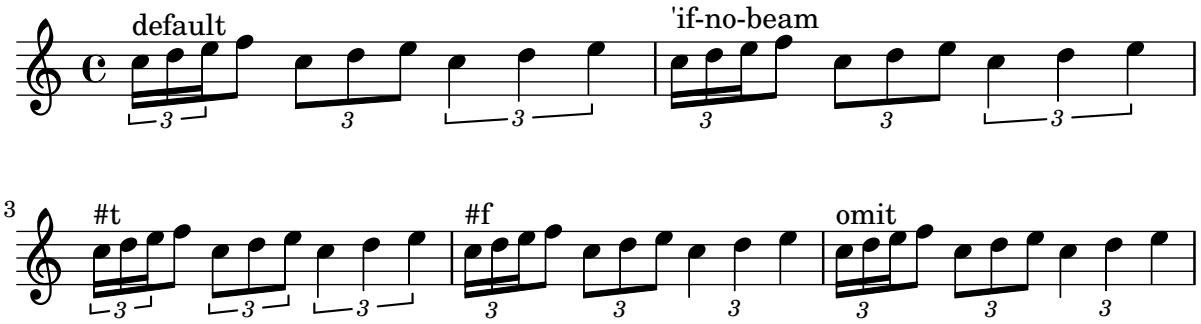
Contrôle de l'impression des crochets de n-olet

Selon la tradition, les crochets indicateurs de n-olet sont toujours imprimés, sauf dans le cas où ils seraient de la même longueur qu'une ligature.

LilyPond permet, au travers de la propriété `bracket-visibility`, de contôler précisément leur affichage : déterminée à `#t`, ils seront toujours imprimés ; `#f` permet de ne jamais les imprimer – donc omettre l'objet `TupletBracket` –, et `#'if-no-beam` les imprimera en l'absence de ligature (comportement par défaut).

```
music = \relative c'' {
  \tuplet 3/2 { c16[ d e } f8]
  \tuplet 3/2 { c8 d e }
  \tuplet 3/2 { c4 d e }
}

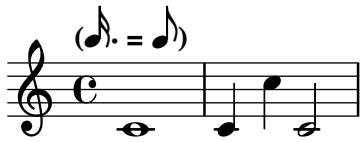
\new Voice {
  \relative c' {
    << \music s4^"default" >>
    \override TupletBracket.bracket-visibility = #'if-no-beam
    << \music s4^"'if-no-beam" >>
    \override TupletBracket.bracket-visibility = ##t
    << \music s4^"#t" >>
    \override TupletBracket.bracket-visibility = ##f
    << \music s4^"#f" >>
    \omit TupletBracket
    << \music s4^"omit" >>
  }
}
```



Création d'une indication métronomique sous forme d'étiquette

Vous pouvez créer des indications de tempo sous la forme d'étiquettes textuelles – des objets *markup* –, notamment des équivalences. Cependant, elles n'apparaîtront pas dans le fichier MIDI.

```
\relative c' {
  \tempo \markup {
    \concat {
      (
      \smaller \general-align #Y #DOWN \note {16.} #1
      " = "
      \smaller \general-align #Y #DOWN \note {8} #1
      )
    }
  }
  c1
  c4 c' c,2
}
```

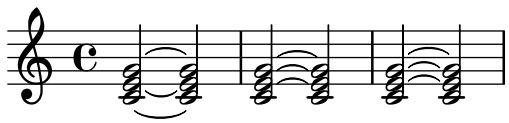


Dessin à main levée de liaisons de tenue

Il est possible de graver manuellement les liaisons de tenue, en modifiant la propriété `tie-configuration`. Pour chaque paire, le premier nombre indique la distance à la portée, en espaces de portée, et le second la direction (1 pour haut, −1 pour bas).

Notez bien que LilyPond fait la distinction, au niveau du premier nombre, entre valeur exacte et valeur inexacte. Dans le cas d'une valeur exacte – autrement dit un entier ou une fraction comme `(/ 4 5)` – celle-ci servira de position verticale brute, ensuite afinée par LilyPond de sorte à éviter les lignes de la portée. Dans le cas d'une valeur inexacte, tel un nombre à virgule flottante, c'est elle qui servira à positionner verticalement, sans ajustement.

```
\relative c' {
  <c e g>2~ <c e g>
  \override TieColumn.tie-configuration =
    #'((0.0 . 1) (-2.0 . 1) (-4.0 . 1))
  <c e g>2~ <c e g>
  \override TieColumn.tie-configuration =
    #'((0 . 1) (-2 . 1) (-4 . 1))
  <c e g>2~ <c e g>
}
```
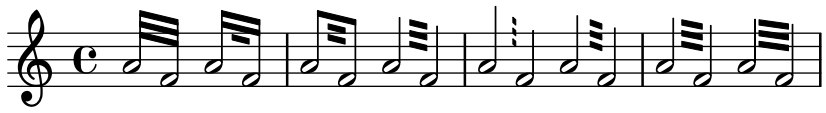


Impression de trémolo avec ligature flottante

Lorsque la durée totale d'un trémolo est inférieure à la noire, égale une blanche, ou bien entre une blanche et une ronde, il est d'usage que toutes les ligatures soient en contact avec les hampes. Certains styles de gravure font cependant apparaître des ligatures détachées, centrées

entre les hampes. Pour ce type de trémolo, le nombre de hampes flottantes se gère au travers de la propriété `gap-count` de l'objet `Beam`, et l'écart entre ligature et hampe se définit par la propriété `gap`.

```
\relative c'' {
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #1
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #2
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #3
  \repeat tremolo 8 { a32 f }

  \override Beam.gap-count = #3
  \override Beam.gap = #1.33
  \repeat tremolo 8 { a32 f }
  \override Beam.gap = #1
  \repeat tremolo 8 { a32 f }
  \override Beam.gap = #0.67
  \repeat tremolo 8 { a32 f }
  \override Beam.gap = #0.33
  \repeat tremolo 8 { a32 f }
}
```

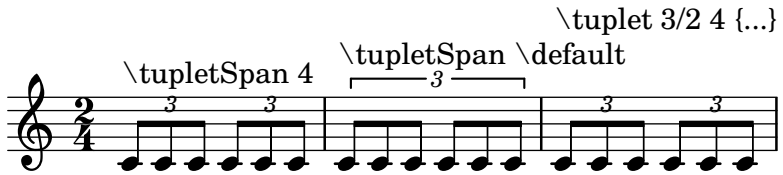


Plusieurs triolets avec une seule commande `\tuplet`

La propriété `tupletSpannerDuration` spécifie la longueur voulue de chaque crochet. Avec elle, vous pouvez faire plusieurs n-olets en ne tapant `\tuplet` qu'une fois, ce qui évite une longue saisie.

Il existe différents moyens de définir `tupletSpannerDuration`. La commande `\tupletSpan` lui affecte une durée arbitraire qui sera réinitialisée dès l'intervention d'une durée à `\default`. Vous pouvez aussi opter pour fournir un argument supplémentaire à la commande `\tuplet`.

```
\relative c' {
  \time 2/4
  \tupletSpan 4
  \tuplet 3/2 { c8^"\\tupletSpan 4" c c c c c }
  \tupletSpan \default
  \tuplet 3/2 { c8^"\\tupletSpan \\default" c c c c c }
  \tuplet 3/2 4 { c8^"\\tuplet 3/2 4 {...}" c c c c c }
}
```



Crochet rectiligne et débordement de ligature

En combinant `stemLeftBeamCount`, `stemRightBeamCount` et des paires de `[]`, vous pourrez obtenir des crochets rectilignes et des ligatures qui débordent à leurs extrémités.

Pour des crochets rectilignes à droite sur des notes isolées, il suffit d'ajouter une paire d'indicateurs de ligature `[]` et de déterminer `stemLeftBeamCount` à zéro, comme dans l'exemple 1.

Pour des crochets rectiligne à gauche, c'est `stemRightBeamCount` qu'il faudra déterminer (exemple 2).

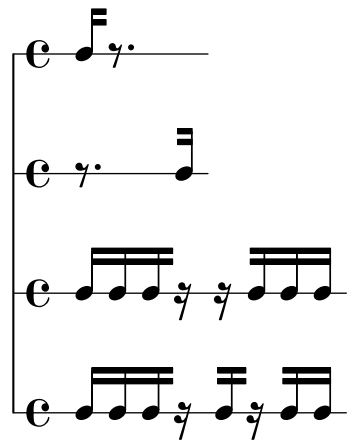
Pour que les barres de ligature débordent sur la droite, `stemRightBeamCount` doit avoir une valeur positive ; pour un débordement à gauche, c'est sur `stemLeftBeamCount` qu'il faut jouer. Tout ceci est illustré par l'exemple 3.

Il est parfois judicieux, lorsqu'une note est encadrée de silences, de l'affubler de crochets rectilignes de part et d'autre. L'exemple 4 montre qu'il suffit d'adjoindre à cette note un `[]`.

(Notez bien que `\set stemLeftBeamCount` sera toujours synonyme de `\once \set`. Autrement dit, la détermination des ligatures n'est pas « permanente » ; c'est la raison pour laquelle les crochets du `16[]` isolé du dernier exemple n'ont rien à voir avec le `\set` indiqué deux notes auparavant.)

```
\score {
  <<
    % Example 1
    \new RhythmicStaff {
      \set stemLeftBeamCount = #0
      c16[]
      r8.
    }
    % Example 2
    \new RhythmicStaff {
      r8.
      \set stemRightBeamCount = #0
      16[]
    }
    % Example 3
    \new RhythmicStaff {
      16 16
      \set stemRightBeamCount = #2
      16 r r
      \set stemLeftBeamCount = #2
      16 16 16
    }
    % Example 4
    \new RhythmicStaff {
      16 16
      \set stemRightBeamCount = #2
      16 r16
      16[]
      r16
      \set stemLeftBeamCount = #2
      16 16
    }
  >>
```

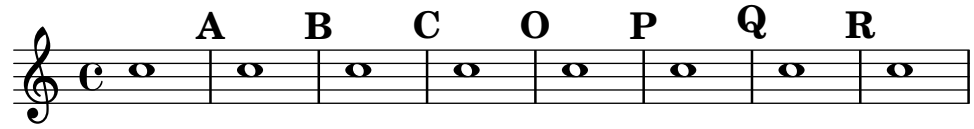
```
}
```



Fixation arbitraire du numéro de repère de départ

Voici comment initialiser arbitrairement une indication automatique de repère, qu'elle soit alphabétique ou numérique :

```
\relative c'' {
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1 \mark #14
  c1 \mark \default
  c1 \mark \default
  c1 \mark \default
  c1
}
```



Génération de crochets personnalisés

Une fonction Scheme personnalisée permet de redéfinir la propriété `stencil` de l'objet `Flag`, de sorte à modifier le glyphe utilisé pour les crochets de croche.

```
#(define-public (weight-flag grob)
   (let* ((stem-grob (ly:grob-parent grob X))
          (log (- (ly:grob-property stem-grob 'duration-log) 2))
          (is-up? (eqv? (ly:grob-property stem-grob 'direction) UP))
          (yext (if is-up? (cons (* log -0.8) 0) (cons 0 (* log 0.8))))
          (flag-stencil (make-filled-box-stencil '(-0.4 . 0.4) yext))
          (stroke-style (ly:grob-property grob 'stroke-style))
          (stroke-stencil (if (equal? stroke-style "grace")
                              (make-line-stencil 0.2 -0.9 -0.4 0.9 -0.4)
                              empty-stencil)))
     (ly:stencil-add flag-stencil stroke-stencil)))
```

```
% Create a flag stencil by looking up the glyph from the font
#(define (inverted-flag grob)
   (let* ((stem-grob (ly:grob-parent grob X))
          (dir (if (eqv? (ly:grob-property stem-grob 'direction) UP) "d" "u"))
          (flag (retrieve-glyph-flag "" dir "" grob))
          (line-thickness (ly:staff-symbol-line-thickness grob))
          (stem-thickness (ly:grob-property stem-grob 'thickness))
          (stem-width (* line-thickness stem-thickness))
          (stroke-style (ly:grob-property grob 'stroke-style))
          (stencil (if (null? stroke-style)
                       flag
                       (add-stroke-glyph flag stem-grob dir stroke-style "")))
          (rotated-flag (ly:stencil-rotate-absolute stencil 180 0 0)))
     (ly:stencil-translate rotated-flag (cons (- (/ stem-width 2)) 0))))
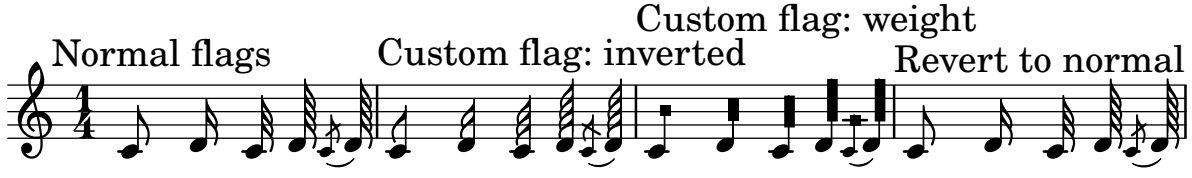
snippetexamplenotes =
{
  \autoBeamOff c'8 d'16 c'32 d'64 \acciaccatura {c'8} d'64
}

{
  \override Score.RehearsalMark.self-alignment-X = #LEFT
  \time 1/4
  \mark "Normal flags"
  \snippetexamplenotes

  \mark "Custom flag: inverted"
  \override Flag.stencil = #inverted-flag
  \snippetexamplenotes

  \mark "Custom flag: weight"
  \override Flag.stencil = #weight-flag
  \snippetexamplenotes

  \mark "Revert to normal"
  \revert Flag.stencil
  \snippetexamplenotes
}
```

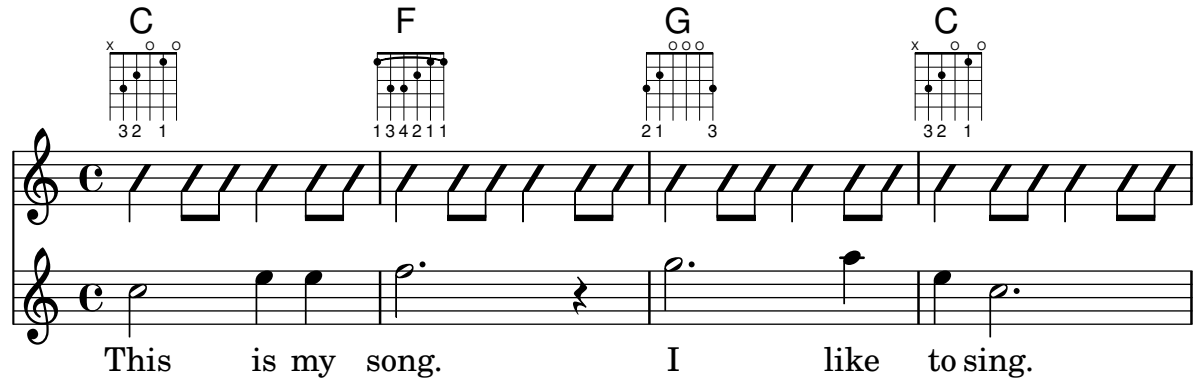


Rythmique et guitare

En matière de notation pour guitare, il arrive que soient indiqués les « coups de gratte » en plus de la mélodie, grilles d'accords et diagrammes de tablature.

```
\include "predefined-guitar-fretboards.ly"
<<
  \new ChordNames {
```

```
    \chordmode {
      c1 | f | g | c
    }
  }
  \new FretBoards {
    \chordmode {
      c1 | f | g | c
    }
  }
  \new Voice \with {
    \consists "Pitch_squash_engraver"
  } {
    \relative c'' {
      \improvisationOn
      c4 c8 c c4 c8 c
      f4 f8 f f4 f8 f
      g4 g8 g g4 g8 g
      c4 c8 c c4 c8 c
    }
  }
  \new Voice = "melody" {
    \relative c'' {
      c2 e4 e4
      f2. r4
      g2. a4
      e4 c2.
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
      This is my song.
      I like to sing.
    }
  }
>>
```



Indication personnalisée d'une polymétrie complexe

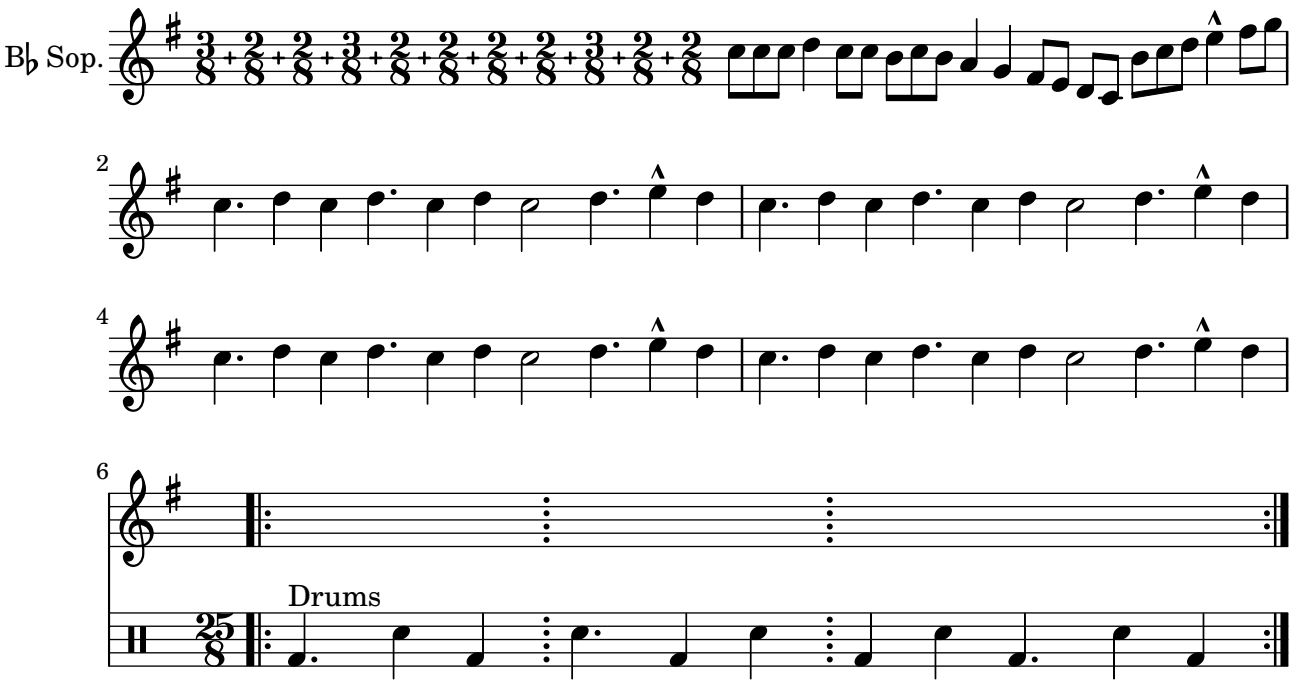
Bien que la métrique complexe ne soit pas l'élément primordial de cet exemple, elle permet d'indiquer la pulsation de cette pièce qui, par ailleurs, constitue le canevas d'une chanson traditionnelle des Balkans.

```
melody = \relative c'' {
  \key g \major
  \compoundMeter #'((3 8) (2 8) (2 8) (3 8) (2 8) (2 8)
                    (2 8) (2 8) (3 8) (2 8) (2 8))
  c8 c c d4 c8 c b c b a4 g fis8 e d c b' c d e4-^ fis8 g \break
  c,4. d4 c4 d4. c4 d c2 d4. e4-^ d4
  c4. d4 c4 d4. c4 d c2 d4. e4-^ d4 \break
  c4. d4 c4 d4. c4 d c2 d4. e4-^ d4
  c4. d4 c4 d4. c4 d c2 d4. e4-^ d4 \break
}

drum = \new DrumStaff \drummode {
  \bar ".|:" bd4.^\markup { Drums } sn4 bd \bar ";" sn4.
  bd4 sn \bar ";" bd sn bd4. sn4 bd \bar ":|."
}

  \new Staff \with {
    instrumentName = \markup { \concat { "B" \flat " Sop." } }
  }

{
  \melody
  \drum
}
```
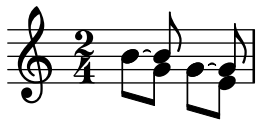


Recours à la propriété `'transparent` pour rendre des objets invisibles

Une modification de la propriété `transparent` permet d'imprimer un objet « à l'encre sympathique » : l'objet n'est pas affiché bien que tous ses attributs soient préservés. L'objet en question occupe donc sa place, est pris en compte lors de collisions, et peut se voir attaché liaisons ou ligatures.

Cet exemple illustre la manière de connecter deux voix par une liaison de prolongation. Les liaisons de prolongation ne peuvent normalement intervenir que dans la même voix. Dès lors que la liaison est entamée dans une autre voix et que la première hampe ascendante est rendue transparente dans cette même voix, la liaison semble passer d'une voix à l'autre.

```
\relative {
  \time 2/4
  <<
    {
      \once \hide Stem
      \once \override Stem.length = #8
      b'8 ~ 8\noBeam
      \once \hide Stem
      \once \override Stem.length = #8
      g8 ~ 8\noBeam
    }
    \\
    {
      b8 g g e
    }
  >>
}
```
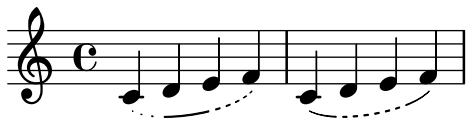


Dessin d'une liaison d'articulation au trait discontinu

Grâce à la propriété `dash-definition`, une liaison d'articulation peut être formée de traits discontinus variables. `dash-definition` se compose d'une liste de `segments-discontinus` (*dash-elements*). Chaque `segment-discontinu` contient une liste de paramètres qui déterminent le comportement du trait pour une section de la liaison.

Cette liaison se définit selon le paramètre de Bézier `t` qui est compris entre 0 (l'extrémité gauche de la liaison) et 1 (l'extrémité droite de la liaison). Chaque `segment-discontinu` se composera selon la liste `(t-début t-fin segment-style segment-taille)`. La portion de liaison allant de `t-début` à `t-fin` aura un trait `segment-style` de longueur `segment-taille`. `segment-taille` est exprimé en espace de portée ; un `segment-style` à 1 donnera un trait plein.
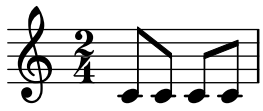
```
\relative c' {
  \once \override
    Slur.dash-definition = #'((0 0.3 0.1 0.75)
                              (0.3 0.6 1 1)
                              (0.65 1.0 0.4 0.75))
  c4( d e f)
  \once \override
    Slur.dash-definition = #'((0 0.25 1 1)
                              (0.3 0.7 0.4 0.75)
                              (0.75 1.0 1 1))
  c4( d e f)
}
```



Contrôle manuel du positionnement des ligatures

Le positionnement et la pente des ligatures peuvent être contrôlés manuellement à l'aide d'une adaptation de la propriété `positions` de l'objet `Beam`.

```
\relative c' {
  \time 2/4
  % from upper staff-line (position 2) to center (position 0)
  \override Beam.positions = #'(2 . 0)
  c8 c
  % from center to one above center (position 1)
  \override Beam.positions = #'(0 . 1)
  c8 c
}
```

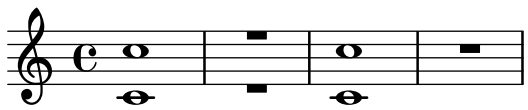


Fusion de silences multimesures dans une partie polyphonique

Dans le cadre d'un portée polyphonique, les silences multimesures sont positionnés différemment selon la voix à laquelle ils appartiennent. Le réglage suivant permet néanmoins de les imprimer sur une même ligne.

```
normalPos = \revert MultiMeasureRest.direction

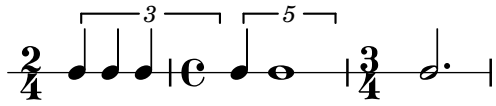
{
  <<
    {
      c''1
      R1
      c''1
      \normalPos
      R1
    }
    \\
    {
      c'1
      R1
      c'1
      \normalPos
      R1
    }
  >>
}
```



Modification de la longueur d'un crochet de n-olet

Les crochets indiquant un n-olet peuvent être prolongés jusqu'à un élément de rupture ou jusqu'à la note suivante. LilyPond termine un crochet de n-olet sur la droite de sa dernière tête de note par défaut ; un crochet de pleine longueur s'étendra plus avant, soit jusqu'à la note suivante et en traversant tous les éléments non rythmiques, soit sur tout l'espace précédant le prochain élément de notation, que ce soit une clef, une métrique, une armure ou une autre note. L'exemple suivant illustre la manière d'activer ces deux fonctionnalités.
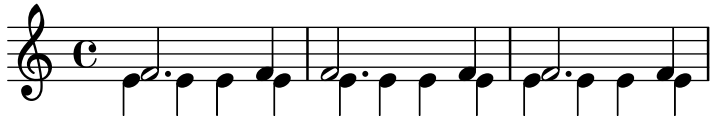
```
\new RhythmicStaff {
  % Set tuplets to be extendable...
  \set tupletFullLength = ##t
  % ...to cover all items up to the next note
  \set tupletFullLengthNote = ##t
  \time 2/4
  \tuplet 3/2 { c4 4 4 }
  % ...or to cover just whitespace
  \set tupletFullLengthNote = ##f
  \time 4/4
  \tuplet 5/4 { 4 1 }
  \time 3/4
  2.
}
```



Déplacement des notes pointées dans une polyphonie

Une note pointée appartenant à la voix supérieure d'une portée polyphonique sera par défaut décalée vers la droite afin d'éviter les collisions avec les autres voix. Ce comportement peut être outrepassé à l'aide de la propriété `prefer-dotted-right` de `NoteCollision`.

```
\new Staff \relative c' <<
  {
    f2. f4
    \override Staff.NoteCollision.prefer-dotted-right = ##f
    f2. f4
    \override Staff.NoteCollision.prefer-dotted-right = ##t
    f2. f4
  }
  \\
  { e4 e e e e e e e e e e }
>>
```

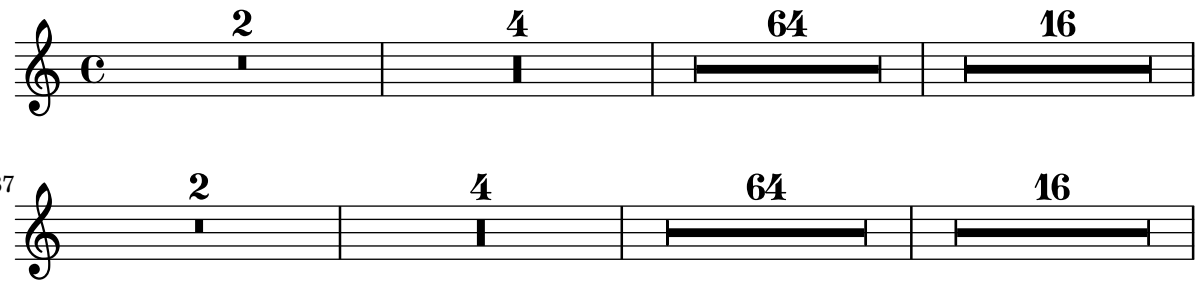


Contrôle de la taille d'un silence multimesure

Les silences multimesure ont une largeur relative à leur durée totale, contrôlée par `MultiMeasureRest.space-increment`. Sa valeur par défaut est fixée à `2.0`.

```
\relative c' {
```

```
  \compressEmptyMeasures
  R1*2 R1*4 R1*64 R1*16
  \override Staff.MultiMeasureRest.space-increment = 2.5
  R1*2 R1*4 R1*64 R1*16
}
```

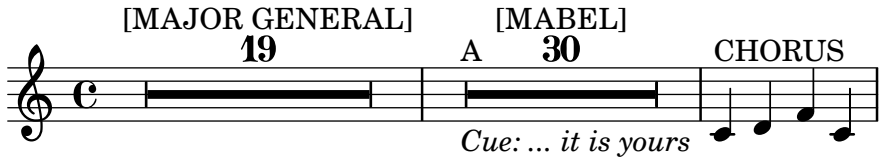


Ajout de texte à un silence multimesure

Lorsque du texte est attaché à un silence multimesure, il sera centré dans la mesure, au-dessus ou en dessous de la portée. Afin d'étirer la mesure dans le cas où ce texte est relativement long, il suffit d'insérer un accord vide auquel on attache le texte en question, avant le silence multimesure.

Le texte attaché à un silence invisible sera aligné sur la gauche de là où serait positionnée la note dans la mesure. Cependant, si la taille de la mesure est déterminée par la longueur du texte, il apparaîtra comme centré.

```
\relative c' {
  \compressMMRests {
    \textLengthOn
    <>^\markup { [MAJOR GENERAL] }
    R1*19
    <>_\markup { \italic { Cue: ... it is yours } }
    <>^\markup { A }
    R1*30^\markup { [MABEL] }
    \textLengthOff
    c4^\markup { CHORUS } d f c
  }
}
```

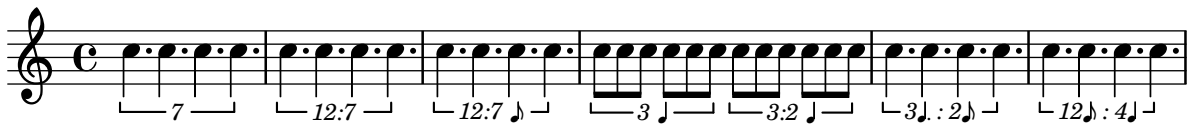


N-olets au chiffrage inhabituel

LilyPond sait aussi gérer des n-olets dont le chiffrage imprimé ne correspond pas exactement à la fraction de mesure à laquelle ils se réfèrent, tout comme ceux auxquels une valeur de note vient en complément du chiffre.

```
\relative c'' {
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-denominator-text 7)
  \tuplet 3/2  { c4. c4. c4. c4. }
```

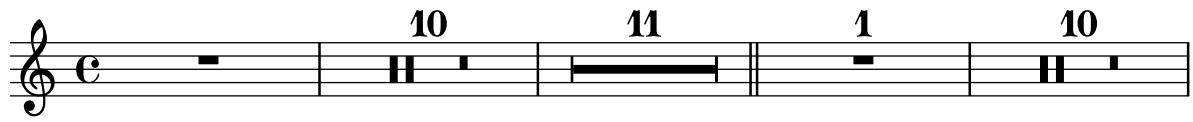
```
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-tuplet-fraction-text 12 7)
  \tuplet 3/2  { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      (tuplet-number::non-default-tuplet-fraction-text 12 7)
      (ly:make-duration 3 0))
  \tuplet 3/2  { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      tuplet-number::calc-denominator-text
      (ly:make-duration 2 0))
  \tuplet 3/2  { c8 c8 c8 c8 c8 c8 }
  \once \override TupletNumber.text =
    #(tuplet-number::append-note-wrapper
      tuplet-number::calc-fraction-text
      (ly:make-duration 2 0))
  \tuplet 3/2  { c8 c8 c8 c8 c8 c8 }
  \once \override TupletNumber.text =
    #(tuplet-number::fraction-with-notes
      (ly:make-duration 2 1) (ly:make-duration 3 0))
  \tuplet 3/2  { c4. c4. c4. c4. }
  \once \override TupletNumber.text =
    #(tuplet-number::non-default-fraction-with-notes 12
      (ly:make-duration 3 0) 4 (ly:make-duration 2 0))
  \tuplet 3/2  { c4. c4. c4. c4. }
}
```

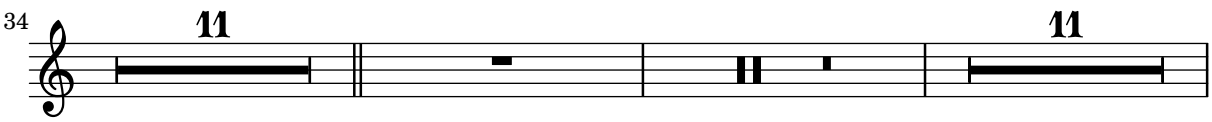


Dénombrer une unique mesure de silence

Les silences multimesures affichent leur longueur sauf s'il n'y a qu'une seule mesure. Ceci peut se modifier en réglant `restNumberThreshold`.

```
{
  \compressEmptyMeasures
  R1 R1*10 R1*11 \bar "||"
  \set restNumberThreshold = 0
  R1 R1*10 R1*11 \bar "||"
  \set restNumberThreshold = 10
  R1 R1*10 R1*11
}
```





PartCombine et autoBeamOff

La fonction `\autoBeamOff` dans le cadre d'un `\partCombine` agit de façon bien particulière ; c'est pourquoi il vaut mieux tout d'abord recourir à

```
\set Staff.autobeaming = ##f
```
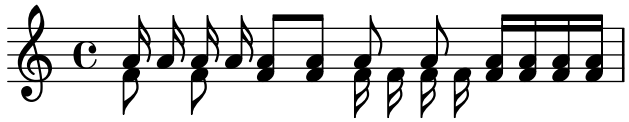
pour désactiver les ligatures automatiques pour l'ensemble de la portée concernée.

L'instruction `\partCombine` fonctionne apparemment sur la base de trois voix : solo hampes montantes, solo hampes descendantes et ensemble hampes montantes.

Lorsque `\autoBeamOff` apparaît dans le premier argument de la combinaison, il s'applique à la voix active à ce moment précis, qu'il s'agisse du solo hampes montantes ou du combiné hampes montantes. Lorsqu'elle est introduite dans le second argument, la commande `\autoBeamOff` s'appliquera au solo hampes descendantes.

Vous devrez donc, afin que `\autoBeamOff` soit pleinement opérationnel dans le cadre d'un `\partCombine`, l'introduire aux **trois** niveaux.

```
{
  %\set Staff.autoBeaming = ##f % turns off all autobeaming
  \partCombine
  {
    \autoBeamOff % applies to split up stems
    \repeat unfold 4 a'16
    %\autoBeamOff % applies to combined up stems
    \repeat unfold 4 a'8
    \repeat unfold 4 a'16
  }
  {
    \autoBeamOff % applies to down stems
    \repeat unfold 4 f'8
    \repeat unfold 8 f'16 |
  }
}
```



Exemple pour percussions

Ce court exemple est tiré de « L'histoire du Soldat » de Stravinsky.

```
#(define mydrums '((bassdrum   default #t  4)
                   (snare      default #t -4)
                   (tambourine default #t  0)))

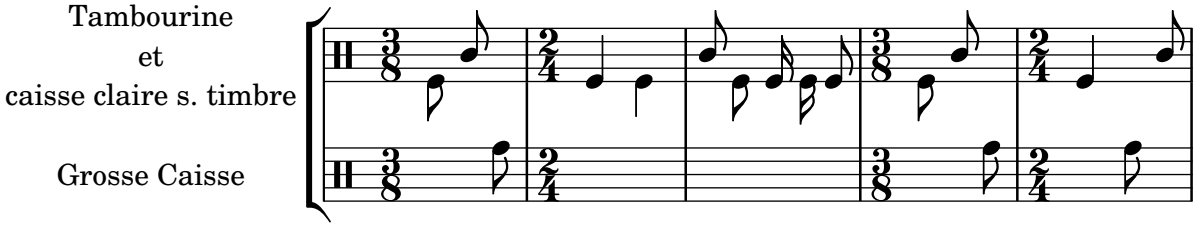
global = {
  \time 3/8 s4.
  \time 2/4 s2*2
  \time 3/8 s4.
  \time 2/4 s2
}
```

```
drumsA = {
  \context DrumVoice <<
    { \global }
    { \drummode {
        \autoBeamOff
        \stemDown sn8 \stemUp tamb s8 |
        sn4 \stemDown sn4 |
        \stemUp tamb8 \stemDown sn8 \stemUp sn16 \stemDown sn \stemUp sn8 |
        \stemDown sn8 \stemUp tamb s8 |
        \stemUp sn4 s8 \stemUp tamb
      }
    }
  >>
}

drumsB = {
  \drummode {
    s4 bd8 s2*2 s4 bd8 s4 bd8 s8
  }
}

\layout {
  indent = #40
}

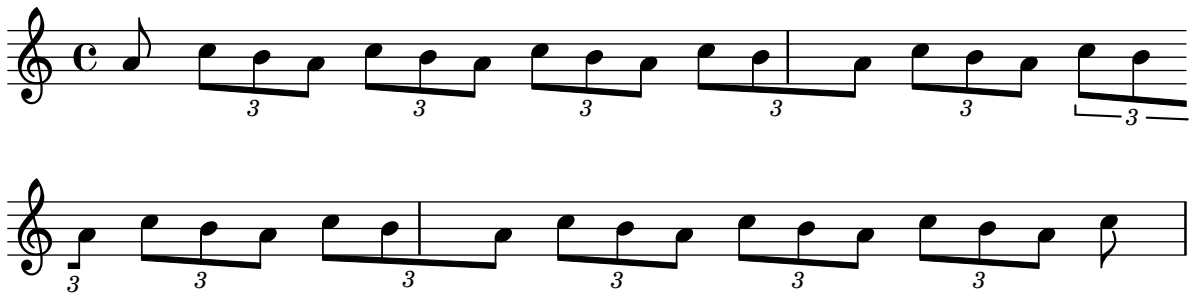
\score {
  \new StaffGroup <<
    \new DrumStaff \with {
      instrumentName = \markup {
        \center-column {
          "Tambourine"
          "et"
          "caisse claire s. timbre"
        }
      }
    drumStyleTable = #(alist->hash-table mydrums)
  }
  \drumsA
  \new DrumStaff \with {
    instrumentName = #"Grosse Caisse"
    drumStyleTable = #(alist->hash-table mydrums)
  }
  \drumsB
  >>
}
```



Saut de ligne au milieu d'un n-olet avec ligature

Cet exemple peu académique démontre comment il est possible d'insérer un saut de ligne dans un n-olet portant une ligature. Ces ligatures doivent toutefois être explicites.

```
\layout {
  \context {
    \Voice
    % Permit line breaks within tuplets
    \remove "Forbid_line_break_engraver"
    % Allow beams to be broken at line breaks
    \override Beam.breakable = ##t
  }
}
\relative c'' {
  a8
  \repeat unfold 5 { \tuplet 3/2 { c[ b a] } }
  % Insert a manual line break within a tuplet
  \tuplet 3/2 { c[ b \bar "" \break a] }
  \repeat unfold 5 { \tuplet 3/2 { c[ b a] } }
  c8
}
```
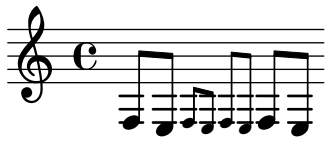


Positionnement des ligatures de notes d'ornement à la hauteur de celles des notes normales

La ligature de notes se trouvant sur des lignes supplémentaires est généralement centrée sur la portée. Les notes d'ornement ayant une hampe raccourcie, leur ligature peut se retrouver en dehors de la portée lorsqu'elles sont sur des lignes supplémentaires. LilyPond permet de rallonger les hampes en pareil cas.

```
\relative c {
  f8[ e]
  \grace {
    f8[ e]
    \override Stem.no-stem-extend = ##f
    f8[ e]
    \revert Stem.no-stem-extend
  }
  f8[ e]
```
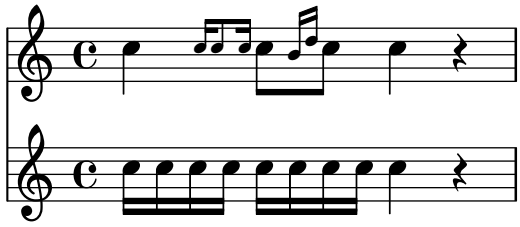
```
}
```



Positionnement des notes d'ornement avec espace flottant

Lorsqu'est activée la propriété `strict-grace-spacing`, l'espacement des notes d'ornement se fera de manière « élastique ». Autrement dit, elles seront décollées de leur note de rattachement : LilyPond commence par espacer les notes normales, puis les ornements sont placés à la gauche de leur note de rattachement.

```
\relative c'' {
  <<
    \override Score.SpacingSpanner.strict-grace-spacing = ##t
    \new Staff \new Voice {
      \afterGrace c4 { c16[ c8 c16] }
      c8[ \grace { b16 d } c8]
      c4 r
    }
    \new Staff {
      c16 c c c c c c c c4 r
    }
  >>
}
```



Positionnement des silences multimesures

Si l'on peut positionner verticalement un silence simple en le rattachant à une note, il n'en va pas de même pour un silence multimesure. Néanmoins, et uniquement dans le cadre de musique polyphonique, les silences multimesures sont positionnés différemment selon qu'ils appartiennent à une voix au numéro pair ou impair. Le positionnement des silences multimesures peut se contrôler ainsi :

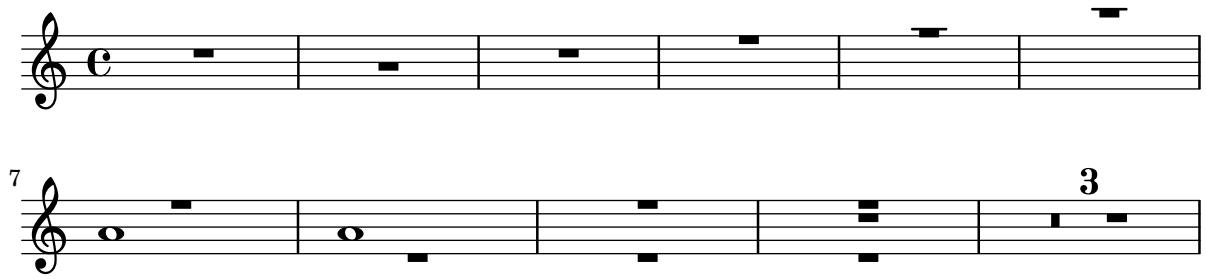
```
\relative c'' {
  % Multi-measure rests by default are set under the fourth line
  R1
  % They can be moved using an override
  \override MultiMeasureRest.staff-position = #-2
  R1
  \override MultiMeasureRest.staff-position = #0
  R1
  \override MultiMeasureRest.staff-position = #2
  R1
```

```
  \override MultiMeasureRest.staff-position = #3
  R1
  \override MultiMeasureRest.staff-position = #6
  R1
  \revert MultiMeasureRest.staff-position
  \break

  % In two Voices, odd-numbered voices are under the top line
  << { R1 } \\ { a1 } >>
  % Even-numbered voices are under the bottom line
  << { a1 } \\ { R1 } >>
  % Multi-measure rests in both voices remain separate
  << { R1 } \\ { R1 } >>

  % Separating multi-measure rests in more than two voices
  % requires an override
  << { R1 } \\ { R1 } \\
     \once \override MultiMeasureRest.staff-position = #0
     { R1 }
  >>

  % Using compressed bars in multiple voices requires another override
  % in all voices to avoid multiple instances being printed
  \compressMMRests
  <<
   \revert MultiMeasureRest.direction
    { R1*3 }
    \\
   \revert MultiMeasureRest.direction
    { R1*3 }
  >>
}
```



Préservation de l'indication de n-olet lors d'un repère final

L'ajout d'une marque-repère en fin de pièce peut entraîner la perte de la dernière indication de n-olet. La désactivation de `TupletBracket.full-length-to-extent` pallie ce problème.
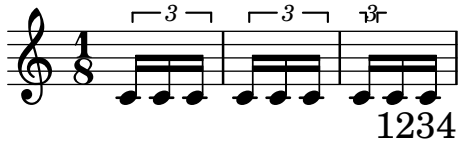
```
% due to issue 2362 a long mark such as
%   \mark "Composed Feb 2007 - Feb 2008"
% cannot be used here.

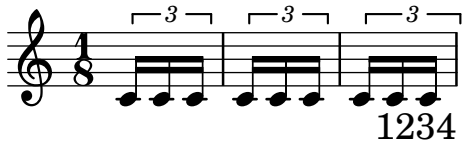
\new Staff {
   \set tupletFullLength = ##t
```

```
   \time 1/8
   \tuplet 3/2 8 { c'16 c' c' c' c' c' c' c' }
   \override Score.RehearsalMark.break-visibility = ##(#t #t #t)
   \override Score.RehearsalMark.direction = #DOWN
   \override Score.RehearsalMark.self-alignment-X = #RIGHT
   \mark "1234"
}

\new Staff {
  \set tupletFullLength = ##t
  \override TupletBracket.full-length-to-extent = ##f

  \time 1/8
   \tuplet 3/2 8 { c'16 c' c' c' c' c' c' c' }
  \override Score.RehearsalMark.break-visibility = ##(#t #t #t)
  \override Score.RehearsalMark.direction = #DOWN
  \override Score.RehearsalMark.self-alignment-X = #RIGHT
   \mark "1234"
}
```

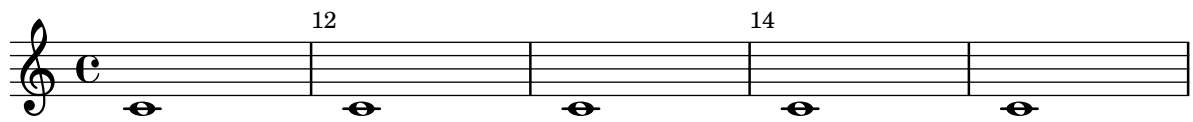


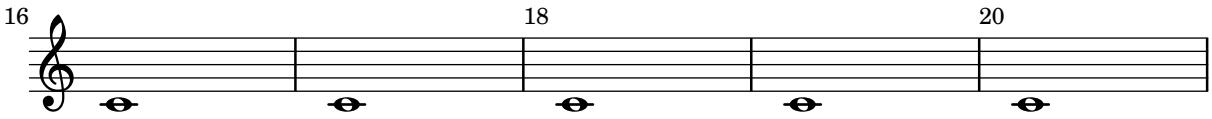


Imprimer les numéros de mesure à intervalle régulier

Vous pouvez imprimer un numéro de mesure à intervalle régulier plutôt qu'en tête de chaque ligne seulement, en recourant à la propriété `barNumberVisibility`. Voici comment afficher le numéro toutes les deux mesures sauf en fin de ligne.

```
\relative c' {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \set Score.currentBarNumber = #11
  % Permit first bar number to be printed
  \bar ""
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
  c1 | c | c | c | c
  \break
  c1 | c | c | c | c
}
```



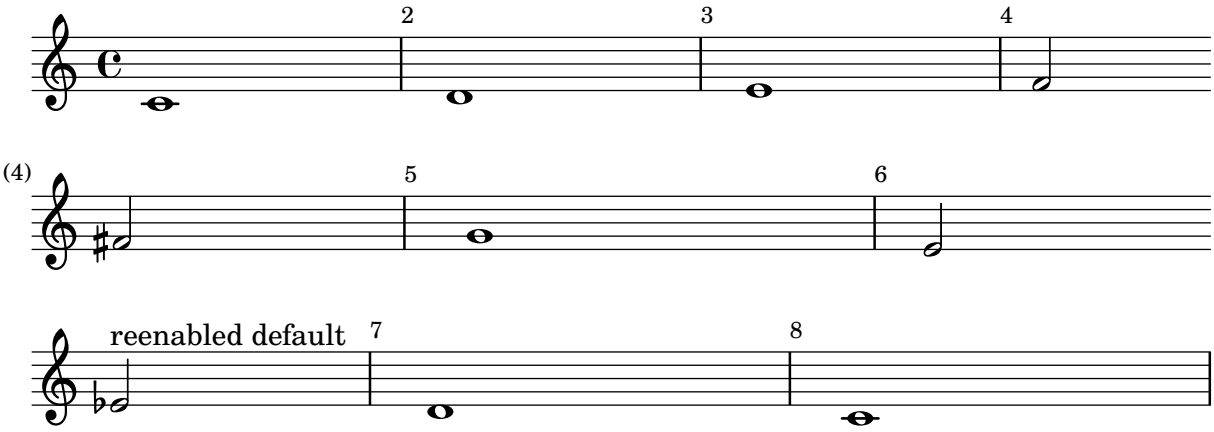


Impression du numéro des mesures tronquées

L'objet `BarNumber` (numéro de mesure) n'est pas répété en début de ligne pour une mesure tronquée. Il apparaîtra, entre parenthèses, dès lors que la propriété `barNumberVisibility` sera affublée de `first-bar-number-invisible-save-broken-bars`.

```
\layout {
  \context {
    \Score
    barNumberVisibility = #first-bar-number-invisible-save-broken-bars
    \override BarNumber.break-visibility = ##(#f #t #t)
  }
}

\relative c' {
  c1 | d | e | f2 \bar "" \break
  fis | g1 | e2 \bar "" \break
  <>^"reenabled default"
  %% back to default, \unset Score.barNumberVisibility would do so as well.
  \set Score.barNumberVisibility =
    #first-bar-number-invisible-and-no-parenthesized-bar-numbers
  es | d1 | c
}
```



Inscrire le numéro de mesure dans un cadre ou un cercle

Les numéros de mesure peuvent être encadrés ou entourés d'un cercle.

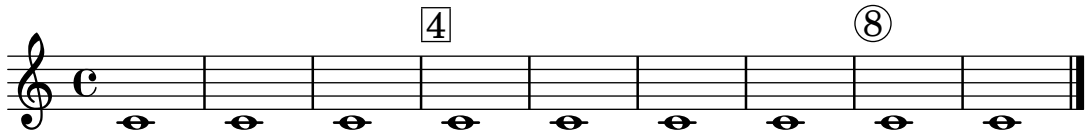
```
\relative c' {
  % Prevent bar numbers at the end of a line and permit them elsewhere
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 4)

  % Increase the size of the bar number by 2
  \override Score.BarNumber.font-size = #2

  % Draw a box round the following bar number(s)
  \override Score.BarNumber.stencil
    = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
```

```
  \repeat unfold 5 { c1 }

  % Draw a circle round the following bar number(s)
  \override Score.BarNumber.stencil
    = #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
  \repeat unfold 4 { c1 } \bar "|."
}
```



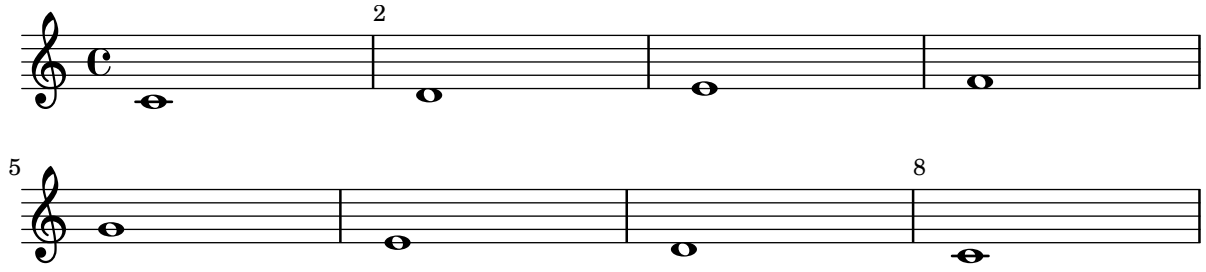
Impression du numéro de mesure selon `modulo-bar-number-visible`

Lorsque le reste de la division du numéro de la mesure courante par le premier argument de `modulo-bar-number-visible` égale le deuxième argument, LilyPond imprime le `BarNumber`. Ceci permet d'imprimer le numéro de mesure à un intervalle donné, par exemple :

- `(modulo-bar-number-visible 3 2)` -> affichage à 2, 5, 8...
- `(modulo-bar-number-visible 4 2)` -> affichage à 2, 6, 10...
- `(modulo-bar-number-visible 3 1)` -> affichage à 3, 5, 7...
- `(modulo-bar-number-visible 5 2)` -> affichage à 2, 7, 12...

```
\layout {
  \context {
    \Score
    \override BarNumber.break-visibility = ##(#f #t #t)
    barNumberVisibility = #(modulo-bar-number-visible 3 2)
  }
}

\relative c' {
  \bar ""
  c1 | d | e | f \break
  g1 | e | d | c
}
```

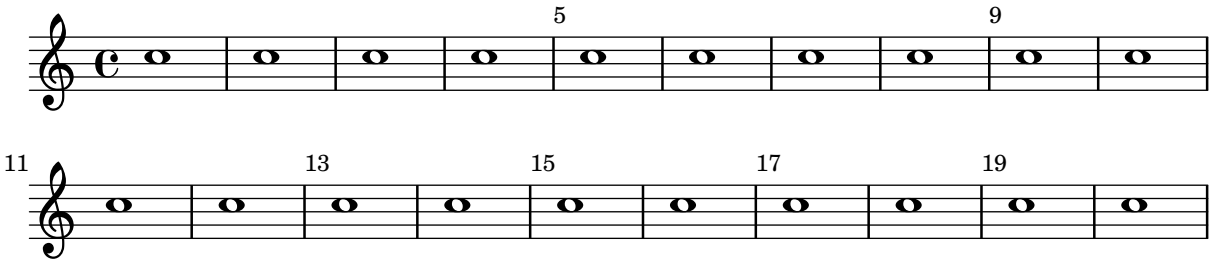


Changement de la fréquence d'impression du numéro de mesure

La fonction de contexte `{set-bar-number-visibility}` permet de modifier la fréquence à laquelle les numéros de mesures s'impriment.

```
\relative c' {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
```

```
  \context Score \applyContext #(set-bar-number-visibility 4)
  \repeat unfold 10 c'1
  \context Score \applyContext #(set-bar-number-visibility 2)
  \repeat unfold 10 c
}
```



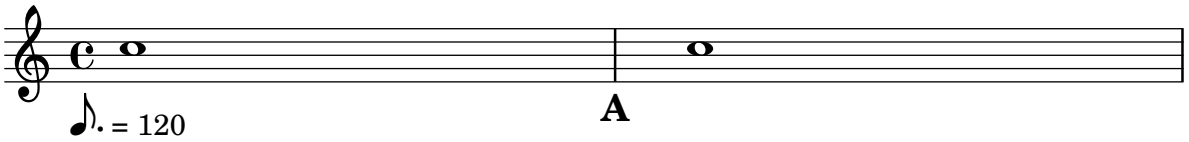
Impression du métronome et des repères sous la portée

Les indications de tempo et les marques de repère s'impriment par défaut au-dessus de la portée. Le fait de régler en conséquence la propriété `direction` des objets `MetronomeMark` et `RehearsalMark` les placera au-dessous de la portée.

```
\layout {
  indent = 0
  ragged-right = ##f
}

{
  % Metronome marks below the staff
  \override Score.MetronomeMark.direction = #DOWN
  \tempo 8. = 120
  c''1

  % Rehearsal marks below the staff
  \override Score.RehearsalMark.direction = #DOWN
  \mark \default
  c''1
}
```



Impression de musique aux métriques différentes

Bien qu'ayant des métriques bien différentes, les deux parties ci-dessous présentées restent synchrones.

Les barres de mesure ne peuvent plus être gérées au niveau du contexte `Score` ; les `Default_barline_engraver` et `Timing_translator` doivent être déplacés du contexte `Score` au contexte `Staff` afin de permettre des barres de mesure individualisées.

Le `Bar_number_engraver` devra lui aussi être déplacé, puisqu'il dépend de propriétés attachées au `Timing_translator`, afin de numéroter les mesures. L'utilisation d'un bloc `\with` dans la portée concernée permettra un affichage des numéros de mesure.

```
\paper {
```

```
  indent = #0
  ragged-right = ##t
}

global = { \time 3/4 { s2.*3 } \bar "" \break { s2.*3 } }

\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Default_bar_line_engraver"
    \remove "Bar_number_engraver"
    \override SpacingSpanner.uniform-stretching = ##t
    \override SpacingSpanner.strict-note-spacing = ##t
    proportionalNotationDuration = #(ly:make-moment 1/64)
  }
  \context {
    \Staff
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  }
  \context {
    \Voice
    \remove "Forbid_line_break_engraver"
    tupletFullLength = ##t
  }
}

Bassklarinette = \new Staff \with {
  \consists "Bar_number_engraver"
  barNumberVisibility = #(every-nth-bar-number-visible 2)
  \override BarNumber.break-visibility = #end-of-line-invisible
} <<
  \global {
    \bar "|"
    \clef treble
    \time 3/8
    d''4.

    \bar "|"
    \time 3/4
    r8 des''2( c''8)

    \bar "|"
    \time 7/8
    r4. ees''2 ~

    \bar "|"
    \time 2/4
    \tupletUp
    \tuplet 3/2 { ees''4 r4 d''4 ~ }
```

```
    \bar "|"
    \time 3/8
    \tupletUp
    \tuplet 4/3 { d''4 r4 }

    \bar "|"
    \time 2/4
    e''2

    \bar "|"
    \time 3/8
    es''4.

    \bar "|"
    \time 3/4
    r8 d''2 r8
    \bar "|"
  }
>>

Perkussion = \new StaffGroup <<
  \new Staff <<
    \global {
      \bar "|"
      \clef percussion
      \time 3/4
      r4 c'2 ~

      \bar "|"
      c'2.

      \bar "|"
      R2.

      \bar "|"
      r2 g'4 ~

      \bar "|"
      g'2. ~

      \bar "|"
      g'2.
    }
  >>
  \new Staff <<
    \global {
      \bar "|"
      \clef percussion
      \time 3/4
      R2.

      \bar "|"
```

```
      g'2. ~
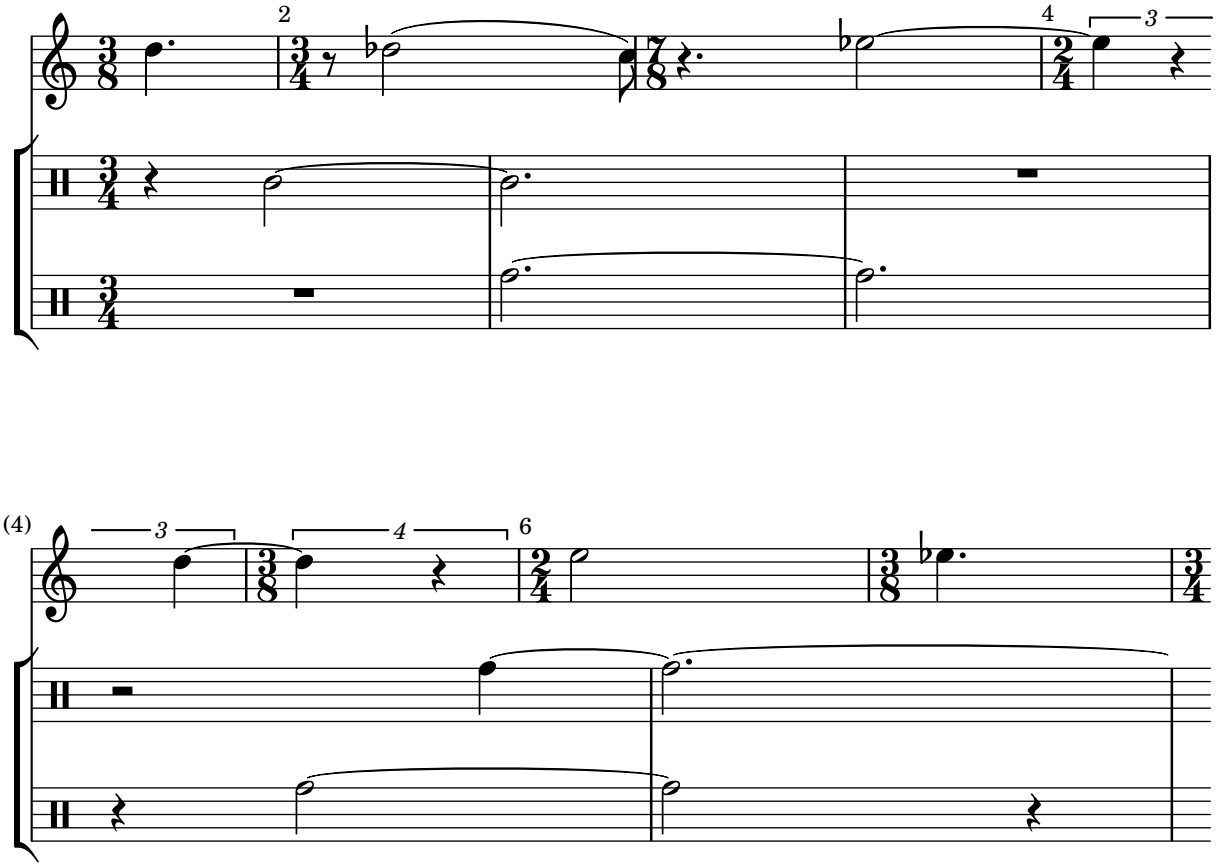
      \bar "|"
      g'2.

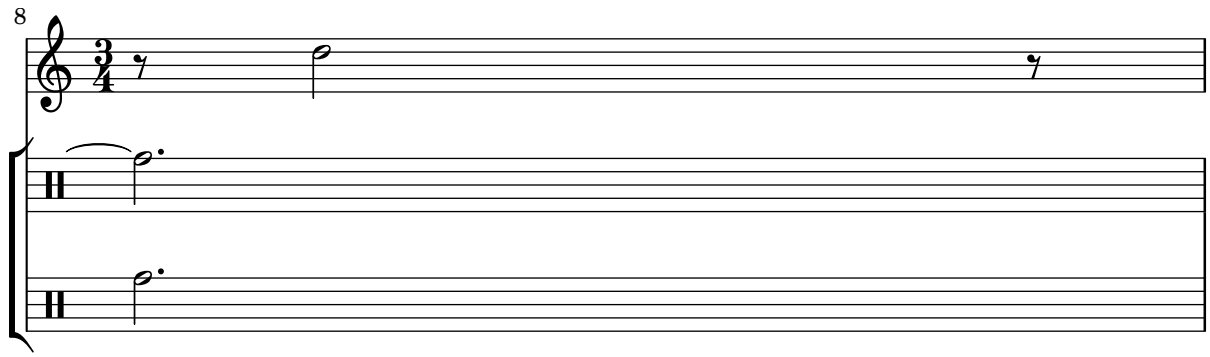
      \bar "|"
      r4 g'2 ~

      \bar "|"
      g'2 r4

      \bar "|"
      g'2.
    }
  >>
>>

\score {
  <<
    \Bassklarinette
    \Perkussion
  >>
}
```



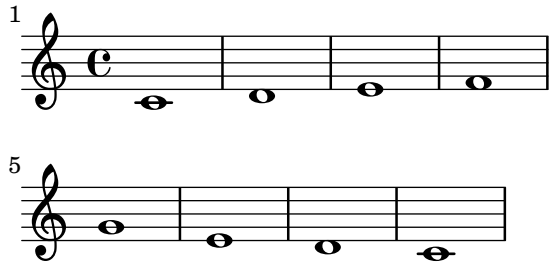


Afficher le numéro de la première mesure

Par défaut, LilyPond n'affiche pas le premier numéro de mesure s'il est inférieur à 2. Le fait de définir `barNumberVisibility` à `all-bar-numbers-visible` vous permettra d'imprimer n'importe quel numéro pour la première mesure. Notez que l'impression d'un numéro de mesure ne peut intervenir que s'il y a une barre. Aussi, pour pouvoir le faire au début d'un morceau, devrez-vous ajouter une barre vide avant la première note.

```
\layout {
  indent = 0
  ragged-right = ##t
}

\relative c' {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  \bar ""
  c1 | d | e | f \break
  g1 | e | d | c
}
```



Redéfinition des réglages de mise en forme par défaut des notes d'ornement

Les réglages par défaut des notes d'ornement sont stockés dans les variables suivantes :

```
startGraceMusic
stopGraceMusic
startAcciaccaturaMusic
stopAcciaccaturaMusic
startAppoggiaturaMusic
stopAppoggiaturaMusic
```
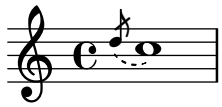
Ces variables sont définies dans le fichier `ly/grace-init.ly`. Amender leur définition permet d'en varier les effets.

```
startAcciaccaturaMusic = {
  <>(
  \override Flag.stroke-style = #"grace"
  \slurDashed
```

```
}

stopAcciaccaturaMusic = {
  \revert Flag.stroke-style
  \slurSolid
  <>)
}

\relative c'' {
  \acciaccatura d8 c1
}
```

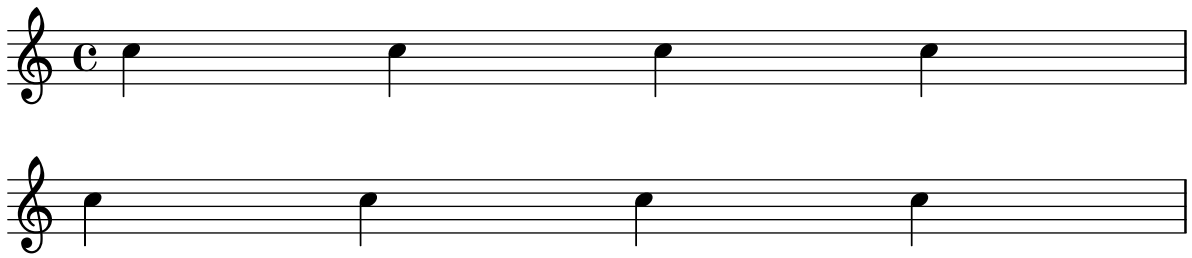


Suppression des numéros de mesure d'une partition

Désactiver le graveur concerné – `Bar_number_engraver` – donnera une partition – contexte `Score` – sans numéros de mesure.

```
\layout {
  \context {
    \Score
    \omit BarNumber
    % or:
    %\remove "Bar_number_engraver"
  }
}

\relative c'' {
  c4 c c c \break
  c4 c c c
}
```

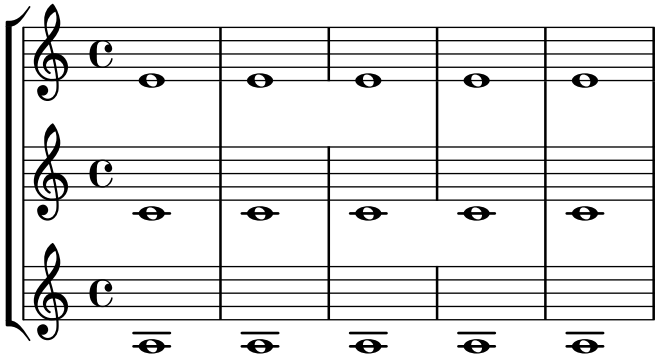


Suppression de la partie interportée des barres de mesure d'un regroupement autre que `ChoirStaff`

Les barres de mesure des regroupements `StaffGroup`, `PianoStaff` et `GrandStaff` sont par défaut d'un seul tenant. La portion entre les portées – l'objet `SpanBar` – peut néanmoins être supprimée, portée par portée.

```
\relative c' {
  \new StaffGroup <<
    \new Staff {
      e1 | e
```

```
      \once \override Staff.BarLine.allow-span-bar = ##f
      e1 | e | e
    }
    \new Staff {
      c1 | c | c
      \once \override Staff.BarLine.allow-span-bar = ##f
      c1 | c
    }
    \new Staff {
      a1 | a | a | a | a
    }
  >>
}
```



Styles de silences

Les silences peuvent être gravés selon différents styles.

```
\new Staff \relative c {
  \omit Score.TimeSignature
  \cadenzaOn

  \override Staff.Rest.style = #'mensural
  r\maxima^\markup \typewriter { mensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""
  \break

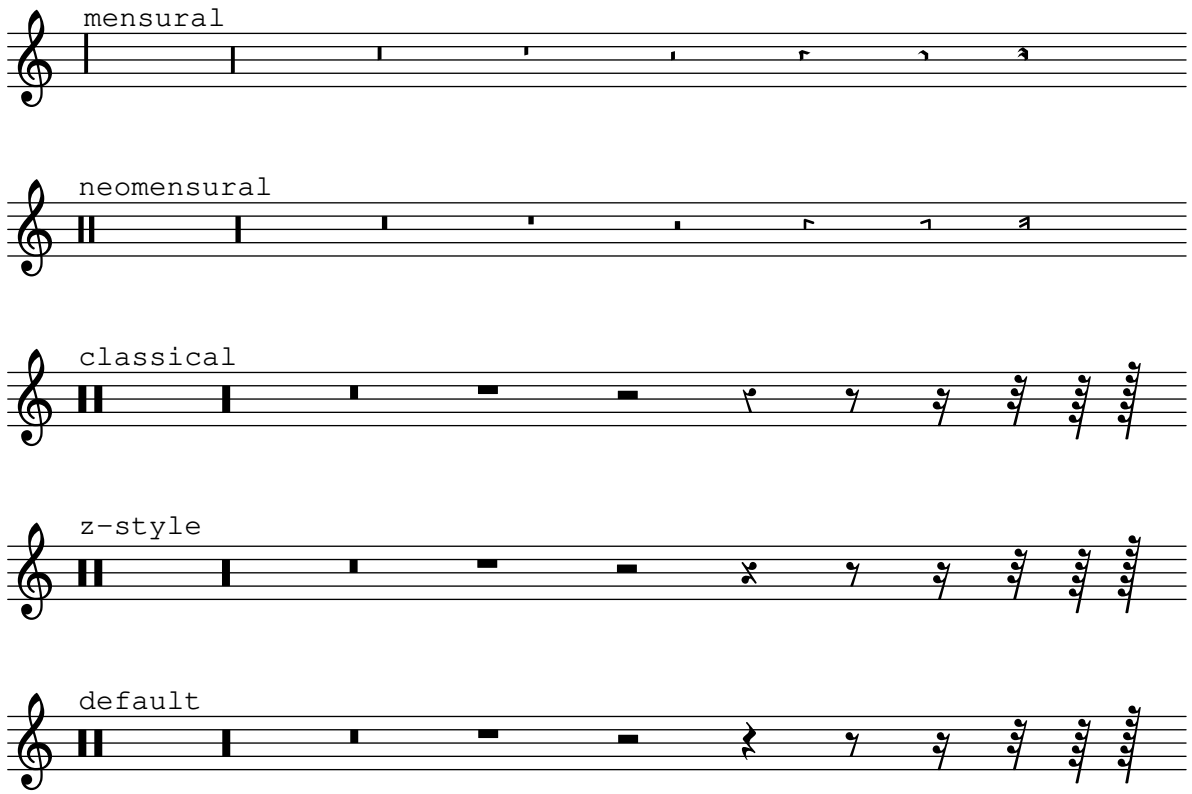
  \override Staff.Rest.style = #'neomensural
  r\maxima^\markup \typewriter { neomensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'classical
  r\maxima^\markup \typewriter { classical }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'z
  r\maxima^\markup \typewriter { z-style }
```

```
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'default
  r\maxima^\markup \typewriter { default }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
}
```
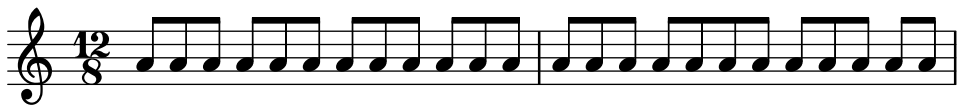


Annulation des règles de ligature par défaut

Pour obtenir des ligatures en groupes de `3-4-3-2` croches, dans une mesure à 12/8, il faudra préalablement annuler les réglages par défaut relatifs à 12/8, puis ajouter les règles adaptées :

```
\relative c'' {
  \time 12/8

  % Default beaming
  a8 a a a a a a a a a a a

  % Set new values for beam endings
  \set Score.beatStructure = 3,4,3,2
  a8 a a a a a a a a a a a
}
```



Barres rythmiques

Il arrive, dans une feuille de chant « simple », que les notes soient remplacées par une « pulsation » et que la structure de la chanson soit indiquée par les accords au-dessus des mesures. Ceci peut être utile lorsque l'on crée ou retranscrit la structure d'une chanson, ainsi que pour donner au guitariste et musiciens de jazz une pseudo partition.

Le fonctionnement standard d'un `\repeat percent` n'est pas ici applicable puisque le premier temps doit être une note ou un silence.

Le code ci-dessous propose deux alternatives à ce problème en redéfinissant l'aspect d'un silence. Si la durée d'un temps ne correspond pas à la noire, le `r4` inclus dans la définition devra être remplacé par un silence de durée appropriée.

```
% Macro to print single slash
rs = {
  \once \override Rest.stencil = #ly:percent-repeat-item-interface::beat-slash
  \once \override Rest.thickness = #0.48
  \once \override Rest.slope = #1.7
  r4
}

% Function to print a specified number of slashes
comp = #(define-music-function (count) (integer?)
  #{
    \override Rest.stencil = #ly:percent-repeat-item-interface::beat-slash
    \override Rest.thickness = #0.48
    \override Rest.slope = #1.7
    \repeat unfold $count { r4 }
    \revert Rest.stencil
  #}
)

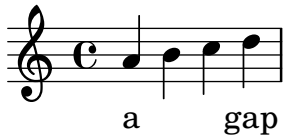
\score {
  \relative c' {
    c4 d e f |
    \rs \rs \rs \rs |
    \comp #4 |
  }
}
```



Sauts de notes en mode paroles (2)

Sachant qu'un `s` au sein d'un bloc `\lyricmode` est interprété non comme de l'espace mais comme une syllabe, l'utilisation de paires de guillemets informatiques (`""`) ou un caractère souligné simple (`_`) permet de « sauter » une note.

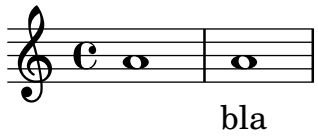
```
<<
  \relative c'' { a4 b c d }
  \new Lyrics \lyricmode { a4 "" _ gap }
>>
```



Sauts de notes en mode paroles

La syntaxe `s` qui permet de « faire un saut dans le temps » n'est disponible qu'en mode notes et en mode accords. Dans les autres situations, comme en mode paroles par exemple, la commande `\skip` produit les mêmes effets.

```
<<
  \relative c'' { a1 | a }
  \new Lyrics \lyricmode { \skip 1 bla1 }
>>
```



Moignons de hampe

Certaines conventions en matière de notation autorisent les ligatures à enjamber des silences. Dans certains cas, des moignons de hampe accrochés à la ligature offrent une meilleure visibilité du rythme ; certaines éditions modernes vont même alors jusqu'à omettre le silence.

Cet exemple illustre la progression : notation traditionnelle, ligature enjambant le silence, silence surplombé d'un moignon et enfin seule une hampe tronquée. Les moignons s'obtiennent par amendement de la propriété `stemlet-length` de l'objet `Stem`, alors que les silences sont masqués par activation de la propriété de transparence.
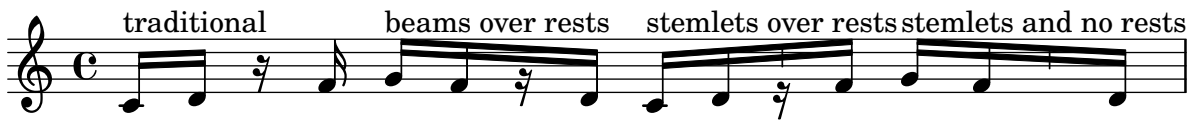
Les *markups* ajoutés au code ci-dessous mettent en exergue les différentes notations.

```
\paper { ragged-right = ##f }

{
  c'16^\markup { traditional } d' r f'
  g'16[^\markup { beams over rests } f' r d']

  % N.B. use Score.Stem to set for the whole score.
  \override Staff.Stem.stemlet-length = #0.75

  c'16[^\markup { stemlets over rests } d' r f']
  g'16[^\markup { stemlets and no rests } f'
  \once \hide Rest
  r16 d']
}
```
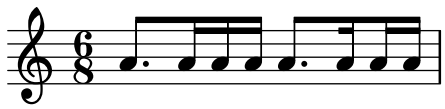


Ligature à la pulsation

Une sous-ligature tronquée peut pointer en direction de la pulsation à laquelle elle se rattache. Dans l'exemple suivant, la première ligature évite toute troncature (comportement par défaut), alors que la deuxième respecte rigoureusement la pulsation.

```
\relative c'' {
```

```
  \time 6/8
  a8. a16 a a
  \set strictBeatBeaming = ##t
  a8. a16 a a
}
```



Subdivision des ligatures

Les ligatures d'une succession de notes de durée inférieure à la croche ne sont pas subdivisées par défaut. Autrement dit, tous les traits de ligature (deux ou plus) seront continus. Ce comportement peut être modifié afin de diviser la ligature en sous-groupes grâce à la propriété `subdivideBeams`. Lorsqu'elle est activée, les ligatures seront subdivisées selon un intervalle défini par `baseMoment` ; il n'y aura alors plus que le nombre de traits de ligature déterminé entre chaque sous-groupe. Si le groupe qui suit la division est plus court que la valeur pour la métrique en cours – généralement lorsque la ligature est incomplète –, le nombre de traits de ligature correspond au regroupement de la bubdivision la plus longue. Cette restriction ne sera toutefois pas appliquée dans le cas où ne reste qu'une note après la division. Par défaut, `baseMoment` fixe la valeur de référence par rapport à la métrique en vigueur. Il faudra donc lui fournir, à l'aide de la fonction `ly:make-moment`, une fraction correspondant à la durée du sous-groupe désiré comme dans l'exemple ci-dessous. Gardez à l'esprit que, si vous venez à modifier `baseMoment`, vous devrez probablement adapter `beatStrusture` afin qu'il reste en adéquation avec les nouvelles valeurs de `baseMoment`.
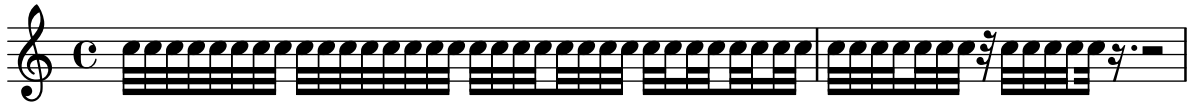
```
\relative c'' {
  c32[ c c c c c c]
  \set subdivideBeams = ##t
  c32[ c c c c c c]

  % Set beam sub-group length to an eighth note
  \set baseMoment = #(ly:make-moment 1/8)
  \set beatStructure = 2,2,2,2
  c32[ c c c c c c]

  % Set beam sub-group length to a sixteenth note
  \set baseMoment = #(ly:make-moment 1/16)
  \set beatStructure = 4,4,4,4
  c32[ c c c c c c]

  % Shorten beam by 1/32
  \set baseMoment = #(ly:make-moment 1/8)
  \set beatStructure = 2,2,2,2
  c32[ c c c c c] r32

  % Shorten beam by 3/32
  \set baseMoment = #(ly:make-moment 1/8)
  \set beatStructure = 2,2,2,2
  c32[ c c c] r16.
  r2
}
```



Exemple pour tam-tam

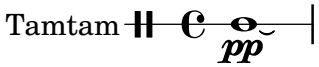
Un simple exemple pour tam-tam, saisi « tt ».

```
#(define mydrums '((tamtam default #t 0)))

\new DrumStaff \with { instrumentName = #"Tamtam" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \override Staff.StaffSymbol.line-positions = #'( 0 )
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)

  tt 1 \pp \laissezVibrer
}
```



Encadrement sur trois côtés

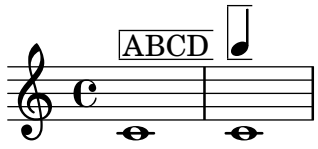
Voici comment construire une commande de *markup* chargée d'agrémenter du texte ou autre annotation, d'une bordure sur trois côtés.

```
% New command to add a three sided box, with sides north, west and south
% Based on the box-stencil command defined in scm/stencil.scm
% Note that ";;" is used to comment a line in Scheme
#(define-public (NWS-box-stencil stencil thickness padding)
   "Add a box around STENCIL, producing a new stencil."
   (let* ((x-ext (interval-widen (ly:stencil-extent stencil X) padding))
          (y-ext (interval-widen (ly:stencil-extent stencil Y) padding))
          (y-rule (make-filled-box-stencil (cons 0 thickness) y-ext))
          (x-rule (make-filled-box-stencil
                   (interval-widen x-ext thickness) (cons 0 thickness))))
     ;; (set! stencil (ly:stencil-combine-at-edge stencil X 1 y-rule padding))
     (set! stencil (ly:stencil-combine-at-edge stencil X LEFT y-rule padding))
     (set! stencil (ly:stencil-combine-at-edge stencil Y UP x-rule 0.0))
     (set! stencil (ly:stencil-combine-at-edge stencil Y DOWN x-rule 0.0))
     stencil))

% The corresponding markup command, based on the \box command defined
% in scm/define-markup-commands.scm
#(define-markup-command (NWS-box layout props arg) (markup?)
   #:properties ((thickness 0.1) (font-size 0) (box-padding 0.2))
   "Draw a box round @var{arg}.  Looks at @code{thickness},
@code{box-padding} and @code{font-size} properties to determine line
thickness and padding around the markup."
   (let ((pad (* (magstep font-size) box-padding))
         (m (interpret-markup layout props arg)))
     (NWS-box-stencil m thickness pad)))

% Test it:
```
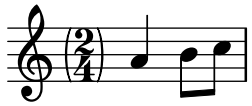
```
\relative c' {
  c1^\markup { \NWS-box ABCD }
  c1^\markup { \NWS-box \note {4} #1.0 }
}
```



Métrique entre parenthèses – méthode 3

Autre manière de mettre la métrique entre parenthèses.
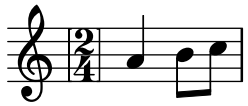
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (parenthesize-stencil (ly:time-signature::print grob) 0.1 0.4 0.4 0.1 ))
  \time 2/4
  a4 b8 c
}
```



Métrique entre parenthèses

Une métrique peut être mise entre parenthèses.

```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (bracketify-stencil (ly:time-signature::print grob) Y 0.1 0.2 0.1))
  \time 2/4
  a4 b8 c
}
```
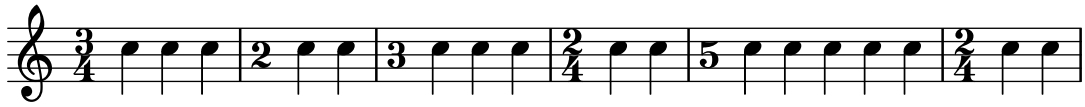


Affichage seulement du numérateur d'une métrique (au lieu d'une fraction)

La métrique est parfois indiquée non pas par une fraction (par ex. 7/4) mais simplement par son numérateur (7 dans ce cas). L'instruction `\override Staff.TimeSignature.style = #'single-digit` permet de déroger au style par défaut de manière permanente – un `\revert Staff.TimeSignature.style` annulera ces modifications. Lorsque cette métrique sous la forme d'un seul chiffre ne se présente qu'une seule fois, il suffit de faire précéder l'instruction `\override` d'un simple `\once`.

```
\relative c'' {
  \time 3/4
  c4 c c
  % Change the style permanently
```

```
  \override Staff.TimeSignature.style = #'single-digit
  \time 2/4
  c4 c
  \time 3/4
  c4 c c
  % Revert to default style:
  \revert Staff.TimeSignature.style
  \time 2/4
  c4 c
  % single-digit style only for the next time signature
  \once \override Staff.TimeSignature.style = #'single-digit
  \time 5/4
  c4 c c c c
  \time 2/4
  c4 c
}
```
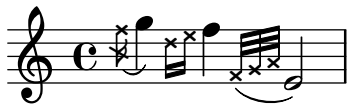


Mise en forme des notes d'ornement

Il est possible de changer globalement la mise en forme des notes d'ornement dans un morceau, au moyen des fonctions `add-grace-property` et `remove-grace-property`.

Ici, par exemple, on ôte la définition de l'orientation des objets `Stem` pour toutes les petites notes, afin que les hampes ne soient pas toujours orientées vers le haut, et on leur préfère des têtes en forme de croix.

```
\relative c'' {
  \new Staff {
    $(remove-grace-property 'Voice 'Stem 'direction)
    $(add-grace-property 'Voice 'NoteHead 'style 'cross)
    \new Voice {
       \acciaccatura { f16 } g4
       \grace { d16 e } f4
       \appoggiatura { f,32 g a } e2
    }
  }
}
```
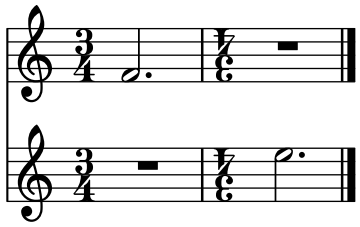


Personnalisation de la métrique

Il est possible de définir de nouveaux styles de métrique. La métrique de la deuxième mesure devrait être sens dessus dessous sur les deux portées.

```
#(add-simple-time-signature-style 'topsy-turvy
   (lambda (fraction)
     (make-rotate-markup 180 (make-compound-meter-markup fraction))))
```

```
<<
  \new Staff {
    \time 3/4 f'2.
    \override Score.TimeSignature.style = #'topsy-turvy
    \time 3/4 R2. \bar "|."
  }
  \new Staff {
    R2. e''
  }
>>
```



Crochets de style alternatif

Une dérogation à la propriété `stencil` de l'objet `Flag` permet aux croches et notes de durée inférieure d'adopter une autre forme de crochet. Sont disponibles les variantes `modern-straight-flag`, `old-straight-flag` et `flat-flag`.

```
testnotes = {
  \autoBeamOff
  c8 d16 c32 d64 \acciaccatura { c8 } d64 r4
}

\score {
  \relative c' {
    \time 2/4
    \testnotes
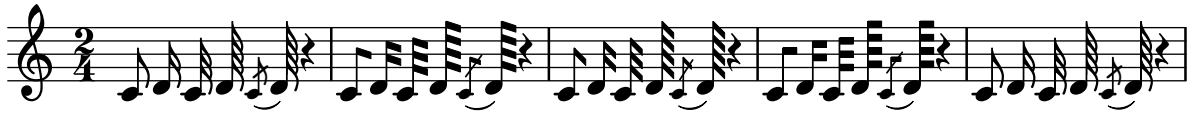
    \override Flag.stencil = #modern-straight-flag
    \testnotes

    \override Flag.stencil = #old-straight-flag
    \testnotes

    \override Flag.stencil = #flat-flag
    \testnotes

    \revert Flag.stencil
    \testnotes
  }
  \layout {
    indent = 0
    \context {
      \Score
      \override NonMusicalPaperColumn.line-break-permission = ##f
    }
```
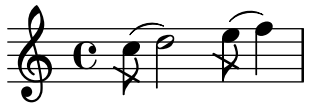
```
  }
}
```



Utilisation de hampe barrée pour une note normale

Le trait que l'on trouve sur les hampes des acciaccatures peut être appliqué dans d'autres situations.

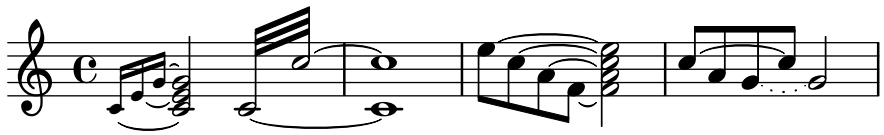
```
\relative c'' {
  \override Flag.stroke-style = #"grace"
  c8( d2) e8( f4)
}
```



Liaison de tenue et arpège

Les liaisons de tenue servent parfois à rendre un accord arpégé. Dans ce cas, les notes liées ne sont pas toutes consécutives. Il faut alors assigner à la propriété `tieWaitForNote` la valeur `#t` (*true* pour « vrai »). Cette même méthode peut servir, par exemple, à lier un trémolo à un accord.

```
\relative c' {
  \set tieWaitForNote = ##t
  \grace { c16[ ~ e ~ g] ~ } <c, e g>2
  \repeat tremolo 8 { c32 ~ c' ~ } <c c,>1
  e8 ~ c ~ a ~ f ~ <e' c a f>2
  \tieUp
  c8 ~ a
  \tieDown
  \tieDotted
  g8 ~ c g2
}
```



Expressive marks

Section “Expressive marks” dans *Manuel de notation*
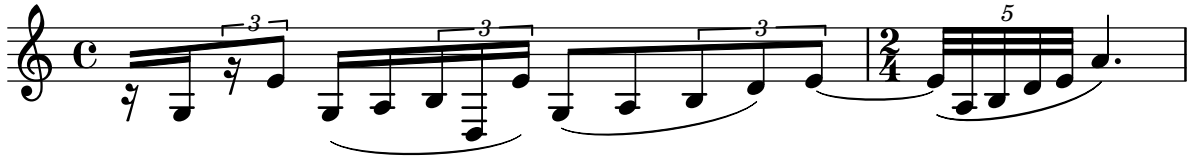
Coexistence de ligatures et liaisons avec des rythmes comprenant des n-olets

La syntaxe de LilyPond demande parfois un positionnement inhabituel des parenthèses, crochets, etc. qui peuvent s’entrelacer.

Par exemple, le crochet ouvrant une ligature manuelle se place **à la suite** de la note de départ et sa durée, non pas avant. De même, le crochet fermant se place à la fin de la ligature, y compris lorsque la dernière note se trouve incluse dans un n-olet.

Cet extrait illustre la manière de combiner ligatures manuelles, liaisons d’articulation, de prolongation ou de phrasé, avec des nolets bornées par des accolades.

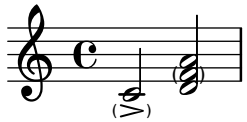
```
{
  r16[ g16 \tuplet 3/2 { r16 e'8] }
  g16( a \tuplet 3/2 { b d e') }
  g8[( a \tuplet 3/2 { b d') e'] ~ }
  \time 2/4
  \tuplet 5/4 { e'32\( a b d' e' } a'4.\)
}
```



Mise entre parenthèses d’un signe d’interprétation ou d’une note d’un accord

La fonction `\parenthesize`, qui permet de mettre un objet entre parenthèses, a ceci de particulier qu’elle est associée à un objet graphique `ParenthesesItem`.

```
\relative c' {
  c2-\parenthesize ->
  \override ParenthesesItem.padding = #0.1
  \override ParenthesesItem.font-size = #-4
  <d \parenthesize f a>2
}
```



Ajout de marques temporelles à un long glissando

Lorsqu’un glissando s’étend dans la durée, on trouve parfois des indications temporelles, matérialisées par des hampes sans tête de note. De telles hampes permettent aussi d’indiquer des éléments intermédiaires.

L’alignement des hampes avec la ligne de glissando peut requérir quelques aménagements.

```
glissandoSkipOn = {
```
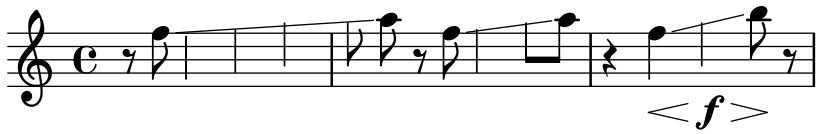
```
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

glissandoSkipOff = {
  \revert NoteColumn.glissando-skip
  \undo \hide NoteHead
  \revert NoteHead.no-ledgers
}

\relative c'' {
  r8 f8\glissando
  \glissandoSkipOn
  f4 g a a8\noBeam
  \glissandoSkipOff
  a8

  r8 f8\glissando
  \glissandoSkipOn
  g4 a8
  \glissandoSkipOff
  a8 |

  r4 f\glissando \<
  \glissandoSkipOn
  a4\f \>
  \glissandoSkipOff
  b8\! r |
}
```



Ajustement du galbe des chutes ou sauts

La propriété `shortest-duration-space` peut devoir être retouchée pour ajuster l'apparence des chutes ou sauts.

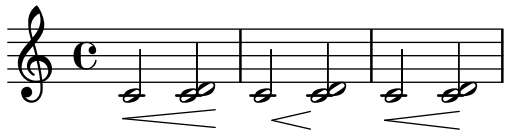
```
\relative c'' {
  \override Score.SpacingSpanner.shortest-duration-space = #4.0
  c2-\bendAfter #5
  c2-\bendAfter #-4.75
  c2-\bendAfter #8.5
  c2-\bendAfter #-6
}
```



Alignement des bornes de soufflet relativement aux NoteColumns

Les terminaisons des soufflets peuvent s'aligner sur la gauche, au centre ou sur la droite des *grobs* NoteColumn à l'aide d'une dérogation à la propriété endpoint-alignments – LEFT, CENTER ou RIGHT – sous forme de paire représentant les extrémités gauche et droite du soufflet. Les endpoint-alignments devraient être des directions (soit −1, 0 ou 1), les autres valeurs entraînant l'émission d'un avertissement. L'extrémité droite d'un soufflet se terminant sur un silence ne sera pas affectée et s'alignera toujours sur le bord gauche de ce silence.

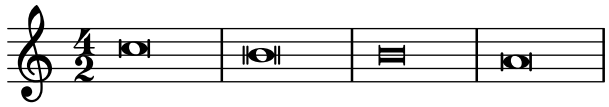
```
{
  c'2\< <c' d'>\! |
  \override Hairpin.endpoint-alignments = #'(1 . -1)
  c'2\< <c' d'>\! |
  \override Hairpin.endpoint-alignments = #`(,LEFT . ,CENTER)
  c'2\< <c' d'>\! |
}
```



Brève alternative, avec deux barres verticales

Voici comment obtenir une brève – aussi appelée note carée – flanquée de deux barres verticales, au lieu d'une comme habituellement.

```
\relative c'' {
  \time 4/2
  c\breve |
  \override Staff.NoteHead.style = #'altdefault
  b\breve
  \override Staff.NoteHead.style = #'baroque
  b\breve
  \revert Staff.NoteHead.style
  a\breve
}
```

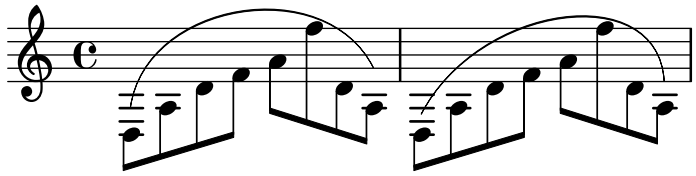


Liaison asymétrique

Une liaison peut adopter une courbe asymétrique afin de s'adapter au mieux à la ligne mélodique.

```
slurNotes = { d,8( a' d f a f' d, a) }

\relative c' {
  \stemDown
  \slurUp
  \slurNotes
  \once \override Slur.eccentricity = #3.0
  \slurNotes
}
```



Signes de respiration
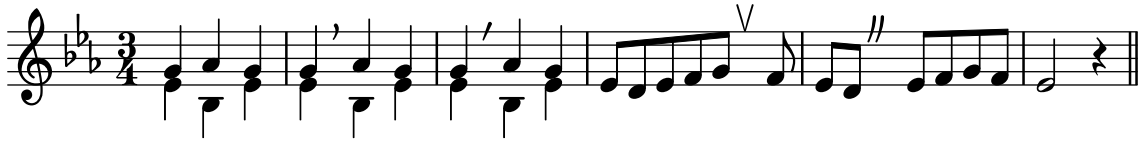
Les indications de respiration sont disponibles sous différentes formes : virgule (par défaut), trait, en V ou en « voie de chemin de fer » (césure).

```
\new Staff \relative c'' {
  \key es \major
  \time 3/4
  % this bar contains no \breathe
  << { g4 as g } \\ { es4 bes es } >> |
  % Modern notation:
  % by default, \breathe uses the rcomma, just as if saying:
  % \override BreathingSign.text = #(make-musicglyph-markup "scripts.rcomma")
  << { g4 as g } \\ { es4 \breathe bes es } >> |

  % rvarcomma and lvarcomma are variations of the default rcomma and lcomma
  % N.B.: must use Staff context here, since we start a Voice below
  \override Staff.BreathingSign.text =
    \markup { \musicglyph "scripts.rvarcomma" }
  << { g4 as g } \\ { es4 \breathe bes es } >> |

  % vee
  \override BreathingSign.text = \markup { \musicglyph "scripts.upbow" }
  es8[ d es f g] \breathe f |

  % caesura
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.caesura.curved" }
  es8[ d] \breathe es[ f g f] |
  es2 r4 \bar "||"
}
```



Soufflet de crescendo partiellement interrompu

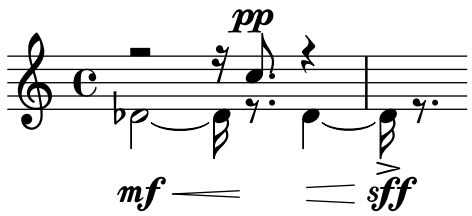
Une portion d'un soufflet de crescendo peut être rendue invisible. Il suffit pour cela de dessiner un rectangle par dessus ce tronçon, ce qui aura pour effet de le rendre invisible. Ce rectangle est défini au sein d'un *markup* par du code PostScript.

La commande de *markup* `with-dimensions` indique à LilyPond de ne prendre en considération que l'extrémité inférieure du rectangle lors de son positionnement par rapport au soufflet. L'ajustement de la propriété `staff-padding` permet d'éviter au rectangle de venir s'intercaler entre le soufflet et la portée.

Le soufflet doit se trouver à un niveau inférieur à celui du *markup* afin que le dessin du rectangle puisse effectivement le recouvrir.

```
\relative c' {
```
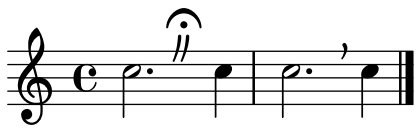
```
  <<
    {
      \dynamicUp
      r2 r16 c'8.\pp r4
    }
    \\
    {
      \override DynamicLineSpanner.layer = #0
      des,2\mf\< ~
      \override TextScript.layer = #2
      \once\override TextScript.staff-padding = #6
      \once\override TextScript.vertical-skylines = #'()
      des16_\markup \with-dimensions #'(2 . 7) #'(0 . 0)
                    \with-color #white
                    \filled-box #'(2 . 7) #'(0 . 2) #0
      r8. des4 ~ des16->\sff r8.
    }
  >>
}
```



Césure en forme de « voie de chemin de fer » avec point d'orgue

Une césure peut parfois s'indiquer par une double respiration surmontée d'un point d'orgue. Le code ci-dessous permet d'obtenir la combinaison répondant à cette esthétique.

```
\relative c'' {
  c2.
  % construct the symbol
  \override BreathingSign.text = \markup {
    \override #'(direction . 1)
    \override #'(baseline-skip . 1.8)
    \dir-column {
      \translate #'(0.155 . 0)
        \center-align \musicglyph "scripts.caesura.curved"
      \center-align \musicglyph "scripts.ufermata"
    }
  }
  \breathe c4
  % set the breathe mark back to normal
  \revert BreathingSign.text
  c2. \breathe c4
  \bar "|."
}
```



Texte centré sous un soufflet

La fonction comprise dans l'extrait suivant permet d'ajouter du texte – comme « molto » ou « poco » – en dessous d'un soufflet de (de)crescendo. Cet exemple présente aussi comment, à l'aide de code Scheme, influencer la manière dont un objet est normalement imprimé.

```
\paper { tagline = ##f }

hairpinWithCenteredText =
#(define-music-function (text) (markup?)
  #{
    \once \override Voice.Hairpin.after-line-breaking =
      #(lambda (grob)
        (let* ((stencil (ly:hairpin::print grob))
               (par-y (ly:grob-parent grob Y))
               (dir (ly:grob-property par-y 'direction))
               (staff-line-thickness
                 (ly:output-def-lookup (ly:grob-layout grob) 'line-thickness))
               (new-stencil (ly:stencil-aligned-to
                 (ly:stencil-combine-at-edge
                   (ly:stencil-aligned-to stencil X CENTER)
                   Y dir
                   (ly:stencil-aligned-to
                     (grob-interpret-markup
                       grob
                       (make-fontsize-markup
                         (magnification->font-size
                           (+ (ly:staff-symbol-staff-space grob)
                              (/ staff-line-thickness 2)))
                           text)) X CENTER))
                 X LEFT))
               (staff-space (ly:output-def-lookup
                 (ly:grob-layout grob) 'staff-space))
               (par-x (ly:grob-parent grob X))
               (dyn-text (grob::has-interface par-x 'dynamic-text-interface))
               (dyn-text-stencil-x-length
                 (if dyn-text
                   (interval-length
                     (ly:stencil-extent (ly:grob-property par-x 'stencil) X))
                   0))
               (x-shift
                 (if dyn-text
                   (-
                     (+ staff-space dyn-text-stencil-x-length)
                     (* 0.5 staff-line-thickness)) 0)))

          (ly:grob-set-property! grob 'Y-offset 0)
          (ly:grob-set-property! grob 'stencil
             (ly:stencil-translate-axis
              new-stencil
              x-shift X))))
```

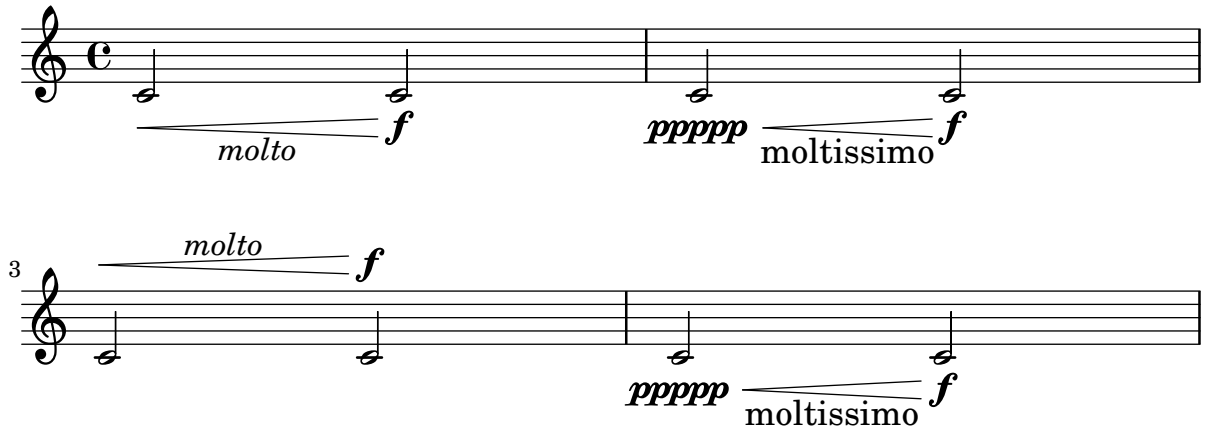
```
  #})

hairpinMolto =
\hairpinWithCenteredText \markup { \italic molto }

hairpinMore =
\hairpinWithCenteredText \markup { \larger moltissimo }

\layout { ragged-right = ##f }

\relative c' {
  \hairpinMolto
  c2\< c\f
  \hairpinMore
  c2\ppppp\< c\f
  \break
  \hairpinMolto
  c2^\< c\f
  \hairpinMore
  c2\ppppp\< c\f
}
```



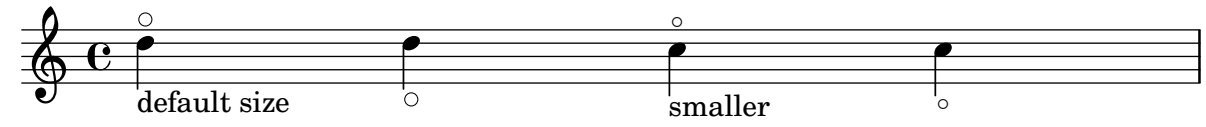
Modifier la taille d'un \flageolet

Il est possible de rapetisser le cercle d'un `\flageolet` grâce à un `\tweak`.

```
smallFlageolet = \tweak font-size -3 \flageolet

\layout { ragged-right = ##f }
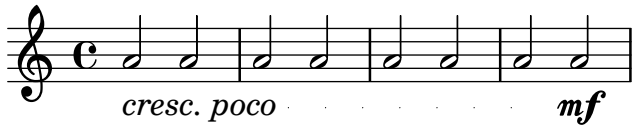
\relative c'' {
  d4^\flageolet_\markup { default size } d_\flageolet
  c4^\smallFlageolet_\markup { smaller } c_\smallFlageolet
}
```



Modification du texte et de l'extension de nuances textuelles

Le texte par défaut des crescendos et decrescendos se change en modifiant les propriétés de contexte `crescendoText` et `decrescendoText`. L'aspect de la ligne d'extension est fonction de la propriété `style` du `DynamicTextSpanner`. Sa valeur par défaut est `'hairpin`, mais d'autres valeurs sont disponibles, comme `'line`, `'dashed-line` et `'dotted-line`.
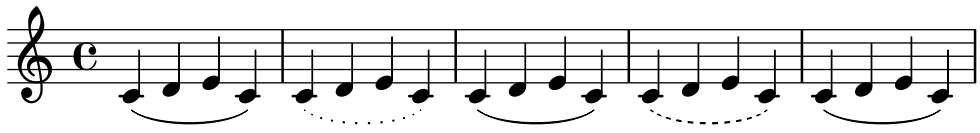
```
\relative c'' {
  \set crescendoText = \markup { \italic { cresc. poco } }
  \set crescendoSpanner = #'text
  \override DynamicTextSpanner.style = #'dotted-line
  a2\< a
  a2 a
  a2 a
  a2 a\mf
}
```



Modification de l'aspect des liaisons d'articulation

Une liaison d'articulation peut se présenter sous la forme d'un trait continu ou discontinu, voire en pointillé.

```
\relative c' {
  c4( d e c)
  \slurDotted
  c4( d e c)
  \slurSolid
  c4( d e c)
  \slurDashed
  c4( d e c)
  \slurSolid
  c4( d e c)
}
```



Modification de l'indicateur de respiration

On peut choisir le glyphe imprimé par cette commande, en modifiant la propriété `text` de l'objet `BreathingSign`, pour lui affecter n'importe quelle indication textuelle.
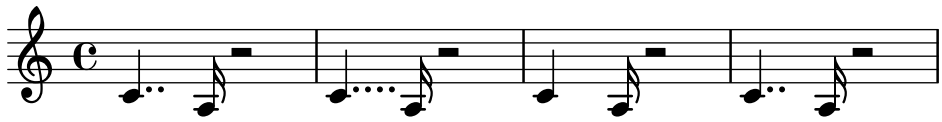
```
\relative c'' {
  c2
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.rvarcomma" }
  \breathe
  d2
}
```



Spécification du nombre de points d'augmentation d'une note

Le nombre de points d'augmentation affectés à une note en particulier peut se modifier indépendamment des points placés après la note.

```
\relative c' {
  c4.. a16 r2 |
  \override Dots.dot-count = #4
  c4.. a16 r2 |
  \override Dots.dot-count = #0
  c4.. a16 r2 |
  \revert Dots.dot-count
  c4.. a16 r2 |
}
```



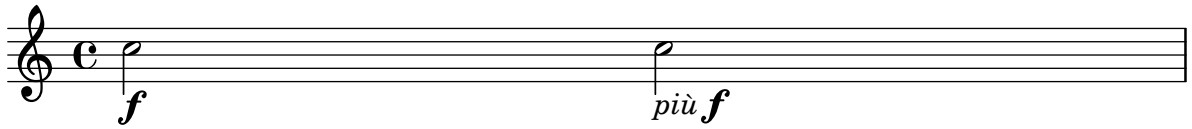
Combinaison de nuance et de texte

Certaines indications de nuance peuvent requérir un complément textuel, comme « più forte » ou « piano subito ». Elles se réalisent aisément à l'aide d'un bloc `\markup`.

```
piuF = \markup { \italic più \dynamic f }

\layout { ragged-right = ##f }

\relative c'' {
  c2\f c-\piuF
}
```
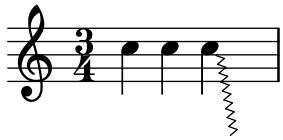


Glissando contemporain

De nos jours, il peut arriver que la note d'arrivée d'un glissando soit absente de la partition. Pour ce faire, il vous faudra utiliser une cadence et « masquer » la note d'arrivée.

```
\relative c'' {
  \time 3/4
  \override Glissando.style = #'zigzag
  c4 c
  \cadenzaOn
  c4\glissando
  \hideNotes
  c,,4
  \unHideNotes
  \cadenzaOff
  \bar "|"
```

```
}
```



Contrôle de la visibilité des extensions d'objet après saut de ligne
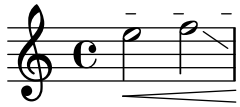
La visibilité des extensions qui se terminent sur la première note après un saut de ligne est contrôlée par un appel de `after-line-breaking` à la fonction `ly:spanner::kill-zero-spanned-time`.

Pour des objets tels qu'un glissando ou un soufflet, le comportement par défaut est de ne pas être reportés après un saut de ligne. L'extension sera donc reprise dès lors que l'appel aura été désactivé.

Il en va inversement pour les extensions qui, par défaut pour les textes notamment, sont reportées après un saut de ligne ; il faudra alors activer l'appel pour empêcher leur report.

```
\paper { ragged-right = ##t }

\relative c'' {
  \override Hairpin.to-barline = ##f
  \override Glissando.breakable = ##t
  % show hairpin
  \override Hairpin.after-line-breaking = ##t
  % hide text span
  \override TextSpanner.after-line-breaking =
    #ly:spanner::kill-zero-spanned-time
  e2\<\startTextSpan
  % show glissando
  \override Glissando.after-line-breaking = ##t
  f2\glissando
  \break
  f,1\!\stopTextSpan
}
```
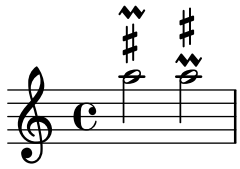




Contrôle de l'ordre vertical des articulations et ornements

Les symboles s'ordonnent verticalement suivant la propriété `script-priority`. Plus sa valeur numérique est faible, plus le symbole sera proche de la note. Dans l'exemple suivant, l'objet `TextScript` – le dièse – a d'abord la propriété la plus basse et se voit donc placé au plus près de la note ; ensuite, c'est l'objet `Script` – le mordant – qui a la propriété la plus basse, et se

place alors sous le dièse. Lorsque deux objets ont la même priorité, c'est l'ordre dans lequel ils sont indiqués qui détermine lequel sera placé en premier.

```
\relative c''' {
  \once \override TextScript.script-priority = #-100
  a2^\prall^\markup { \sharp }
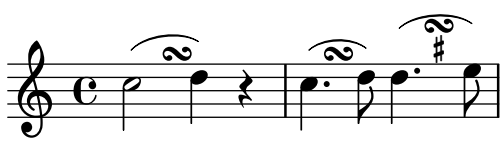
  \once \override Script.script-priority = #-100
  a2^\prall^\markup { \sharp }
}
```



Création d'un groupetto retardé

Obtenir un groupetto retardé et dans lequel la note la plus basse est altérée requiert quelques surcharges. La propriété `outside-staff-priority` doit être désactivée (`#f`) pour éviter qu'elle prenne le pas sur la propriété `avoid-slur`. L'ajustement du positionnement horizontal s'effectue en jouant sur les fractions `2/3` et `1/3`.
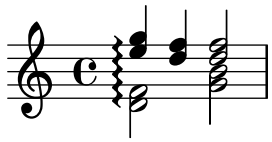
```
\relative c'' {
  c2*2/3 ( s2*1/3\turn d4) r
  <<
    { c4.( d8) }
    { s4 s\turn }
  >>
  \transpose c d \relative c'' <<
    { c4.( d8) }
    {
      s4
      \once \set suggestAccidentals = ##t
      \once \override AccidentalSuggestion.outside-staff-priority = ##f
      \once \override AccidentalSuggestion.avoid-slur = #'inside
      \once \override AccidentalSuggestion.font-size = -3
      \once \override AccidentalSuggestion.script-priority = -1
      \single \hideNotes
      b8-\turn \noBeam
      s8
    }
  >>
}
```



Arpège distribué sur plusieurs voix

Affecter le graveur `Span_arpeggio_engraver` au contexte de la portée (`Staff`) permet de distribuer un arpège sur plusieurs voix.

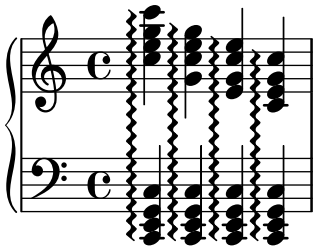
```
\new Staff \with {
  \consists "Span_arpeggio_engraver"
}
\relative c' {
  \set Staff.connectArpeggios = ##t
  <<
    { <e' g>4\arpeggio <d f> <d f>2 }
    \\
    { <d, f>2\arpeggio <g b>2 }
  >>
}
```



Arpège distribué sur une partition pour piano

Dans une double portée pour piano (`PianoStaff`), un arpège peut s'étendre sur les deux portées grâce à la propriété `PianoStaff.connectArpeggios`.

```
\new PianoStaff \relative c'' <<
  \set PianoStaff.connectArpeggios = ##t
  \new Staff {
    <c e g c>4\arpeggio
    <g c e g>4\arpeggio
    <e g c e>4\arpeggio
    <c e g c>4\arpeggio
  }
  \new Staff {
    \clef bass
    \repeat unfold 4 {
      <c,, e g c>4\arpeggio
    }
  }
>>
```

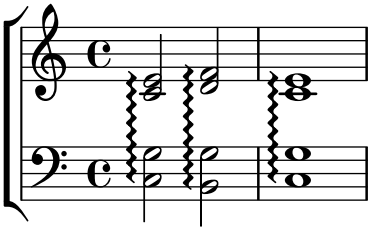


Arpège distribué pour un autre contexte que le piano

Il est possible de distribuer un arpège sur plusieurs portées d'un système autre que le `PianoStaff` dès lors que vous incluez le `Span_arpeggio_engraver` au contexte `Score`.

```
\score {
  \new ChoirStaff {
    \set Score.connectArpeggios = ##t
```

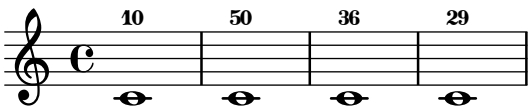
```
    <<
      \new Voice \relative c' {
        <c e>2\arpeggio
        <d f>2\arpeggio
        <c e>1\arpeggio
      }
      \new Voice \relative c {
        \clef bass
        <c g'>2\arpeggio
        <b g'>2\arpeggio
        <c g'>1\arpeggio
      }
    >>
  }
  \layout {
    \context {
      \Score
      \consists "Span_arpeggio_engraver"
    }
  }
}
```



Création de doigtés sur deux caractères

Il est tout à fait possible de noter un doigté supérieur à 5.

```
\relative c' {
  c1-10
  c1-50
  c1-36
  c1-29
}
```



Indications de nuance vraiment entre parenthèses
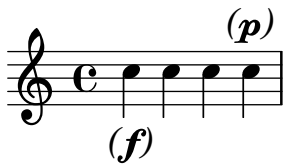
Bien que le moyen le plus simple pour ajouter des parenthèses à une indication de nuance consiste à utiliser un bloc `\markup`, cette pratique a un inconvénient : les objets ainsi créés seront considérés comme des annotations textuelles, non comme des nuances.

Il est néanmoins possible de créer des objets particuliers en partant de code Scheme – comme expliqué dans le manuel de notation – avec la fonction `make-dynamic-script`. Les *markups*

ainsi créés seront alors considérés comme étant des indications de nuance, et de ce fait pourront se voir appliquer les effets des commandes `\dynamicUp` et `\dynamicDown`.

```
paren =
#(define-event-function (dyn) (ly:event?)
   (make-dynamic-script
    #{ \markup \concat {
         \normal-text \italic \fontsize #2 (
         \pad-x #0.2 #(ly:music-property dyn 'text)
         \normal-text \italic \fontsize #2 )
       }
    #}))

\relative c'' {
  c4\paren\f c c \dynamicUp c\paren\p
}
```



Création de repères simultanés

Les indications de repère, à la différence des scripts textuels, ne peuvent s'empiler en un même endroit de la partition : il n'est possible de créer qu'un seul objet `RehearsalMark` à la fois. Le recours à une mesure invisible et à une barre de mesure permet cependant d'ajouter une autre indication de repère, donnant ainsi l'impression d'un double repère sur la même colonne.
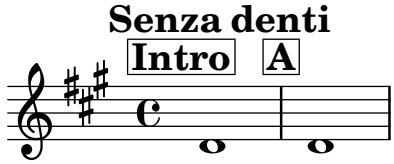
Cette méthode permet aussi de positionner un repère en fin de ligne et un autre au début de la ligne suivante.

```
{
  \key a \major
  \set Score.markFormatter = #format-mark-box-letters
  \once \override Score.RehearsalMark.outside-staff-priority = #5000
  \once \override Score.RehearsalMark.self-alignment-X = #LEFT
  \once \override Score.RehearsalMark.break-align-symbols = #'(key-signature)
  \mark \markup { \bold { Senza denti } }

  % the hidden measure and bar line
  % \cadenzaOn turns off automatic calculation of bar numbers
  \cadenzaOn
  \once \override Score.TimeSignature.stencil = ##f
  \time 1/16
  s16 \bar ""
  \cadenzaOff

  \time 4/4
  \once \override Score.RehearsalMark.self-alignment-X = #LEFT
  \mark \markup { \box \bold Intro }
  d'1
  \mark \default
  d'1
```
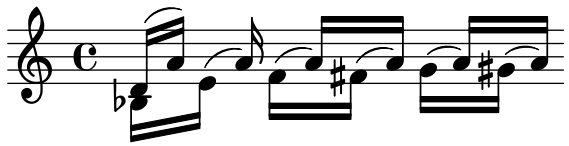
```
}
```



Création d'une liaison entre plusieurs voix

Dans certaines situations, il peut être nécessaire de lier des notes appartenant à des voix différentes. La solution consiste à ajouter une note invisible dans l'une des voix, à l'aide de la commande `\hideNotes`.

Voici par exemple la mesure 235 de la chaconne de la seconde partita pour violon solo, BWV 1004, de Bach.

```
\relative c' {
  <<
    {
      d16( a') s a s a[ s a] s a[ s a]
    }
    \\
    {
      \slurUp
      bes,16[ s e](
      \hideNotes a)
      \unHideNotes f[(
      \hideNotes a)
      \unHideNotes fis](
      \hideNotes a)
      \unHideNotes g[(
      \hideNotes a)
      \unHideNotes gis](
      \hideNotes a)
    }
  >>
}
```



Création d'extensions de texte

Les commandes `\startTextSpan` et `\stopTextSpan` permettent d'ajouter une ligne de prolongation aux indications textuelles, à l'instar des indications de pédale ou d'octaviation. Jouer sur les propriétés de l'objet `TextSpanner` permet d'en modifier le rendu.

```
\paper { ragged-right = ##f }

\relative c'' {
  \override TextSpanner.bound-details.left.text = #"bla"
  \override TextSpanner.bound-details.right.text = #"blu"
```

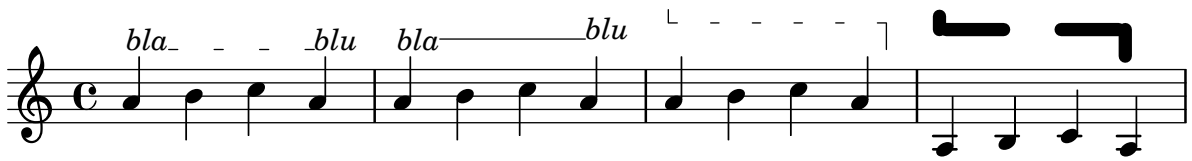
```
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.style = #'line
  \once \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.style = #'dashed-line
  \override TextSpanner.bound-details.left.text =
    \markup { \draw-line #'(0 . 1) }
  \override TextSpanner.bound-details.right.text =
    \markup { \draw-line #'(0 . -2) }
  \once \override TextSpanner.bound-details.right.padding = #-2

  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \set Staff.middleCPosition = #-13
  \override TextSpanner.dash-period = #10
  \override TextSpanner.dash-fraction = #0.5
  \override TextSpanner.thickness = #10
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan
}
```



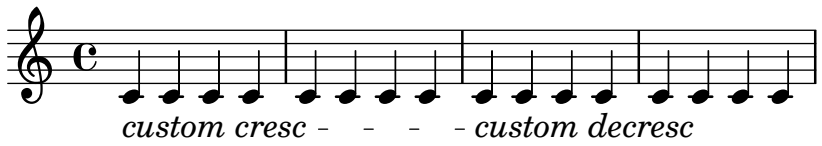
Personnalisation des extenseurs de nuance postfix

Il s'agit de fonctions postfix pour personnaliser l'extension des crescendos textuels. L'extension devrait débuter sur la première note de la mesure. Il faut utiliser `-\mycresc` – comme une articulation – sous peine que le départ de l'extension n'apparaisse qu'à la note suivante.

```
% Two functions for (de)crescendo spanners where you can explicitly
% give the spanner text.
mycresc =
#(define-music-function (mymarkup) (markup?)
   (make-music 'CrescendoEvent
               'span-direction START
               'span-type 'text
               'span-text mymarkup))
mydecresc =
#(define-music-function (mymarkup) (markup?)
   (make-music 'DecrescendoEvent
```

```
               'span-direction START
               'span-type 'text
               'span-text mymarkup))

\relative c' {
  c4-\mycresc "custom cresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4-\mydecresc "custom decresc" c4 c4 c4 |
  c4 c4\! c4 c4
}
```
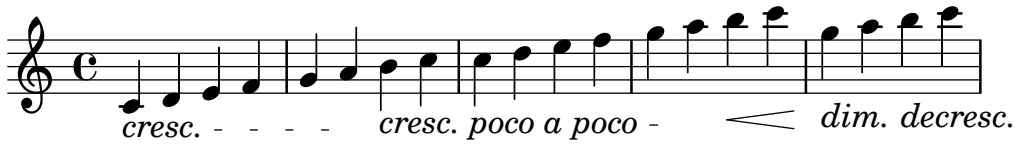


Extensions de nuance postfix

Les lignes d'extension des commandes `\cresc`, `\dim` et `\decresc` peuvent désormais être personnalisées facilement sous forme d'opérateurs postfix. Soufflets et (de)crescendos peuvent cohabiter. `\<` et `\>` produiront par défaut des soufflets, alors que `\cresc`, etc. produiront une indication textuelle avec extension.

```
% Some sample text dynamic spanners, to be used as postfix operators
crpoco =
#(make-music 'CrescendoEvent
             'span-direction START
             'span-type 'text
             'span-text "cresc. poco a poco")
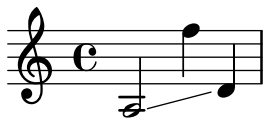
\relative c' {
  c4\cresc d4 e4 f4 |
  g4 a4\! b4\crpoco c4 |
  c4 d4 e4 f4 |
  g4 a4\! b4\< c4 |
  g4\dim a4 b4\decresc c4\!
}
```



Glissando par dessus un objet graphique

Un glissando peut sauter un objet `NoteColumn`.

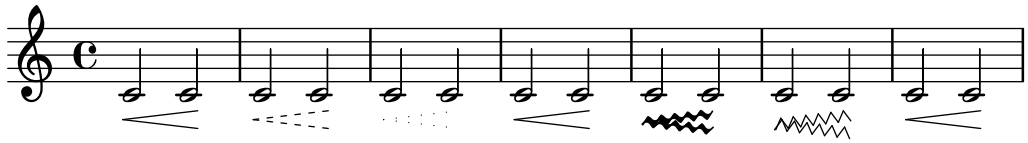
```
\relative c' {
  a2 \glissando
  \once \override NoteColumn.glissando-skip = ##t
  f''4 d,
}
```



Stylisation des lignes de soufflet

Les lignes d'un soufflet peuvent adopter tous les styles permis par la `line-interface` : discontinu, pointillé, continu, ondulé ou en zigzag.

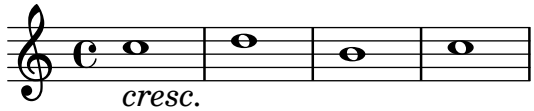
```
\relative c' {
  c2\< c\!
  \override Hairpin.style = #'dashed-line
  c2\< c\!
  \override Hairpin.style = #'dotted-line
  c2\< c\!
  \override Hairpin.style = #'line
  c2\< c\!
  \override Hairpin.style = #'trill
  c2\< c\!
  \override Hairpin.style = #'zigzag
  c2\< c\!
  \revert Hairpin.style
  c2\< c\!
}
```



Masquage de l'extension des nuances textuelles

Les crescendos et decrescendos indiqués textuellement – tels que *cresc.* ou *dim.* – sont suivis de pointillés qui montrent leur étendue. On peut empêcher l'impression de ces pointillés.

```
\relative c'' {
  \override DynamicTextSpanner.style = #'none
  \crescTextCresc
  c1\< | d | b | c\!
}
```



Alignement de nuances personnalisées comme « sempre pp », « più f » ou « subito p »

Certaines indications de nuance requièrent un complément textuel, comme « sempre pp ». Dans la mesure où les nuances sont habituellement centrées sous la note, le *pp* se trouvera repoussé loin après la note à laquelle il s'applique.

Différentes approches permettent de correctement aligner horizontalement ce « sempre pp » :

– Un simple `\once \override DynamicText.X-offset = #-9.2` avant la note considérée de telle sorte que la nuance soit décalée manuellement à la bonne place. Inconvénient : il faut le faire manuellement à chaque fois qu'intervient ce *markup*.

– L'intégration d'un rembourrage (`#:hspace 7.1`) à la définition de cette nuance personnalisée afin qu'une fois centrée par LilyPond elle soit correctement alignée. Inconvénient : le rembourrage occupera exactement cet espace et ne permettra à aucun autre *markup* ou nuance d'apparaître à cet endroit.

– L'application d'un décalage au script de nuance `\once \override ... X-offset = ...`. Inconvénient : là aussi il faut le faire à chaque fois.

– L'attribution arbitraire d'une dimension à 0 du texte additionnel à l'aide d'un `#:with-dimensions '(0 . 0) '(0 . 0)`. Inconvénient : LilyPond considère que « sempre » n'occupe pas d'espace, et donc pourra mettre à cet endroit d'autres éléments, ce qui pourrait générer des collisions qui ne seront pas détectées par les routines *ad hoc*. D'autre part, il semble persister un espacement, ce qui laisse l'impression d'un alignement différent en l'absence de texte additionnel.

– L'ajout, explicite, du décalage directement dans la fonction Scheme du script de nuance.

– La définition d'un alignement explicite au sein du script de nuance. Ceci ne sera suivi d'effet, par défaut, qu'en jouant sur `X-offset`. Inconvénient : il faut définir `DynamicText.X-offset`, ce qui s'appliquera à toutes les nuances textuelles. Par ailleurs, l'alignement sera réalisé sur le bord droit du texte additionnel, non sur le milieu de *pp*.

```
\paper {
  ragged-right = ##f
  indent = 2.5\cm
}

% Solution 1: Using a simple markup with a particular halign value
% Drawback: It's a markup, not a dynamic command, so \dynamicDown
%           etc. will have no effect
semppMarkup = \markup { \halign #1.4 \italic "sempre" \dynamic "pp" }

% Solution 2: Using a dynamic script & shifting with
%             \once \override ...X-offset = ..
% Drawback: \once \override needed for every invocation
semppK =
#(make-dynamic-script
  (markup #:line
          (#:normal-text
           #:italic "sempre"
           #:dynamic "pp")))

% Solution 3: Padding the dynamic script so the center-alignment
%             puts it at the correct position
% Drawback: the padding really reserves the space, nothing else can be there
semppT =
#(make-dynamic-script
  (markup #:line
          (#:normal-text
           #:italic "sempre"
           #:dynamic "pp"
           #:hspace 7.1)))

% Solution 4: Dynamic, setting the dimensions of the additional text to 0
% Drawback: To lilypond "sempre" has no extent, so it might put
%           other stuff there => collisions
```

```
% Drawback: Also, there seems to be some spacing, so it's not exactly the
%           same alignment as without the additional text
semppM =
#(make-dynamic-script
  (markup #:line
          (#:with-dimensions '(0 . 0) '(0 . 0)
                             #:right-align
                             #:normal-text
                             #:italic "sempre"
                             #:dynamic "pp")))
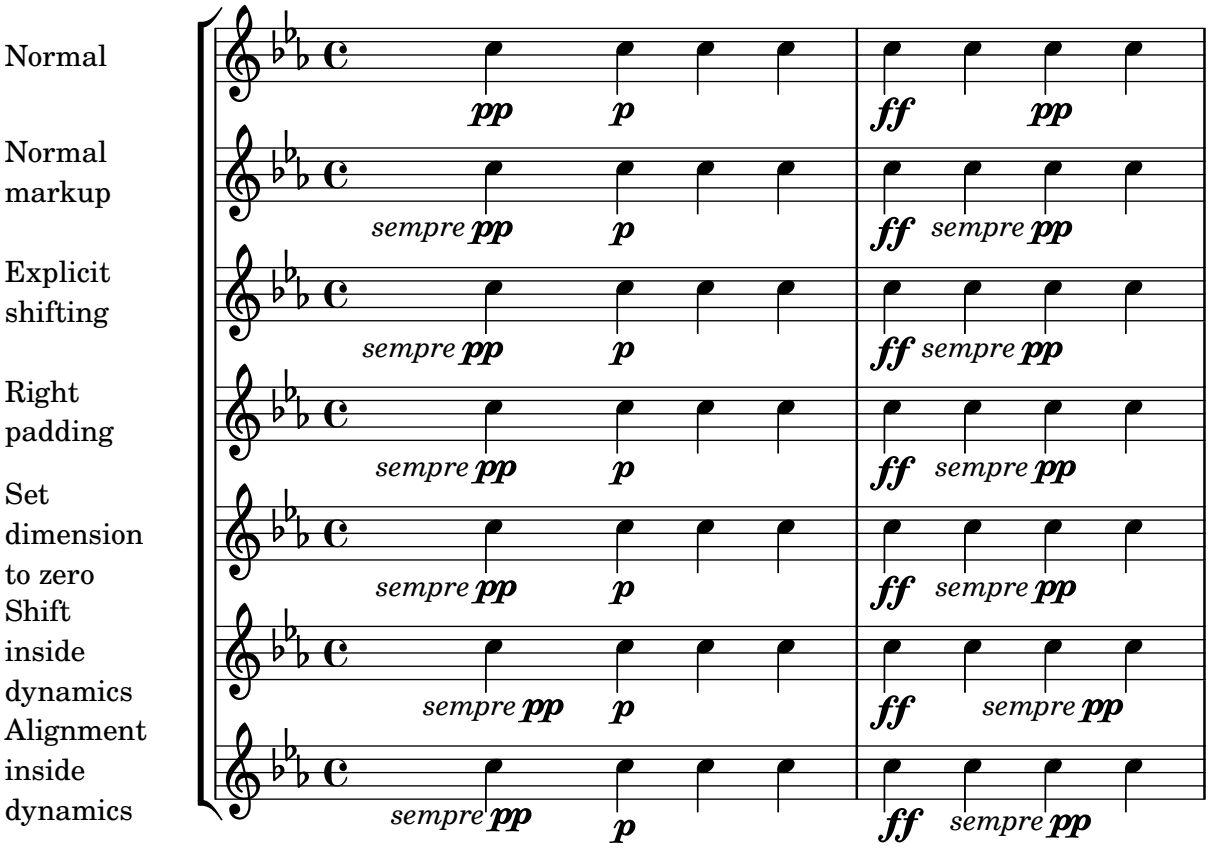
% Solution 5: Dynamic with explicit shifting inside the scheme function
semppG =
#(make-dynamic-script
  (markup #:hspace 0
          #:translate '(-18.85 . 0)
          #:line (#:normal-text
                  #:italic "sempre"
                  #:dynamic "pp")))

% Solution 6: Dynamic with explicit alignment. This has only effect
%             if one sets X-offset!
% Drawback: One needs to set DynamicText.X-offset!
% Drawback: Aligned at the right edge of the additional text,
%           not at the center of pp
semppMII =
#(make-dynamic-script
  (markup #:line (#:right-align
                  #:normal-text
                  #:italic "sempre"
                  #:dynamic "pp")))

\new StaffGroup <<
  \new Staff = "s" \with { instrumentName = \markup \column { Normal } }
  <<
    \relative c'' {
      \key es \major
      c4\pp c\p c c | c\ff c c\pp c
    }
  >>
  \new Staff = "sMarkup" \with {
    instrumentName = \markup \column { Normal markup }
  }
  <<
    \relative c'' {
      \key es \major
      c4-\semppMarkup c\p c c | c\ff c c-\semppMarkup c
    }
  >>
  \new Staff = "sK" \with {
    instrumentName = \markup \column { Explicit shifting }
  }
```

```
  <<
    \relative c'' {
      \key es \major
      \once \override DynamicText.X-offset = #-9.2
      c4\semppK c\p c c
      c4\ff c
      \once \override DynamicText.X-offset = #-9.2
      c4\semppK c
    }
  >>
  \new Staff = "sT" \with {
    instrumentName = \markup \column { Right padding }
  }
  <<
    \relative c'' {
      \key es \major
      c4\semppT c\p c c | c\ff c c\semppT c
    }
  >>
  \new Staff = "sM" \with {
    instrumentName = \markup \column { Set dimension "to zero" }
  }
  <<
    \relative c'' {
      \key es \major
      c4\semppM c\p c c | c\ff c c\semppM c
    }
  >>
  \new Staff = "sG" \with {
    instrumentName = \markup \column { Shift inside dynamics}
  }
  <<
    \relative c'' {
      \key es \major
      c4\semppG c\p c c | c\ff c c\semppG c
    }
  >>
  \new Staff = "sMII" \with {
    instrumentName = \markup \column { Alignment inside dynamics }
  }
  <<
    \relative c'' {
      \key es \major
      % Setting to ##f (false) gives the same result
      \override DynamicText.X-offset = #0
      c4\semppMII c\p c c | c\ff c c\semppMII c
    }
  >>
>>

\layout { \override Staff.InstrumentName.self-alignment-X = #LEFT }
```



Impression d'un double repère sur une même barre (méthode 1)

Cette méthode permet d'imprimer deux marques-repères l'une au-dessus de l'autre. Le repère inférieur est décalé sous la portée, puis s'insère au-dessus de lui un espace pour positionner le repère supérieur en surplomb de la portée.

L'ajustement des valeurs de `extra-offset` et `baseline-skip` permet d'accroître ou réduire globalement l'espace entre le repère et la portée.

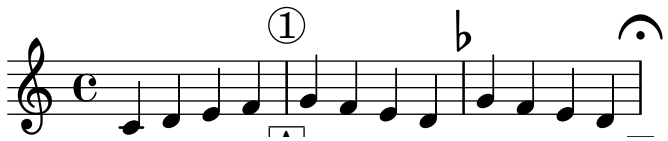
La capacité de pratiquement tout type de glyphe ou chaîne de caractères à faire l'objet d'un repère les rend capables de venir se centrer au-dessus ou en dessous d'une barre de mesure.

L'adjonction de la propriété `break-visibility`, comme dans cet autre cas (`\%22http://lsr.di.unimi.it/LSR/Item?id=1\%22`), permet de placer un double repère en fin de ligne.

Cette première méthode, bien que moins compliquée que la seconde, ne permet pas d'ajuster le positionnement d'un repère indépendamment de l'autre. Elle peut aussi générer quelques problèmes quant à l'espacement vertical, puisque le recours à `extra-offset` ne modifie en rien la valeur originelle de la boîte englobant le repère.

```
\relative c'{
  c d e f |
  \once \override Score.RehearsalMark.extra-offset = #'(0 . -8.5)
  \once \override Score.RehearsalMark.baseline-skip = #9
  \mark \markup \center-column { \circle 1 \box A }
  g f e d |
  \once \override Score.RehearsalMark.extra-offset = #'(0 . -8.5)
  \once \override Score.RehearsalMark.baseline-skip = #9
  \mark \markup \center-column { \flat { \bold \small \italic Fine. } }
  g f e d |
  \once \override Score.RehearsalMark.extra-offset = #'(0 . -8.5)
  \once \override Score.RehearsalMark.baseline-skip = #9
```

```
  \override Score.RehearsalMark.break-visibility = #begin-of-line-invisible
  \mark \markup \center-column { \fermata \box z }
}
```



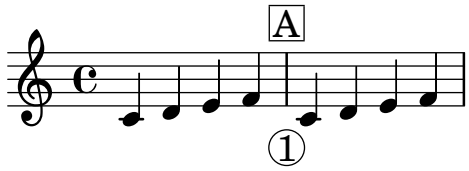
Impression d'un double repère sur une même barre (méthode 2)

Cette méthode permet d'imprimer deux marques-repères – l'une au-dessus de la portée et l'autre en dessous – à l'aide de deux voix séparées. Le graveur de repères est ensuite déplacé dans chacune des voix, sans quoi aucun repère ne sera imprimé. Enfin, chaque voix se voit attribuer un positionnement pour les repères.

Cette méthode, bien que plus complexe que la première, apporte plus de flexibilité dans l'ajustement individuel des deux composants du repère.

```
\score {
  \relative c'
  <<
    \new Staff {
      <<
        \new Voice \with {
          \consists Mark_engraver
          \consists "Staff_collecting_engraver"
        }
        { c4 d e f
          \mark \markup { \box A }
          c4 d e f
        }
        \new Voice \with {
          \consists Mark_engraver
          \consists "Staff_collecting_engraver"
          \override RehearsalMark.direction = #DOWN
        }
        { s4 s s s
          \mark \markup { \circle 1 }
          s4 s s s
        }
      >>
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Mark_engraver"
      \remove "Staff_collecting_engraver"
    }
```
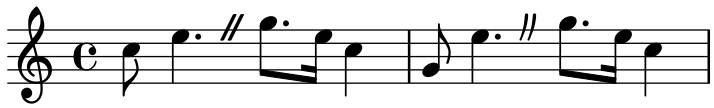
```
  }
}
```



Insertion d'une césure

Une surcharge de la propriété `text` de l'objet `BreathingSign` permet de créer une marque de césure.

LilyPond dispose également d'une variante courbée.

```
\relative c'' {
  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.straight"
  }
  c8 e4. \breathe g8. e16 c4

  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.curved"
  }
  g8 e'4. \breathe g8. e16 c4
}
```
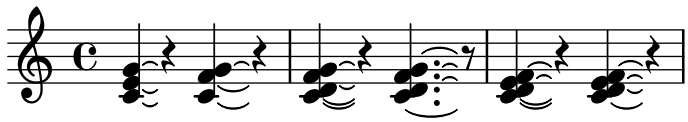


Liaison « Laissez vibrer »

Les indications de « laisser vibrer » ont une taille fixe. Leur formatage est accessible au travers de la propriété `tie-configuration`.

```
\relative c' {
  <c e g>4\laissezVibrer r <c f g>\laissezVibrer r
  <c d f g>4\laissezVibrer r <c d f g>4.\laissezVibrer r8

  <c d e f>4\laissezVibrer r
  \override LaissezVibrerTieColumn.tie-configuration
     = #`((-7 . ,DOWN)
          (-5 . ,DOWN)
          (-3 . ,UP)
          (-1 . ,UP))
  <c d e f>4\laissezVibrer r
}
```



Terminaison de ligne en flèche

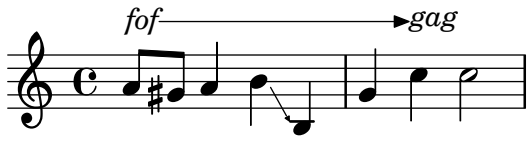
Les extensions de texte, tout comme les indications sous forme de ligne tel un glissando, peuvent se voir pourvues d'une extrémité en flèche.

```
\relative c'' {
  \override TextSpanner.bound-padding = #1.0
  \override TextSpanner.style = #'line
  \override TextSpanner.bound-details.right.arrow = ##t
  \override TextSpanner.bound-details.left.text = #"fof"
  \override TextSpanner.bound-details.right.text = #"gag"
  \override TextSpanner.bound-details.right.padding = #0.6

  \override TextSpanner.bound-details.right.stencil-align-dir-y = #CENTER
  \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER

  \override Glissando.bound-details.right.arrow = ##t
  \override Glissando.arrow-length = #0.5
  \override Glissando.arrow-width = #0.25

  a8\startTextSpan gis a4 b\glissando b,
  g'4 c\stopTextSpan c2
}
```
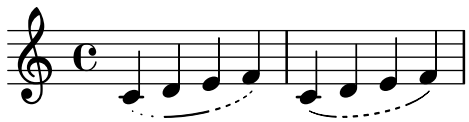


Dessin d'une liaison d'articulation au trait discontinu

Grâce à la propriété `dash-definition`, une liaison d'articulation peut être formée de traits discontinus variables. `dash-definition` se compose d'une liste de `segments-discontinus` (*dash-elements*). Chaque `segment-discontinu` contient une liste de paramètres qui déterminent le comportement du trait pour une section de la liaison.

Cette liaison se définit selon le paramètre de Bézier `t` qui est compris entre 0 (l'extrémité gauche de la liaison) et 1 (l'extrémité droite de la liaison). Chaque `segment-discontinu` se composera selon la liste `(t-début t-fin segment-style segment-taille)`. La portion de liaison allant de `t-début` à `t-fin` aura un trait `segment-style` de longueur `segment-taille`. `segment-taille` est exprimé en espace de portée ; un `segment-style` à 1 donnera un trait plein.

```
\relative c' {
  \once \override
    Slur.dash-definition = #'((0 0.3 0.1 0.75)
                              (0.3 0.6 1 1)
                              (0.65 1.0 0.4 0.75))
  c4( d e f)
  \once \override
    Slur.dash-definition = #'((0 0.25 1 1)
                              (0.3 0.7 0.4 0.75)
                              (0.75 1.0 1 1))
  c4( d e f)
}
```



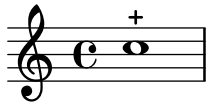
Modification de la signification des raccourcis pour les signes d'articulation
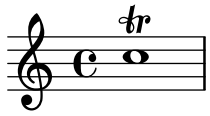
Les raccourcis sont répertoriés dans le fichier '`ly/script-init.ly`', dans lequel on retrouve les variables `dashHat`, `dashPlus`, `dashDash`, `dashBang`, `dashLarger`, `dashDot` et `dashUnderscore` ainsi que leur valeur par défaut. Ces valeurs peuvent être modifiées selon vos besoins. Il suffit par exemple, pour affecter au raccourci `-+` (`dashPlus`) le symbole du trille en lieu et place du `+` (caractère plus), d'assigner la valeur `trill` à la variable `dashPlus` :

```
\relative c'' { c1-+ }

dashPlus = "trill"

\relative c'' { c1-+ }
```



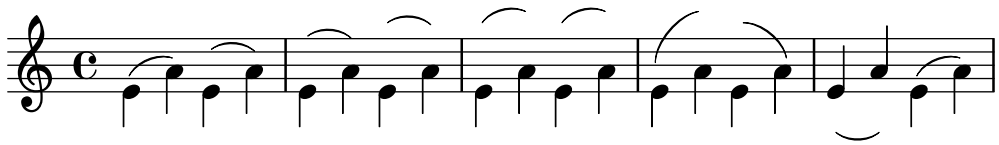


Déplacement vertical des liaisons d'articulation

Le positionnement vertical d'une liaison se contrôle par la propriété `positions` de l'objet `Slur`. Cette propriété dispose de deux paramètres : le premier affecte l'extrémité gauche de la liaison, le second son extrémité droite. La valeur des paramètres n'aura aucune influence sur le galbe de la liaison. LilyPond ne s'en servira que pour adapter le positionnement de la courbe. Des valeurs positives décalent la liaison vers le haut et s'utilisent pour des hampes descendantes. Des valeurs négatives entraînent un décalage vers le bas.

```
\relative c' {
  \stemDown
  e4( a)
  \override Slur.positions = #'(1 . 1)
  e4( a)
  \override Slur.positions = #'(2 . 2)
  e4( a)
  \override Slur.positions = #'(3 . 3)
  e4( a)
  \override Slur.positions = #'(4 . 4)
  e4( a)
  \override Slur.positions = #'(5 . 5)
  e4( a)
  \override Slur.positions = #'(0 . 5)
  e4( a)
  \override Slur.positions = #'(5 . 0)
  e4( a)
  \stemUp
  \override Slur.positions = #'(-5 . -5)
  e4( a)
```

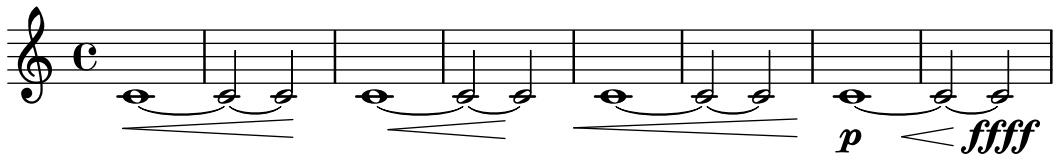
```
  \stemDown
  \revert Slur.positions
  e4( a)
}
```



Déplacer les extrémités de soufflets

Les terminaisons des soufflets peuvent être décalées en jouant sur la propriété `shorten-pair` de l'objet`Hairpin`. Les valeurs positives déplacent vers l'intérieur, les négatives vers l'extérieur. Contrairement à la propriété `minimum-length`, cette propriété n'affecte que l'apparence du soufflet sans ajuster l'espacement horizontal (y compris avec les nuances textuelles). Cette méthode est donc appropriée aux situations dans lesquelles un soufflet requiert un ajustement fin dans l'espace qui lui est alloué.

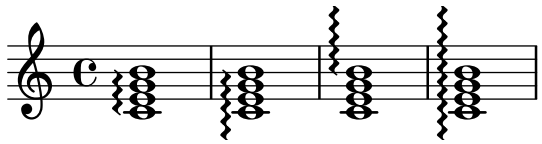
```
{
  c'1~\<
  c'2~ c'\!
  \once \override Hairpin.shorten-pair = #'(2 . 2)
  c'1~\<
  c'2~ c'\!
  \once \override Hairpin.shorten-pair = #'(-2 . -2)
  c'1~\<
  c'2~ c'\!
  c'1~\p-\tweak shorten-pair #'(2 . 0)\<
  c'2~ c'\ffff
}
```



Positionnement des arpeggios

L'ajustement de la taille d'une indication d'arpeggio s'effectue au travers du positionnement de ses extrémités haute ou basse.
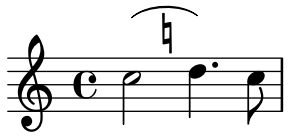
```
\relative c' {
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(-5 . 0)
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(0 . 5)
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(-5 . 5)
  <c e g b>1\arpeggio
}
```



Positionnement d'une annotation à l'intérieur d'une liaison

Lorsqu'une annotation doit s'incrire à l'intérieur d'une liaison, la propriété `outside-staff-priority` doit être désactivée.
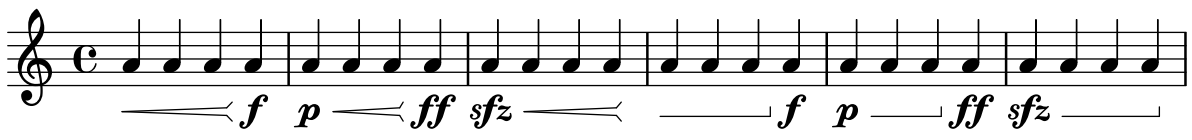
```
\relative c'' {
  \override TextScript.avoid-slur = #'inside
  \override TextScript.outside-staff-priority = ##f
  c2(^\markup { \halign #-10 \natural } d4.) c8
}
```
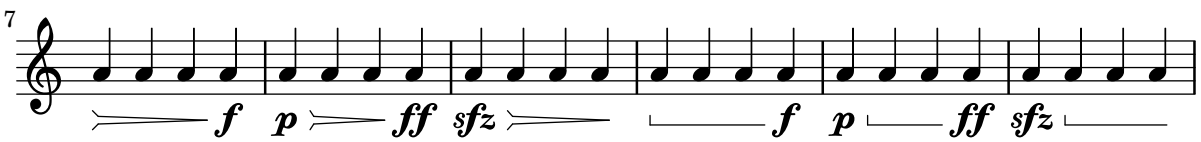


Différents styles de soufflet

Les soufflets de nuance peuvent adopter des styles différents.

```
\relative c'' {
  \override Hairpin.stencil = #flared-hairpin
  a4\< a a a\f
  a4\p\< a a a\ff
  a4\sfz\< a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4\< a a a\f
  a4\p\< a a a\ff
  a4\sfz\< a a a\!
  \override Hairpin.stencil = #flared-hairpin
  a4\> a a a\f
  a4\p\> a a a\ff
  a4\sfz\> a a a\!
  \override Hairpin.stencil = #constante-hairpin
  a4\> a a a\f
  a4\p\> a a a\ff
  a4\sfz\> a a a\!
}
```

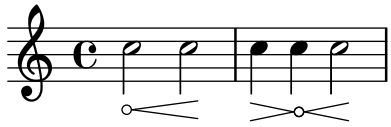




Impression de soufflets « al niente »

Des crescendos ou decrescendos *al niente* peuvent être indiqués de manière graphique, en assignant *vrai* (`#t`) à la propriété `circled-tip`, ce qui affiche un cercle à leur extrémité.

```
\relative c'' {
  \override Hairpin.circled-tip = ##t
  c2\< c\!
  c4\> c\< c2\!
}
```



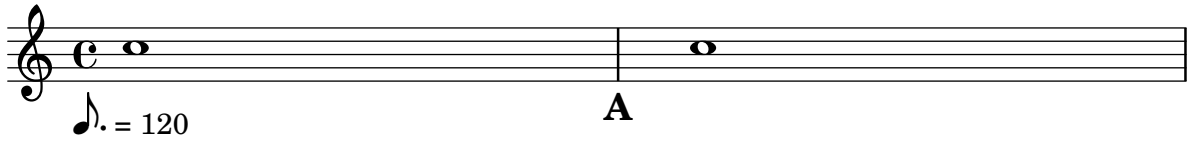
Impression du métronome et des repères sous la portée

Les indications de tempo et les marques de repère s'impriment par défaut au-dessus de la portée. Le fait de régler en conséquence la propriété `direction` des objets `MetronomeMark` et `RehearsalMark` les placera au-dessous de la portée.

```
\layout {
  indent = 0
  ragged-right = ##f
}

{
  % Metronome marks below the staff
  \override Score.MetronomeMark.direction = #DOWN
  \tempo 8. = 120
  c''1

  % Rehearsal marks below the staff
  \override Score.RehearsalMark.direction = #DOWN
  \mark \default
  c''1
}
```

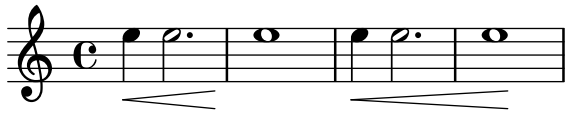


Soufflets et barres de mesure

En principe, un soufflet – (de)crescendo imprimé sous forme graphique – commence au bord gauche de la note de départ, et se termine au bord droit de la note d'arrivée. Cependant, si la note d'arrivée est sur un premier temps, le soufflet s'arrêtera au niveau de la barre de mesure qui la précède. Ce comportement peut être annulé en assignant *faux* (`#f`) à la propriété `to-barline`.

```
\relative c'' {
  e4\< e2.
  e1\!
  \override Hairpin.to-barline = ##f
  e4\< e2.
```

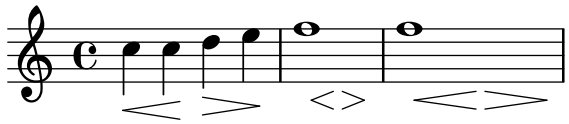
```
  e1\!
}
```



Ajustement de la longueur d'un soufflet

Si un soufflet est trop court, il suffit d'ajuster la propriété `minimum-length` de l'objet `Hairpin` pour l'allonger.

```
\relative c'' {
  c4\< c\! d\> e\!
  << f1 { s4 s\< s\> s\! } >>
  \override Hairpin.minimum-length = #5
  << f1 { s4 s\< s\> s\! } >>
}
```



Impression d'une même articulation des deux côtés d'une note ou d'un accord

LilyPond ne permet pas, par défaut, qu'une même articulation (accent, flageolet, point d'orgue, etc.) se retrouve à la fois au-dessus et au-dessous d'une note. Par exemple, `c4_\fermata^\fermata` ne donnera qu'un seul point d'orgue en dessous du do ; celui du dessus sera tout bonnement ignoré.

On peut néanmoins accoler des scripts, tels des doigtés, à l'intérieur d'un accord ; il peut donc y avoir autant d'articulations que de besoin, ce qui, par voie de conséquence, permet de s'affranchir de la présence des hampes et de positionner l'articulation relativement à la tête de note comme dans le cas du flageolet ci-dessous. L'imitation du traitement d'un script externe à un accord requérant un `'add-stem-support` demande de libeller la note comme étant un accord et d'ajouter les articulations au sein de la construction `<...>`.

Un simple amendement permettra de rectifier le positionnement habituel en surplomb : `<c-\tweak direction #DOWN-\fermata^\fermata>`

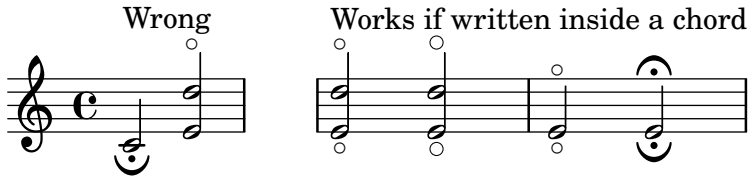
```
% The same as \flageolet, just a little smaller
smallFlageolet = \tweak font-size #-2 \flageolet

\relative c' {
  <>^"Wrong"
  c2_\fermata^\fermata % The second fermata is ignored!
  <e d'>2^\smallFlageolet_\smallFlageolet

  \stopStaff s1 \startStaff

  <>^"Works if written inside a chord"
  <e_\smallFlageolet d'^\smallFlageolet>2
  <e_\flageolet d'^\flageolet>2
  <e_\smallFlageolet^\smallFlageolet>2
```

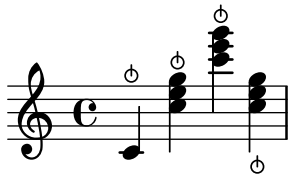
```
  <e_\fermata^\fermata>2
}
```



Slap ou pizzicato Bartok

Un *slap*, aussi appelé *pizzicato à la Bartók*, est un pizzicato où, au lieu de tirer la corde de côté comme à l'accoutumée, « on la pince en la soulevant plus fortement, et à la verticale, en la lâchant violemment. Celle-ci frappe alors la touche, et produit à la fois la note et le son percussif » (Wikipedia). Il s'indique par un cercle flanqué d'un trait vertical.
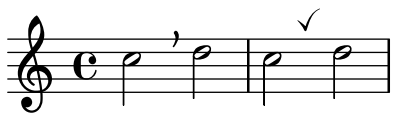
```
\relative c' {
  c4\snappizzicato
  <c' e g>4\snappizzicato
  <c' e g>4^\snappizzicato
  <c, e g>4_\snappizzicato
}
```



Remplacement du signe de respiration par une coche

Les musiques vocales ou pour vents utilisent souvent une coche en tant que signe de respiration. Ceci indique une respiration qui enlève une fraction à la note précédente plutôt qu'une véritable pause comme le fait un signe sous forme de virgule. La coche peut être remontée un peu afin de l'isoler de la portée.

```
\relative c'' {
  c2
  \breathe
  d2
  \override BreathingSign.Y-offset = #2.6
  \override BreathingSign.text =
    \markup { \musicglyph "scripts.tickmark" }
  c2
  \breathe
  d2
}
```

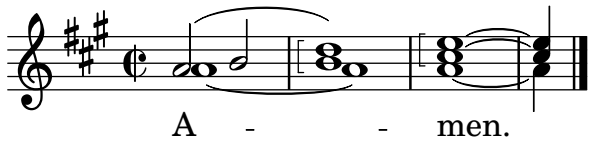


Utilisation d'un `arpeggioBracket` pour rendre les divisions plus évidentes

Un crochet d'arpège (`arpeggioBracket`) permet de mettre en évidence les divisions d'un pupitre en l'absence de hampe, comme on le voit régulièrement dans les partitions pour chœur.

```
\include "english.ly"

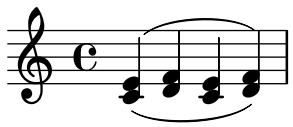
\score {
  \relative c'' {
    \key a \major
    \time 2/2
    <<
      \new Voice = "upper"
      <<
        { \voiceOne \arpeggioBracket
          a2( b2
          <b d>1\arpeggio)
          <cs e>\arpeggio ~
          <cs e>4
        }
        \addlyrics { \lyricmode { A -- men. } }
      >>
      \new Voice = "lower"
      { \voiceTwo
        a1 ~
        a
        a ~
        a4 \bar "|."
      }
    >>
  }
  \layout { ragged-right = ##t }
}
```



Accords et double liaison d'articulation

Certains compositeurs utilisent deux liaisons lorsqu'ils veulent lier des accords. Dans LilyPond, il faut pour cela activer la propriété `doubleSlurs`.

```
\relative c' {
  \set doubleSlurs = ##t
  <c e>4( <d f> <c e> <d f>)
}
```
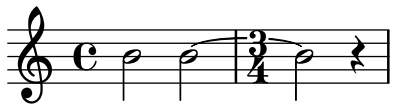


Utilisation de la propriété `whiteout`

Tout objet graphique peut s'imprimer sur un fond blanc afin de masquer une partie des objets qu'il recouvre. Ceci trouve toute son utilité pour améliorer certaines collisions, notamment dans des situations où un repositionnement d'objets est irréaliste. Il faut alors explicitement définir la propriété `layer` afin de contrôler quels objets seront masqués par le fond blanc.

Dans l'exemple ci-dessous, la liaison est en collision avec la métrique ; la situation est améliorée dès lors que la portion de liaison qui traverse la métrique est masquée par l'affectation de la propriété `whiteout` à l'objet `TimeSignature`. Pour ce faire, `TimeSignature` est déplacé sur un calque au-dessus de celui de `Tie` – il reste sur le calque par défaut (1) –, puis le `StaffSymbol` est placé sur un calque supérieur à celui de `TimeSignature` pour ne pas être masqué.

```
{
  \override Score.StaffSymbol.layer = #4
  \override Staff.TimeSignature.layer = #3
  b'2 b'~
  \once \override Staff.TimeSignature.whiteout = ##t
  \time 3/4
  b' r4
}
```
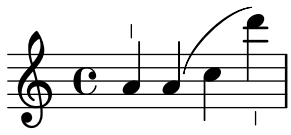


Articulation baroque en forme de coche

On trouve régulièrement, en musique baroque, cette courte ligne verticale. Sa signification peut varier, mais elle indique le plus souvent une note plus « appuyée ». Voici comment générer ce signe particulier.

```
upline =
\tweak stencil
  #(lambda (grob)
    (grob-interpret-markup grob #{ \markup \draw-line #'(0 . 1) #}))
  \stopped

\relative c' {
  a'4^\upline a( c d')_\upline
}
```

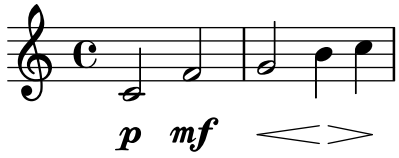


Alignement vertical des nuances sur plusieurs notes

Des nuances qui commencent, finissent ou se produisent sur une même note auront le même alignement vertical. L'augmentation de la propriété `staff-padding` de l'objet `DynamicLineSpanner` permet d'aligner différentes nuances affectées à différentes notes.

```
\relative c' {
  \override DynamicLineSpanner.staff-padding = #4
  c2\p f\mf
```

```
  g2\< b4\> c\!
}
```



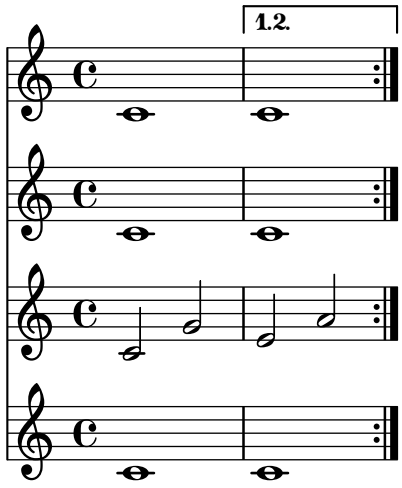
Repeats

Section “Repeats” dans *Manuel de notation*

Ajout du crochet de reprise à d’autres portées

D’ordinaire, le graveur `Volta_engraver` réside dans le contexte `Score` ; les crochets précédant une reprise s’impriment donc seulement au-dessus de la portée du haut. On peut ajuster cela en déplaçant ce graveur vers les contextes de portée (`Staff`) qui doivent comporter ces crochets.

```
<<
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
  \new Staff \with { \consists "Volta_engraver" } { c'2 g' e' a' }
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
>>
```



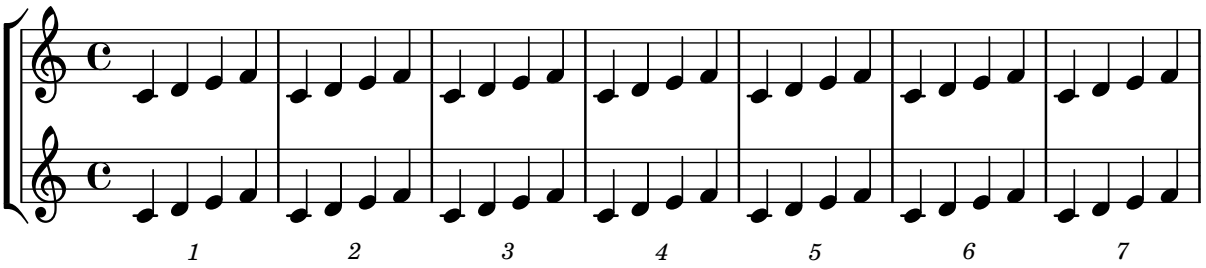
Numéros de mesure centrés

Les partitions pour grands ensembles disposent souvent de la numérotation des mesures centrée en dessous du système. L’extrait ci-dessous indique comment le `Measure_counter_engraver` permet de répondre à cette pratique. Le graveur est ici ajouté à un contexte `Dynamics`.

```
\layout {
  \context {
    \Dynamics
    \consists #Measure_counter_engraver
    \override MeasureCounter.direction = #DOWN
    \override MeasureCounter.font-encoding = #'latin1
    \override MeasureCounter.font-shape = #'italic
    % to control the distance of the Dynamics context from the staff:
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing.padding = #2
  }
  \context {
    \Score
    \remove "Bar_number_engraver"
  }
}
```

```
pattern = \repeat unfold 7 { c'4 d' e' f' }

\new StaffGroup <<
  \new Staff {
    \pattern
  }
  \new Staff {
    \pattern
  }
  \new Dynamics {
    \startMeasureCount
    s1*7
    \stopMeasureCount
  }
>>
```



Modification des barres de mesure par défaut

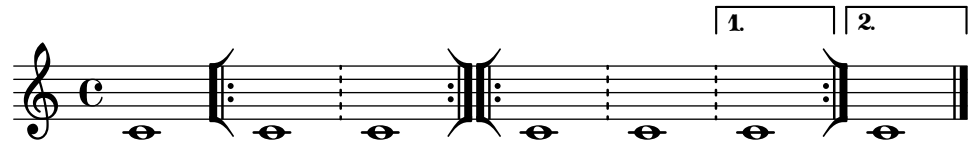
Les barres de mesure par défaut peuvent se modifier grâce à leur redéfinition au sein d'un contexte `Score`.

```
% http://lsr.di.unimi.it/LSR/Item?id=964
%%=> http://lists.gnu.org/archive/html/lilypond-user/2014-03/msg00126.html
%%=> http://lilypond.1069038.n5.nabble.com/Changing-the-default-end-repeat-bracket-tc169357

\layout {
  \context {
    \Score
    %% Changing the defaults from engraver-init.ly
    defaultBarType = #"!"
    startRepeatType = #"[|:"
    endRepeatType = #":|]"
    doubleRepeatType = #":|][|:"
  }
}

%% example:
{
  c'1
  \repeat volta 2 { \repeat unfold 2 c' }
  \repeat volta 2 { \repeat unfold 2 c' }
  \alternative {
    { c' }
    {
```
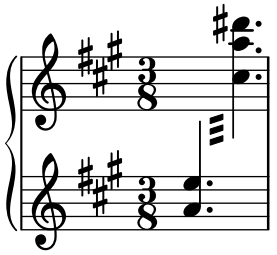
```
      %% v2.18 workaround
      \once\override Score.VoltaBracket.shorten-pair = #'(1 . -1)
      c'
    }
  }
  \bar "|."
}
```



Trémolo et changement de portée

Dans la mesure où `\repeat tremolo` requiert deux arguments musicaux pour un trémolo d'accords, la note ou l'accord de la portée opposée doit être encadré par des accolades et se voir adjoindre la commande `\change Staff`.

```
\new PianoStaff <<
  \new Staff = "up" \relative c'' {
    \key a \major
    \time 3/8
    s4.
  }
  \new Staff = "down" \relative c'' {
    \key a \major
    \time 3/8
    \voiceOne
    \repeat tremolo 6 {
      <a e'>32
      {
        \change Staff = "up"
        \voiceTwo
        <cis a' dis>32
      }
    }
  }
>>
```
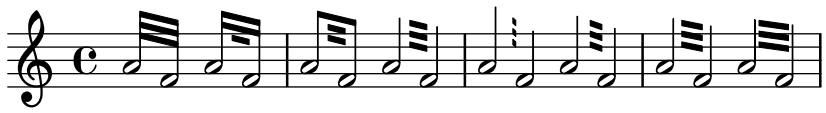


Impression de trémolo avec ligature flottante

Lorsque la durée totale d'un trémolo est inférieure à la noire, égale une blanche, ou bien entre une blanche et une ronde, il est d'usage que toutes les ligatures soient en contact avec les hampes. Certains styles de gravure font cependant apparaître des ligatures détachées, centrées

entre les hampes. Pour ce type de trémolo, le nombre de hampes flottantes se gère au travers de la propriété `gap-count` de l'objet `Beam`, et l'écart entre ligature et hampe se définit par la propriété `gap`.

```
\relative c'' {
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #1
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #2
  \repeat tremolo 8 { a32 f }
  \override Beam.gap-count = #3
  \repeat tremolo 8 { a32 f }

  \override Beam.gap-count = #3
  \override Beam.gap = #1.33
  \repeat tremolo 8 { a32 f }
  \override Beam.gap = #1
  \repeat tremolo 8 { a32 f }
  \override Beam.gap = #0.67
  \repeat tremolo 8 { a32 f }
  \override Beam.gap = #0.33
  \repeat tremolo 8 { a32 f }
}
```
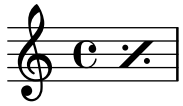


Répétition en pourcent isolée

Des symboles de pourcentage isolés peuvent aussi être obtenus au moyen d'un silence multimesure dont on modifie l'aspect :

```
makePercent =
#(define-music-function (note) (ly:music?)
   "Make a percent repeat the same length as NOTE."
   (make-music 'PercentEvent
               'length (ly:music-length note)))

\relative c'' {
  \makePercent s1
}
```
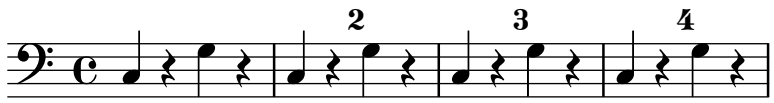


Compteur de mesures

Le code ci-dessous permet, de façon détournée, d'afficher un compteur de mesures à l'aide d'une répétition en pourcent rendue transparente.

```
<<
  \context Voice = "foo" {
```

```
    \clef bass
    c4 r g r
    c4 r g r
    c4 r g r
    c4 r g r
  }
  \context Voice = "foo" {
    \set countPercentRepeats = ##t
    \hide PercentRepeat
    \override PercentRepeatCounter.staff-padding = #1
    \repeat percent 4 { s1 }
  }
>>
```



Numérotation de groupes de mesures

Cet extrait illustre l'utilisation du `Measure_counter_engraver` aux fins de numéroter un groupe de mesures successives. La section à numéroter peut éventuellement comporter des reprises.

Le graveur doit être ajouté au contexte approprié. Il est ici attaché au contexte `Staff`, mais pourrait l'être à un contexte `Dynamics`.

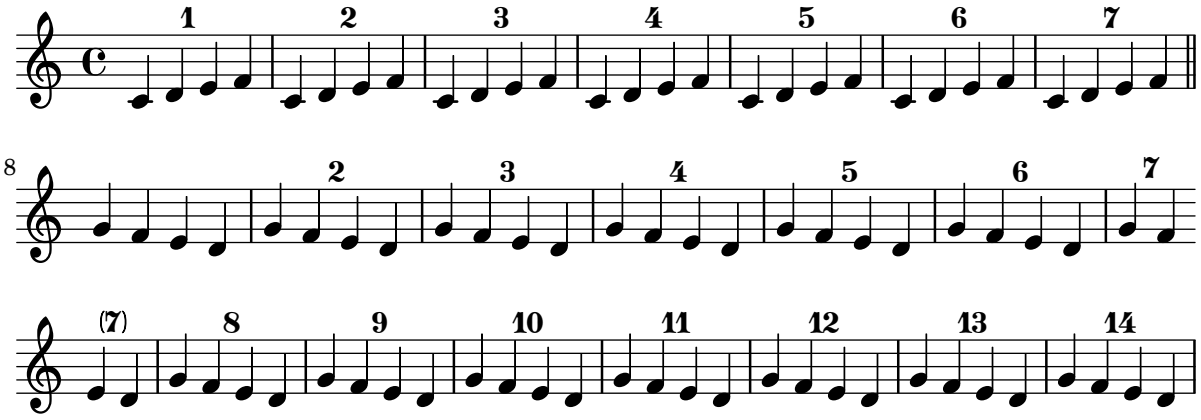
Le compteur est lancé par `\startMeasureCount` et arrêté par `\stopMeasureCount`. La numérotation commence à `1` par défaut, comportement qui peut se modifier à l'aide de la propriété `count-from`.

Lorsqu'une mesure s'étend après un saut de ligne, le numéro y est alors répété, entre parenthèses.

```
\layout {
  \context {
    \Staff
    \consists #Measure_counter_engraver
  }
}

\new Staff {
  \startMeasureCount
  \repeat unfold 7 {
    c'4 d' e' f'
  }
  \stopMeasureCount
  \bar "||"
  g'4 f' e' d'
  \override Staff.MeasureCounter.count-from = #2
  \startMeasureCount
  \repeat unfold 5 {
    g'4 f' e' d'
  }
  g'4 f'
  \bar ""
  \break
```

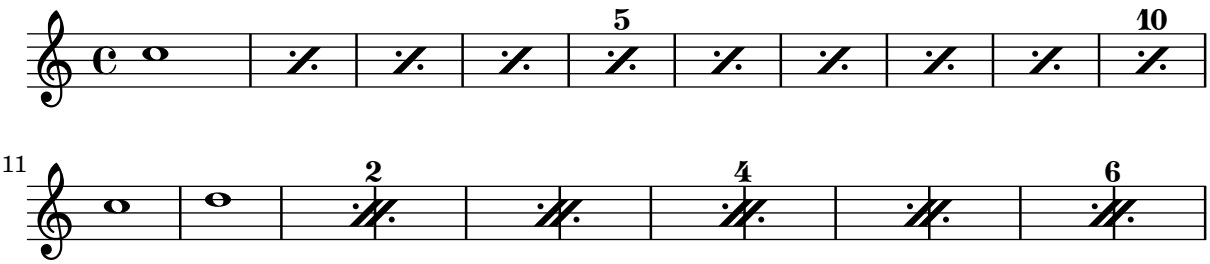
```
  e'4 d'
  \repeat unfold 7 {
    g'4 f' e' d'
  }
  \stopMeasureCount
}
```



Affichage du numéro de répétition en pourcent

Le numéro de mesure répétée sera imprimé à intervalle régulier si vous déterminez la propriété de contexte `repeatCountVisibility`.
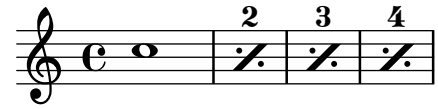
```
\relative c'' {
  \set countPercentRepeats = ##t
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
  \repeat percent 10 { c1 } \break
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
  \repeat percent 6 { c1 d1 }
}
```



Compteur de répétition en pourcent

Les répétitions de plus de deux mesures sont surmontées d'un compteur, si l'on active la propriété `countPercentRepeats` comme le montre l'exemple suivant :

```
\relative c'' {
  \set countPercentRepeats = ##t
  \repeat percent 4 { c1 }
}
```



Positionnement des segno et coda (avec saut de ligne)

Le code ci-dessous permet d'adjoindre à un signe *segno* un texte *D.S. al Coda*, là où se trouverait normalement un bout de portée. La *coda* entamera une nouvelle ligne. Une variante, indiquée ici même, permet de laisser la *coda* sur la même ligne.

```
{
  \clef treble
  \key g \major
  \time 4/4
  \relative c'' {
    \repeat unfold 4 {
      c4 c c c
    }

    % Set segno sign as rehearsal mark and adjust size if needed
    % \once \override Score.RehearsalMark.font-size = #3
    \mark \markup { \musicglyph "scripts.segno" }
    \repeat unfold 2 {
      c4 c c c
    }

    % Set coda sign as rehearsal mark and adjust size if needed
    \once \override Score.RehearsalMark.font-size = #4
    \mark \markup { \musicglyph "scripts.coda" }
    \repeat unfold 2 {
      c4 c c c
    }

    % Should Coda be on anew line?
    % Coda NOT on new line: use \nobreak
    % Coda on new line: DON'T use \nobreak
    % \noBreak

    \bar "||"

    % Set segno sign as rehearsal mark and adjust size if needed
    \once \override Score.RehearsalMark.break-visibility =
      #begin-of-line-invisible
    % \once \override Score.RehearsalMark.font-size = #3
    \mark \markup { \musicglyph "scripts.segno" }

    % Here begins the trickery!
    % \cadenzaOn will suppress the bar count
    % and \stopStaff removes the staff lines.
    \cadenzaOn
    \stopStaff
        % Some examples of possible text-displays

        % text line-aligned
        % ==================
        % Move text to the desired position
        % \once \override TextScript.extra-offset = #'( 2 . -3.5 )
```

```
    % | <>^\markup { D.S. al Coda } }

    % text center-aligned
    % ====================
    % Move text to the desired position
    % \once \override TextScript.extra-offset = #'( 6 . -5.0 )
    % | <>^\markup { \center-column { D.S. "al Coda" } }

    % text and symbols center-aligned
    % ===============================
    % Move text to the desired position
    % and tweak spacing for optimum text alignment
    \repeat unfold 1 {
      s1
      \bar ""
    }
    \once \override TextScript.extra-offset = #'( 0 . -3.0 )
    \once \override TextScript.word-space = #1.5
    <>^\markup { \center-column { "D.S. al Coda"
      \line {
        \musicglyph "scripts.coda"
        \musicglyph "scripts.tenuto"
        \musicglyph "scripts.coda"} } }

    % Increasing the unfold counter will expand the staff-free space
    \repeat unfold 3 {
      s1
      \bar ""
    }
    % Resume bar count and show staff lines again
\startStaff
\cadenzaOff

% Should Coda be on new line?
% Coda NOT on new line: DON'T use \break
% Coda on new line: use \break
\break

% Show up, you clef and key!
\once \override Staff.KeySignature.break-visibility = #end-of-line-invisible
\once \override Staff.Clef.break-visibility = #end-of-line-invisible

% Set coda sign as rehearsal mark and adjust size and position

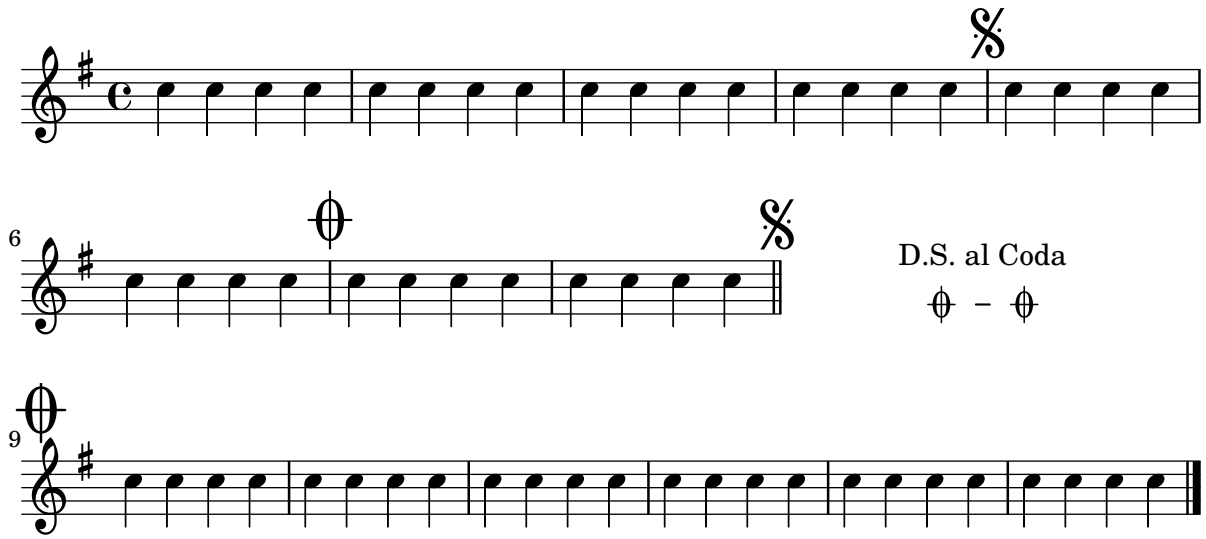
% Put the coda sign on top of the (treble-)clef
% depending on coda's line-position

% Coda NOT on new line, use this:
% \once \override Score.RehearsalMark.extra-offset = #'( -2 . 1.75 )

% Coda on new line, use this:
\once \override Score.RehearsalMark.extra-offset = #'( -5 . .5 )
```

```
    \once \override Score.RehearsalMark.font-size = #5
    \mark \markup { \musicglyph "scripts.coda" }

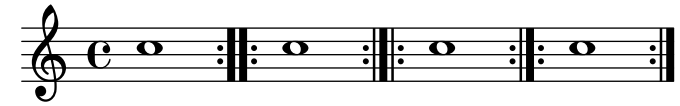
    % The coda
    \repeat unfold 6 {
      c4 c c c
    }
    \bar"|."
  }
}
```



Succession de reprises et style de barre par défaut

LilyPond dispose de trois différents styles de barre pour indiquer une succession de reprises. Vous devez opter pour un style par défaut, à l'aide de la propriété `doubleRepeatType`.

```
\relative c'' {
  \repeat volta 1 { c1 }
  \set Score.doubleRepeatType = #":..:"
  \repeat volta 1 { c1 }
  \set Score.doubleRepeatType = #":|.|:"
  \repeat volta 1 { c1 }
  \set Score.doubleRepeatType = #":|.:"
  \repeat volta 1 { c1 }
}
```
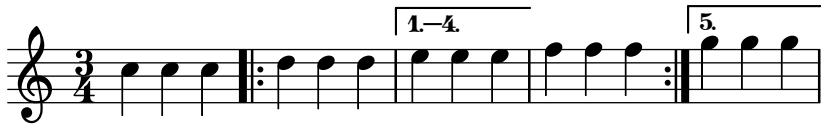


Diminution de la taille du crochet d'alternative

Les crochets indiquant les fins alternatives s'étalent tout au long de celles-ci. On peut les raccourcir en jouant sur la propriété `voltaSpannerDuration`. Dans l'exemple suivant, le crochet ne se prolonge que sur une mesure à 3/4.

```
\relative c'' {
```

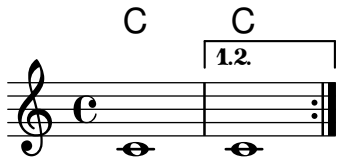
```
  \time 3/4
  c4 c c
  \set Score.voltaSpannerDuration = #(ly:make-moment 3/4)
  \repeat volta 5 { d4 d d }
  \alternative {
    {
      e4 e e
      f4 f f
    }
    { g4 g g }
  }
}
```



Crochet de reprise sous les chiffrages d'accord

L'ajout du `Volta_engraver` à la bonne portée permet d'imprimer les crochets de reprise entre les chiffrages et la portée.

```
\score {
  <<
    \chords {
      c1
      c1
    }
    \new Staff \with {
      \consists "Volta_engraver"
    }
    {
      \repeat volta 2 { c'1 }
      \alternative { c' }
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_engraver"
    }
  }
}
```

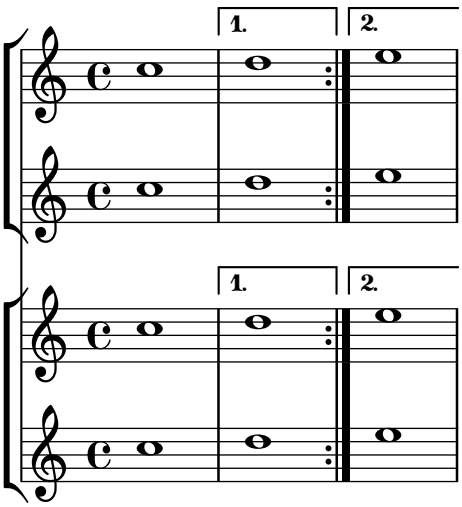


Volta multiportée

L'ajout du `Volta_engraver` à la portée appropriée permet de répéter les indications de reprise qui normalement n'apparaissent que sur la portée supérieure de la partition.

```
voltaMusic = \relative c'' {
  \repeat volta 2 {
    c1
  }
  \alternative {
    d1
    e1
  }
}

<<
  \new StaffGroup <<
    \new Staff \voltaMusic
    \new Staff \voltaMusic
  >>
  \new StaffGroup <<
    \new Staff \with { \consists "Volta_engraver" }
      \voltaMusic
    \new Staff \voltaMusic
  >>
>>
```



Indication de reprise avec texte grâce à `repeatCommands`
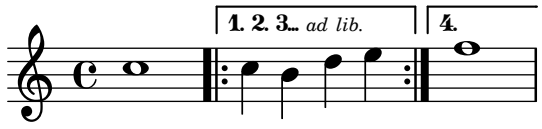
La commande `\repeat volta` permet d'indiquer facilement des reprises. Il est certains cas où l'adjonction d'un texte sous forme de `\markup` nécessite cependant de recourir à la propriété de contexte `repeatCommands`.

Dans la mesure où l'argument de `repeatCommands` est constitué d'une liste, le plus simple est de définir le texte dans une variable qui sera ensuite incorporée dans la liste en respectant la syntaxe Scheme `#(list (list 'volta texteIdentificateur))`. Les commandes de début et de fin de reprise peuvent s'ajouter séparément à la liste des éléments.

```
voltaAdLib = \markup { 1. 2. 3... \text \italic { ad lib. } }

\relative c'' {
  c1
  \set Score.repeatCommands = #(list (list 'volta voltaAdLib) 'start-repeat)
```

```
  c4 b d e
  \set Score.repeatCommands = #'((volta #f) (volta "4.") end-repeat)
  f1
  \set Score.repeatCommands = #'((volta #f))
}
```



Simultaneous notes
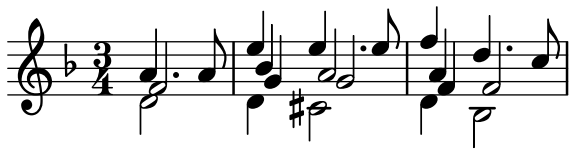
Section “Simultaneous notes” dans *Manuel de notation*

Ajout de voix pour éviter les collisions

Dans certains cas de musique polyphonique complexe, une voix supplémentaire peut permettre d’éviter les risques de collision. Lorsque quatre voix parallèles ne suffisent pas, la fonction Scheme `context-spec-music` permet d’ajouter encore d’autres voix.

```
voiceFive = #(context-spec-music (make-voice-props-set 4) 'Voice)

\relative c'' {
  \time 3/4
  \key d \minor
  \partial 2
  <<
    \new Voice  {
      \voiceOne
      a4. a8
      e'4 e4. e8
      f4 d4. c8
    }
    \new Voice {
      \voiceTwo
      d,2
      d4 cis2
      d4 bes2
    }
    \new Voice {
      \voiceThree
      f'2
      bes4 a2
      a4 s2
    }
    \new Voice {
      \voiceFive
      s2
      g4 g2
      f4 f2
    }
  >>
}
```

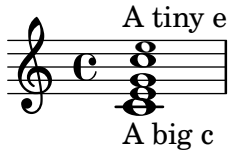


Modification de la taille d’une note particulière d’un accord

La commande `\tweak` permet de modifier la propriété `font-size` d’une note particulière d’un accord.

Il s'agit de placer, à l'intérieur même de l'accord (dans la construction `< >`) et avant la note considérée, l'instruction `\tweak` suivie de `font-size` et de définir la taille voulue – comme `#-2` pour une petite tête.
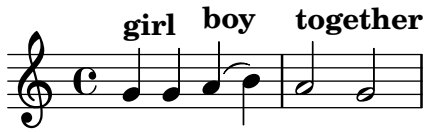
```
\relative c' {
  <\tweak font-size #+2 c e g c
   \tweak font-size #-2 e>1
   ^\markup { A tiny e }_\markup { A big c }
}
```



Modification des indications de parties combinées

Lorsque vous regroupez automatiquement des parties, vous pouvez modifier le texte qui sera affiché pour les solos et pour les parties à l'unisson :

```
\new Staff <<
  \set Staff.soloText = #"girl"
  \set Staff.soloIIText = #"boy"
  \set Staff.aDueText = #"together"
  \partCombine
    \relative c'' {
      g4 g r r
      a2 g
    }
    \relative c'' {
      r4 r a( b)
      a2 g
    }
>>
```

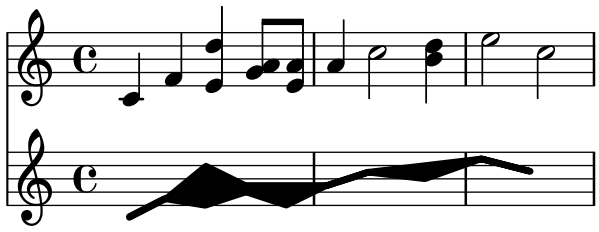


Clusters

Un cluster indique à l'instrumentiste qu'il doit jouer toute une suite de sons.

```
fragment = \relative c' {
  c4 f <e d'>4
  <g a>8 <e a> a4 c2 <d b>4
  e2 c
}

<<
  \new Staff \fragment
  \new Staff \makeClusters \fragment
>>
```



Combinaison de deux parties sur une même portée

L'outil de combinaison de parties (la commande `\partCombine`) permet d'avoir deux parties différentes sur une même portée. LilyPond ajoute automatiquement des indications textuelles, telles que « solo » ou « a2 ». Si votre intention n'est que de fusionner les parties, sans ajouter de texte, assignez faux à la propriété `printPartCombineTexts`. Dans le cas de partitions vocales, et plus particulièrement d'hymnes, ces « solo/a2 » ne sont d'aucune utilité, aussi vaut-il mieux les désactiver. Dans le cas où il y aurait alternance entre *solo* et *tutti*, il vaut mieux faire appel à de la musique polyphonique standard.

Voici trois moyens d'imprimer deux parties sur une même portée : en polyphonie normale, avec `\partCombine` sans indication supplémentaire, et avec `\partCombine` commentée.

```
%% Combining pedal notes with clef changes

musicUp = \relative c'' {
  \time 4/4
  a4 c4.( g8) a4 |
  g4 e' g,( a8 b) |
  c b a2.
}

musicDown = \relative c'' {
  g4 e4.( d8) c4 |
  r2 g'4( f8 e) |
  d2 \stemDown a
}

\score {
    <<
    \new Staff \with { instrumentName = "Standard polyphony" }
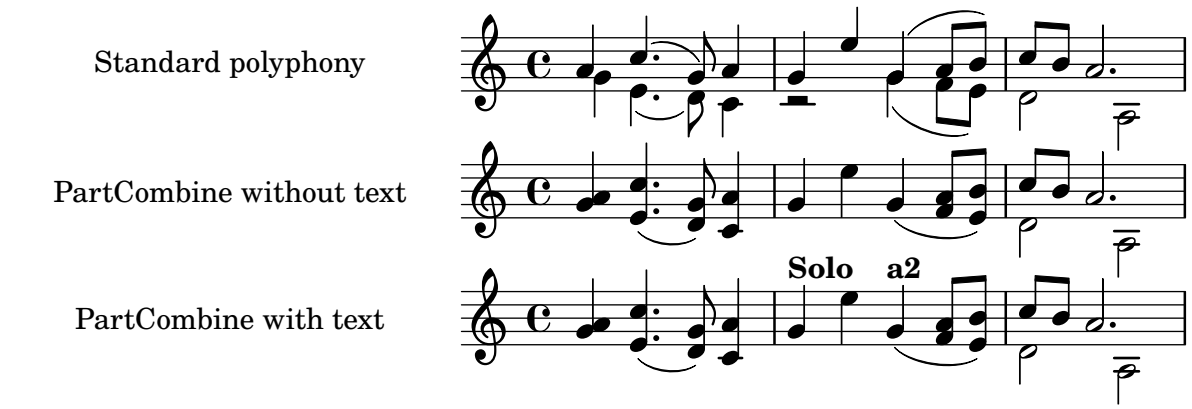
      << \musicUp \\ \musicDown >>

    \new Staff \with {
      instrumentName = "PartCombine without text"
      printPartCombineTexts = ##f
    }

    \partCombine \musicUp \musicDown

    \new Staff \with { instrumentName = "PartCombine with text" }
      \partCombine \musicUp \musicDown
    >>
  \layout {
    indent = 6.0\cm
    \context {
      \Score
      \override SystemStartBar.collapse-height = #30
```
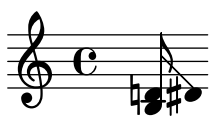
```
    }
  }
}
```



Impression d'accords complexes

Voici comment obtenir l'impression d'un accord au sein duquel une même note est jouée deux fois avec des altérations différentes.

```
fixA = {
  \once \override Stem.length = #11
}

fixB = {
  \once \override NoteHead.X-offset = #1.7
  \once \override Stem.length = #7
  \once \override Stem.rotation = #'(45 0 0)
  \once \override Stem.extra-offset = #'(-0.1 . -0.2)
  \once \override Flag.style = #'no-flag
  \once \override Accidental.extra-offset = #'(4 . -.1)
}


\relative c' {
  << { \fixA <b d!>8 } \\ { \voiceThree \fixB dis } >> s
}
```

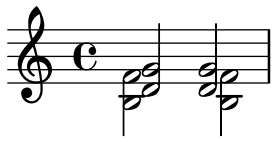


Décalage horizontal forcé

Quand LilyPond est dépassé, la propriété `force-hshift` de l'objet `NoteColumn` et des silences à hauteur déterminée peuvent s'avérer utiles pour dicter au programme les choix de placement. On travaille ici en espace de portée.

```
\relative c' <<
  {
    <d g>2 <d g>
  }
```

```
  \\
  {
    <b f'>2
    \once \override NoteColumn.force-hshift = #1.7
    <b f'>2
  }
>>
```
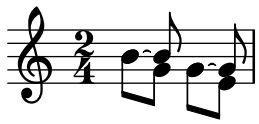


Recours à la propriété `'transparent` pour rendre des objets invisibles

Une modification de la propriété `transparent` permet d'imprimer un objet « à l'encre sympathique » : l'objet n'est pas affiché bien que tous ses attributs soient préservés. L'objet en question occupe donc sa place, est pris en compte lors de collisions, et peut se voir attaché liaisons ou ligatures.

Cet exemple illustre la manière de connecter deux voix par une liaison de prolongation. Les liaisons de prolongation ne peuvent normalement intervenir que dans la même voix. Dès lors que la liaison est entamée dans une autre voix et que la première hampe ascendante est rendue transparente dans cette même voix, la liaison semble passer d'une voix à l'autre.

```
\relative {
  \time 2/4
  <<
    {
      \once \hide Stem
      \once \override Stem.length = #8
      b'8 ~ 8\noBeam
      \once \hide Stem
      \once \override Stem.length = #8
      g8 ~ 8\noBeam
    }
    \\
    {
      b8 g g e
    }
  >>
}
```
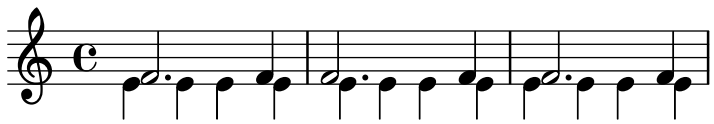


Déplacement des notes pointées dans une polyphonie

Une note pointée appartenant à la voix supérieure d'une portée polyphonique sera par défaut décalée vers la droite afin d'éviter les collisions avec les autres voix. Ce comportement peut être outrepassé à l'aide de la propriété `prefer-dotted-right` de `NoteCollision`.

```
\new Staff \relative c' <<
```

```
  {
    f2. f4
    \override Staff.NoteCollision.prefer-dotted-right = ##f
    f2. f4
    \override Staff.NoteCollision.prefer-dotted-right = ##t
    f2. f4
  }
  \\
  { e4 e e e e e e e e e e }
>>
```
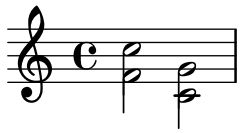


Suppression des avertissements de chevauchement

If notes from two voices with stems in the same direction are placed at the same position, and both voices have no shift or the same shift specified, the error message '`warning: ignoring too many clashing note columns`' will appear when compiling the LilyPond file. This message can be suppressed by setting the `'ignore-collision` property of the `NoteColumn` object to `#t`. Please note that this does not just suppress warnings but stops LilyPond trying to resolve collisions at all and so may have unintended results unless used with care.

```
ignore = \override NoteColumn.ignore-collision = ##t

\relative c' {
  \new Staff <<
    \new Voice { \ignore \stemDown f2 g }
    \new Voice { c2 \stemDown c, }
  >>
}
```



Deux \partCombine sur une même portée

La fonction `\partCombine` prend deux expressions musicales contenant chacune une partie et les distribue dans quatre contextes `Voice` nommés « two », « one », « solo » et « chords » selon le moment et la manière dont ces parties fusionnent en une voix commune. Les voix issues de `\partCombine` peuvent disposer de leurs propres propriétés de mise en forme définies de manière habituelle. Sont ici définies des extensions à `\partCombine` de sorte à faciliter le placement de quatre voix sur une seule portée.

```
soprano = { d'4 | cis'  b  e'  d'8 cis' | cis'2 b }
alto = { fis4 | e8 fis gis ais b4 b | b ais fis2 }
tenor = { a8 b | cis' dis' e'4 b8 cis' d'4 | gis cis' dis'2 }
bass = { fis8 gis | a4 gis g fis | eis fis b,2 }

\new Staff <<
  \key b\minor
```

```
  \clef alto
  \partial 4
  \transpose b b'
  \partCombineUp \soprano \alto
  \partCombineDown \tenor \bass
>>

\layout {
  \context {
    \Staff
    \accepts "VoiceBox"
  }
  \context {
    \name "VoiceBox"
    \type "Engraver_group"
    \defaultchild "Voice"
    \accepts "Voice"
    \accepts "NullVoice"
  }
}

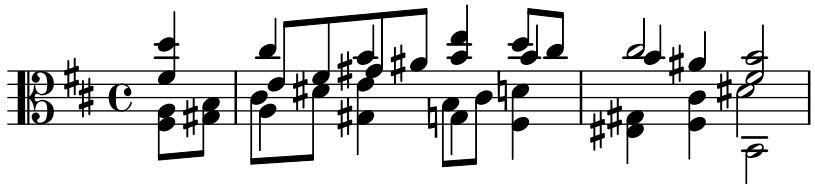
customPartCombineUp =
#(define-music-function (partOne partTwo)
  (ly:music? ly:music?)
"Take the music in @var{partOne} and @var{partTwo} and return
a @code{VoiceBox} named @q{Up} containing @code{Voice}s
that contain @var{partOne} and @var{partTwo} merged into one
voice where feasible.  This variant sets the default voicing
in the output to use upward stems."
#{
  \new VoiceBox = "Up" <<
    \context Voice = "one" { \voiceOne }
    \context Voice = "two" { \voiceThree }
    \context Voice = "shared" { \voiceOne }
    \context Voice = "solo" { \voiceOne }
    \context NullVoice = "null" {}
    \partCombine #partOne #partTwo
  >>
#})

customPartCombineDown = #
(define-music-function (partOne partTwo)
  (ly:music? ly:music?)
"Take the music in @var{partOne} and @var{partTwo} and return
a @code{VoiceBox} named @q{Down} containing @code{Voice}s
that contain @var{partOne} and @var{partTwo} merged into one
voice where feasible.  This variant sets the default voicing
in the output to use downward stems."
#{
  \new VoiceBox = "Down" <<
    \set VoiceBox.soloText = #"Solo III"
    \set VoiceBox.soloIIText = #"Solo IV"
```

```
    \context Voice ="one" { \voiceFour }
    \context Voice ="two" { \voiceTwo }
    \context Voice ="shared" { \voiceFour }
    \context Voice ="solo" { \voiceFour }
    \context NullVoice = "null" {}
    \partCombine #partOne #partTwo
  >>
#})

soprano = { d'4 | cis'  b  e'  d'8 cis' | cis'2 b }
alto = { fis4 | e8 fis gis ais b4 b | b ais fis2 }
tenor = { a8 b | cis' dis' e'4 b8 cis' d'4 | gis cis' dis'2 }
bass = { fis8 gis | a4 gis g fis | eis fis b,2 }

\new Staff <<
  \key b\minor
  \clef alto
  \partial 4
  \transpose b b'
  \customPartCombineUp \soprano \alto
  \customPartCombineDown \tenor \bass
>>
```
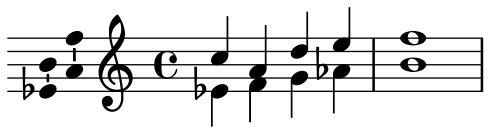


Staff notation

Section “Staff notation” dans *Manuel de notation*

Un ambitus par voix

L’`ambitus` peut être individualisé par voix. Il faut en pareil cas éviter qu’ils se chevauchent.

```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus.X-offset = #2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```

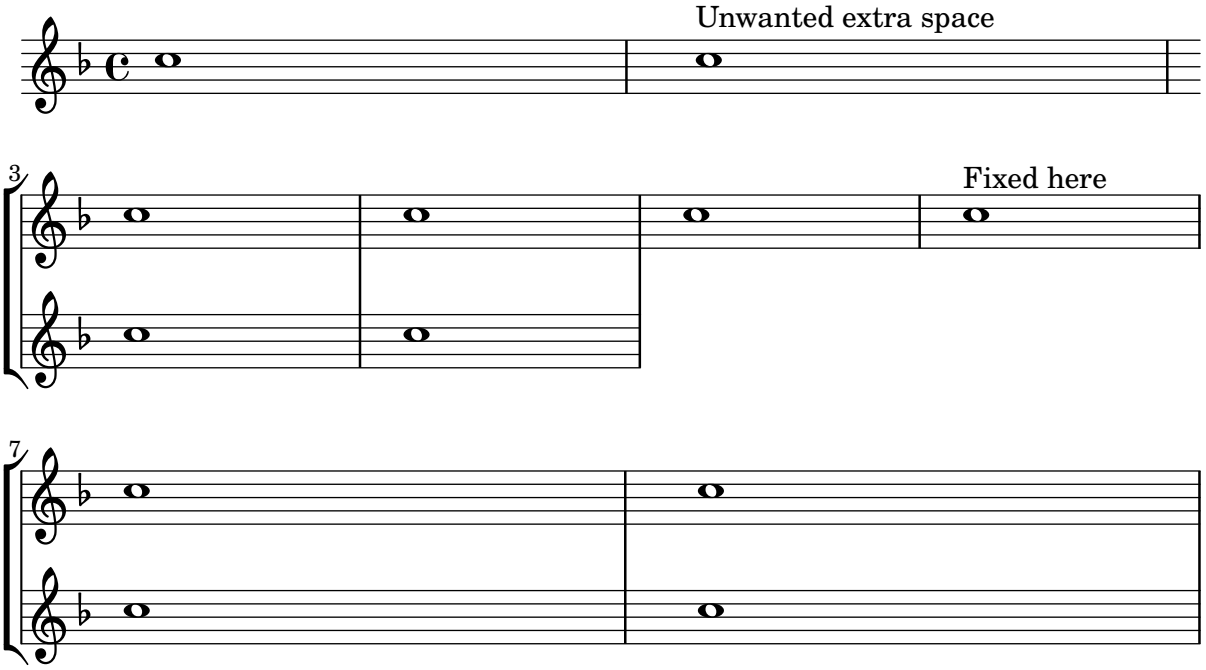


Ajout d’une portée supplémentaire après un saut de ligne

Lorsqu’une nouvelle portée vient s’ajouter après un saut de ligne, LilyPond préserve un espace juste avant le saut de ligne – pour un éventuel changement d’armure qui, quoi qu’il en soit, ne sera pas imprimé. L’astuce consiste alors, comme indiqué dans l’exemple suivant, à ajuster `Staff.explicitKeySignatureVisibility`.

```
\score {
  \new StaffGroup \relative c'' {
    \new Staff
    \key f \major
    c1 c^"Unwanted extra space" \break
    << { c1 | c }
       \new Staff {
         \key f \major
         \once \omit Staff.TimeSignature
         c1 | c
       }
    >>
    c1 | c^"Fixed here" \break
    << { c1 | c }
       \new Staff {
         \once \set Staff.explicitKeySignatureVisibility = #end-of-line-invisible
```

```
        \key f \major
        \once \omit Staff.TimeSignature
        c1 | c
      }
    >>
  }
}
```

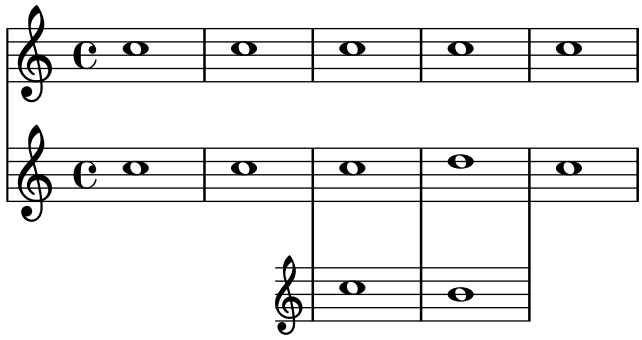


Ajout d'une portée supplémentaire

Vous pouvez ajouter une nouvelle portée, éventuellement de manière temporaire, après le début d'un morceau.

```
\score {
  <<
    \new Staff \relative c'' {
      c1 | c | c | c | c
    }
    \new StaffGroup \relative c'' {
      \new Staff {
        c1 | c
        <<
          {
            c1 | d
          }
          \new Staff {
            \once \omit Staff.TimeSignature
            c1 | b
          }
        >>
        c1
      }
    }
```

```
  >>
}
```



Ajout d'indicateurs à une portée dédoublée après un saut de ligne

Dans cet extrait sont définies les commandes `\splitStaffBarLine`, `convUpStaffBarLine` et `convDownStaffBarLine`. Ces commandes ajoutent une double flèche après la dernière barre de mesure d'une portée, indiquant par là que ses différentes voix disposeront de leur propre portée à la ligne suivante.

```
#(define-markup-command (arrow-at-angle layout props angle-deg length fill)
   (number? number? boolean?)
   (let* (
           (PI-OVER-180 (/ (atan 1 1) 34))
           (degrees->radians (lambda (degrees) (* degrees PI-OVER-180)))
           (angle-rad (degrees->radians angle-deg))
           (target-x (* length (cos angle-rad)))
           (target-y (* length (sin angle-rad))))
     (interpret-markup layout props
       (markup
        #:translate (cons (/ target-x 2) (/ target-y 2))
        #:rotate angle-deg
        #:translate (cons (/ length -2) 0)
        #:concat (#:draw-line (cons length 0)
                   #:arrow-head X RIGHT fill)))))


splitStaffBarLineMarkup = \markup \with-dimensions #'(0 . 0) #'(0 . 0) {
  \combine
  \arrow-at-angle #45 #(sqrt 8) ##t
  \arrow-at-angle #-45 #(sqrt 8) ##t
}

splitStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob splitStaffBarLineMarkup)
      0))
```

```
  \break
}

convDownStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . -.13)\arrow-at-angle #-45 #(sqrt 8) ##t
        }#})
      0))
  \break
}

convUpStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . .14)\arrow-at-angle #45 #(sqrt 8) ##t
        }#})
      0))
  \break
}


\paper {
  ragged-right = ##t
  short-indent = 10\mm
}

separateSopranos = {
  \set Staff.instrumentName = "AI AII"
  \set Staff.shortInstrumentName = "AI AII"
  \splitStaffBarLine
  \change Staff = "up"
}
convSopranos = {
  \convDownStaffBarLine
  \change Staff = "shared"
  \set Staff.instrumentName = "S A"
  \set Staff.shortInstrumentName = "S A"
}

sI = {
  \voiceOne
```

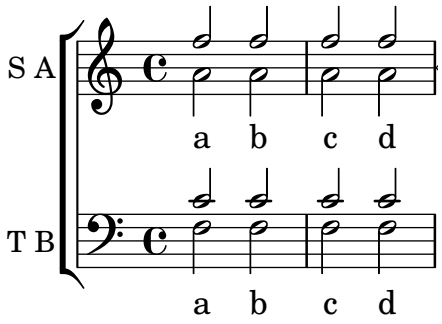
```
  \repeat unfold 4 f''2
  \separateSopranos
  \repeat unfold 4 g''2
  \convSopranos
  \repeat unfold 4 c''2
}
sII = {
  s1*2
  \voiceTwo
  \change Staff = "up"
  \repeat unfold 4 d''2
}
aI = {
  \voiceTwo
  \repeat unfold 4 a'2
  \voiceOne
  \repeat unfold 4 b'2
  \convUpStaffBarLine
  \voiceTwo
  \repeat unfold 4 g'2
}
aII = {
  s1*2
  \voiceTwo
  \repeat unfold 4 g'2
}
ten = {
  \voiceOne
  \repeat unfold 4 c'2
  \repeat unfold 4 d'2
  \repeat unfold 4 c'2
}
bas = {
  \voiceTwo
  \repeat unfold 4 f2
  \repeat unfold 4 g2
  \repeat unfold 4 c2
}

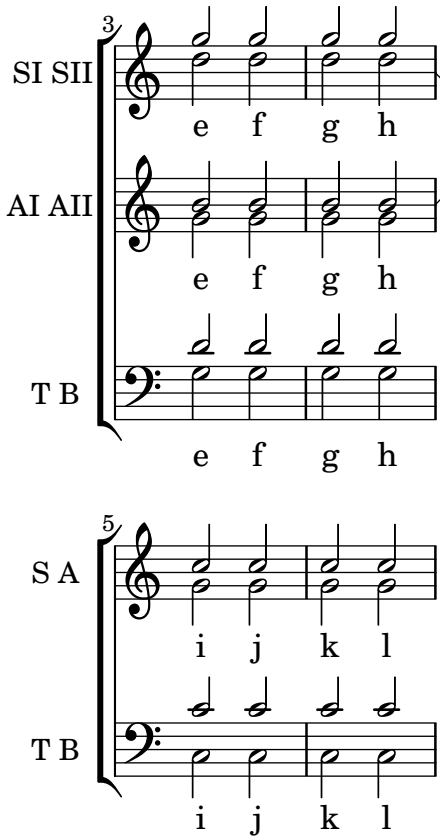
\score {
  <<
    \new ChoirStaff <<
      \new Staff = up \with {
        instrumentName = "SI SII"
        shortInstrumentName = "SI SII"
      } {
        s1*4
      }

      \new Staff = shared \with {
        instrumentName = "S A"
        shortInstrumentName = "S A"
```

```
      } <<
        \new Voice = sopI \sI
        \new Voice = sopII \sII
        \new Voice = altI \aI
        \new Voice = altII \aII
      >>
      \new Lyrics \with {
        alignBelowContext = up
      }
      \lyricsto sopII { e f g h }
      \new Lyrics \lyricsto altI { a b c d e f g h i j k l }

      \new Staff = men \with {
        instrumentName = "T B"
        shortInstrumentName = "T B"
      } <<
        \clef F
        \new Voice = ten \ten
        \new Voice = bas \bas
      >>
      \new Lyrics \lyricsto bas { a b c d e f g h i j k l }
    >>
  >>
  \layout {
    \context {
      \Staff \RemoveEmptyStaves
      \override VerticalAxisGroup.remove-first = ##t
    }
  }
}
```





Indication de l'instrument cité dans l'accompagnement d'une partition pour chœur

Lorsque le nombre d'instruments cités dans la réduction pour piano se multiplie, vous pourriez avoir intérêt à créer votre propre fonction pour gérer ces repères. La fonction musicale `\cueWhile` prend quatre arguments : la musique d'où provient la citation, telle que définie par `\addQuote`, le nom qui sera mentionné en regard de cette citation, son positionnement – `UP` ou `DOWN` selon qu'il sera attribué à `\voiceOne` et placé au-dessus ou `\voiceTwo` et placé en dessous – et enfin la musique du piano qui interviendra en parallèle. Le nom de l'instrument en question viendra s'aligner sur la gauche de la citation. Bien que vous puissiez effectuer plusieurs citations, elle ne peuvent être simultanées.

```
cueWhile =
#(define-music-function
   (instrument name dir music)
   (string? string? ly:dir? ly:music?)
   #{
     \cueDuring $instrument #dir {
       \once \override TextScript.self-alignment-X = #RIGHT
       \once \override TextScript.direction = $dir
       <>-\markup { \tiny #name }
       $music
     }
   #})
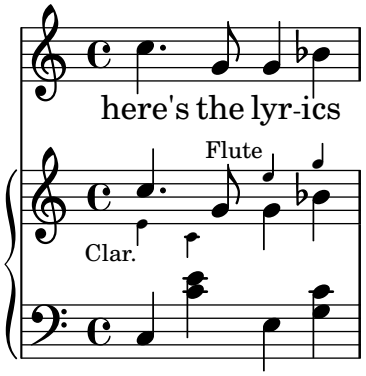
flute = \relative c'' {
  \transposition c'
  s4 s4 e g
}
\addQuote "flute" { \flute }
```

```
clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

singer = \relative c'' { c4. g8 g4 bes4 }
words = \lyricmode { here's the lyr -- ics }

pianoRH = \relative c'' {
  \transposition c'
  \cueWhile "clarinet" "Clar." #DOWN { c4. g8 }
  \cueWhile "flute" "Flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  <<
    \new Staff {
      \new Voice = "singer" {
        \singer
      }
    }
    \new Lyrics {
      \lyricsto "singer"
      \words
    }
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH
        }
      }
      \new Staff {
        \clef "bass"
        \pianoLH
      }
    >>
  >>
}
```



Ajout de marques temporelles à un long glissando

Lorsqu'un glissando s'étend dans la durée, on trouve parfois des indications temporelles, matérialisées par des hampes sans tête de note. De telles hampes permettent aussi d'indiquer des éléments intermédiaires.

L'alignement des hampes avec la ligne de glissando peut requérir quelques aménagements.
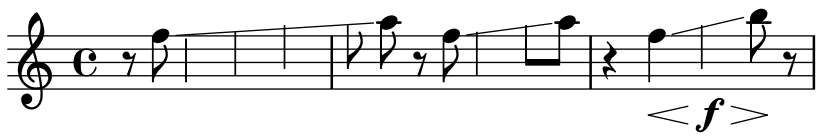
```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

glissandoSkipOff = {
  \revert NoteColumn.glissando-skip
  \undo \hide NoteHead
  \revert NoteHead.no-ledgers
}

\relative c'' {
  r8 f8\glissando
  \glissandoSkipOn
  f4 g a a8\noBeam
  \glissandoSkipOff
  a8

  r8 f8\glissando
  \glissandoSkipOn
  g4 a8
  \glissandoSkipOff
  a8 |

  r4 f\glissando \<
  \glissandoSkipOn
  a4\f \>
  \glissandoSkipOff
  b8\! r |
}
```
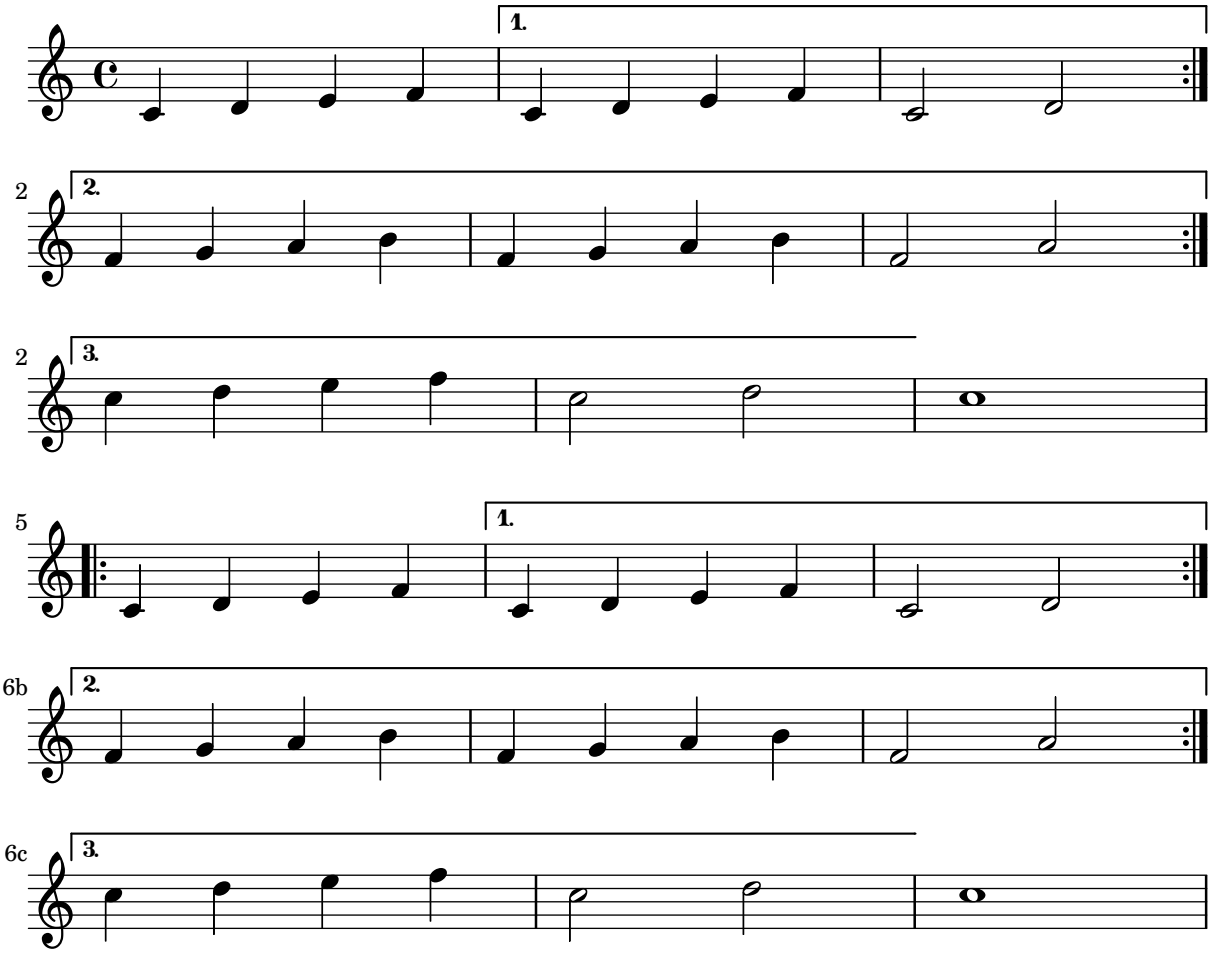


Numérotation des mesures et alternatives

Deux méthodes alternatives vous permettent de gérer la numérotation des mesures en cas de reprises.

```
\relative c'{
  \set Score.alternativeNumberingStyle = #'numbers
  \repeat volta 3 { c4 d e f | }
    \alternative {
      { c4 d e f | c2 d \break }
```

```
      { f4 g a b | f4 g a b | f2 a | \break }
      { c4 d e f | c2 d }
    }
  c1 \break
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \repeat volta 3 { c,4 d e f | }
    \alternative {
      { c4 d e f | c2 d \break }
      { f4 g a b | f4 g a b | f2 a | \break }
      { c4 d e f | c2 d }
    }
  c1
}
```
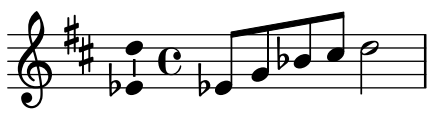


Ambitus après l'armure

L'ambitus se place par défaut à gauche de la clef. La fonction `\ambitusAfter` permet cependant de modifier ce positionnement ; sa syntaxe est `\ambitusAfter` *`grob-interface`* – voir Section "Graphical Object Interfaces" dans *Référence des propriétés internes* pour une liste des valeurs de *`grob-interface`* possibles. L'un des cas d'usage est d'insérer l'ambitus entre l'armure et la métrique.

```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
```

```
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```



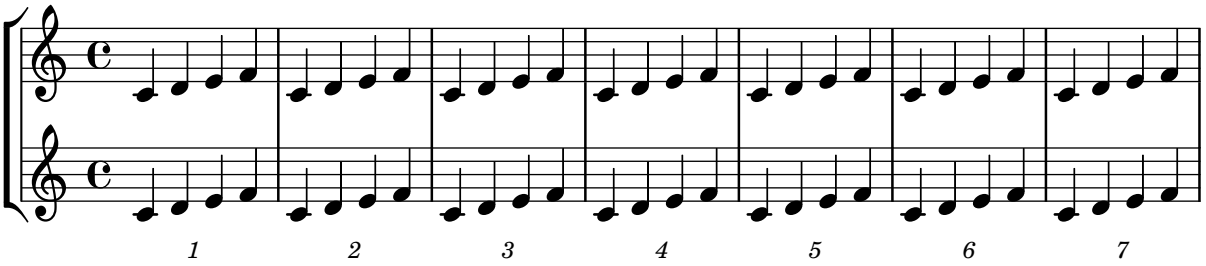
Numéros de mesure centrés

Les partitions pour grands ensembles disposent souvent de la numérotation des mesures centrée en dessous du système. L'extrait ci-dessous indique comment le `Measure_counter_engraver` permet de répondre à cette pratique. Le graveur est ici ajouté à un contexte `Dynamics`.

```
\layout {
  \context {
    \Dynamics
    \consists #Measure_counter_engraver
    \override MeasureCounter.direction = #DOWN
    \override MeasureCounter.font-encoding = #'latin1
    \override MeasureCounter.font-shape = #'italic
    % to control the distance of the Dynamics context from the staff:
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing.padding = #2
  }
  \context {
    \Score
    \remove "Bar_number_engraver"
  }
}

pattern = \repeat unfold 7 { c'4 d' e' f' }

\new StaffGroup <<
  \new Staff {
    \pattern
  }
  \new Staff {
    \pattern
  }
  \new Dynamics {
    \startMeasureCount
    s1*7
    \stopMeasureCount
  }
>>
```



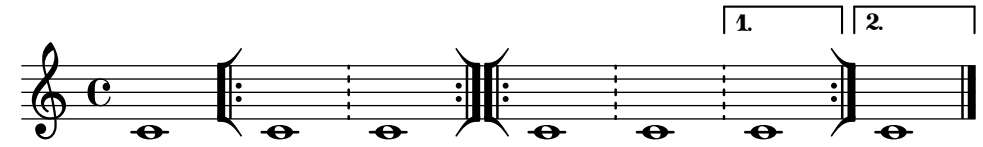
Modification des barres de mesure par défaut

Les barres de mesure par défaut peuvent se modifier grâce à leur redéfinition au sein d'un contexte `Score`.

```
% http://lsr.di.unimi.it/LSR/Item?id=964
%%=> http://lists.gnu.org/archive/html/lilypond-user/2014-03/msg00126.html
%%=> http://lilypond.1069038.n5.nabble.com/Changing-the-default-end-repeat-bracket-tc169357

\layout {
  \context {
    \Score
    %% Changing the defaults from engraver-init.ly
    defaultBarType = #"!"
    startRepeatType = #"[|:"
    endRepeatType = #":|]"
    doubleRepeatType = #":|][|:"
  }
}

%% example:
{
  c'1
  \repeat volta 2 { \repeat unfold 2 c' }
  \repeat volta 2 { \repeat unfold 2 c' }
  \alternative {
    { c' }
    {
      %% v2.18 workaround
      \once\override Score.VoltaBracket.shorten-pair = #'(1 . -1)
      c'
    }
  }
  \bar "|."
}
```



Modification du nombre de lignes de la portée

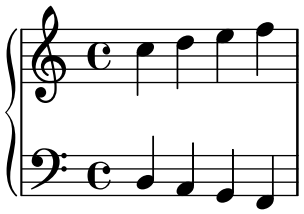
Le nombre de lignes d'une portée se modifie par adaptation de la propriété `line-count` du `StaffSymbol`.

```
upper = \relative c'' {
  c4 d e f
```

```
}

lower = \relative c {
  \clef bass
  c4 b a g
}

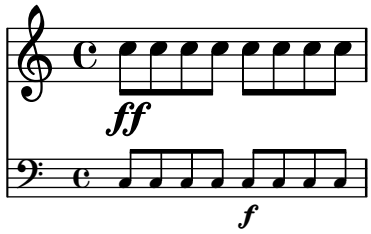
\score {
  \context PianoStaff <<
    \new Staff {
      \upper
    }
    \new Staff {
      \override Staff.StaffSymbol.line-count = #4
      \lower
    }
  >>
}
```



Modification de la taille d'une portée

Bien que le meilleur moyen de définir la taille des portées consiste à utiliser `#(set-global-staff-size xx)`, une portée en particulier peut se redimensionner en affectant d'un coefficient ses propriétés `staff-space` et `fontSize`.

```
<<
  \new Staff {
    \relative c'' {
      \dynamicDown
      c8\ff c c c c c c c
    }
  }
  \new Staff \with {
    fontSize = #-3
    \override StaffSymbol.staff-space = #(magstep -3)
  } {
    \clef bass
    c8 c c c c\f c c c
  }
>>
```



Impression de papier à musique

Rien n'est plus simple, pour obtenir une feuille blanche avec des portées, que de générer des mesures vides, de supprimer le `Bar_number_engraver` du contexte `Score`, et les `Time_signature_engraver`, `Clef_engraver` et `Bar_engraver` du contexte `Staff`.

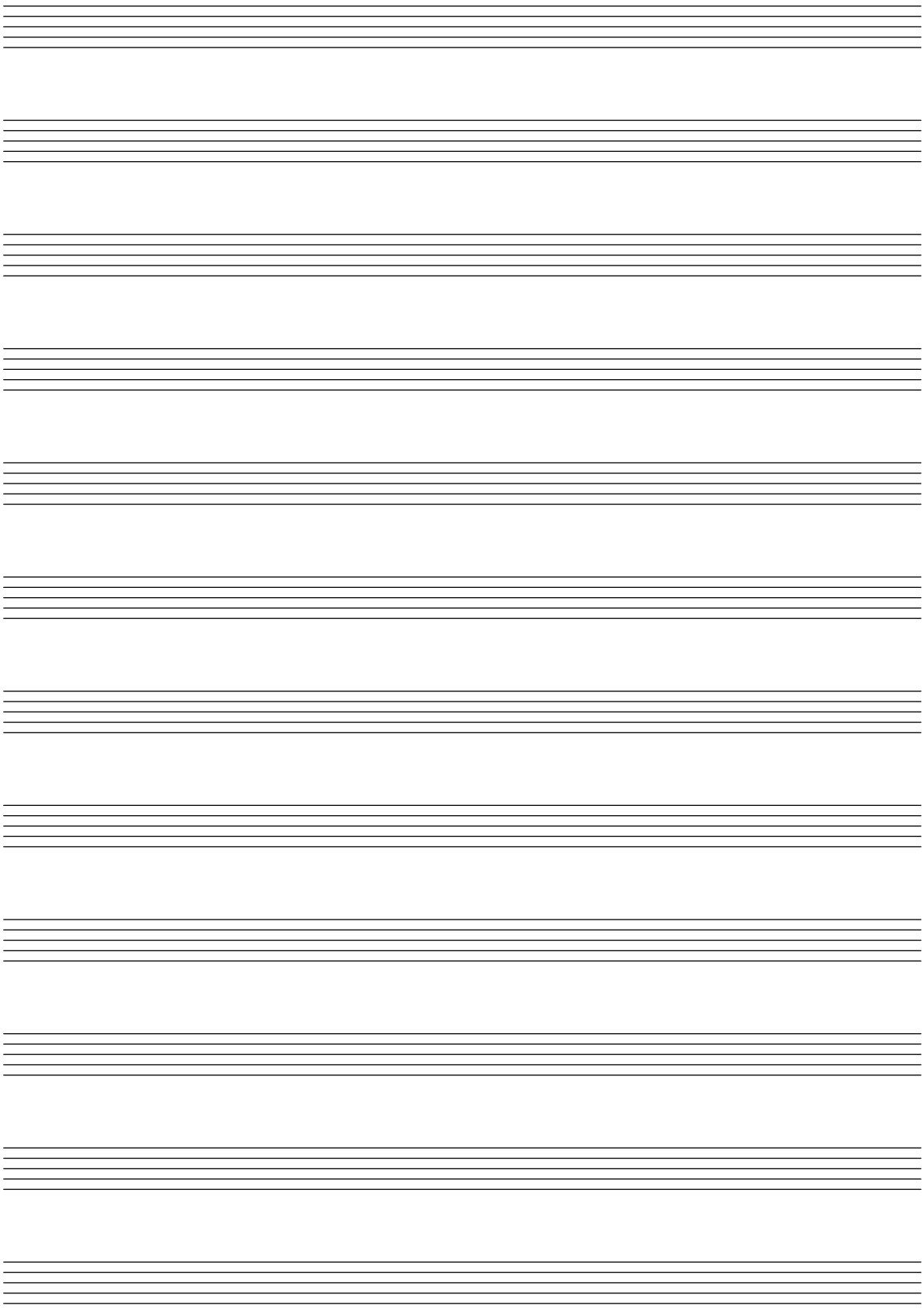
```
#(set-global-staff-size 20)

\score {
  {
    \repeat unfold 12 { s1 \break }
  }
  \layout {
    indent = 0\in
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Clef_engraver"
      \remove "Bar_engraver"
    }
    \context {
      \Score
      \remove "Bar_number_engraver"
    }
  }
}

% uncomment these lines for "letter" size
%{
\paper {
  #(set-paper-size "letter")
  ragged-last-bottom = ##f
  line-width = 7.5\in
  left-margin = 0.5\in
  bottom-margin = 0.25\in
  top-margin = 0.25\in
}
%}

% uncomment these lines for "A4" size
%{
\paper {
  #(set-paper-size "a4")
  ragged-last-bottom = ##f
  line-width = 180
  left-margin = 15
  bottom-margin = 10
  top-margin = 10
```

```
}
%}
```



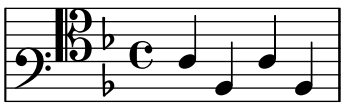
Création d'armures personnalisées

LilyPond accepte les armures personnalisées. Dans cet exemple, il s'agit d'une tonalité de ré mineur à laquelle on rajoute des bémols.

```
\new Staff \with {
  \override StaffSymbol.line-count = #8
  \override KeySignature.flat-positions = #'((-7 . 6))
  \override KeyCancellation.flat-positions = #'((-7 . 6))
  % presumably sharps are also printed in both octaves
  \override KeySignature.sharp-positions = #'((-6 . 7))
  \override KeyCancellation.sharp-positions = #'((-6 . 7))

  \override Clef.stencil = #
  (lambda (grob)(grob-interpret-markup grob
  #{ \markup\combine
    \musicglyph "clefs.C"
    \translate #'(-3 . -2)
    \musicglyph "clefs.F"
   #}))
    clefPosition = #3
    middleCPosition = #3
    middleCClefPosition = #3
}

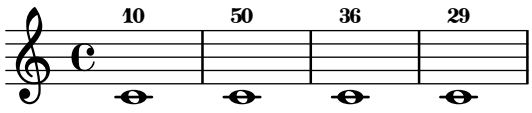
{
  \key d\minor
  f bes, f bes,
}
```



Création de doigtés sur deux caractères

Il est tout à fait possible de noter un doigté supérieur à 5.

```
\relative c' {
  c1-10
  c1-50
  c1-36
  c1-29
}
```



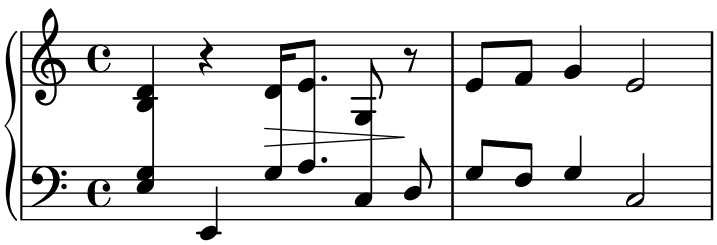
Hampes interportées

L'exemple ci-dessous illustre l'utilisation du `Span_stem_engraver` et de la commande `\crossStaff` afin de connecter des hampes entre les portées.

Nul n'est besoin de spécifier la taille des hampes ; le graveur calcule automatiquement la distance relative des têtes de note avec les portées.

```
\layout {
  \context {
    \PianoStaff
    \consists #Span_stem_engraver
  }
}

{
  \new PianoStaff <<
    \new Staff {
      <b d'>4 r d'16\> e'8. g8 r\!
      e'8 f' g'4 e'2
    }
    \new Staff {
      \clef bass
      \voiceOne
      \autoBeamOff
      \crossStaff { <e g>4 e, g16 a8. c8} d
      \autoBeamOn
      g8 f g4 c2
    }
  >>
}
```



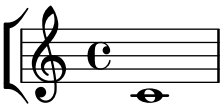
Indicateur de regroupement et portée unique

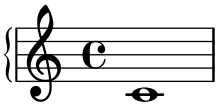
Lorsque, dans des regroupements de type `ChoirStaff` ou `StaffGroup`, une seule portée est active, aucune indication n'est donnée en début de ligne. Affecter à la propriété `collapse-height` un nombre de lignes inférieur à celui de la portée permet de modifier ce comportement par défaut.

Notez bien que dans le cas des `PianoStaff` et `GrandStaff`, pour lesquels le délimiteur de système est une accolade et non un crochet, il ne s'agit pas de la même propriété – voir le deuxième système de l'exemple.

```
\score {
  \new StaffGroup <<
    % Must be lower than the actual number of staff lines
    \override StaffGroup.SystemStartBracket.collapse-height = #4
    \override Score.SystemStartBar.collapse-height = #4
    \new Staff {
      c'1
    }
  >>
```

```
}
\score {
  \new PianoStaff <<
    \override PianoStaff.SystemStartBrace.collapse-height = #4
    \override Score.SystemStartBar.collapse-height = #4
    \new Staff {
      c'1
    }
  >>
}
```



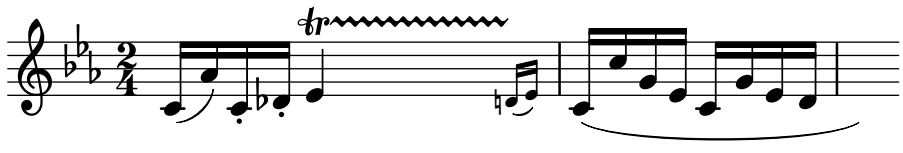


Allongement d'une marque de trille (`TrillSpanner`)

La propriété `minimum-length` d'une extension de trille – objet `TrillSpanner` – n'est effective qu'après un appel explicite à la procédure `set-spacing-rods`.

Pour ce faire, la propriété `set-spacing-rods` doit être définie à `ly:spanner::set-spacing-rods`.

```
\relative c' {
\key c\minor
  \time 2/4
  c16( as') c,-. des-.
  \once\override TrillSpanner.minimum-length = #15
  \once\override TrillSpanner.springs-and-rods = #ly:spanner::set-spacing-rods
  \afterGrace es4
  \startTrillSpan { d16[( \stopTrillSpan es)] }
  c( c' g es c g' es d
  \hideNotes
  c8)
}
```



Rappel du glissando à l'occasion d'une alternative

Un glissando qui se prolonge sur plusieurs sections `\alternative` peut se rappeler à l'aide d'une note d'ornement supplémentaire et masquée, à laquelle sera attaché le départ du glissando, ce dans chaque bloc `\alternative`. Cette note d'ornement devrait avoir la même hauteur que la note où commençait le glissando originel. Ceci est géré par une fonction musicale qui prendra en argument la hauteur de la note d'ornement.

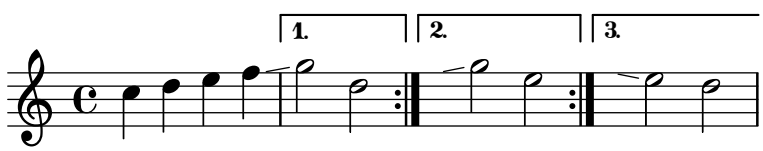
Dans le cadre d'un musique polyphonique, il ne faudra pas oublier d'ajouter une note d'ornement dans toutes les autres voix afin de préserver la synchronisation.

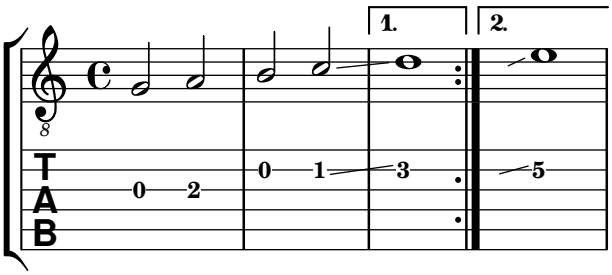
```
repeatGliss = #(define-music-function (grace)
  (ly:pitch?)
  #{
    % the next two lines ensure the glissando is long enough
    % to be visible
    \once \override Glissando.springs-and-rods
      = #ly:spanner::set-spacing-rods
    \once \override Glissando.minimum-length = #3.5
    \once \hideNotes
    \grace $grace \glissando
  #})

\score {
  \relative c'' {
    \repeat volta 3 { c4 d e f\glissando }
    \alternative {
      { g2 d }
      { \repeatGliss f g2 e }
      { \repeatGliss f e2 d }
    }
  }
}

music =  \relative c' {
  \voiceOne
  \repeat volta 2 {
    g a b c\glissando
  }
  \alternative {
    { d1 }
    { \repeatGliss c \once \omit StringNumber e1\2 }
  }
}

\score {
  \new StaffGroup <<
    \new Staff <<
      \new Voice { \clef "G_8" \music }
    >>
    \new TabStaff  <<
      \new TabVoice { \clef "moderntab" \music }
    >>
  >>
}
```





Liaison de prolongation aplatie

La fonction ici présentée prend en argument le `Tie.stencil` par défaut et en recalcule le résultat d'après les étendues de cette valeur par défaut.

Des ajustements plus poussés sont possibles, en apportant des dérogations à `Tie.details.height-limit` ou à l'aide de la fonction `\shape`. La définition personnalisée peut aussi se modifier à la volée.

```
%% http://lsr.di.unimi.it/LSR/Item?id=1031

#(define ((flared-tie coords) grob)

  (define (pair-to-list pair)
     (list (car pair) (cdr pair)))

  (define (normalize-coords goods x y dir)
    (map
      (lambda (coord)
        ;(coord-scale coord (cons x (* y dir)))
        (cons (* x (car coord)) (* y dir (cdr coord))))
      goods))

  (define (my-c-p-s points thick)
    (make-connected-path-stencil
      points
      thick
      1.0
      1.0
      #f
      #f))

  ;; outer let to trigger suicide
  (let ((sten (ly:tie::print grob)))
    (if (grob::is-live? grob)
        (let* ((layout (ly:grob-layout grob))
               (line-thickness (ly:output-def-lookup layout 'line-thickness))
               (thickness (ly:grob-property grob 'thickness 0.1))
               (used-thick (* line-thickness thickness))
               (dir (ly:grob-property grob 'direction))
               (xex (ly:stencil-extent sten X))
               (yex (ly:stencil-extent sten Y))
               (lenx (interval-length xex))
               (leny (interval-length yex))
               (xtrans (car xex))
               (ytrans (if (> dir 0)(car yex) (cdr yex)))
               (uplist
```

```
                (map pair-to-list
                     (normalize-coords coords lenx (* leny 2) dir))))

   (ly:stencil-translate
       (my-c-p-s uplist used-thick)
     (cons xtrans ytrans)))
   '()))

#(define flare-tie
  (flared-tie '((0 . 0)(0.1 . 0.2) (0.9 . 0.2) (1.0 . 0.0))))

\layout {
  \context {
    \Voice
    \override Tie.stencil = #flare-tie
  }
}

\paper { ragged-right = ##f }

\relative c' {
  a4~a
  \override Tie.height-limit = 4
  a'4~a
  a'4~a
  <a,, c e a c e a c e>~ q

  \break

  a'4~a
  \once \override Tie.details.height-limit = 14
  a4~a

  \break

  a4~a
  \once \override Tie.details.height-limit = 0.5
  a4~a
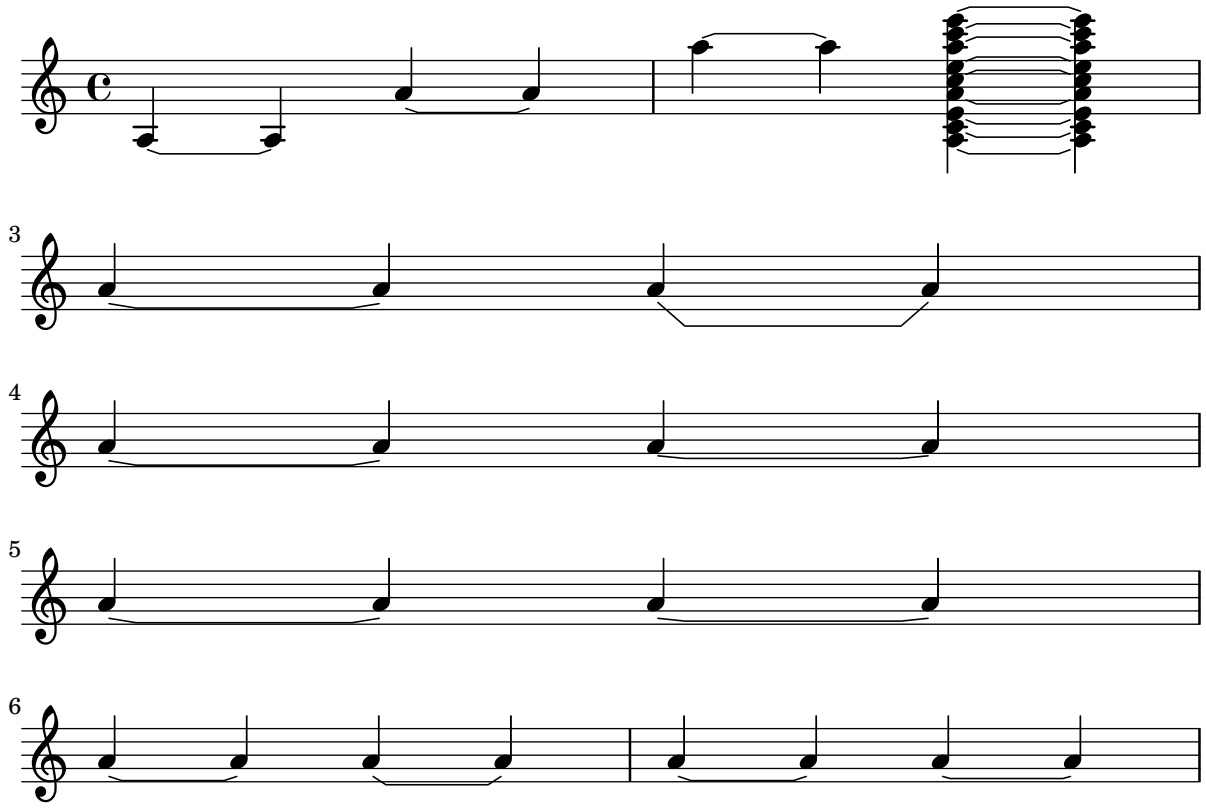
  \break

  a4~a
  \shape #'((0 . 0) (0 . 0.4) (0 . 0.4) (0 . 0)) Tie
  a4~a

  \break

  a4~a
  \once \override Tie.stencil =
    #(flared-tie '((0 . 0)(0.1 . 0.4) (0.9 . 0.4) (1.0 . 0.0)))
  a4~a
```

```
  a4~a
  \once \override Tie.stencil =
    #(flared-tie '((0 . 0)(0.06 . 0.1) (0.94 . 0.1) (1.0 . 0.0)))
  a4~a
}
```

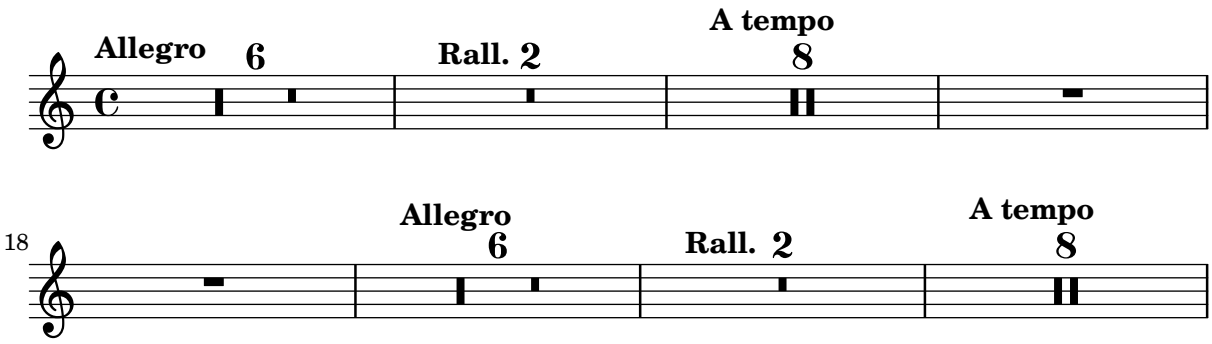


Adaptation de la largeur de mesure selon le `MetronomeMark`

Par défaut, les indications métronomiques n'influencent en rien l'espacement horizontal. Une simple dérogation, comme dans la deuxième partie de cet exemple, fournit la solution.

```
example = {
  \tempo "Allegro"
  R1*6
  \tempo "Rall."
  R1*2
  \tempo "A tempo"
  R1*8
}
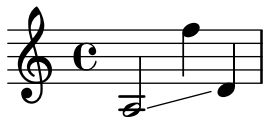
{
  \compressMMRests {
    \example
    R1
    R1
    \override Score.MetronomeMark.extra-spacing-width = #'(-3 . 0)
    \example
  }
}
```



Glissando par dessus un objet graphique

Un glissando peut sauter un objet `NoteColumn`.

```
\relative c' {
  a2 \glissando
  \once \override NoteColumn.glissando-skip = ##t
  f''4 d,
}
```



Impression d'un double repère sur une même barre (méthode 1)

Cette méthode permet d'imprimer deux marques-repères l'une au-dessus de l'autre. Le repère inférieur est décalé sous la portée, puis s'insère au-dessus de lui un espace pour positionner le repère supérieur en surplomb de la portée.

L'ajustement des valeurs de `extra-offset` et `baseline-skip` permet d'accroître ou réduire globalement l'espace entre le repère et la portée.

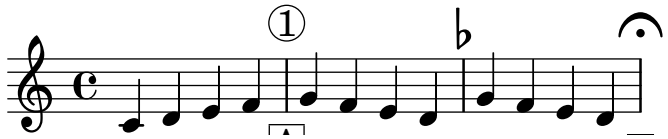
La capacité de pratiquement tout type de glyphe ou chaîne de caractères à faire l'objet d'un repère les rend capables de venir se centrer au-dessus ou en dessous d'une barre de mesure.

L'adjonction de la propriété `break-visibility`, comme dans cet autre cas (`\%22http://lsr.di.unimi.it/LSR/Item?id=1\%22`), permet de placer un double repère en fin de ligne.

Cette première méthode, bien que moins compliquée que la seconde, ne permet pas d'ajuster le positionnement d'un repère indépendamment de l'autre. Elle peut aussi générer quelques problèmes quant à l'espacement vertical, puisque le recours à `extra-offset` ne modifie en rien la valeur originelle de la boîte englobant le repère.

```
\relative c'{
  c d e f |
  \once \override Score.RehearsalMark.extra-offset = #'(0 . -8.5)
  \once \override Score.RehearsalMark.baseline-skip = #9
  \mark \markup \center-column { \circle 1 \box A }
  g f e d |
  \once \override Score.RehearsalMark.extra-offset = #'(0 . -8.5)
  \once \override Score.RehearsalMark.baseline-skip = #9
  \mark \markup \center-column { \flat { \bold \small \italic Fine. } }
  g f e d |
  \once \override Score.RehearsalMark.extra-offset = #'(0 . -8.5)
  \once \override Score.RehearsalMark.baseline-skip = #9
  \override Score.RehearsalMark.break-visibility = #begin-of-line-invisible
  \mark \markup \center-column { \fermata \box z }
```

```
}
```

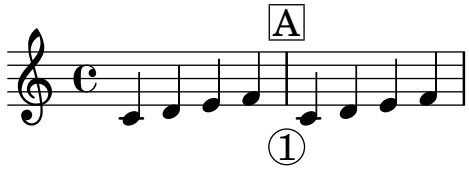


Impression d'un double repère sur une même barre (méthode 2)

Cette méthode permet d'imprimer deux marques-repères – l'une au-dessus de la portée et l'autre en dessous – à l'aide de deux voix séparées. Le graveur de repères est ensuite déplacé dans chacune des voix, sans quoi aucun repère ne sera imprimé. Enfin, chaque voix se voit attribuer un positionnement pour les repères.

Cette méthode, bien que plus complexe que la première, apporte plus de flexibilité dans l'ajustement individuel des deux composants du repère.

```
\score {
  \relative c'
  <<
    \new Staff {
      <<
        \new Voice \with {
          \consists Mark_engraver
          \consists "Staff_collecting_engraver"
        }
        { c4 d e f
          \mark \markup { \box A }
          c4 d e f
        }
        \new Voice \with {
          \consists Mark_engraver
          \consists "Staff_collecting_engraver"
          \override RehearsalMark.direction = #DOWN
        }
        { s4 s s s
          \mark \markup { \circle 1 }
          s4 s s s
        }
      >>
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Mark_engraver"
      \remove "Staff_collecting_engraver"
    }
  }
}
```



Incipit

Dans le cadre de la transcription de musique mensurale, un « incipit » en début de pièce permet d'indiquer tonalité et tempo originaux. Les musiciens sont maintenant habitués aux barres de mesure, ce qui n'était pas le cas à l'époque. Dans une forme de compromis, les barres de mesures sont souvent imprimées entre les portées, mise en forme que l'on appelle « lignes de mensuration ».

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% A short excerpt from the Jubilate Deo by Orlande de Lassus
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global = {
  \set Score.skipBars = ##t
  \key g \major
  \time 4/4

  % the actual music
  \skip 1*8

  % let finis bar go through all staves
  \override Staff.BarLine.transparent = ##f

  % finis bar
  \bar "|."
}

discantusIncipit = {
  \clef "neomensural-c1"
  \key f \major
  \time 2/2
  c''1.
}

discantusNotes = {
  \transpose c' c'' {
    \clef "treble"
    d'2. d'4 |
    b e' d'2 |
    c'4 e'4.( d'8 c' b |
    a4) b a2 |
    b4.( c'8 d'4) c'4 |
    \once \hide NoteHead
    c'1 |
    b\breve |
  }
}

discantusLyrics = \lyricmode {
  Ju -- bi -- la -- te De -- o,
```

```
  om -- nis ter -- ra, __ om-
  "..."
  -us.
}

altusIncipit = {
  \clef "neomensural-c3"
  \key f \major
  \time 2/2
  r1 f'1.
}

altusNotes = {
  \transpose c' c'' {
    \clef "treble"
    r2 g2. e4 fis g |
    a2 g4 e |
    fis g4.( fis16 e fis4) |
    g1 |
    \once \hide NoteHead
    g1 |
    g\breve |
  }
}

altusLyrics = \lyricmode {
  Ju -- bi -- la -- te
  De -- o, om -- nis ter -- ra,
  "..."
  -us.
}

tenorIncipit = {
  \clef "neomensural-c4"
  \key f \major
  \time 2/2
  r\longa
  r\breve
  r1 c'1.
}

tenorNotes = {
  \transpose c' c' {
    \clef "treble_8"
    R1 |
    R1 |
    R1 |
    % two measures
    r2 d'2. d'4 b e' |
    \once \hide NoteHead
    e'1 |
    d'\breve |
```

```
  }
}

tenorLyrics = \lyricmode {
  Ju -- bi -- la -- te
  "..."
  -us.
}
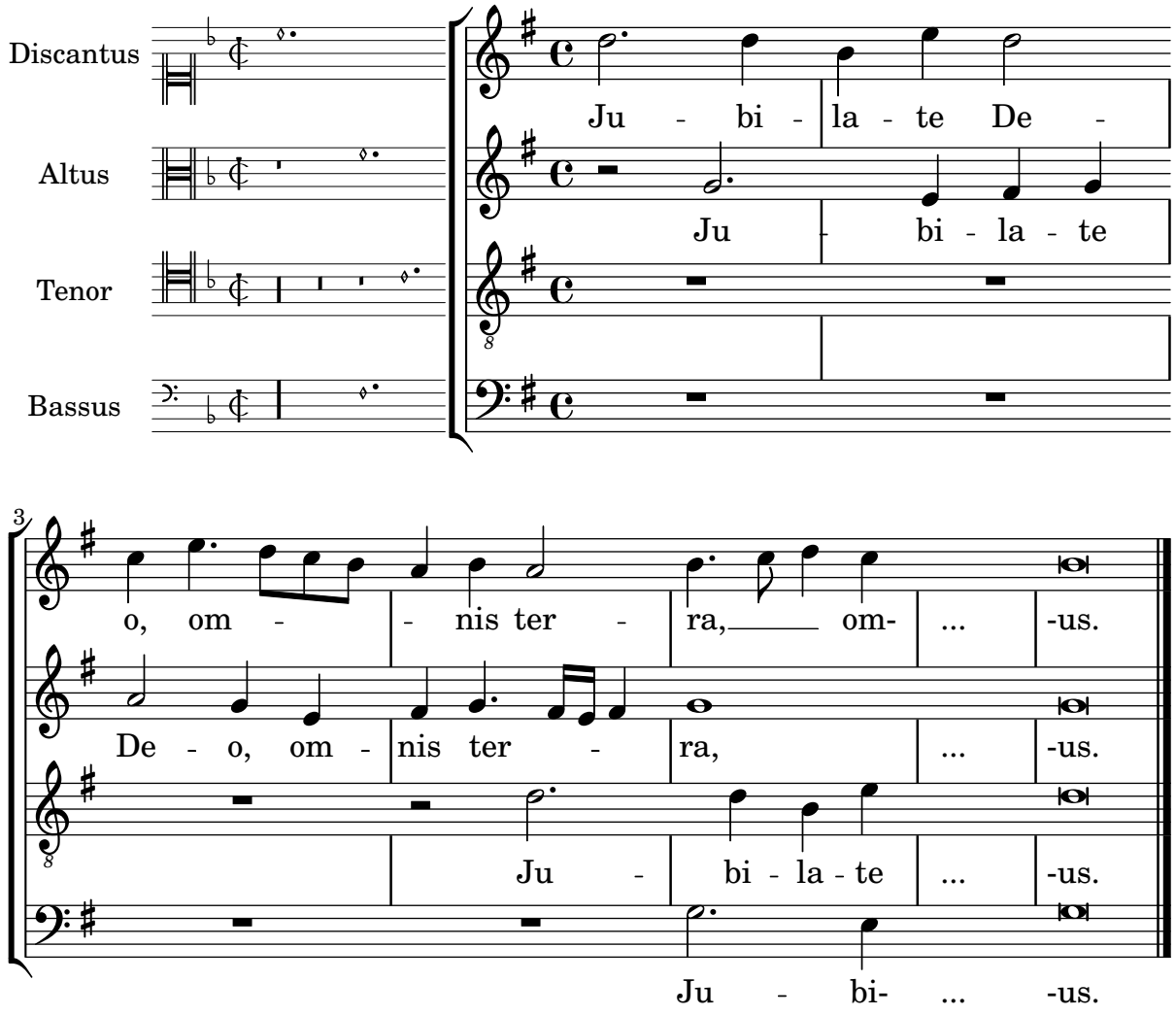
bassusIncipit = {
  \clef "mensural-f"
  \key f \major
  \time 2/2
  r\maxima
  f1.
}

bassusNotes = {
  \transpose c' c' {
    \clef "bass"
    R1 |
    R1 |
    R1 |
    R1 |
    g2. e4 |
    \once \hide NoteHead
    e1 |
    g\breve |
  }
}

bassusLyrics = \lyricmode {
  Ju -- bi-
  "..."
  -us.
}

\score {
  <<
    \new StaffGroup = choirStaff <<
      \new Voice = "discantusNotes" <<
        \set Staff.instrumentName = "Discantus"
        \incipit \discantusIncipit
        \global
        \discantusNotes
      >>
      \new Lyrics \lyricsto discantusNotes { \discantusLyrics }
      \new Voice = "altusNotes" <<
        \set Staff.instrumentName = "Altus"
        \global
        \incipit \altusIncipit
        \altusNotes
```

```
      >>
      \new Lyrics \lyricsto altusNotes { \altusLyrics }
      \new Voice = "tenorNotes" <<
        \set Staff.instrumentName = "Tenor"
        \global
        \incipit \tenorIncipit
        \tenorNotes
      >>
      \new Lyrics \lyricsto tenorNotes { \tenorLyrics }
      \new Voice = "bassusNotes" <<
        \set Staff.instrumentName = "Bassus"
        \global
        \incipit \bassusIncipit
        \bassusNotes
      >>
      \new Lyrics \lyricsto bassusNotes { \bassusLyrics }
    >>
  >>
  \layout {
    \context {
      \Score
      %% no bar lines in staves or lyrics
      \hide BarLine
    }
    %% the next two instructions keep the lyrics between the bar lines
    \context {
      \Lyrics
      \consists "Bar_engraver"
      \consists "Separating_line_group_engraver"
    }
    \context {
      \Voice
      %% no slurs
      \hide Slur
      %% Comment in the below "\remove" command to allow line
      %% breaking also at those bar lines where a note overlaps
      %% into the next measure.  The command is commented out in this
      %% short example score, but especially for large scores, you
      %% will typically yield better line breaking and thus improve
      %% overall spacing if you comment in the following command.
      %%\remove "Forbid_line_break_engraver"
    }
    indent = 6\cm
    incipit-width = 4\cm
  }
}
```



Insertion d'un fragment dans un *markup* au-dessus de la portée

La commande `\markup` est polyvalente. Dans cet exemple, elle contient un bloc `\score` plutôt que du texte ou une marque.

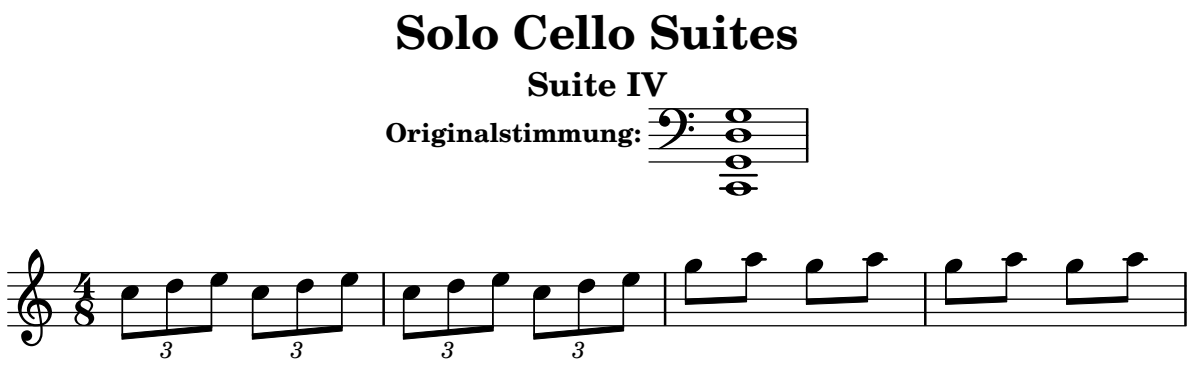
```
tuning = \markup {
  \score {
    \new Staff \with { \remove "Time_signature_engraver" }
    {
      \clef bass
      <c, g, d g>1
    }
    \layout { ragged-right = ##t  indent = 0\cm }
  }
}

\header {
  title = "Solo Cello Suites"
  subtitle = "Suite IV"
  subsubtitle = \markup { Originalstimmung: \raise #0.5 \tuning }
}

\layout { ragged-right = ##f }

\relative c'' {
```

```
  \time 4/8
  \tuplet 3/2 { c8 d e } \tuplet 3/2 { c d e }
  \tuplet 3/2 { c8 d e } \tuplet 3/2 { c d e }
  g8 a g a
  g8 a g a
}
```



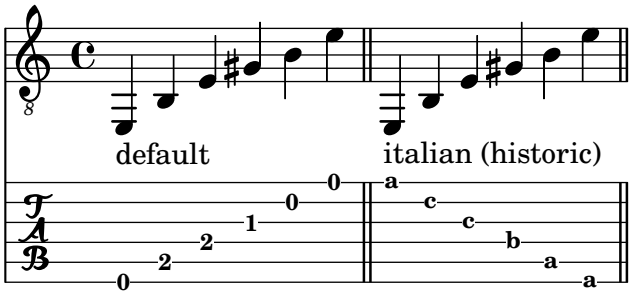
Impression de la corde aiguë d'un `TabStaff` en bas

Il est d'usage, en matière de tablature, d'imprimer la première corde en haut. Il est toutefois possible de la positionner en bas grâce à une modification de la propriété de contexte `stringOneTopmost`. Définir ce réglage au sein d'un bloc `\layout` l'appliquera à l'intégralité du contexte.

```
%
%\layout {
%  \context {
%    \Score
%    stringOneTopmost = ##f
%  }
%  \context {
%    \TabStaff
%    tablatureFormat = #fret-letter-tablature-format
%  }
%}

m = {
  \cadenzaOn
  e, b, e gis! b e'
  \bar "||"
}

<<
  \new Staff { \clef "G_8" <>_"default" \m <>_"italian (historic)"\m }
  \new TabStaff
  {
    \m
    \set Score.stringOneTopmost = ##f
    \set TabStaff.tablatureFormat = #fret-letter-tablature-format
    \m
  }
>>
```

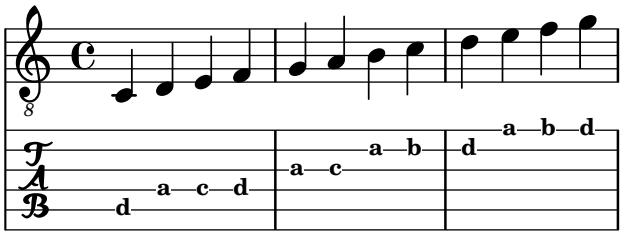


Tablature en lettres

Une tablature peut comporter des lettres en lieu et place des chiffres.

```
music = \relative c {
  c4 d e f
  g4 a b c
  d4 e f g
}

<<
  \new Staff {
    \clef "G_8"
    \music
  }
  \new TabStaff \with {
    tablatureFormat = #fret-letter-tablature-format
  }
  {
    \music
  }
>>
```



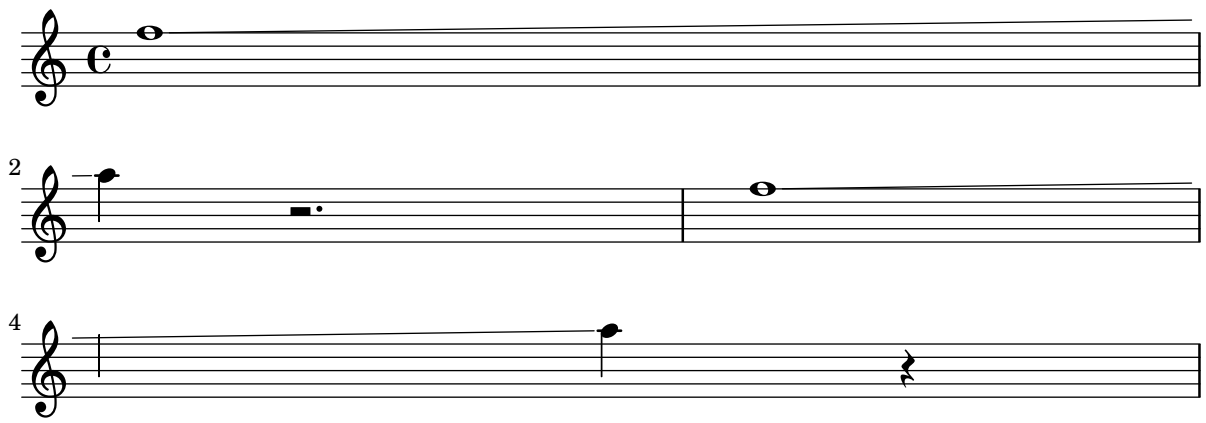
Saut de ligne et glissando

L'affectation de la valeur `#t` à la propriété `breakable`, combinée à `after-line-breaking`, permet la rupture d'une indication de glissando lors d'un saut de ligne.

```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

\relative c'' {
  \override Glissando.breakable = ##t
  \override Glissando.after-line-breaking = ##t
  f1\glissando |
  \break
```
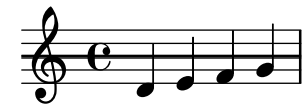
```
  a4 r2. |
  f1\glissando
  \once \glissandoSkipOn
  \break
  a2 a4 r4 |
}
```



Empâtement de certaines lignes d'une portée

Vous pourriez avoir envie, dans un but pédagogique, de rendre certaines lignes d'une portée plus épaisses que les autres, comme la ligne médiane, ou bien pour mettre en exergue la ligne portant la clef de sol. Il suffit pour cela d'ajouter une ligne qui sera accolée à celle qui doit être mise en évidence, grâce à la propriété `line-positions` de l'objet `StaffSymbol`.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```
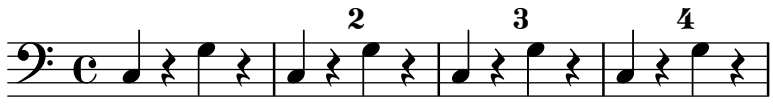


Compteur de mesures

Le code ci-dessous permet, de façon détournée, d'afficher un compteur de mesures à l'aide d'une répétition en pourcent rendue transparente.

```
<<
  \context Voice = "foo" {
    \clef bass
    c4 r g r
    c4 r g r
    c4 r g r
    c4 r g r
  }
  \context Voice = "foo" {
    \set countPercentRepeats = ##t
    \hide PercentRepeat
    \override PercentRepeatCounter.staff-padding = #1
```

```
    \repeat percent 4 { s1 }
  }
>>
```

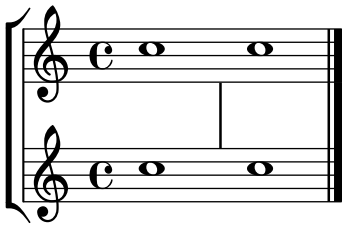


Présentation à l'ancienne (barres de mesure entre les portées)

En musique mensurale, les barres de mesure ne traversent pas les portées. Pour obtenir ce résultat avec un `StaffGroup` plutôt qu'en utilisant un `ChoirStaff`, il faudra masquer les portions de barre qui recouvrent les portées à l'aide d'un `\hide`.

```
global = {
  \hide Staff.BarLine
  s1 s
  % the final bar line is not interrupted
  \undo \hide Staff.BarLine
  \bar "|."
}

\new StaffGroup \relative c'' {
  <<
    \new Staff { << \global { c1 c } >> }
    \new Staff { << \global { c c } >> }
  >>
}
```

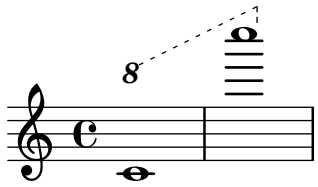


Modification de la pente de l'extension d'octaviation

Il est possible d'adapter la pente d'une indication d'octaviation.

```
\relative c'' {
  \override Staff.OttavaBracket.stencil = #ly:line-spanner::print
  \override Staff.OttavaBracket.bound-details =
    #`((left . ((Y . 0) ; Change the integer here
                (attach-dir . ,LEFT)
                (padding . 0)
                (stencil-align-dir-y . ,CENTER)))
       (right . ((Y . 5) ; Change the integer here
                 (padding . 0)
                 (attach-dir . ,RIGHT)
                 (text . ,(make-draw-dashed-line-markup
                           (cons 0 -1.2))))))
  \override Staff.OttavaBracket.left-bound-info =
     #ly:line-spanner::calc-left-bound-info-and-text
```

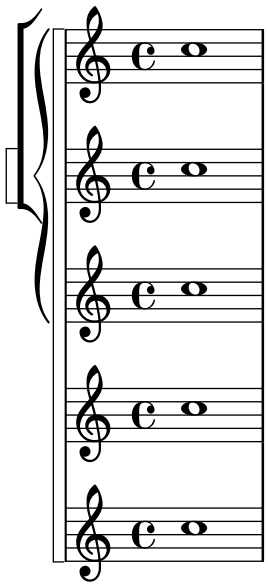
```
  \override Staff.OttavaBracket.right-bound-info =
     #ly:line-spanner::calc-right-bound-info
  \ottava #1
  c1
  c'''1
}
```



Imbrications de regroupements de portées

La propriété `systemStartDelimiterHierarchy` permet de créer des regroupements imbriqués complexes. La commande `\set StaffGroup.systemStartDelimiterHierarchy` prend en argument la liste alphabétique des sous-groupes à hiérarchiser. Chaque sous-groupe peut être affublé d'un délimiteur particulier. Chacun des regroupements intermédiaires doit être borné par des parenthèses. Bien que des éléments de la liste puissent être omis, le premier délimiteur embrassera toujours l'intégralité des portées. Vous disposez des quatre délimiteurs `SystemStartBar`, `SystemStartBracket`, `SystemStartBrace` et `SystemStartSquare`.

```
\new StaffGroup
\relative c'' <<
  \override StaffGroup.SystemStartSquare.collapse-height = #4
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare (SystemStartBrace (SystemStartBracket a
                              (SystemStartSquare b)  ) c ) d)
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
>>
```



Armures inhabituelles

La commande `\key` détermine la propriété `keyAlterations` d'un contexte `Staff`. Des armures inhabituelles peuvent être spécifiées en modifiant directement cette propriété.
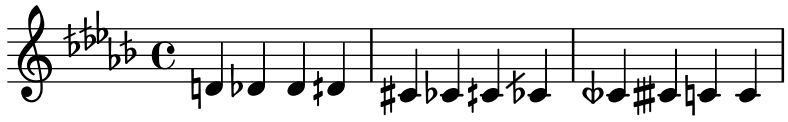
Il s'agit en l'occurrence de définir une liste :

```
\set Staff.keyAlterations =
  #`(((octave . pas) . altération) ((octave . pas) . altération)
  @dots{})
```

dans laquelle, et pour chaque élément, `octave` spécifie l'octave (0 pour celle allant du do médium au si supérieur), `pas` la note dans cette octave (0 pour do et 6 pour si), et `altération` sera `,SHARP` ou `,FLAT` ou `,DOUBLE-SHARP`, etc. (attention à la virgule en préfixe).

Une formulation abrégée – `(pas . altération)` – signifie que l'altération de l'élément en question sera valide quelle que soit l'octave. En ce qui concerne les gammes microtonales dans lesquelles un « dièse » n'est pas d'un centième, `altération` se réfère à un deux-centième de ton entier.

```
\include "arabic.ly"
\relative do' {
  \set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
                                 (1 . ,SEMI-FLAT)
                                 (2 . ,FLAT)
                                 (5 . ,FLAT)
                                 (6 . ,SEMI-FLAT))
%\set Staff.extraNatural = ##f
  re reb \dwn reb resd
  dod dob dosd \dwn dob |
  dobsb dodsd do do |
}
```



Numérotation de groupes de mesures

Cet extrait illustre l'utilisation du `Measure_counter_engraver` aux fins de numéroter un groupe de mesures successives. La section à numéroter peut éventuellement comporter des reprises.

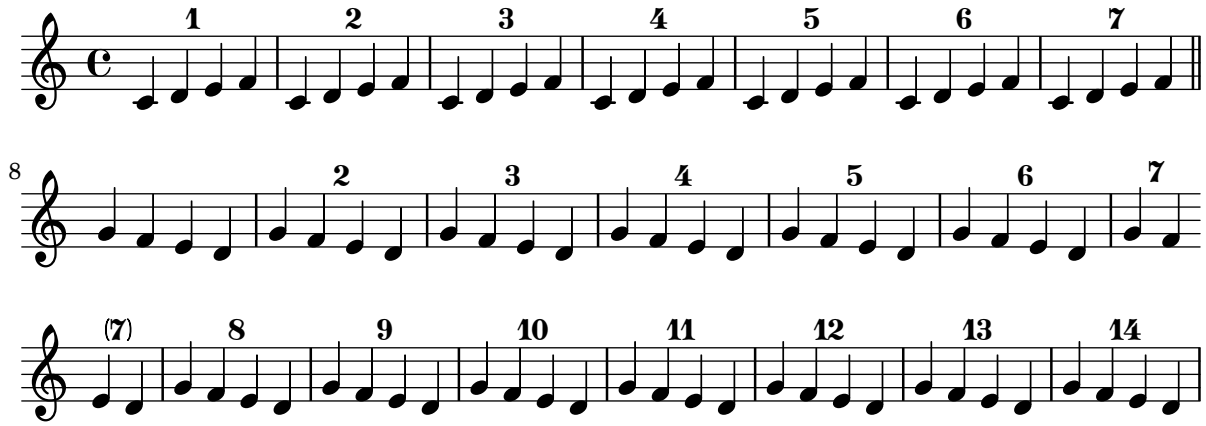
Le graveur doit être ajouté au contexte approprié. Il est ici attaché au contexte `Staff`, mais pourrait l'être à un contexte `Dynamics`.

Le compteur est lancé par `\startMeasureCount` et arrêté par `\stopMeasureCount`. La numérotation commence à `1` par défaut, comportement qui peut se modifier à l'aide de la propriété `count-from`.

Lorsqu'une mesure s'étend après un saut de ligne, le numéro y est alors répété, entre parenthèses.

```
\layout {
  \context {
    \Staff
    \consists #Measure_counter_engraver
  }
}
```

```
\new Staff {
  \startMeasureCount
  \repeat unfold 7 {
    c'4 d' e' f'
  }
  \stopMeasureCount
  \bar "||"
  g'4 f' e' d'
  \override Staff.MeasureCounter.count-from = #2
  \startMeasureCount
  \repeat unfold 5 {
    g'4 f' e' d'
  }
  g'4 f'
  \bar ""
  \break
  e'4 d'
  \repeat unfold 7 {
    g'4 f' e' d'
  }
  \stopMeasureCount
}
```



Orchestre, chœur et piano

Ce canevas illustre l'utilisation de contextes `StaffGroup` pour regrouper les instruments selon leur famille, imbriqués dans un `GrandStaff`, ainsi que le recours à la fonction `\transpose` pour les instruments transpositeurs. Dans tous les identificateurs, la musique est stockée en ut. Les notes peuvent tout aussi bien être saisies en ut ou dans la tonalité particulière de l'instrument avant d'être transposées puis affectées à une variable.

```
#(set-global-staff-size 17)
\paper {
  indent = 3.0\cm  % add space for instrumentName
  short-indent = 1.5\cm  % add less space for shortInstrumentName
}

fluteMusic = \relative c' { \key g \major g'1 b }

% Pitches as written on a manuscript for Clarinet in A
```

```
% are transposed to concert pitch.

clarinetMusic = \transpose c' a
  \relative c'' { \key bes \major bes1 d }

trumpetMusic = \relative c { \key g \major g''1 b }

% Key signature is often omitted for horns

hornMusic = \transpose c' f
  \relative c { d'1 fis }

percussionMusic = \relative c { \key g \major g1 b }

sopranoMusic = \relative c'' { \key g \major g'1 b }

sopranoLyrics = \lyricmode { Lyr -- ics }

altoIMusic = \relative c' { \key g \major g'1 b }

altoIIMusic = \relative c' { \key g \major g'1 b }

altoILyrics =  \sopranoLyrics

altoIILyrics = \lyricmode { Ah -- ah }

tenorMusic = \relative c' { \clef "treble_8" \key g \major g1 b }

tenorLyrics = \sopranoLyrics

pianoRHMusic = \relative c { \key g \major g''1 b }

pianoLHMusic = \relative c { \clef bass \key g \major g1 b }

violinIMusic = \relative c' { \key g \major g'1 b }

violinIIMusic = \relative c' { \key g \major g'1 b }

violaMusic = \relative c { \clef alto \key g \major g'1 b }

celloMusic = \relative c { \clef bass \key g \major g1 b }

bassMusic = \relative c { \clef "bass_8" \key g \major g,1 b }

\score {
  <<
    \new StaffGroup = "StaffGroup_woodwinds" <<
      \new Staff = "Staff_flute" \with { instrumentName = "Flute" }
      \fluteMusic

      \new Staff = "Staff_clarinet" \with {
        instrumentName = \markup { \concat { "Clarinet in B" \flat } }
```

```
  }

  % Declare that written Middle C in the music
  % to follow sounds a concert B flat, for
  % output using sounded pitches such as MIDI.
  %\transposition bes

  % Print music for a B-flat clarinet
  \transpose bes c' \clarinetMusic
>>

\new StaffGroup = "StaffGroup_brass" <<
  \new Staff = "Staff_hornI" \with { instrumentName = "Horn in F" }
   % \transposition f
    \transpose f c' \hornMusic

  \new Staff = "Staff_trumpet" \with { instrumentName = "Trumpet in  C" }
  \trumpetMusic

>>
\new RhythmicStaff = "RhythmicStaff_percussion"
\with { instrumentName = "Percussion" }
<<
  \percussionMusic
>>
\new PianoStaff \with { instrumentName = "Piano" }
<<
  \new Staff { \pianoRHMusic }
  \new Staff { \pianoLHMusic }
>>
\new ChoirStaff = "ChoirStaff_choir" <<
  \new Staff = "Staff_soprano" \with { instrumentName = "Soprano" }
  \new Voice = "soprano"
  \sopranoMusic

  \new Lyrics \lyricsto "soprano" { \sopranoLyrics }
  \new GrandStaff = "GrandStaff_altos"
  \with { \accepts Lyrics } <<
    \new Staff = "Staff_altoI"  \with { instrumentName = "Alto I" }
    \new Voice = "altoI"
    \altoIMusic

    \new Lyrics \lyricsto "altoI" { \altoILyrics }
    \new Staff = "Staff_altoII" \with { instrumentName = "Alto II" }
    \new Voice = "altoII"
    \altoIIMusic

    \new Lyrics \lyricsto "altoII" { \altoIILyrics }
  >>

  \new Staff = "Staff_tenor" \with { instrumentName = "Tenor" }
    \new Voice = "tenor"
```

```
        \tenorMusic

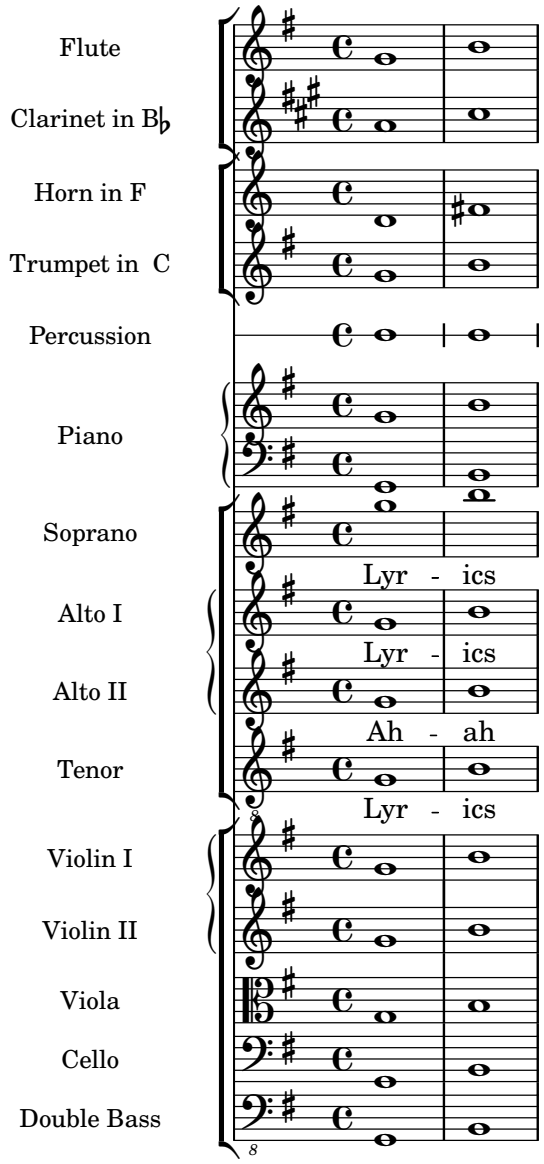
      \new Lyrics \lyricsto "tenor" { \tenorLyrics }
    >>
    \new StaffGroup = "StaffGroup_strings" <<
      \new GrandStaff = "GrandStaff_violins" <<
        \new Staff = "Staff_violinI" \with { instrumentName = "Violin I" }
        \violinIMusic

        \new Staff = "Staff_violinII" \with { instrumentName = "Violin II" }
        \violinIIMusic
      >>

      \new Staff = "Staff_viola" \with { instrumentName = "Viola" }
      \violaMusic

      \new Staff = "Staff_cello" \with { instrumentName = "Cello" }
      \celloMusic
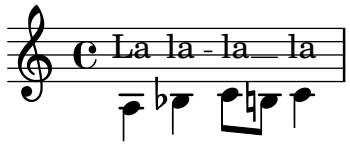
      \new Staff = "Staff_bass" \with { instrumentName = "Double Bass" }
      \bassMusic
    >>
  >>
  \layout { }
}
```



Impression des paroles dans la portée

Des paroles peuvent venir s'inscrire dans la portée même. Ces paroles sont décalées par la dérogation `\override LyricText.extra-offset = #'(0 . dimension)` ; des commandes similaires s'occuperont des extenseurs et des tirets. Le décalage optimal ne peut s'obtenir qu'en procédant à tâtons.

```
<<
  \new Staff <<
    \new Voice = "voc" \relative c' { \stemDown a bes c8 b c4 }
  >>
  \new Lyrics \with {
    \override LyricText.extra-offset = #'(0 . 8.6)
    \override LyricExtender.extra-offset = #'(0 . 8.6)
    \override LyricHyphen.extra-offset = #'(0 . 8.6)
  } \lyricsto "voc" { La la -- la __ _ la }
>>
```



Citation d'une autre voix et transposition
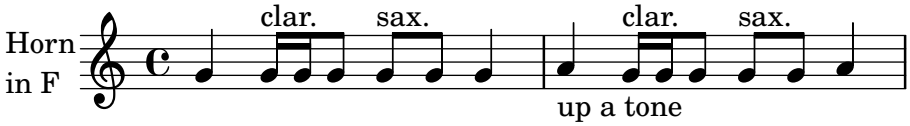
Les citations tiennent compte de la transposition, aussi bien celle de l'instrument d'origine que celle de la partie où elles interviennent. Dans l'exemple suivant, tous les instruments sont en tonalité de concert et seront repris par un instrument en fa. Le destinataire de la citation peut à son tour transposer à l'aide de la commande `\transpose`. En pareil cas, toutes les hauteurs seront transposées, y compris celles de la citation.

```
\addQuote clarinet {
  \transposition bes
  \repeat unfold 8 { d'16 d' d'8 }
}

\addQuote sax {
  \transposition es'
  \repeat unfold 16 { a8 }
}

quoteTest = {
  % french horn
  \transposition f
  g'4
  << \quoteDuring "clarinet" { \skip 4 } s4^"clar." >>
  << \quoteDuring "sax" { \skip 4 } s4^"sax." >>
  g'4
}

{
  \new Staff \with {
    instrumentName = \markup { \column { Horn "in F" } }
  }
  \quoteTest
  \transpose c' d' << \quoteTest s4_"up a tone" >>
}
```



Citation d'une autre voix

Les types d'événements pris en charge pour la citation sont déterminés par la propriété `quotedEventTypes`. Par défaut, sa valeur est fixée à `(note-event rest-event tie-event beam-event tuplet-span-event)`, ce qui signifie que seuls les notes, silences, liaisons, ligatures et n-olets seront mentionnés par `\quoteDuring`. Dans l'exemple suivant, le quart de soupir n'est pas reproduit puisqu'il n'est pas mentionné parmi les `quotedEventTypes`.

Pour connaître la liste des types d'événements, reportez-vous à la rubrique *Music classes* de la référence des propriétés internes.

```
quoteMe = \relative c' {
```
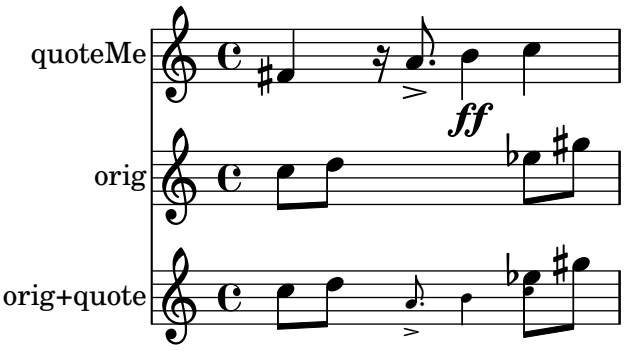
```
  fis4 r16 a8.-> b4\ff c
}
\addQuote quoteMe \quoteMe

original = \relative c'' {
  c8 d s2
  \once \override NoteColumn.ignore-collision = ##t
  es8 gis8
}

<<
  \new Staff \with { instrumentName = "quoteMe" }
  \quoteMe

  \new Staff \with { instrumentName = "orig" }
  \original

  \new Staff \with {
    instrumentName = "orig+quote"
    quotedEventTypes = #'(note-event articulation-event)
  }
  \relative c''
  <<
    \original
    \new Voice {
      s4
      \set fontSize = #-4
      \override Stem.length-fraction = #(magstep -4)
      \quoteDuring "quoteMe" { \skip 2. }
    }
  >>
>>
```



Retrait de l'accolade à la première ligne d'une pièce pour piano

Dans cet extrait est supprimée la première accolade d'un `PianoStaff` ou d'un `GrandStaff`.

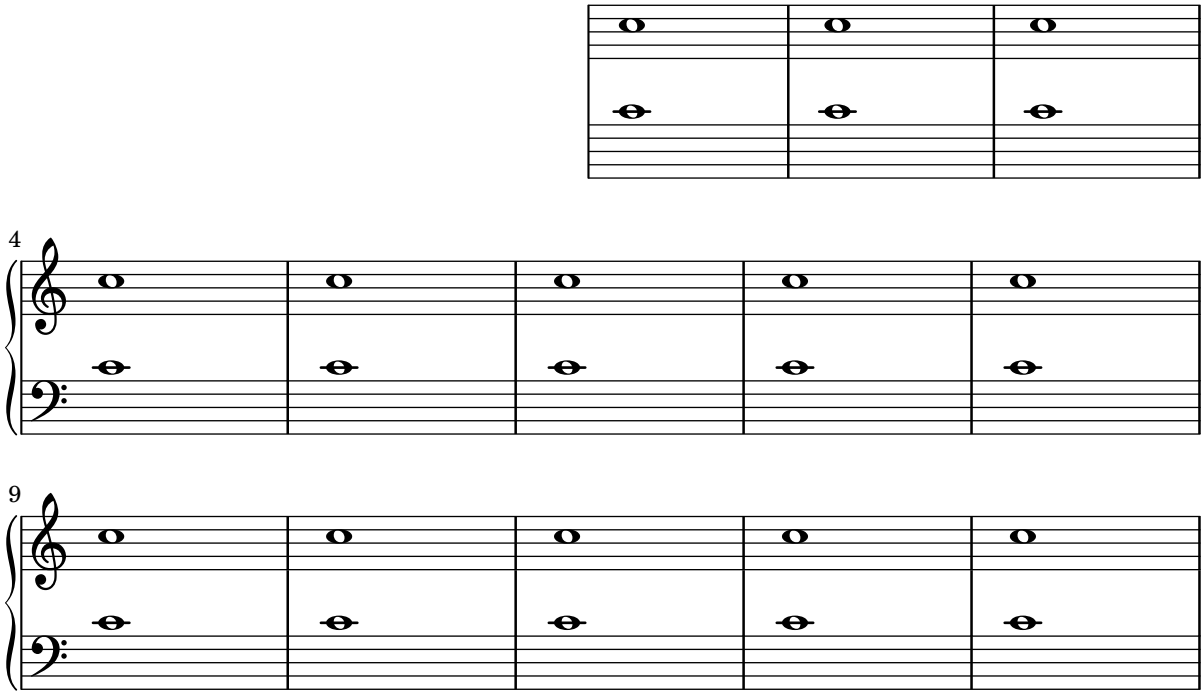
Ceci peut s'avérer utile pour couper et coller l'image générée dans de la musique préexistante.

Est utilisée la fonction `\alterBroken`.

```
someMusic =  {
  \once \override Staff.Clef.stencil = ##f
  \once \override Staff.TimeSignature.stencil = ##f
```

```
  \repeat unfold 3 c1 \break
  \repeat unfold 5 c1 \break
  \repeat unfold 5 c1
}

\score {
  \new PianoStaff
  <<
    \new Staff = "right" \relative c'' \someMusic
    \new Staff = "left" \relative c' { \clef F \someMusic }
  >>
  \layout {
    indent=75
    \context {
      \PianoStaff
      \alterBroken transparent #'(#t) SystemStartBrace
    }
  }
}
```



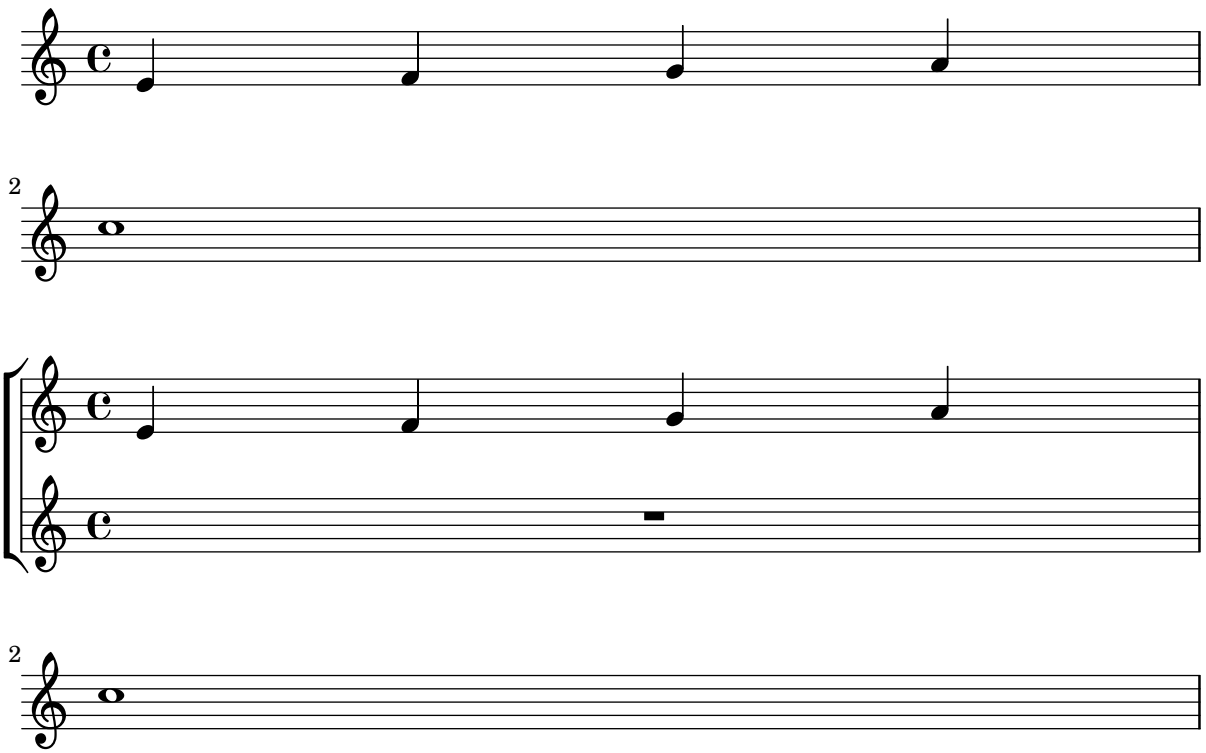
Masquage de la première ligne si elle est vide

Par défaut, le premier système comportera absolument toutes les portées. Si vous préférez masquer les portées vides y compris pour le premier système, vous devrez activer la propriété `remove-first` du `VerticalAxisGroup`. Mentionnée dans un bloc `\layout`, cette commande agira de manière globale. Pour qu'elle ne soit effective que pour une portée particulière, vous devrez également spécifier le contexte (`Staff` pour qu'il ne concerne que la portée en cours) en préfixe de la propriété.

La première ligne inférieure du deuxième `StaffGroup` est bien présente, pour la simple raison que le réglage en question ne s'applique qu'à la portée dans laquelle il a été inscrit.

```
\layout {
```

```
  \context {
    \Staff \RemoveEmptyStaves
    % To use the setting globally, uncomment the following line:
    % \override VerticalAxisGroup.remove-first = ##t
  }
}
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    % To use the setting globally, comment this line,
    % uncomment the line in the \layout block above
    \override Staff.VerticalAxisGroup.remove-first = ##t
    R1 \break
    R
  }
>>
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    R1 \break
    R
  }
>>
```



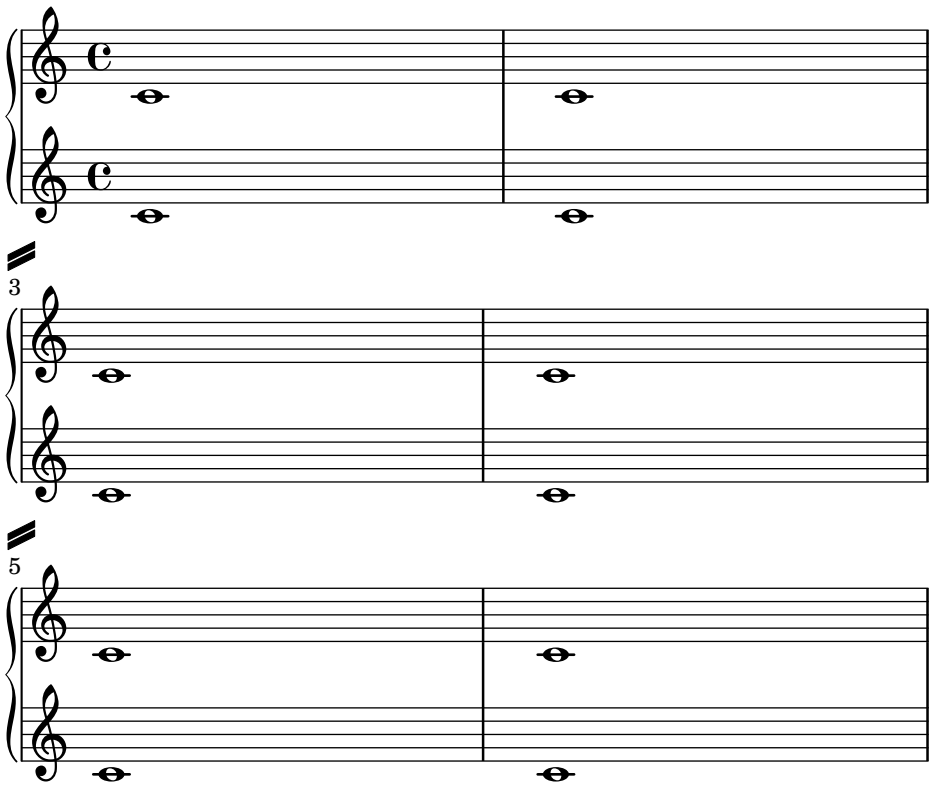
Séparation visuelle entre les systèmes

La séparation entre deux systèmes consécutifs peut être mise en évidence par n'importe quel *markup*. LilyPond dispose à cet effet d'une double oblique inversée : `\slashSeparator`.

```
\paper {
  system-separator-markup = \slashSeparator
  line-width = 120
}

notes = \relative c' {
  c1 | c \break
  c1 | c \break
  c1 | c
}
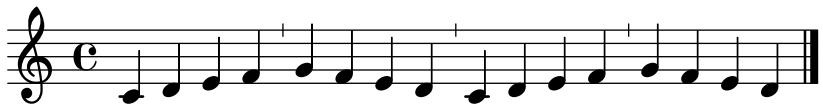
\book {
  \score {
    \new GrandStaff <<
      \new Staff \notes
      \new Staff \notes
    >>
  }
}
```


3
5

Barre de mesure en encoche

Lorsque les barres de mesure ne sont là que dans un but de coordination et non pour accentuer le rythme, il arrive souvent qu'elles se présentent sous la forme d'une simple encoche.
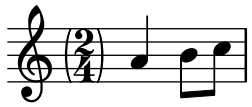
```
\relative c' {
  \set Score.defaultBarType = #"'"
  c4 d e f
  g4 f e d
  c4 d e f
  g4 f e d
  \bar "|."
}
```



Métrique entre parenthèses – méthode 3

Autre manière de mettre la métrique entre parenthèses.
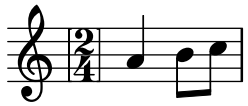
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (parenthesize-stencil (ly:time-signature::print grob) 0.1 0.4 0.4 0.1 ))
  \time 2/4
  a4 b8 c
}
```



Métrique entre parenthèses

Une métrique peut être mise entre parenthèses.

```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (bracketify-stencil (ly:time-signature::print grob) Y 0.1 0.2 0.1))
  \time 2/4
  a4 b8 c
}
```



Affinage des propriétés d'une clef

Modifier le glyphe, la position de la clef ou son octaviation ne changeront pas la position des notes ; il faut pour y parvenir modifier aussi la position du do médium. La redéfinition préalable de `middleCClefPosition` permet de placer l'armure sur les bonnes lignes. Le positionnement est relatif à la ligne médiane, un nombre positif faisant monter, un nombre négatif abaissant.

Par exemple, la commande `\clef "treble_8"` équivaut à définir `clefGlyph`, `clefPosition` – qui contrôle la position verticale de la clef – `middleCPosition` et `clefOctavation`. Une nouvelle

clef apparaîtra dès lors que l'une de ces propriétés, à l'exception de `middleCPosition`, aura été modifiée.

Les exemples qui suivent illustrent les différentes possibilités de définir ces propriétés manuellement. Sur la première ligne, la position relative des notes par rapport aux clefs est préservée, ce qui n'est pas le cas pour la deuxième ligne.

```
{
  % The default treble clef
  \key f \major
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  \set Staff.middleCPosition = #6
  \set Staff.middleCClefPosition = #6
  \key g \major
  c'1
  % The baritone clef
  \set Staff.clefGlyph = #"clefs.C"
  \set Staff.clefPosition = #4
  \set Staff.middleCPosition = #4
  \set Staff.middleCClefPosition = #4
  \key f \major
  c'1
  % The standard choral tenor clef
  \set Staff.clefGlyph = #"clefs.G"
  \set Staff.clefPosition = #-2
  \set Staff.clefTransposition = #-7
  \set Staff.middleCPosition = #1
  \set Staff.middleCClefPosition = #1
  \key f \major
  c'1
  % A non-standard clef
  \set Staff.clefPosition = #0
  \set Staff.clefTransposition = #0
  \set Staff.middleCPosition = #-4
  \set Staff.middleCClefPosition = #-4
  \key g \major
  c'1 \break

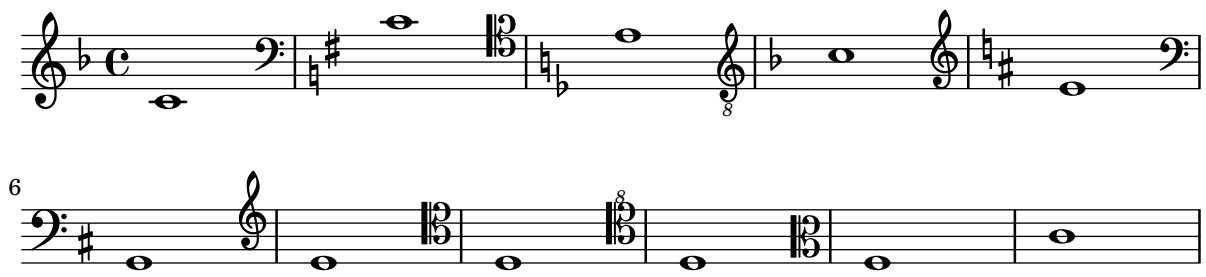
  % The following clef changes do not preserve
  % the normal relationship between notes, key signatures
  % and clefs:

  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  c'1
  \set Staff.clefGlyph = #"clefs.G"
  c'1
  \set Staff.clefGlyph = #"clefs.C"
  c'1
  \set Staff.clefTransposition = #7
```

```
  c'1
  \set Staff.clefTransposition = #0
  \set Staff.clefPosition = #0
  c'1

  % Return to the normal clef:

  \set Staff.middleCPosition = #0
  c'1
}
```



Deux \partCombine sur une même portée

La fonction `\partCombine` prend deux expressions musicales contenant chacune une partie et les distribue dans quatre contextes `Voice` nommés « two », « one », « solo » et « chords » selon le moment et la manière dont ces parties fusionnent en une voix commune. Les voix issues de `\partCombine` peuvent disposer de leurs propres propriétés de mise en forme définies de manière habituelle. Sont ici définies des extensions à `\partCombine` de sorte à faciliter le placement de quatre voix sur une seule portée.

```
soprano = { d'4 | cis'  b  e'  d'8 cis' | cis'2 b }
alto = { fis4 | e8 fis gis ais b4 b | b ais fis2 }
tenor = { a8 b | cis' dis' e'4 b8 cis' d'4 | gis cis' dis'2 }
bass = { fis8 gis | a4 gis g fis | eis fis b,2 }

\new Staff <<
  \key b\minor
  \clef alto
  \partial 4
  \transpose b b'
  \partCombineUp \soprano \alto
  \partCombineDown \tenor \bass
>>
\layout {
  \context {
    \Staff
    \accepts "VoiceBox"
  }
  \context {
    \name "VoiceBox"
    \type "Engraver_group"
    \defaultchild "Voice"
    \accepts "Voice"
    \accepts "NullVoice"
```

```
  }
}

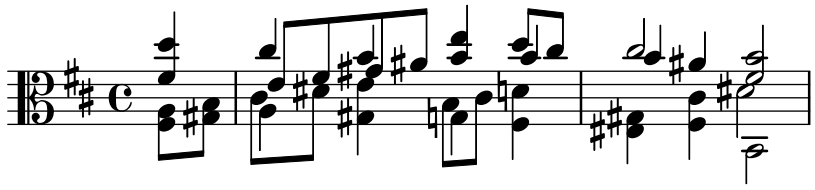
customPartCombineUp =
#(define-music-function (partOne partTwo)
  (ly:music? ly:music?)
"Take the music in @var{partOne} and @var{partTwo} and return
a @code{VoiceBox} named @q{Up} containing @code{Voice}s
that contain @var{partOne} and @var{partTwo} merged into one
voice where feasible.  This variant sets the default voicing
in the output to use upward stems."
#{
  \new VoiceBox = "Up" <<
    \context Voice = "one" { \voiceOne }
    \context Voice = "two" { \voiceThree }
    \context Voice = "shared" { \voiceOne }
    \context Voice = "solo" { \voiceOne }
    \context NullVoice = "null" {}
    \partCombine #partOne #partTwo
  >>
#})

customPartCombineDown = #
(define-music-function (partOne partTwo)
  (ly:music? ly:music?)
"Take the music in @var{partOne} and @var{partTwo} and return
a @code{VoiceBox} named @q{Down} containing @code{Voice}s
that contain @var{partOne} and @var{partTwo} merged into one
voice where feasible.  This variant sets the default voicing
in the output to use downward stems."
#{
  \new VoiceBox = "Down" <<
    \set VoiceBox.soloText = #"Solo III"
    \set VoiceBox.soloIIText = #"Solo IV"
    \context Voice ="one" { \voiceFour }
    \context Voice ="two" { \voiceTwo }
    \context Voice ="shared" { \voiceFour }
    \context Voice ="solo" { \voiceFour }
    \context NullVoice = "null" {}
    \partCombine #partOne #partTwo
  >>
#})

soprano = { d'4 | cis'  b  e'  d'8 cis' | cis'2 b }
alto = { fis4 | e8 fis gis ais b4 b | b ais fis2 }
tenor = { a8 b | cis' dis' e'4 b8 cis' d'4 | gis cis' dis'2 }
bass = { fis8 gis | a4 gis g fis | eis fis b,2 }

\new Staff <<
  \key b\minor
  \clef alto
  \partial 4
```

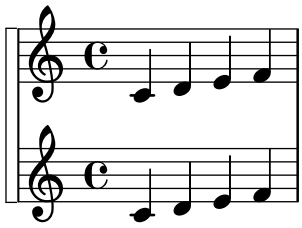
```
  \transpose b b'
  \customPartCombineUp \soprano \alto
  \customPartCombineDown \tenor \bass
>>
```



Indication de regroupement de portées par un rectangle

Un regroupement de portées sera indiqué par un simple rectangle – `SystemStartSquare` – en début de ligne dès lors que vous le mentionnerez explicitement au sein d'un contexte `StaffGroup` ou `ChoirStaff`.

```
\score {
  \new StaffGroup { <<
  \set StaffGroup.systemStartDelimiter = #'SystemStartSquare
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  >> }
}
```



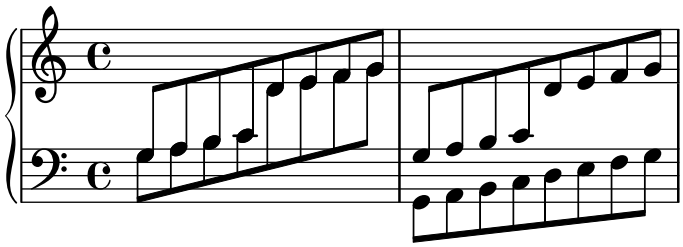
Utilisation d'autoChange avec plus d'une voix

L'instruction `\autoChange` est opérationnelle y compris en présence de voix multiples.

```
\score
{
  \new PianoStaff
  <<
    \new Staff = "up" {
      <<
        \set Timing.beamExceptions = #'()
        \set Timing.beatStructure = #'(4)
        \new Voice {
          \voiceOne
          \autoChange
          \relative c' {
            g8 a b c d e f g
            g,8 a b c d e f g
          }
        }
```

```
        \new Voice {
          \voiceTwo
          \autoChange
          \relative c' {
            g8 a b c d e f g
            g,,8 a b c d e f g
          }
        }
      >>
    }

    \new Staff = "down" {
      \clef bass
    }
  >>
}
```



Utilisation de lignes de repères dans une partition « à la française »

Le recours à des contextes `MarkLine` (tel qu'indiqué dans le LSR1010 (`http://lsr.di.unimi.it/LSR/Item?id=1010`)) dans une partition « à la française » peut s'avérer problématique lorsque toutes les portées entre deux contextes `Markline` sont absentes d'un système. La présence du `Keep_alive_together_engraver` au sein de chacun des `StaffGroup` permet de maintenir actif le `MarkLine` uniquement lorsque les autres portées du regroupement sont elles-mêmes actives.

```
bars = {
  \tempo "Allegro" 4=120
  s1*2
  \repeat unfold 5 { \mark \default s1*2 }
  \bar "||"
  \tempo "Adagio" 4=40
  s1*2
  \repeat unfold 8 { \mark \default s1*2 }
  \bar "|."
}
winds = \repeat unfold 120 { c''4 }
trumpet = { \repeat unfold 8 g'2 R1*16 \repeat unfold 4 g'2 R1*8 }
trombone = { \repeat unfold 4 c'1 R1*8 d'1 R1*17 }
strings = \repeat unfold 240 { c''8 }
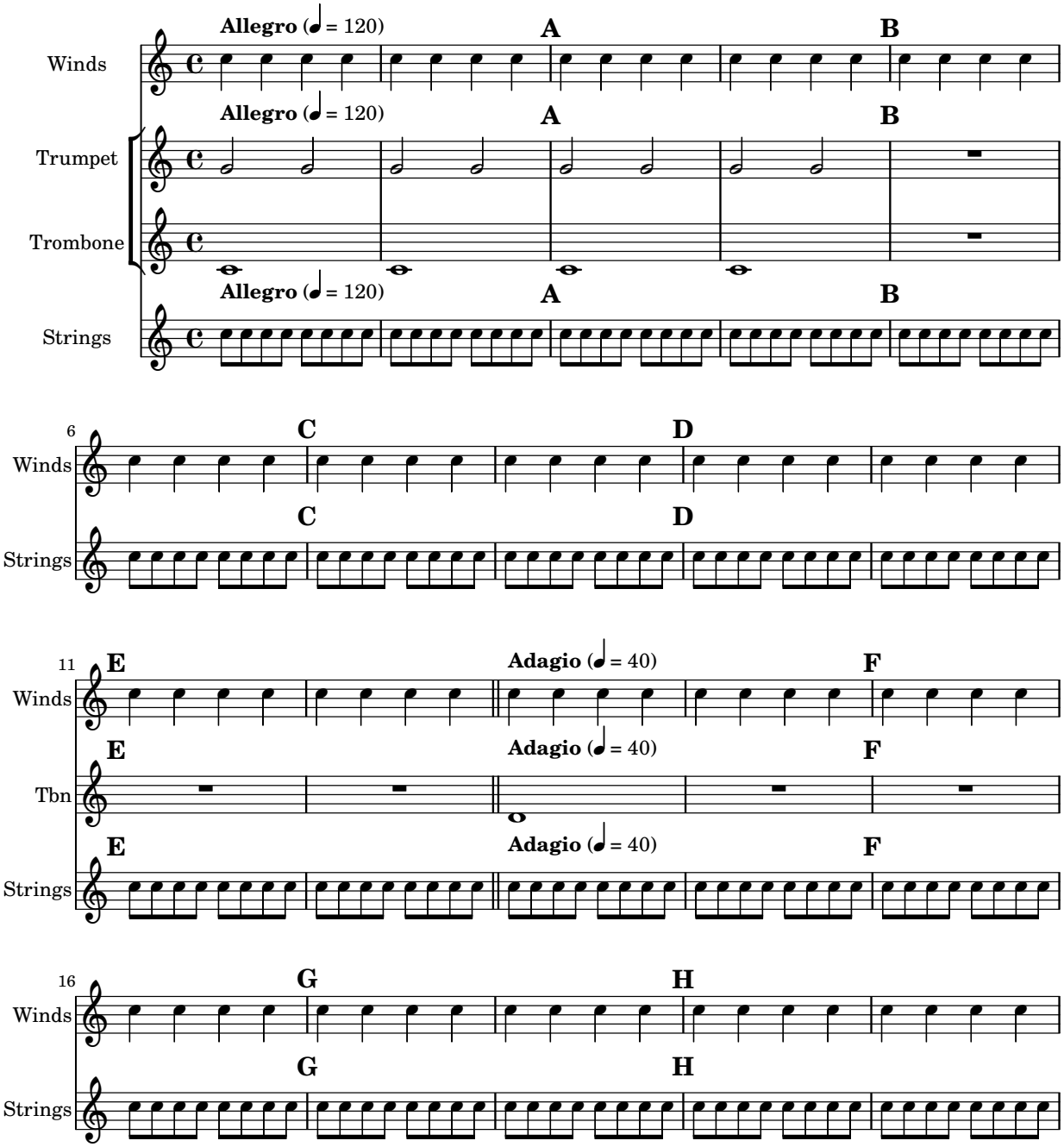
#(set-global-staff-size 16)
\paper {
  systems-per-page = 5
```

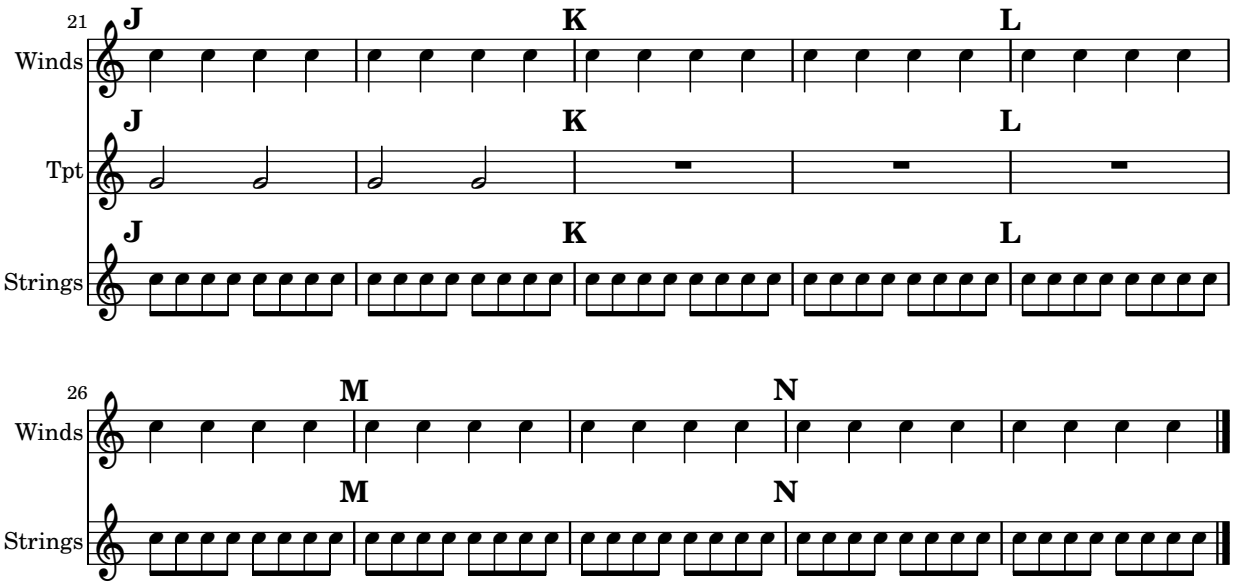
```
  ragged-last-bottom = ##f
}

\layout {
  indent = 15\mm
  short-indent = 5\mm
  \context {
    \name MarkLine
    \type Engraver_group
    \consists Output_property_engraver
    \consists Axis_group_engraver
    \consists Mark_engraver
    \consists Metronome_mark_engraver
    \override VerticalAxisGroup.remove-empty = ##t
    \override VerticalAxisGroup.remove-layer = #'any
    \override VerticalAxisGroup.staff-affinity = #DOWN
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing.basic-distance = 1
    keepAliveInterfaces = #'()
  }
  \context {
    \Staff
    \override VerticalAxisGroup.remove-empty = ##t
    \override VerticalAxisGroup.remove-layer = ##f
  }
  \context {
    \StaffGroup
    \accepts MarkLine
    \consists Keep_alive_together_engraver
  }
  \context {
    \Score
    \remove Mark_engraver
    \remove Metronome_mark_engraver
  }
}

\score {
  <<
    \new StaffGroup = "winds" \with {
      instrumentName = "Winds"
      shortInstrumentName = "Winds"
    } <<
      \new MarkLine \bars
      \new Staff \winds
    >>
    \new StaffGroup = "brass" <<
      \new MarkLine \bars
      \new Staff = "trumpet" \with {
        instrumentName = "Trumpet"
        shortInstrumentName = "Tpt"
      } \trumpet
      \new Staff = "trombone" \with {
```

```
        instrumentName = "Trombone"
        shortInstrumentName = "Tbn"
      } \trombone
    >>
    \new StaffGroup = "strings" \with {
      instrumentName = "Strings"
      shortInstrumentName = "Strings"
    } <<
      \new MarkLine \bars
      \new Staff = "strings" { \strings }
    >>
  >>
}
```





Alignement vertical de StaffGroups sans SystemStartBar

Cet extrait indique comment obtenir des regroupements `StaffGroup` alignés verticalement, disposant chacun de leur indication de groupe `SystemStartBar`, sans qu'il ne soient connectés entre eux.

```
% by Thomas Morley

#(set-global-staff-size 18)

\paper {
  indent = 0
  ragged-right = ##f
  print-all-headers = ##t
}

\layout {
  \context {
    \Staff
    \consists "Mark_engraver"
    \override RehearsalMark.self-alignment-X = #LEFT
  }
  \context {
    \StaffGroup
    systemStartDelimiterHierarchy =
      #'(SystemStartBrace (SystemStartBracket a b))
  }
  \context {
    \Score
    \override SystemStartBrace.style = #'bar-line
    \omit SystemStartBar
    \override SystemStartBrace.padding = #-0.1
    \override SystemStartBrace.thickness = #1.6
    \remove "Mark_engraver"
    \override StaffGrouper.staffgroup-staff-spacing.basic-distance = #15
  }
}
```

```
%%%% EXAMPLE

txt =
\lyricmode {
  Wer4 nur den lie -- ben Gott läßt wal2 -- ten4
  und4 hof -- fet auf ihn al -- le Zeit2.
}

% First StaffGroup "exercise"

eI =
\relative c' {
        \mark \markup {
                \bold Teacher:
                This is a simple setting of the choral. Please improve it.
                }
        \key a \minor
        \time 4/4
        \voiceOne

        \partial 4
        e4
        a b c b
        a b gis2
        e4\fermata g! g f
        e a a gis
        a2.\fermata
        \bar ":|."
}

eII =
\relative c' {
        \key a \minor
        \time 4/4
        \voiceTwo
        \partial 4
        c4
        e e e gis
        a f e2
        b4 b d d
        c c d d
        c2.
        \bar ":|."
}

eIII =
\relative c' {
        \key a \minor
        \time 4/4
        \clef bass
        \voiceOne
```

```
        \partial 4
        a4
        c b a b
        c d b2
        gis4 g g b
        c a f e
        e2.
}

eIV =
\relative c' {
        \key a \minor
        \time 4/4
        \clef bass
        \voiceTwo

        \partial 4
        a,4
        a' gis a e
        a, d e2
        e,4\fermata e' b g
        c f d e
        a,2.\fermata
        \bar ":|."
}

exercise =
\new StaffGroup = "exercise"
<<

  \new Staff
    <<
      \new Voice \eI
      \new Voice \eII
    >>

  \new Lyrics \txt

  \new Staff
    <<
      \new Voice \eIII
      \new Voice \eIV
    >>
>>

% Second StaffGRoup "simple Bach"

sbI =
\relative c' {
        \mark \markup { \bold" Pupil:" Here's my version! }
        \key a \minor
        \time 4/4
```

```
        \voiceOne

        \partial 4
        e4
        a b c b
        a b gis2
        e4\fermata g! g f
        e a a gis
        a2.\fermata
        \bar ":|."
}

sbII =
\relative c' {
        \key a \minor
        \time 4/4
        \voiceTwo
        \partial 4
        c8 d
        e4 e e8 f g4
        f f e2
        b4 b8 c d4 d
        e8 d c4 b8 c d4
        c2.
        \bar ":|."
}

sbIII =
\relative c' {
        \key a \minor
        \time 4/4
        \clef bass
        \voiceOne

        \partial 4
        a8 b
        c4 b a b8 c
        d4 d8 c b2
        gis4 g g8 a b4
        b a8 g f4 e
        e2.
}

sbIV =
\relative c' {
        \key a \minor
        \time 4/4
        \clef bass
        \voiceTwo

        \partial 4
        a,4
```

```
        a' gis a e
        f8 e d4 e2
        e,4\fermata e' b a8 g
        c4 f8 e d4 e
        a,2.\fermata
        \bar ":|."
}

simpleBach =
\new StaffGroup = "simple Bach"
<<

  \new Staff
    <<
      \new Voice \sbI
      \new Voice \sbII
    >>

  \new Lyrics \txt

  \new Staff
    <<
      \new Voice \sbIII
      \new Voice \sbIV
    >>
>>

% Third StaffGroup "chromatic Bach"

cbI =
\relative c' {
        \mark \markup {
          \bold "Teacher:"
          \column {
            "Well, you simply copied and transposed a version of J.S.Bach."
            "Do you know this one?"
          }
        }
        \key a \minor
        \time 4/4
        \voiceOne

        \partial 4
        e4
        a b c b
        a b gis4. fis8
        e4\fermata g! g f
        e a a8 b gis4
        a2.\fermata
        \bar ":|."
}
```

```
cbII =
\relative c' {
        \key a \minor
        \time 4/4
        \voiceTwo
        \partial 4
        c8 d
        e4 e e8 fis gis4
        a8 g! f!4 e2
        b4 e e d
        d8[ cis] d dis e fis e4
        e2.
        \bar ":|."
}

cbIII =
\relative c' {
        \key a \minor
        \time 4/4
        \clef bass
        \voiceOne

        \partial 4
        a8 b
        c[ b] a gis8 a4 d,
        e8[ e'] d c b4. a8
        gis4 b c d8 c
        b[ a] a b c b b c16 d
        c2.
}

cbIV =
\relative c' {
        \key a \minor
        \time 4/4
        \clef bass
        \voiceTwo

        \partial 4
        a4
        c, e a, b
        c d e2
        e4\fermata e a b8 c
        gis[ g] fis f e dis e4
        a,2.\fermata
        \bar ":|."
}

chromaticBach =
\new StaffGroup = "chromatic Bach"
<<
```

```
  \new Staff
    <<
      \new Voice \cbI
      \new Voice \cbII
    >>

  \new Lyrics \txt

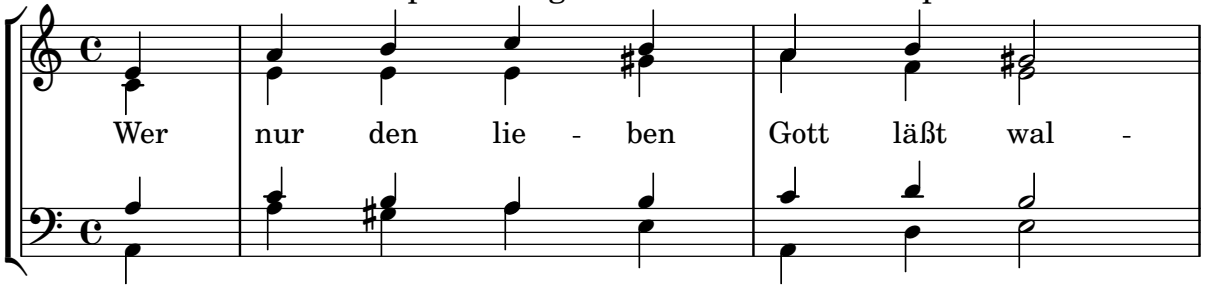
  \new Staff
    <<
      \new Voice \cbIII
      \new Voice \cbIV
    >>
>>


% Score

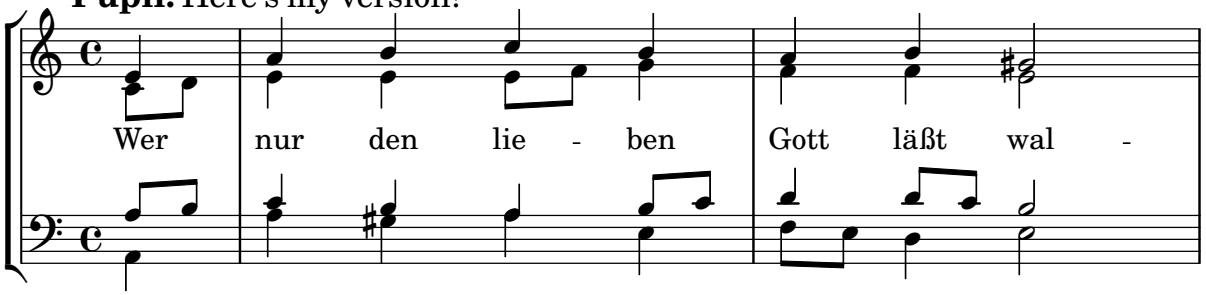
\score {
        <<
        \exercise
        \simpleBach
        \chromaticBach
        >>
        \header {
                title = \markup
                           \column {
                             \combine \null \vspace #1
                             "Exercise: Improve the given choral"
                             " "
                            }
        }
        \layout {
                \context {
                        \Lyrics
                        \override LyricText.X-offset = #-1
                }
        }
}
```

Exercise: Improve the given choral
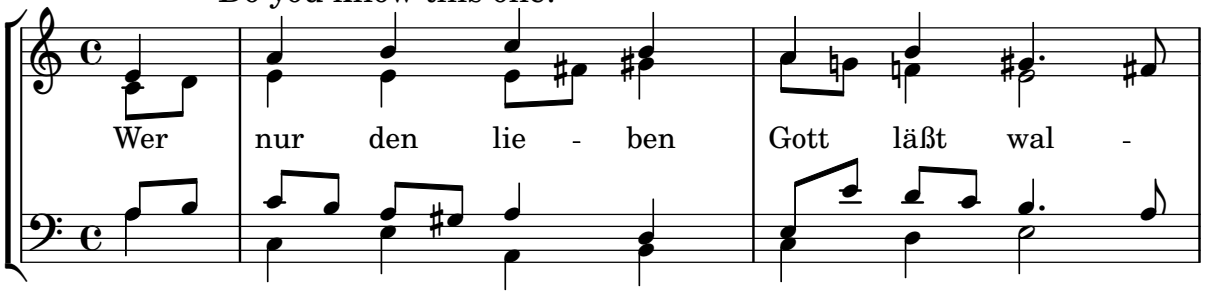
Teacher: This is a simple setting of the choral. Please improve it.
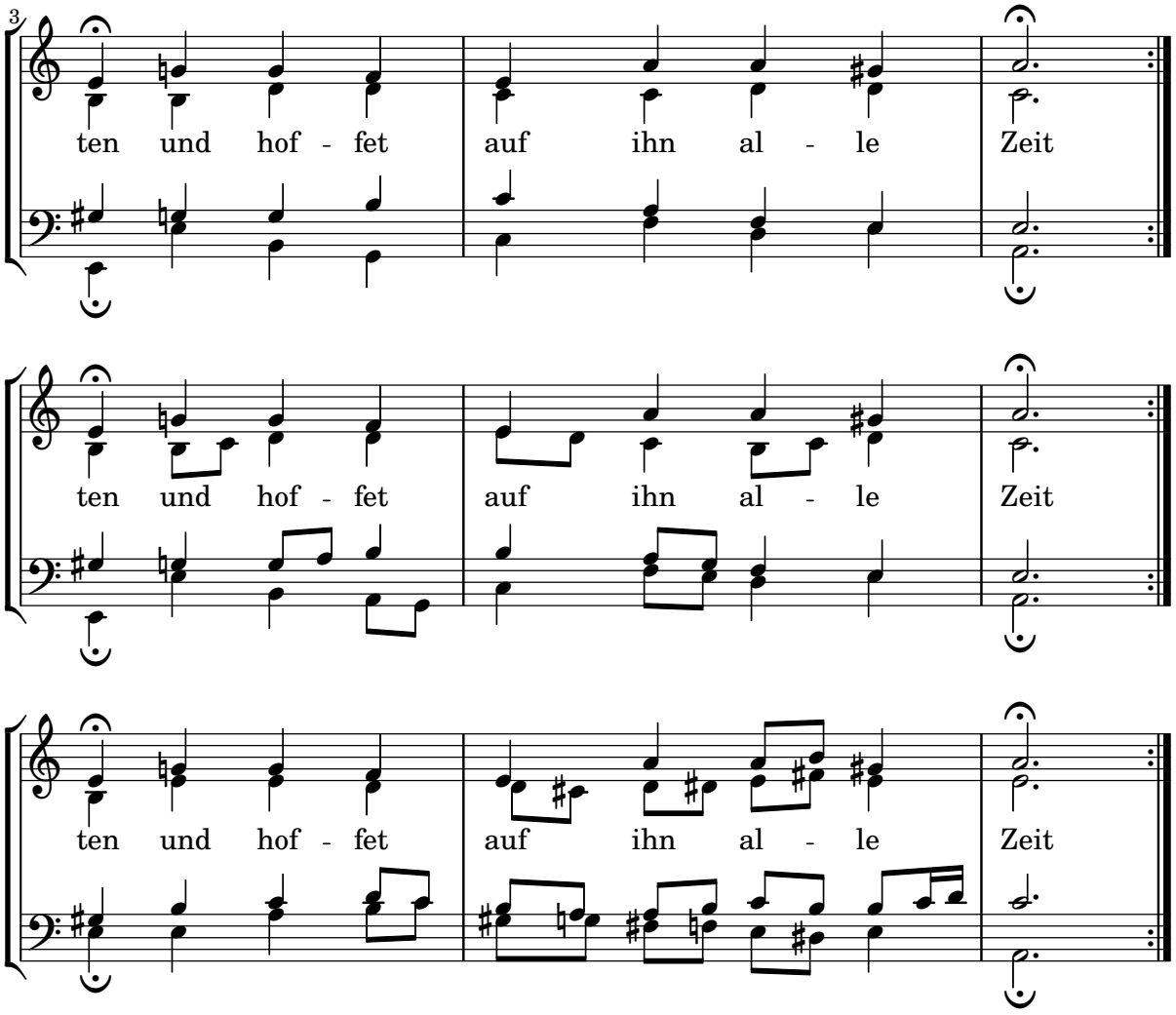


Pupil: Here's my version!



Teacher: Well, you simply copied and transposed a version of J.S.Bach. Do you know this one?



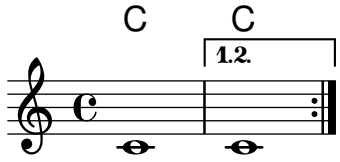


Crochet de reprise sous les chiffrages d'accord

L'ajout du `Volta_engraver` à la bonne portée permet d'imprimer les crochets de reprise entre les chiffrages et la portée.

```
\score {
  <<
    \chords {
      c1
      c1
    }
    \new Staff \with {
      \consists "Volta_engraver"
    }
    {
      \repeat volta 2 { c'1 }
      \alternative { c' }
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_engraver"
    }
  }
```

```
}
```

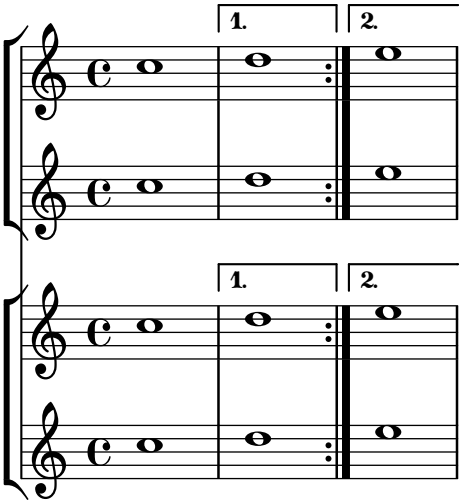


Volta multiportée

L'ajout du `Volta_engraver` à la portée appropriée permet de répéter les indications de reprise qui normalement n'apparaissent que sur la portée supérieure de la partition.

```
voltaMusic = \relative c'' {
  \repeat volta 2 {
    c1
  }
  \alternative {
    d1
    e1
  }
}

<<
  \new StaffGroup <<
    \new Staff \voltaMusic
    \new Staff \voltaMusic
  >>
  \new StaffGroup <<
    \new Staff \with { \consists "Volta_engraver" }
      \voltaMusic
    \new Staff \voltaMusic
  >>
>>
```
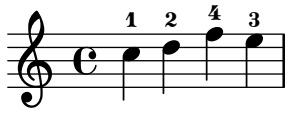


Editorial annotations

Section “Editorial annotations” dans *Manuel de notation*

Ajout de doigtés à une partition

Les instructions de doigtés se saisissent selon une syntaxe très simple.

```
\relative c'' {
  c4-1 d-2 f-4 e-3
}
```



Ajout de liens à des objets

La fonction `add-link`, telle que définie ci-dessous, permet d’ajouter un lien au stencil d’un objet graphique. Elle s’emploie au sein d’un `\override` ou d’un `\tweak`.

À noter que le fonctionnemment du `point-and-click` est perturbé sur les objets ainsi liés.

Cette fonction n’est opérationnelle que pour une sortie PDF.

La coloration des objets liés s’obtient par une commande séparée.

```
#(define (add-link url-strg)
  (lambda (grob)
    (let* ((stil (ly:grob-property grob 'stencil)))
      (if (ly:stencil? stil)
          (let* ((x-ext (ly:stencil-extent stil X))
                 (y-ext (ly:stencil-extent stil Y))
                 (url-expr `(url-link ,url-strg ,x-ext ,y-ext))
                 (new-stil
                   (ly:stencil-add
                     (ly:make-stencil url-expr x-ext y-ext)
                     stil)))
          (ly:grob-set-property! grob 'stencil new-stil))))))

%%%% test

%% For easier maintenance of this snippet the URL is formatted to use the
%% actually used LilyPond version.
%% Of course a literal URL would work as well.

#(define major.minor-version
  (string-join (take (string-split (lilypond-version) #\.) 2) "."))

urlI =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/writing-pitches"
  major.minor-version)

urlII =
#(format #f
```

```
  "http://lilypond.org/doc/v~a/Documentation/notation/rhythms"
  major.minor-version)

urlIII =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/note-heads"
  major.minor-version)

urlIV =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/beams"
  major.minor-version)

urlV =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/note-head-styles"
  major.minor-version)

urlVI =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/writing-pitches"
  major.minor-version)

\relative c' {
   \key cis \minor
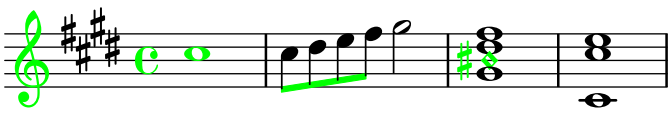
   \once \override Staff.Clef.color = #green
   \once \override Staff.Clef.after-line-breaking =
     #(add-link urlI)

   \once \override Staff.TimeSignature.color = #green
   \once \override Staff.TimeSignature.after-line-breaking =
     #(add-link urlII)

   \once \override NoteHead.color = #green
   \once \override NoteHead.after-line-breaking =
     #(add-link urlIII)

   cis'1
   \once \override Beam.color = #green
   \once \override Beam.after-line-breaking =
     #(add-link urlIV)
   cis8 dis e fis gis2
   <gis,
    \tweak Accidental.color #green
    \tweak Accidental.after-line-breaking #(add-link urlVI)
    \tweak color #green
    \tweak after-line-breaking #(add-link urlV)
    \tweak style #'harmonic
    bis
    dis
    fis
```

```
   >1
   <cis, cis' e>
}
```



Ajout de *markups* à une tablature

Par défaut, les *markups* n'apparaissent pas dans une tablature.

Il suffit, pour les voir apparaître, d'une simple commande `\revert TabStaff.TextScript.stencil`.

```
%% http://lsr.di.unimi.it/LSR/Item?id=919
% by P.P.Schneider on June 2014

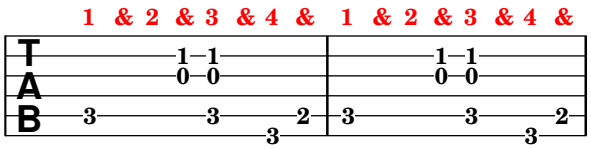
high  = { r4 r8 <g c'> q r8 r4 }

low = { c4 r4 c8 r8 g,8 b, }

pulse = { s8^"1" s^"&" s^"2" s^"&" s^"3" s^"&" s^"4" s^"&" }

\score {
 \new TabStaff {
   \repeat unfold 2 << \high \\ \low \\ \pulse >>
  }
  \layout {
    \context {
      \TabStaff
      \clef moderntab
      \revert TextScript.stencil
      \override TextScript.font-series = #'bold
      \override TextScript.font-size = #-2
      \override TextScript.color = #red

    }
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1/8)
    }
  }
}
```
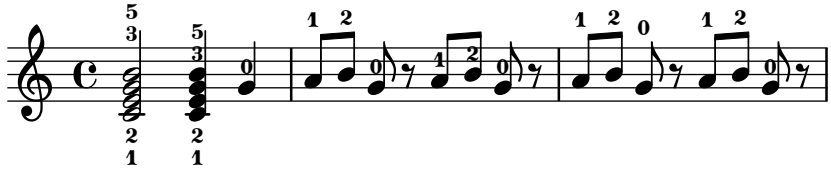


Impression des doigtés à l'intérieur de la portée

L'empilement des indications de doigté se fait par défaut à l'extérieur de la portée. Il est néanmoins possible d'annuler ce comportement. Une attention particulière doit toutefois être

portée dans les cas où doigté et hampe vont dans la même direction : les indications de doigté n'évitent les hampe qu'en présence de ligature. Ce réglage peut s'adapter pour éviter toutes les hampes ou aucune d'elles. L'exemple suivant illustre ces deux options, ainsi que la manière de revenir au comportement par défaut.
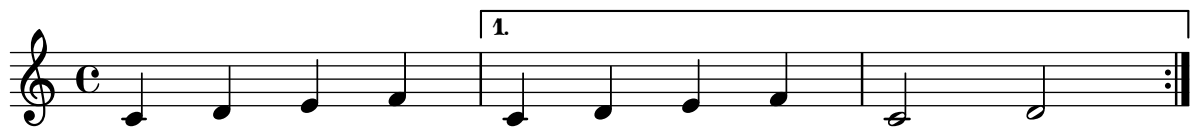
```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 g'-0
  a8[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##f
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##t
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = #only-if-beamed
  a[-1 b]-2 g-0 r
}
```
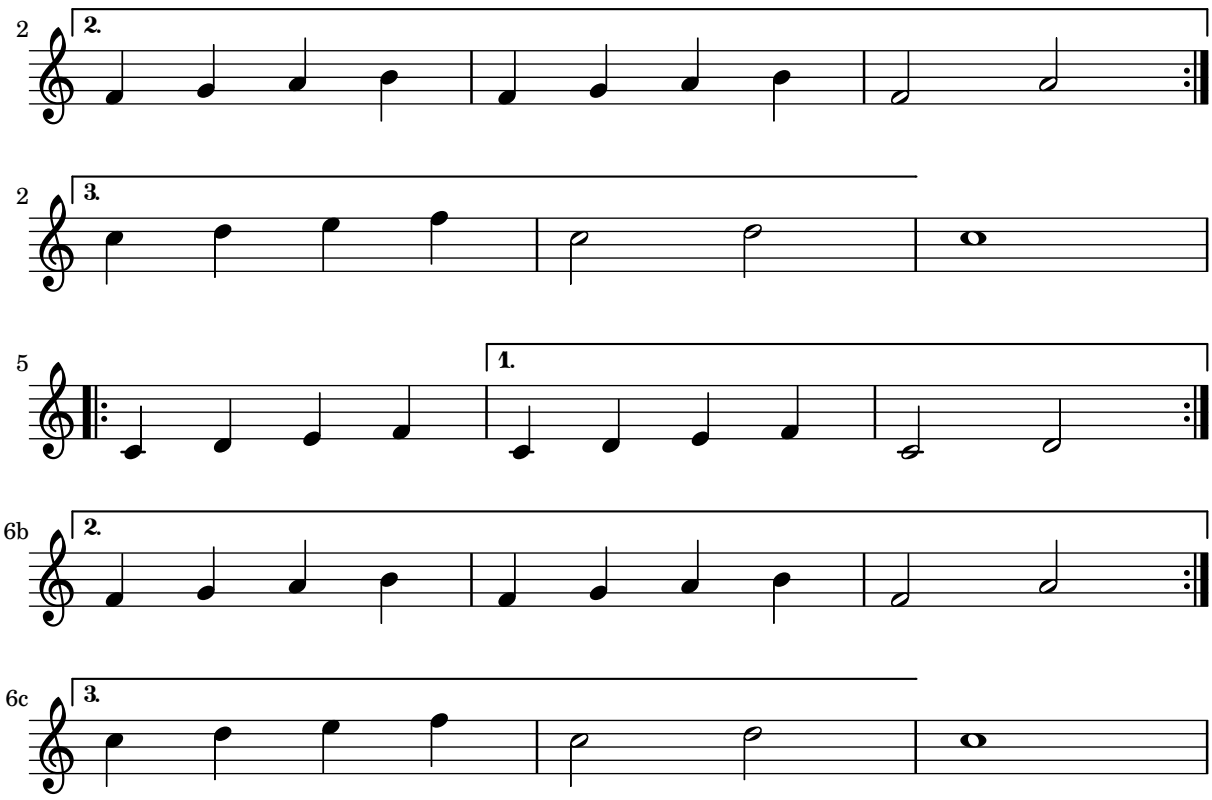


Numérotation des mesures et alternatives

Deux méthodes alternatives vous permettent de gérer la numérotation des mesures en cas de reprises.

```
\relative c'{
  \set Score.alternativeNumberingStyle = #'numbers
  \repeat volta 3 { c4 d e f | }
    \alternative {
      { c4 d e f | c2 d \break }
      { f4 g a b | f4 g a b | f2 a | \break }
      { c4 d e f | c2 d }
    }
  c1 \break
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \repeat volta 3 { c,4 d e f | }
    \alternative {
      { c4 d e f | c2 d \break }
      { f4 g a b | f4 g a b | f2 a | \break }
      { c4 d e f | c2 d }
    }
  c1
}
```
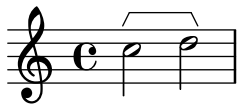




Crochets d'analyse au-dessus de la portée

Les crochets d'analyse viennent par défaut se positionner au-dessous de la portée. L'exemple suivant vous indique comment les faire apparaître en surplomb de la portée.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}

\relative c'' {
  \once \override HorizontalBracket.direction = #UP
  c2\startGroup
  d2\stopGroup
}
```



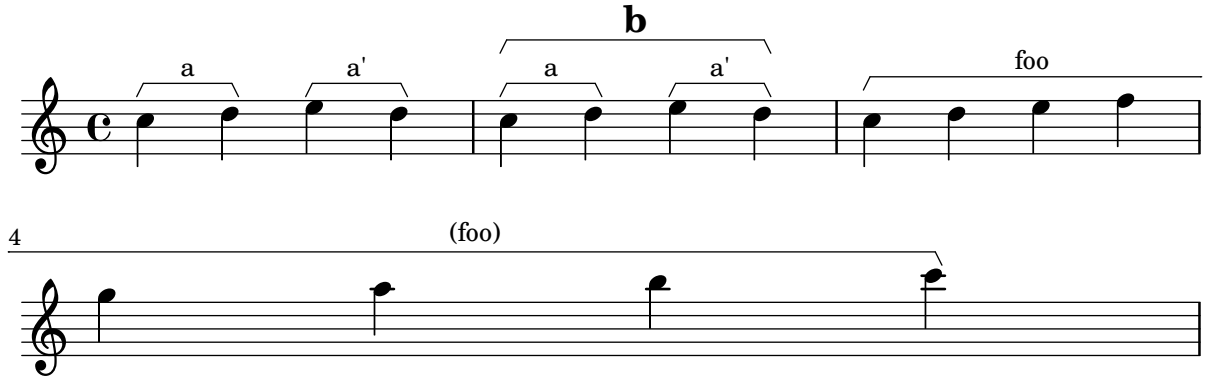
Crochet d'analyse avec texte

Du texte peut venir s'ajouter aux crochets d'analyse grâce à la propriété `texte` de l'objet graphique `HorizontalBracketText`. Plusieurs crochets présents en un même moment requièrent d'utiliser la commande `\tweak`. Le texte ajouté sera répété, entre parenthèse, après un saut de ligne.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
```

```
    \override HorizontalBracket.direction = #UP
  }
}

{
  \once\override HorizontalBracketText.text = "a"
    c''\startGroup d''\stopGroup
    \once\override HorizontalBracketText.text = "a'"
    e''\startGroup d''\stopGroup |
  c''-\tweak HorizontalBracketText.text
        \markup \bold \huge "b" \startGroup
     -\tweak HorizontalBracketText.text "a" \startGroup
    d''\stopGroup
    e''-\tweak HorizontalBracketText.text "a'" \startGroup
    d''\stopGroup\stopGroup |
  c''-\tweak HorizontalBracketText.text foo \startGroup
    d'' e'' f'' | \break
  g'' a'' b'' c'''\stopGroup
}
```



Profilage des notes selon leur degré dans la gamme

La propriété `shapeNoteStyles` permet d'affecter un profil particulier à chaque degré de la gamme – à partir de l'armure ou de la propriété `tonic`. Ses valeurs sont constituées d'une liste de symboles, qu'il s'agisse de formes géométriques (`triangle`, `cross` ou `xcircle`) ou basés sur la tradition des graveurs américains (avec quelques noms de note latins).

LilyPond dispose de deux raccourcis, `\aikenHeads` et `\sacredHarpHeads`, permettant de reproduire d'anciens recueils de chansons américaines.

L'exemple suivant montre plusieurs manières de profiler les têtes de note, ainsi que la capacité de transposer tout en respectant la fonction harmonique de chaque note dans la gamme.

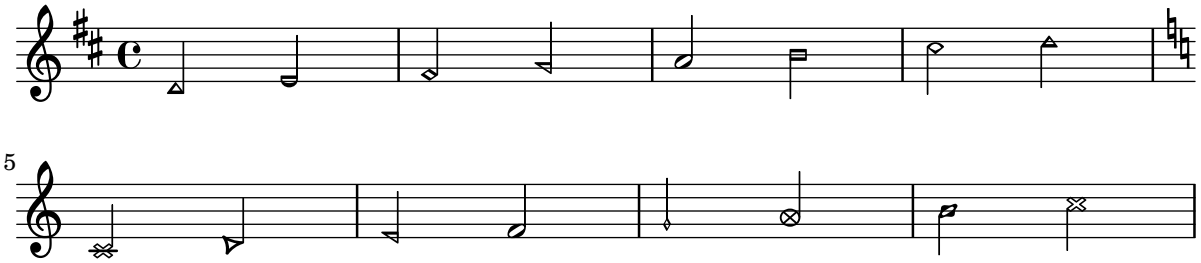
```
fragment = {
  \key c \major
  c2 d
  e2 f
  g2 a
  b2 c
}

\new Staff {
  \transpose c d
```

```
  \relative c' {
    \set shapeNoteStyles = ##(do re mi fa
                              #f la ti)
    \fragment
  }

  \break

  \relative c' {
    \set shapeNoteStyles = ##(cross triangle fa #f
                              mensural xcircle diamond)
    \fragment
  }
}
```

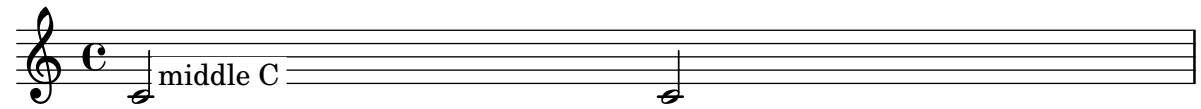


Blanchiment de lignes de portée avec la commande `\whiteout`

L'instruction `\whiteout` intercale un rectangle blanc contenant un *markup*. Dans la mesure où les lignes de la portée sont à un niveau inférieur à la plupart des autres objets graphiques, ce cache ne devrait pas masquer d'autres éléments.

```
\layout {
  ragged-right = ##f
}

\relative c' {
  \override TextScript.extra-offset = #'(2 . 4)
  c2-\markup { \whiteout \pad-markup #0.5 "middle C" } c
}
```



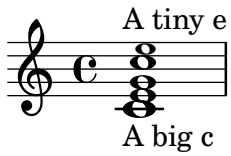
Modification de la taille d'une note particulière d'un accord

La commande `\tweak` permet de modifier la propriété `font-size` d'une note particulière d'un accord.

Il s'agit de placer, à l'intérieur même de l'accord (dans la construction `< >`) et avant la note considérée, l'instruction `\tweak` suivie de `font-size` et de définir la taille voulue – comme `#-2` pour une petite tête.

```
\relative c' {
  <\tweak font-size #+2 c e g c
```

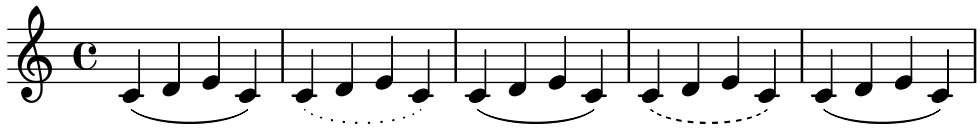
```
   \tweak font-size #-2 e>1
   ^\markup { A tiny e }_\markup { A big c }
}
```



Modification de l'aspect des liaisons d'articulation

Une liaison d'articulation peut se présenter sous la forme d'un trait continu ou discontinu, voire en pointillé.

```
\relative c' {
  c4( d e c)
  \slurDotted
  c4( d e c)
  \slurSolid
  c4( d e c)
  \slurDashed
  c4( d e c)
  \slurSolid
  c4( d e c)
}
```



Coloration des notes selon leur hauteur

Les têtes de note peuvent adopter une couleur différente selon leur hauteur ou leur nom ; la fonction utilisée ici fait même la distinction entre enharmoniques.

```
%Association list of pitches to colors.
#(define color-mapping
   (list
    (cons (ly:make-pitch 0 0 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 0 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 1 FLAT) (x11-color 'green))
    (cons (ly:make-pitch 0 2 NATURAL) (x11-color 'red))
    (cons (ly:make-pitch 0 2 SHARP) (x11-color 'green))
    (cons (ly:make-pitch 0 3 FLAT) (x11-color 'red))
    (cons (ly:make-pitch 0 3 NATURAL) (x11-color 'green))
    (cons (ly:make-pitch 0 4 SHARP) (x11-color 'red))
    (cons (ly:make-pitch 0 5 NATURAL) (x11-color 'green))
    (cons (ly:make-pitch 0 5 FLAT) (x11-color 'red))
    (cons (ly:make-pitch 0 6 SHARP) (x11-color 'red))
    (cons (ly:make-pitch 0 1 NATURAL) (x11-color 'blue))
    (cons (ly:make-pitch 0 3 SHARP) (x11-color 'blue))
    (cons (ly:make-pitch 0 4 FLAT) (x11-color 'blue))
    (cons (ly:make-pitch 0 5 SHARP) (x11-color 'blue))
```
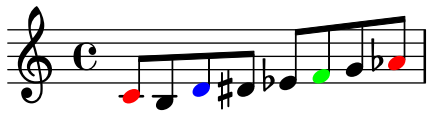
```
    (cons (ly:make-pitch 0 6 FLAT) (x11-color 'blue))))

%Compare pitch and alteration (not octave).
#(define (pitch-equals? p1 p2)
   (and
    (= (ly:pitch-alteration p1) (ly:pitch-alteration p2))
    (= (ly:pitch-notename p1) (ly:pitch-notename p2))))

#(define (pitch-to-color pitch)
   (let ((color (assoc pitch color-mapping pitch-equals?)))
     (if color
         (cdr color))))

#(define (color-notehead grob)
   (pitch-to-color
    (ly:event-property (event-cause grob) 'pitch)))

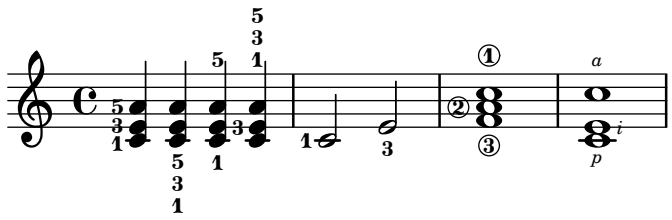
\score {
  \new Staff \relative c' {
    \override NoteHead.color = #color-notehead
    c8 b d dis ees f g aes
  }
}
```



Contrôler la position des doigtés dans un accord

Le positionnement des doigtés peut être contrôlé de manière très précise. Afin que l'orientation soit prise en compte, il est nécessaire d'utiliser une syntaxe d'accord `< >`, même s'il ne s'agit que d'une seule note. Le positionnement des numéros de corde et doigtés main droite se règle de manière analogue.
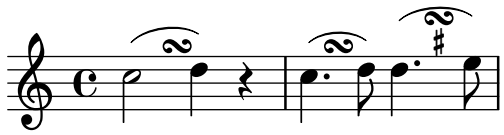
```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger #1 e\rightHandFinger #2 c'\rightHandFinger #4 >
}
```



Création d'un groupetto retardé

Obtenir un groupetto retardé et dans lequel la note la plus basse est altérée requiert quelques surcharges. La propriété `outside-staff-priority` doit être désactivée (`#f`) pour éviter qu'elle prenne le pas sur la propriété `avoid-slur`. L'ajustement du positionnement horizontal s'effectue en jouant sur les fractions `2/3` et `1/3`.

```
\relative c'' {
  c2*2/3 ( s2*1/3\turn d4) r
  <<
    { c4.( d8) }
    { s4 s\turn }
  >>
  \transpose c d \relative c'' <<
    { c4.( d8) }
    {
      s4
      \once \set suggestAccidentals = ##t
      \once \override AccidentalSuggestion.outside-staff-priority = ##f
      \once \override AccidentalSuggestion.avoid-slur = #'inside
      \once \override AccidentalSuggestion.font-size = -3
      \once \override AccidentalSuggestion.script-priority = -1
      \single \hideNotes
      b8-\turn \noBeam
      s8
    }
  >>
}
```



Impression de papier à musique

Rien n'est plus simple, pour obtenir une feuille blanche avec des portées, que de générer des mesures vides, de supprimer le `Bar_number_engraver` du contexte `Score`, et les `Time_signature_engraver`, `Clef_engraver` et `Bar_engraver` du contexte `Staff`.

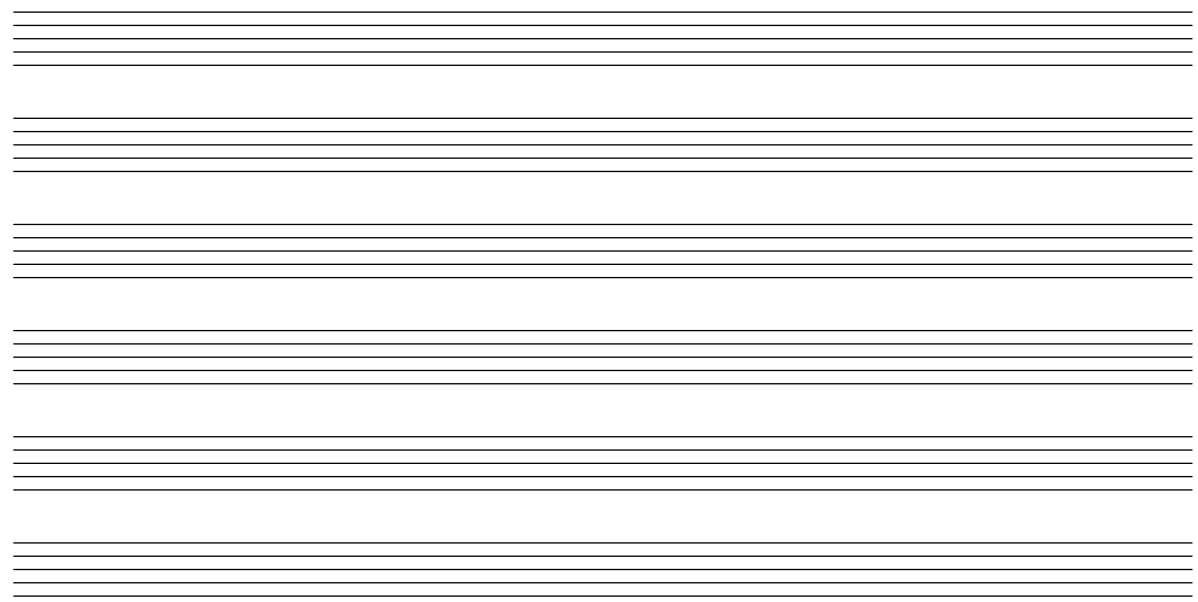
```
#(set-global-staff-size 20)

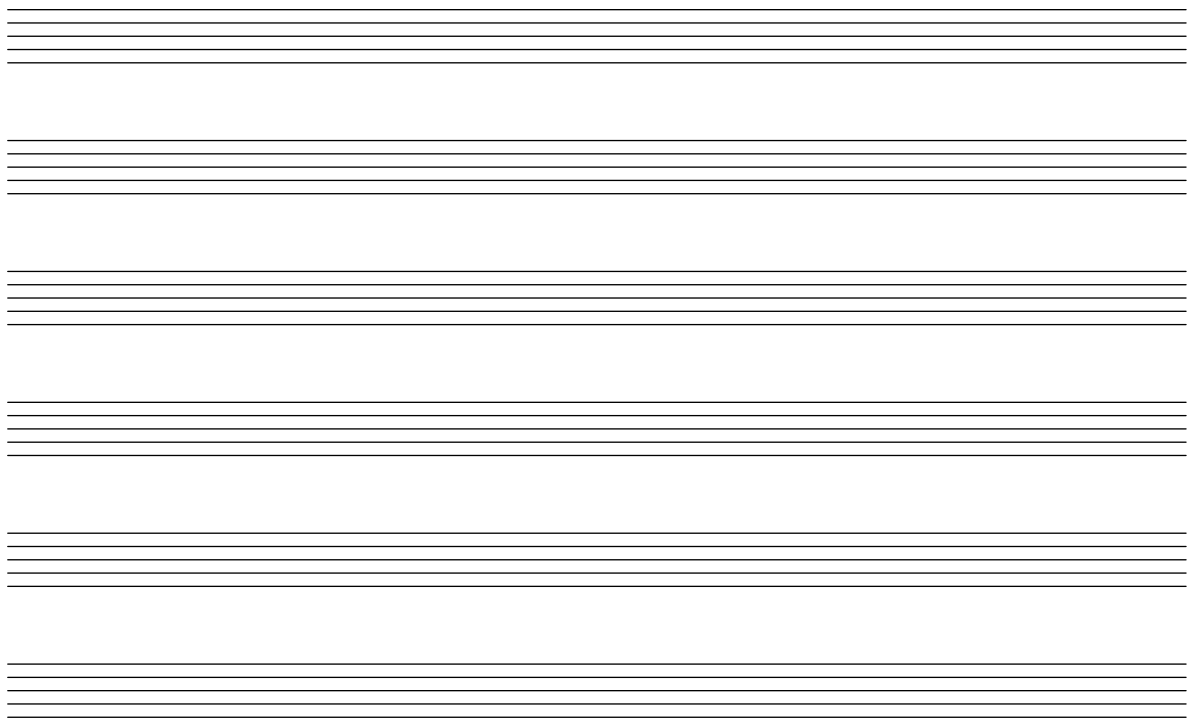
\score {
  {
    \repeat unfold 12 { s1 \break }
  }
  \layout {
    indent = 0\in
    \context {
      \Staff
```

```
      \remove "Time_signature_engraver"
      \remove "Clef_engraver"
      \remove "Bar_engraver"
    }
    \context {
      \Score
      \remove "Bar_number_engraver"
    }
  }
}

% uncomment these lines for "letter" size
%{
\paper {
  #(set-paper-size "letter")
  ragged-last-bottom = ##f
  line-width = 7.5\in
  left-margin = 0.5\in
  bottom-margin = 0.25\in
  top-margin = 0.25\in
}
%}

% uncomment these lines for "A4" size
%{
\paper {
  #(set-paper-size "a4")
  ragged-last-bottom = ##f
  line-width = 180
  left-margin = 15
  bottom-margin = 10
  top-margin = 10
}
%}
```

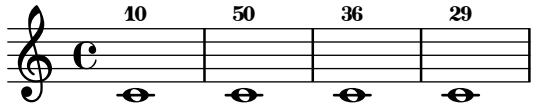




Création de doigtés sur deux caractères

Il est tout à fait possible de noter un doigté supérieur à 5.
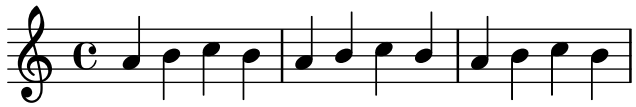
```
\relative c' {
  c1-10
  c1-50
  c1-36
  c1-29
}
```



Direction par défaut des hampes de la ligne médiane

La direction des hampes des notes placées sur la ligne médiane de la portée est gérée par la propriété `neutral-direction` de l'objet `Stem`.

```
\relative c'' {
  a4 b c b
  \override Stem.neutral-direction = #up
  a4 b c b
  \override Stem.neutral-direction = #down
  a4 b c b
}
```



Taille de police différente pour `instrumentName` et `shortInstrumentName`

Une simple dérogation à un contexte particulier permet de différencier la taille des variables *instrumentName* et *shortInstrumentName*.

```
InstrumentNameFontSize =
#(define-music-function (font-size-pair)(pair?)
"Sets the @code{font-size} of @code{InstrumentName}.
The font-size for the initial @code{instrumentName} is taken from the first
value in @var{font-size-pair}.  @code{shortInstrumentName} will get the second
value of @var{font-size-pair}.
"

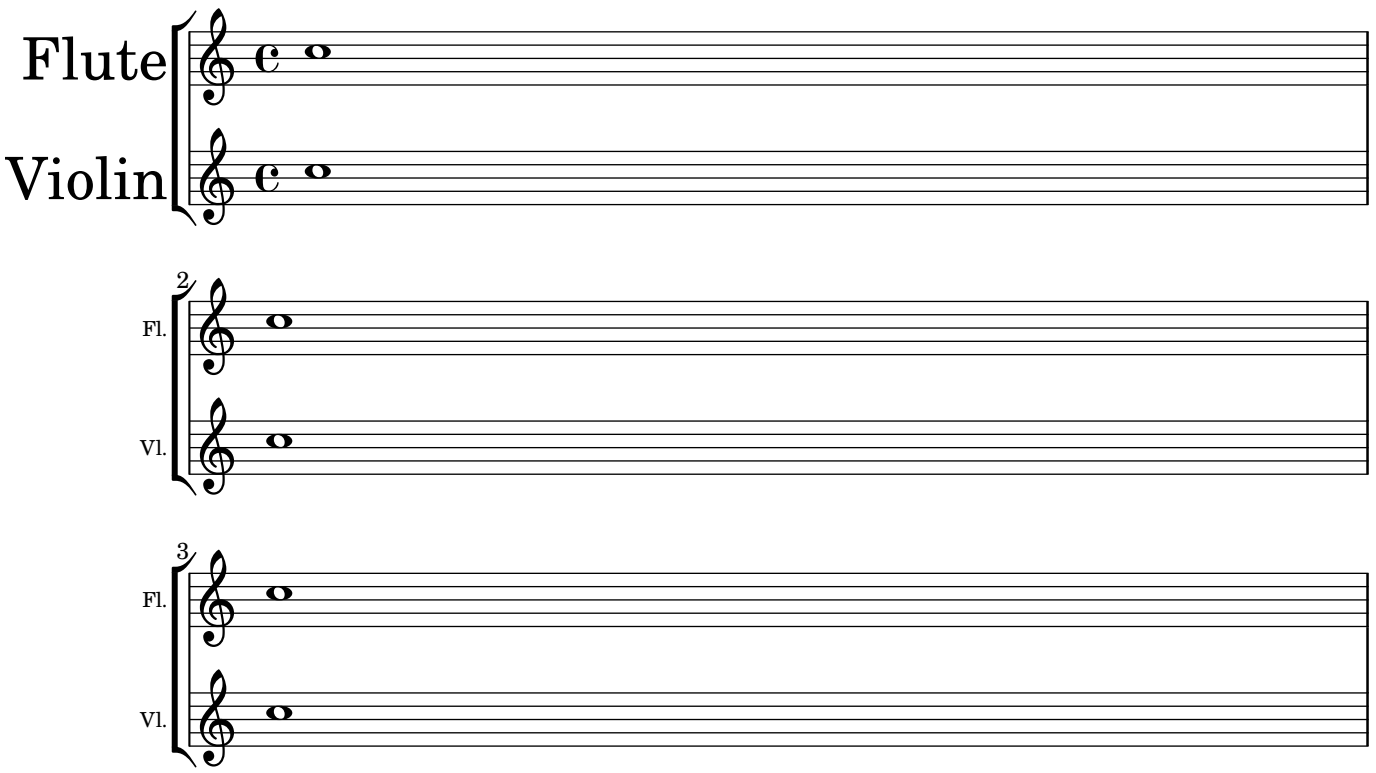
;; This code could be changed/extended to set different values for each
;; occurance of `shortInstrumentName'

#{
  \override InstrumentName.after-line-breaking =
    #(lambda (grob)
       (let* ((orig (ly:grob-original grob))
              (siblings (if (ly:grob? orig)
                            (ly:spanner-broken-into orig)
                            '())))
         (if (pair? siblings)
             (begin
               (ly:grob-set-property!
                 (car siblings)
                 'font-size
                 (car font-size-pair))
               (for-each
                 (lambda (g)
                   (ly:grob-set-property! g 'font-size (cdr font-size-pair)))
                 (cdr siblings))))))
#})

\layout {
  \context {
    \Staff
    \InstrumentNameFontSize #'(6 . -3)
  }
}

\new StaffGroup <<
  \new Staff
     \with {
       instrumentName = "Flute"
       shortInstrumentName = "Fl."
     }
     { c''1 \break c'' \break c'' }
  \new Staff
   \with {
     instrumentName = "Violin"
     shortInstrumentName = "Vl."
```

```
   }
   { c''1 \break c'' \break c'' }
>>
```



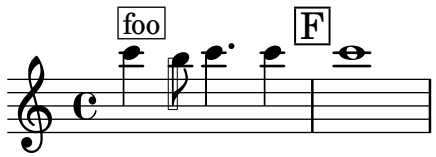
Encadrement d'objets

La fonction `print` peut se modifier pour obtenir l'encadrement de n'importe quel objet.

```
\relative c'' {
  \override TextScript.stencil =
    #(make-stencil-boxer 0.1 0.3 ly:text-interface::print)
  c'4^"foo"

  \override Stem.stencil =
    #(make-stencil-boxer 0.05 0.25 ly:stem::print)
  \override Score.RehearsalMark.stencil =
    #(make-stencil-boxer 0.15 0.3 ly:text-interface::print)
  b8

  \revert Stem.stencil
  \revert Flag.stencil
  c4. c4
  \mark "F"
  c1
}
```
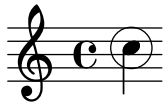


Encerclement de notes

Voici comment entourer d'un cercle une note :

```
circle =
\once \override NoteHead.stencil = #(lambda (grob)
    (let* ((note (ly:note-head::print grob))
           (combo-stencil (ly:stencil-add
               note
               (circle-stencil note 0.1 0.8))))
          (ly:make-stencil (ly:stencil-expr combo-stencil)
            (ly:stencil-extent note X)
            (ly:stencil-extent note Y))))

{ \circle c'' }
```

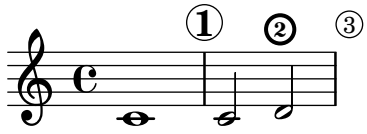


Encercler divers objets

La commande de *markup* `\circle` permet de dessiner un cercle autour de différents objets comme des indications de doigté. D'autres objets nécessitent de faire appel à des techniques spécifiques. Cet exemple illustre deux manières de procéder : pour les repères, et pour les numéros de mesure.

```
\relative c' {
  c1
  \set Score.markFormatter =
    #(lambda (mark context)
              (make-circle-markup (format-mark-numbers mark context)))
  \mark \default

  c2 d^\markup {
    \override #'(thickness . 3) {
      \circle \finger 2
    }
  }
  \override Score.BarNumber.break-visibility = #all-visible
  \override Score.BarNumber.stencil =
    #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
}
```
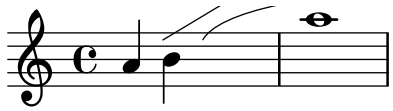


Intégration de PostScript dans un *markup*

Du code PostScript peut directement être intégré dans un bloc `\markup`.

```
% PostScript is a registered trademark of Adobe Systems Inc.
```

```
\relative c'' {
  a4-\markup { \postscript "3 4 moveto 5 3 rlineto stroke" }
  -\markup { \postscript "[ 0 1 ] 0 setdash 3 5 moveto 5 -3 rlineto stroke " }

  b4-\markup { \postscript "3 4 moveto 0 0 1 2 8 4 20 3.5 rcurveto stroke" }
  s2
  a'1
}
```

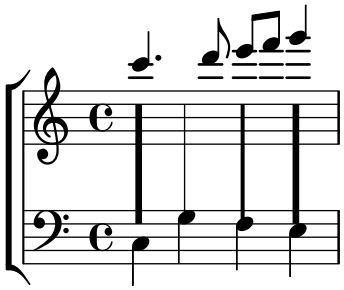


Apparence du quadrillage temporel

Modifier certaines des propriétés du quadrillage temporel aura pour effet d'en changer l'apparence.

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \relative c'' {
        \stemUp
        c'4. d8 e8 f g4
      }
    }
    \new Staff {
      \relative c {
        % this moves them up one staff space from the default position
        \override Score.GridLine.extra-offset = #'(0.0 . 1.0)
        \stemDown
        \clef bass
        \once \override Score.GridLine.thickness = #5.0
        c4
        \once \override Score.GridLine.thickness = #1.0
        g'4
        \once \override Score.GridLine.thickness = #3.0
        f4
        \once \override Score.GridLine.thickness = #5.0
        e4
      }
    }
  >>
  \layout {
    \context {
      \Staff
      % set up grids
      \consists "Grid_point_engraver"
      % set the grid interval to one quarter note
      gridInterval = #(ly:make-moment 1/4)
    }
    \context {
      \Score
```

```
      \consists "Grid_line_span_engraver"
      % this moves them to the right half a staff space
      \override NoteColumn.X-offset = #-0.5
    }
  }
}
```



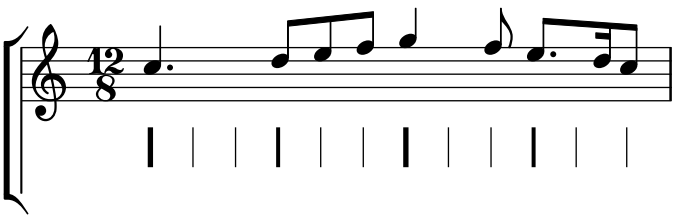
Quadrillage temporel : mise en évidence du rythme et synchronisation des notes

Des lignes verticales entre les portées peuvent s'ajouter dans le but d'indiquer la synchronisation entre des notes. Dans le cas de musique monophonique, on peut toutefois créer une deuxième portée, invisible, et raccourcir les lignes comme ici.

```
\score {
  \new ChoirStaff {
    \relative c'' <<
      \new Staff {
        \time 12/8
        \stemUp
        c4. d8 e8 f g4 f8 e8. d16 c8
      }
      \new Staff {
        % hides staff and notes so that only the grid lines are visible
        \hideNotes
        \hide Staff.BarLine
        \override Staff.StaffSymbol.line-count = #0
        \hide Staff.TimeSignature
        \hide Staff.Clef

        % dummy notes to force regular note spacing
        \once  \override Score.GridLine.thickness = #4.0
        c8 c c
        \once  \override Score.GridLine.thickness = #3.0
        c8 c c
        \once  \override Score.GridLine.thickness = #4.0
        c8 c c
        \once  \override Score.GridLine.thickness = #3.0
        c8 c c
      }
    >>
  }
  \layout {
    \context {
```

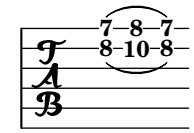
```
      \Score
      \consists "Grid_line_span_engraver"
      % center grid lines horizontally below note heads
      \override NoteColumn.X-offset = #-0.5
    }
    \context {
      \Staff
      \consists "Grid_point_engraver"
      gridInterval = #(ly:make-moment 1/8)
      % set line length and positioning:
      % two staff spaces above center line on hidden staff
      % to four spaces below center line on visible staff
      \override GridPoint.Y-extent = #'(2 . -4)
    }
    ragged-right = ##t
  }
}
```



Hammer on et pull off avec accords

Dans le cadre de notes en accord, les *hammer-on* et *pull-off* sont indiqués par un arc simple. Vous obtiendrez néanmoins un arc double en réglant la propriété `doubleSlurs` sur `#t`.

```
\new TabStaff {
  \relative c' {
    % chord hammer-on and pull-off
    \set doubleSlurs = ##t
    <g' b>8( <a c> <g b>)
  }
}
```

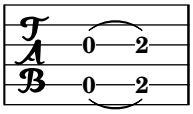


Hammer on et pull off gérés par les voix

L'arc des *hammer-on* et *pull-off* est ascendant dans les voix une et trois, et descendant dans les voix deux et quatre.

```
\new TabStaff {
  \relative c' {
    << { \voiceOne g2( a) }
    \\ { \voiceTwo a,( b) }
    >> \oneVoice
```

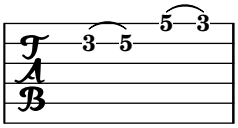
```
  }
}
```



Hammer on et pull off

Hammer-on et *pull-off* peuvent s'indiquer par des liaisons.

```
\new TabStaff {
  \relative c' {
    d4( e\2)
    a( g)
  }
}
```



Impression d'un double repère sur une même barre (méthode 1)

Cette méthode permet d'imprimer deux marques-repères l'une au-dessus de l'autre. Le repère inférieur est décalé sous la portée, puis s'insère au-dessus de lui un espace pour positionner le repère supérieur en surplomb de la portée.

L'ajustement des valeurs de `extra-offset` et `baseline-skip` permet d'accroître ou réduire globalement l'espace entre le repère et la portée.

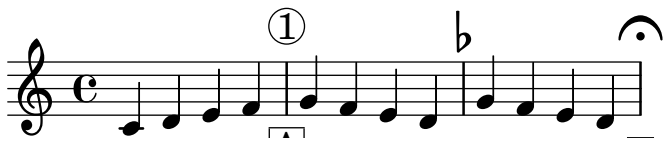
La capacité de pratiquement tout type de glyphe ou chaîne de caractères à faire l'objet d'un repère les rend capables de venir se centrer au-dessus ou en dessous d'une barre de mesure.

L'adjonction de la propriété `break-visibility`, comme dans cet autre cas (`\%22http://lsr.di.unimi.it/LSR/Item?id=1\%22`), permet de placer un double repère en fin de ligne.

Cette première méthode, bien que moins compliquée que la seconde, ne permet pas d'ajuster le positionnement d'un repère indépendamment de l'autre. Elle peut aussi générer quelques problèmes quant à l'espacement vertical, puisque le recours à `extra-offset` ne modifie en rien la valeur originelle de la boîte englobant le repère.

```
\relative c'{
  c d e f |
  \once \override Score.RehearsalMark.extra-offset = #'(0 . -8.5)
  \once \override Score.RehearsalMark.baseline-skip = #9
  \mark \markup \center-column { \circle 1 \box A }
  g f e d |
  \once \override Score.RehearsalMark.extra-offset = #'(0 . -8.5)
  \once \override Score.RehearsalMark.baseline-skip = #9
  \mark \markup \center-column { \flat { \bold \small \italic Fine. } }
  g f e d |
  \once \override Score.RehearsalMark.extra-offset = #'(0 . -8.5)
  \once \override Score.RehearsalMark.baseline-skip = #9
  \override Score.RehearsalMark.break-visibility = #begin-of-line-invisible
```

```
  \mark \markup \center-column { \fermata \box z }
}
```



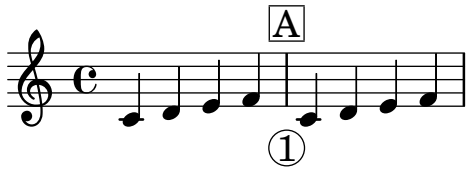
Impression d'un double repère sur une même barre (méthode 2)

Cette méthode permet d'imprimer deux marques-repères – l'une au-dessus de la portée et l'autre en dessous – à l'aide de deux voix séparées. Le graveur de repères est ensuite déplacé dans chacune des voix, sans quoi aucun repère ne sera imprimé. Enfin, chaque voix se voit attribuer un positionnement pour les repères.

Cette méthode, bien que plus complexe que la première, apporte plus de flexibilité dans l'ajustement individuel des deux composants du repère.

```
\score {
  \relative c'
  <<
    \new Staff {
      <<
        \new Voice \with {
          \consists Mark_engraver
          \consists "Staff_collecting_engraver"
        }
        { c4 d e f
          \mark \markup { \box A }
          c4 d e f
        }
        \new Voice \with {
          \consists Mark_engraver
          \consists "Staff_collecting_engraver"
          \override RehearsalMark.direction = #DOWN
        }
        { s4 s s s
          \mark \markup { \circle 1 }
          s4 s s s
        }
      >>
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Mark_engraver"
      \remove "Staff_collecting_engraver"
    }
  }
```
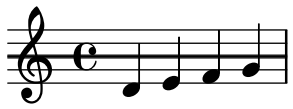
```
}
```



Empâtement de certaines lignes d'une portée

Vous pourriez avoir envie, dans un but pédagogique, de rendre certaines lignes d'une portée plus épaisses que les autres, comme la ligne médiane, ou bien pour mettre en exergue la ligne portant la clef de sol. Il suffit pour cela d'ajouter une ligne qui sera accolée à celle qui doit être mise en évidence, grâce à la propriété `line-positions` de l'objet `StaffSymbol`.

```
{
  \override Staff.StaffSymbol.line-positions =
    #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```



Marking notes of spoken parts with a cross on the stem (Sprechstimme)

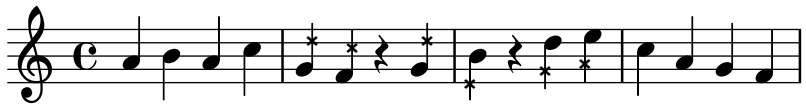
This example shows how to put crosses on stems. Mark the beginning of a spoken section with the `\speakOn` keyword, and end it with the `\speakOff` keyword.

```
speakOn = {
  \override Stem.stencil =
    #(lambda (grob)
       (let* ((x-parent (ly:grob-parent grob X))
              (is-rest? (ly:grob? (ly:grob-object x-parent 'rest))))
         (if is-rest?
             empty-stencil
             (ly:stencil-combine-at-edge
              (ly:stem::print grob)
              Y
              (- (ly:grob-property grob 'direction))
              (grob-interpret-markup grob
                                     (markup #:center-align #:fontsize -4
                                             #:musicglyph "noteheads.s2cross"))
              -2.3))))
}

speakOff = {
  \revert Stem.stencil
  \revert Flag.stencil
}

\score {
```
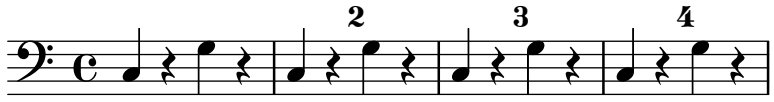
```
  \new Staff {
    \relative c'' {
      a4 b a c
      \speakOn
      g4 f r g
      b4 r d e
      \speakOff
      c4 a g f
    }
  }
}
```



Compteur de mesures

Le code ci-dessous permet, de façon détournée, d'afficher un compteur de mesures à l'aide d'une répétition en pourcent rendue transparente.

```
<<
  \context Voice = "foo" {
    \clef bass
    c4 r g r
    c4 r g r
    c4 r g r
    c4 r g r
  }
  \context Voice = "foo" {
    \set countPercentRepeats = ##t
    \hide PercentRepeat
    \override PercentRepeatCounter.staff-padding = #1
    \repeat percent 4 { s1 }
  }
>>
```



Numérotation de groupes de mesures

Cet extrait illustre l'utilisation du `Measure_counter_engraver` aux fins de numéroter un groupe de mesures successives. La section à numéroter peut éventuellement comporter des reprises.

Le graveur doit être ajouté au contexte approprié. Il est ici attaché au contexte `Staff`, mais pourrait l'être à un contexte `Dynamics`.

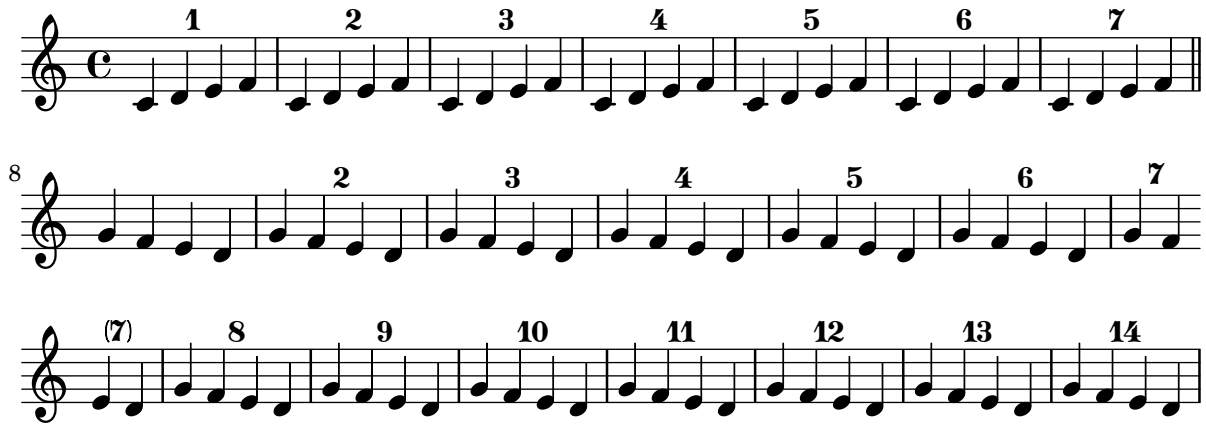
Le compteur est lancé par `\startMeasureCount` et arrêté par `\stopMeasureCount`. La numérotation commence à `1` par défaut, comportement qui peut se modifier à l'aide de la propriété `count-from`.

Lorsqu'une mesure s'étend après un saut de ligne, le numéro y est alors répété, entre parenthèses.

```
\layout {
```

```
  \context {
    \Staff
    \consists #Measure_counter_engraver
  }
}

\new Staff {
  \startMeasureCount
  \repeat unfold 7 {
    c'4 d' e' f'
  }
  \stopMeasureCount
  \bar "||"
  g'4 f' e' d'
  \override Staff.MeasureCounter.count-from = #2
  \startMeasureCount
  \repeat unfold 5 {
    g'4 f' e' d'
  }
  g'4 f'
  \bar ""
  \break
  e'4 d'
  \repeat unfold 7 {
    g'4 f' e' d'
  }
  \stopMeasureCount
}
```

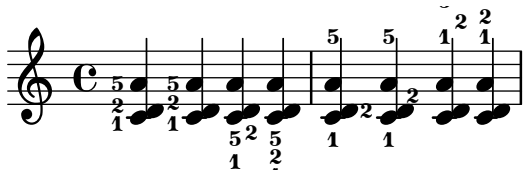


Positionnement précis des indications de doigté

Les options par défaut en matière de positionnement des doigtés d'un accord donnent généralement de bons résultats. Il se peut néanmoins qu'un ajustement soit nécessaire dans certains cas particuliers, notamment en présence d'un intervalle de seconde. L'astuce ici présentée permet d'obtenir un meilleur rendu.

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 d-2 a'-5>4
  <c-1 d-\tweak extra-offset #'(0 . 0.2)-2 a'-5>4
```
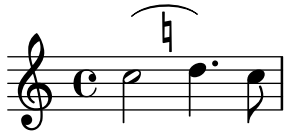
```
  \set fingeringOrientations = #'(down)
  <c-1 d-2 a'-5>4
  <c-\tweak extra-offset #'(0 . -1.1)-1
   d-\tweak extra-offset #'(-1.2 . -1.8)-2 a'-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 d-\tweak extra-offset #'(-0.3 . 0)-2 a'-5>4
  <c-1 d-\tweak extra-offset #'(-1 . 1.2)-2 a'-5>4
  \set fingeringOrientations = #'(up)
  <c-1 d-\tweak extra-offset #'(0 . 1.1)-2
   a'-\tweak extra-offset #'(0 . 1)-5>4
  <c-1 d-\tweak extra-offset #'(-1.2 . 1.5)-2
   a'-\tweak extra-offset #'(0 . 1.4)-5>4
}
```



Positionnement d'une annotation à l'intérieur d'une liaison

Lorsqu'une annotation doit s'incrire à l'intérieur d'une liaison, la propriété `outside-staff-priority` doit être désactivée.

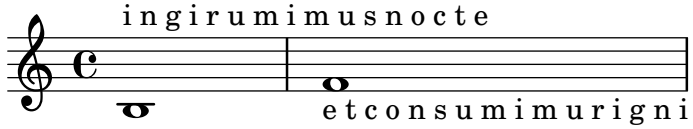
```
\relative c'' {
  \override TextScript.avoid-slur = #'inside
  \override TextScript.outside-staff-priority = ##f
  c2(^\markup { \halign #-10 \natural } d4.) c8
}
```



Impression de texte de droite à gauche

Du texte, inclus dans un objet *markup*, peut s'imprimer de droite à gauche, comme illustré ci-dessous.

```
{
  b1^\markup {
    \line { i n g i r u m i m u s n o c t e }
  }
  f'_\markup {
    \override #'(text-direction . -1)
    \line { i n g i r u m i m u s n o c t e }
  }
}
```

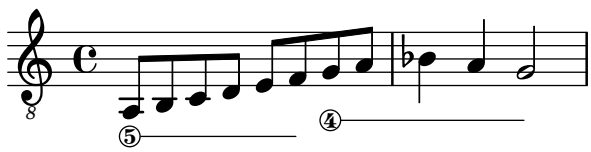


Ligne de prolongation pour numéro de corde

Voici comment ajouter une ligne de prolongation à une indication de numéro de corde, afin de stipuler que les notes qui suivent doivent être jouées sur la corde en question.

```
stringNumberSpanner =
  #(define-music-function (StringNumber) (string?)
    #{
      \override TextSpanner.style = #'solid
      \override TextSpanner.font-size = #-5
      \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
      \override TextSpanner.bound-details.left.text =
        \markup { \circle \number $StringNumber }
    #})


\relative c {
  \clef "treble_8"
  \stringNumberSpanner "5"
  \textSpannerDown
  a8\startTextSpan
  b c d e f\stopTextSpan
  \stringNumberSpanner "4"
  g\startTextSpan a
  bes4 a g2\stopTextSpan
}
```
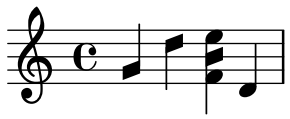


Utilisation de PostScript pour générer des têtes de note à l'allure particulière

Lorsqu'il est impossible d'obtenir facilement une allure particulière pour les têtes de note en recourant à la technique du `\markup`, un code PostScript peut vous tirer d'embarras. Voici comment générer des têtes ressemblant à des parallélogrammes.

```
parallelogram =
  #(ly:make-stencil (list 'embedded-ps
    "gsave
      currentpoint translate
      newpath
      0 0.25 moveto
      1.3125 0.75 lineto
      1.3125 -0.25 lineto
      0 -0.75 lineto
      closepath
      fill
      grestore" )
    (cons 0 1.3125)
    (cons -.75 .75))
```

```
myNoteHeads = \override NoteHead.stencil = \parallelogram
normalNoteHeads = \revert NoteHead.stencil

\relative c'' {
  \myNoteHeads
  g4 d'
  \normalNoteHeads
  <f, \tweak stencil \parallelogram b e>4 d
}
```
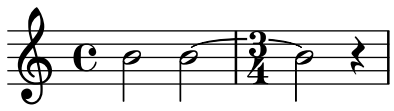


Utilisation de la propriété `whiteout`

Tout objet graphique peut s'imprimer sur un fond blanc afin de masquer une partie des objets qu'il recouvre. Ceci trouve toute son utilité pour améliorer certaines collisions, notamment dans des situations où un repositionnement d'objets est irréaliste. Il faut alors explicitement définir la propriété `layer` afin de contrôler quels objets seront masqués par le fond blanc.

Dans l'exemple ci-dessous, la liaison est en collision avec la métrique ; la situation est améliorée dès lors que la portion de liaison qui traverse la métrique est masquée par l'affectation de la propriété `whiteout` à l'objet `TimeSignature`. Pour ce faire, `TimeSignature` est déplacé sur un calque au-dessus de celui de `Tie` – il reste sur le calque par défaut (1) –, puis le `StaffSymbol` est placé sur un calque supérieur à celui de `TimeSignature` pour ne pas être masqué.

```
{
  \override Score.StaffSymbol.layer = #4
  \override Staff.TimeSignature.layer = #3
  b'2 b'~
  \once \override Staff.TimeSignature.whiteout = ##t
  \time 3/4
  b' r4
}
```



Text

Section “Text” dans *Manuel de notation*

Ajout de *markups* à une tablature

Par défaut, les *markups* n’apparaissent pas dans une tablature.

Il suffit, pour les voir apparaître, d’une simple commande `\revert TabStaff.TextScript.stencil`.

```
%% http://lsr.di.unimi.it/LSR/Item?id=919
% by P.P.Schneider on June 2014

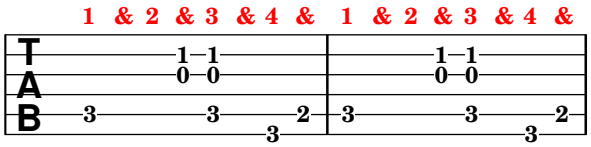
high  = { r4 r8 <g c'> q r8 r4 }

low = { c4 r4 c8 r8 g,8 b, }

pulse = { s8^"1" s^"&" s^"2" s^"&" s^"3" s^"&" s^"4" s^"&" }

\score {
 \new TabStaff {
   \repeat unfold 2 << \high \\ \low \\ \pulse >>
  }
  \layout {
    \context {
      \TabStaff
      \clef moderntab
      \revert TextScript.stencil
      \override TextScript.font-series = #'bold
      \override TextScript.font-size = #-2
      \override TextScript.color = #red

    }
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1/8)
    }
  }
}
```



Ajout de la date du jour à une partition

Avec un peu de code Scheme, voici comment ajouter facilement la date du jour à votre partition.

```
% first, define a variable to hold the formatted date:
date = #(strftime "%d-%m-%Y" (localtime (current-time)))
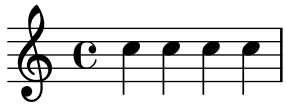
% use it in the title block:
\header {
```

```
  title = "Including the date!"
  subtitle = \date
}

\score {
  \relative c'' {
    c4 c c c
  }
}
% and use it in a \markup block:
\markup {
  \date
}
```

Including the date!

11-04-2021
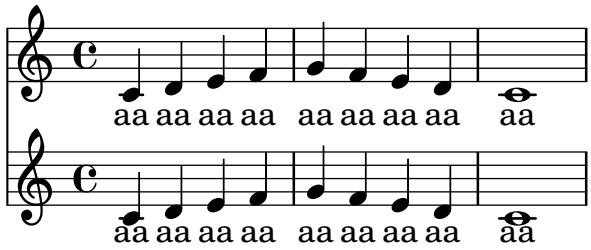


11-04-2021

Ajustement de l'espacement vertical des paroles

Cet extrait illustre la manière de réduire l'espace entre la ligne de paroles et la portée.

```
% Default layout:
<<
  \new Staff \new Voice = melody \relative c' {
    c4 d e f
    g4 f e d
    c1
  }
  \new Lyrics \lyricsto melody { aa aa aa aa aa aa aa aa aa }

  \new Staff {
    \new Voice = melody \relative c' {
      c4 d e f
      g4 f e d
      c1
    }
  }
  % Reducing the minimum space below the staff and above the lyrics:
  \new Lyrics \with {
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing =
      #'((basic-distance . 1))
  }
  \lyricsto melody { aa aa aa aa aa aa aa aa aa }
>>
```



Alignement des noms d'instrument

L'alignement horizontal des noms d'instrument se gère à l'aide de la propriété `Staff.InstrumentName.self-alignment-X`. Les variables `indent` et `short-indent`, attachées au bloc `\layout`, déterminent l'espace alloué à l'alignement des noms d'instrument, respectivement dans leurs formes développée et abrégée.

```
\paper { left-margin = 3\cm }

\score {
  \new StaffGroup <<

    \new Staff \with {
      \override InstrumentName.self-alignment-X = #LEFT
      instrumentName = \markup \left-column {
        "Left aligned"
        "instrument name"
        }
        shortInstrumentName = "Left"
      }

      {  c''1 \break c''1 }

    \new Staff \with {
      \override InstrumentName.self-alignment-X = #CENTER
      instrumentName = \markup \center-column {
        Centered
        "instrument name"
        }
      shortInstrumentName = "Centered"
    }

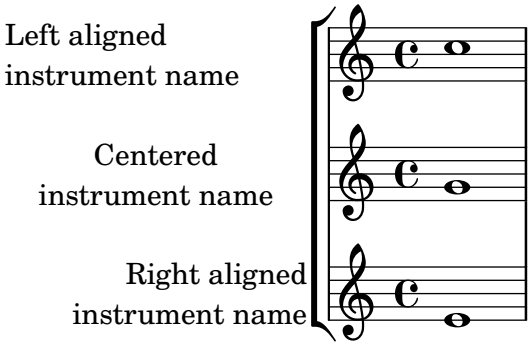
    { g'1 g'1}

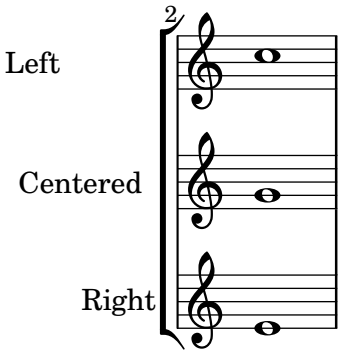
    \new Staff \with {
      \override InstrumentName.self-alignment-X = #RIGHT
      instrumentName = \markup \right-column {
        "Right aligned"
        "instrument name"
      }
      shortInstrumentName = "Right"
    }

    { e'1 e'1 }

  >>
```

```
  \layout {
    ragged-right = ##t
    indent = 4\cm
    short-indent = 2\cm
  }
}
```

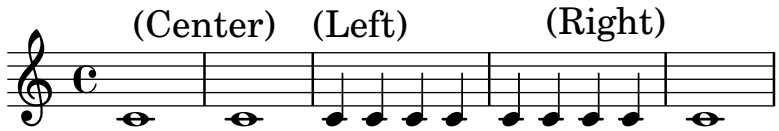




Alignement des objets créés par la commande \mark

Par défaut, la commande **\mark** centre les objets sur la barre de mesure. Ce comportement peut se modifier pour un alignement par la gauche ou par la droite.
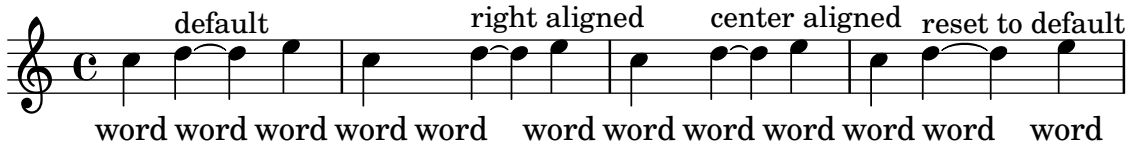
```
\relative c' {
  c1 \mark "(Center)"
  c1
  \once \override Score.RehearsalMark.self-alignment-X = #LEFT
  \mark "(Left)"
  c4 c c c
  c4 c c c
  \once \override Score.RehearsalMark.self-alignment-X = #RIGHT
  \mark "(Right)"
  c1
}
```



Alignement de la première syllabe d'un mélisme

Par défaut, une syllabe qui entame un mélisme est alignée sur la note par la gauche. Cet alignement peut se modifier à l'aide de la propriété `lyricMelismaAlignment`.

```
\score {
  <<
    \new Staff {
      \relative c''
      \new Voice = "vocal" {
        c d~^\markup default  d e
        c d~^\markup "right aligned" d e
        c d~^\markup "center aligned" d e
        c d~^\markup "reset to default" d e
      }
    }
    \new Lyrics \lyricsto "vocal" {
      word word word
      \set lyricMelismaAlignment = #RIGHT
      word word word
      \set lyricMelismaAlignment = #CENTER
      word word word
      \unset lyricMelismaAlignment
      word word word
    }
  >>
}
```

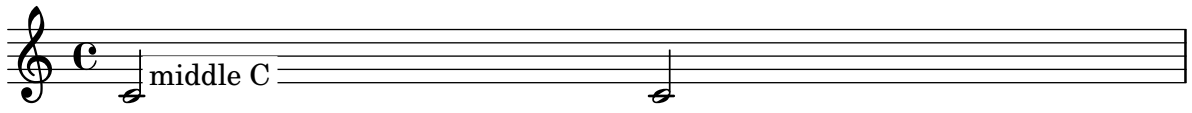


Blanchiment de lignes de portée avec la commande `\whiteout`

L'instruction `\whiteout` intercale un rectangle blanc contenant un *markup*. Dans la mesure où les lignes de la portée sont à un niveau inférieur à la plupart des autres objets graphiques, ce cache ne devrait pas masquer d'autres éléments.

```
\layout {
  ragged-right = ##f
}

\relative c' {
  \override TextScript.extra-offset = #'(2 . 4)
  c2-\markup { \whiteout \pad-markup #0.5 "middle C" } c
}
```



Texte centré sous un soufflet

La fonction comprise dans l'extrait suivant permet d'ajouter du texte – comme « molto » ou « poco » – en dessous d'un soufflet de (de)crescendo. Cet exemple présente aussi comment, à l'aide de code Scheme, influencer la manière dont un objet est normalement imprimé.

```
\paper { tagline = ##f }

hairpinWithCenteredText =
#(define-music-function (text) (markup?)
  #{
    \once \override Voice.Hairpin.after-line-breaking =
      #(lambda (grob)
        (let* ((stencil (ly:hairpin::print grob))
               (par-y (ly:grob-parent grob Y))
               (dir (ly:grob-property par-y 'direction))
               (staff-line-thickness
                 (ly:output-def-lookup (ly:grob-layout grob) 'line-thickness))
               (new-stencil (ly:stencil-aligned-to
                 (ly:stencil-combine-at-edge
                   (ly:stencil-aligned-to stencil X CENTER)
                   Y dir
                   (ly:stencil-aligned-to
                     (grob-interpret-markup
                       grob
                       (make-fontsize-markup
                         (magnification->font-size
                           (+ (ly:staff-symbol-staff-space grob)
                              (/ staff-line-thickness 2)))
                         text)) X CENTER))
                 X LEFT))
               (staff-space (ly:output-def-lookup
                 (ly:grob-layout grob) 'staff-space))
               (par-x (ly:grob-parent grob X))
               (dyn-text (grob::has-interface par-x 'dynamic-text-interface))
               (dyn-text-stencil-x-length
                 (if dyn-text
                   (interval-length
                     (ly:stencil-extent (ly:grob-property par-x 'stencil) X))
                   0))
               (x-shift
                 (if dyn-text
                   (-
                     (+ staff-space dyn-text-stencil-x-length)
                     (* 0.5 staff-line-thickness)) 0)))

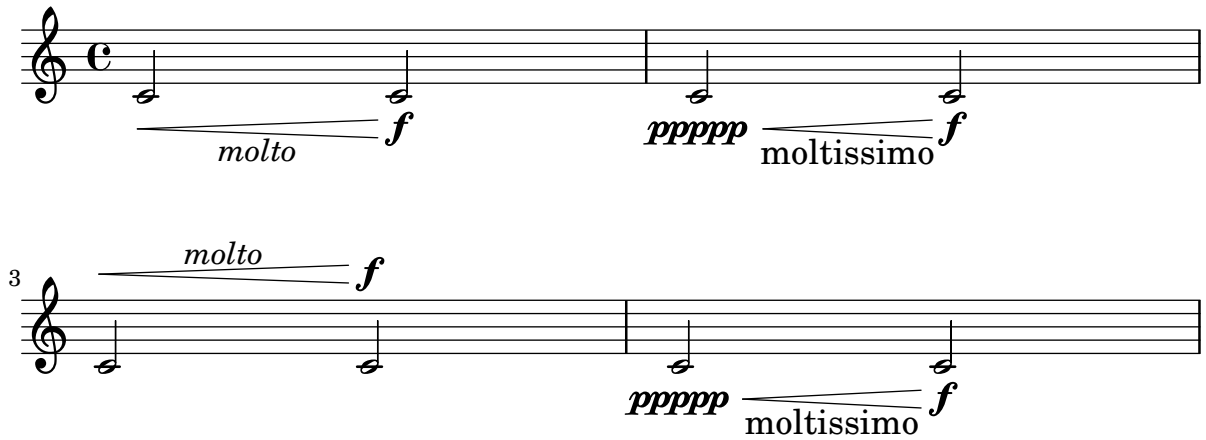
          (ly:grob-set-property! grob 'Y-offset 0)
          (ly:grob-set-property! grob 'stencil
             (ly:stencil-translate-axis
              new-stencil
              x-shift X))))
  #})
```

```
hairpinMolto =
\hairpinWithCenteredText \markup { \italic molto }

hairpinMore =
\hairpinWithCenteredText \markup { \larger moltissimo }

\layout { ragged-right = ##f }

\relative c' {
  \hairpinMolto
  c2\< c\f
  \hairpinMore
  c2\ppppp\< c\f
  \break
  \hairpinMolto
  c2^\< c\f
  \hairpinMore
  c2\ppppp\< c\f
}
```

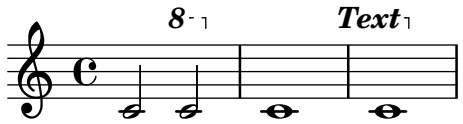


Modification du texte des marques d'octaviation

En interne, la fonction `\ottava` détermine les propriétés `ottavation` (par ex. en `"8va"` ou `"8vb"`) et `middleCPosition`. Vous pouvez modifier le texte d'une marque d'octaviation en définissant `ottavation` après avoir fait appel à `ottava`.

Un texte bref est particulièrement utile lorsque l'octaviation est courte.

```
{
  c'2
  \ottava #1
  \set Staff.ottavation = #"8"
  c''2
  \ottava #0
  c'1
  \ottava #1
  \set Staff.ottavation = #"Text"
  c''1
}
```



Changement des fontes textuelles par défaut

Les familles de fontes par défaut pour le texte peuvent être changées à l'aide de la fonction `make-pango-font-tree`.

```
%{
You may have to install additional fonts.

Red Hat Fedora

  dejavu-fonts-all

Debian GNU/Linux, Ubuntu

  fonts-dejavu-core
  fonts-dejavu-extra
%}

\paper {
  % change for other default global staff size.
  myStaffSize = #20
  %{
     run
         lilypond -dshow-available-fonts
     to show all fonts available in the process log.
  %}

  #(define fonts
    (make-pango-font-tree "DejaVu Serif"
                          "DejaVu Sans"
                          "DejaVu Sans Mono"
     (/ myStaffSize 20)))
}

{
  g'''4^\markup {
    DejaVu Serif: \bold bold
                  \italic italic
                  \italic \bold { bold italic }
  }
  g4_\markup {
    \override #'(font-family . sans) {
      DejaVu Sans: \bold bold
                   \italic italic
                   \italic \bold { bold italic }
    }
  }
  g''2^\markup {
    \override #'(font-family . typewriter) {
      DejaVu Sans Mono: \bold bold
                        \italic italic
```

```
                    \italic \bold { bold italic }
    }
  }
}
```



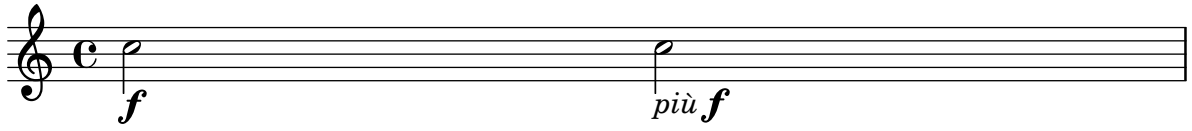
Combinaison de nuance et de texte

Certaines indications de nuance peuvent requérir un complément textuel, comme « più forte » ou « piano subito ». Elles se réalisent aisément à l'aide d'un bloc `\markup`.

```
piuF = \markup { \italic più \dynamic f }

\layout { ragged-right = ##f }

\relative c'' {
  c2\f c-\piuF
}
```



Combinaison de deux parties sur une même portée

L'outil de combinaison de parties (la commande `\partCombine`) permet d'avoir deux parties différentes sur une même portée. LilyPond ajoute automatiquement des indications textuelles, telles que « solo » ou « a2 ». Si votre intention n'est que de fusionner les parties, sans ajouter de texte, assignez faux à la propriété `printPartCombineTexts`. Dans le cas de partitions vocales, et plus particulièrement d'hymnes, ces « solo/a2 » ne sont d'aucune utilité, aussi vaut-il mieux les désactiver. Dans le cas où il y aurait alternance entre *solo* et *tutti*, il vaut mieux faire appel à de la musique polyphonique standard.

Voici trois moyens d'imprimer deux parties sur une même portée : en polyphonie normale, avec `\partCombine` sans indication supplémentaire, et avec `\partCombine` commentée.

```
%% Combining pedal notes with clef changes

musicUp = \relative c'' {
  \time 4/4
  a4 c4.( g8) a4 |
  g4 e' g,( a8 b) |
  c b a2.
}

musicDown = \relative c'' {
  g4 e4.( d8) c4 |
  r2 g'4( f8 e) |
```

```
  d2 \stemDown a
}
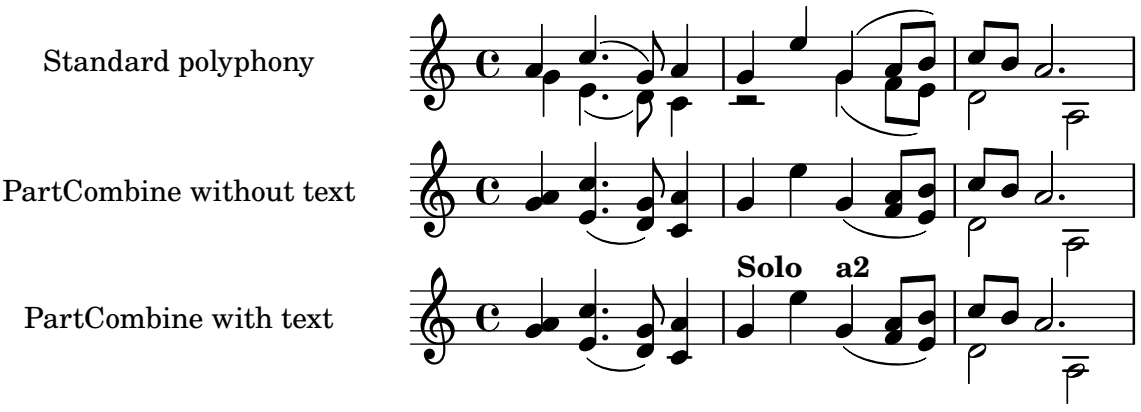
\score {
    <<
    \new Staff \with { instrumentName = "Standard polyphony" }

      << \musicUp \\ \musicDown >>

    \new Staff \with {
      instrumentName = "PartCombine without text"
      printPartCombineTexts = ##f
    }

    \partCombine \musicUp \musicDown

    \new Staff \with { instrumentName = "PartCombine with text" }
      \partCombine \musicUp \musicDown
    >>
  \layout {
    indent = 6.0\cm
    \context {
      \Score
      \override SystemStartBar.collapse-height = #30
    }
  }
}
```



Indications de nuance vraiment entre parenthèses
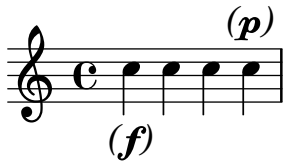
Bien que le moyen le plus simple pour ajouter des parenthèses à une indication de nuance consiste à utiliser un bloc `\markup`, cette pratique a un inconvénient : les objets ainsi créés seront considérés comme des annotations textuelles, non comme des nuances.

Il est néanmoins possible de créer des objets particuliers en partant de code Scheme – comme expliqué dans le manuel de notation – avec la fonction `make-dynamic-script`. Les *markups* ainsi créés seront alors considérés comme étant des indications de nuance, et de ce fait pourront se voir appliquer les effets des commandes `\dynamicUp` et `\dynamicDown`.

```
paren =
#(define-event-function (dyn) (ly:event?)
```

```
   (make-dynamic-script
    #{ \markup \concat {
         \normal-text \italic \fontsize #2 (
         \pad-x #0.2 #(ly:music-property dyn 'text)
         \normal-text \italic \fontsize #2 )
       }
    #}))

\relative c'' {
  c4\paren\f c c \dynamicUp c\paren\p
}
```



Création de repères simultanés
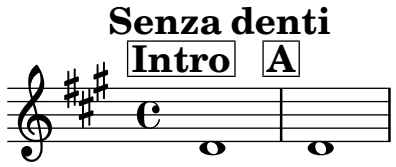
Les indications de repère, à la différence des scripts textuels, ne peuvent s'empiler en un même endroit de la partition : il n'est possible de créer qu'un seul objet `RehearsalMark` à la fois. Le recours à une mesure invisible et à une barre de mesure permet cependant d'ajouter une autre indication de repère, donnant ainsi l'impression d'un double repère sur la même colonne.

Cette méthode permet aussi de positionner un repère en fin de ligne et un autre au début de la ligne suivante.

```
{
  \key a \major
  \set Score.markFormatter = #format-mark-box-letters
  \once \override Score.RehearsalMark.outside-staff-priority = #5000
  \once \override Score.RehearsalMark.self-alignment-X = #LEFT
  \once \override Score.RehearsalMark.break-align-symbols = #'(key-signature)
  \mark \markup { \bold { Senza denti } }

  % the hidden measure and bar line
  % \cadenzaOn turns off automatic calculation of bar numbers
  \cadenzaOn
  \once \override Score.TimeSignature.stencil = ##f
  \time 1/16
  s16 \bar ""
  \cadenzaOff

  \time 4/4
  \once \override Score.RehearsalMark.self-alignment-X = #LEFT
  \mark \markup { \box \bold Intro }
  d'1
  \mark \default
  d'1
}
```



Création d'extensions de texte

Les commandes `\startTextSpan` et `\stopTextSpan` permettent d'ajouter une ligne de prolongation aux indications textuelles, à l'instar des indications de pédale ou d'octaviation. Jouer sur les propriétés de l'objet `TextSpanner` permet d'en modifier le rendu.

```
\paper { ragged-right = ##f }

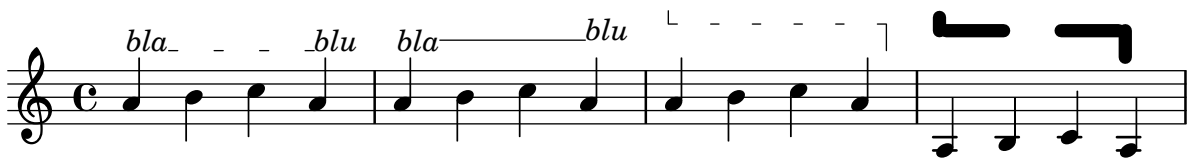
\relative c'' {
  \override TextSpanner.bound-details.left.text = #"bla"
  \override TextSpanner.bound-details.right.text = #"blu"
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.style = #'line
  \once \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.style = #'dashed-line
  \override TextSpanner.bound-details.left.text =
    \markup { \draw-line #'(0 . 1) }
  \override TextSpanner.bound-details.right.text =
    \markup { \draw-line #'(0 . -2) }
  \once \override TextSpanner.bound-details.right.padding = #-2

  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \set Staff.middleCPosition = #-13
  \override TextSpanner.dash-period = #10
  \override TextSpanner.dash-fraction = #0.5
  \override TextSpanner.thickness = #10
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan
}
```



Champs d'entête

Voici la liste de tous les champs d'entête :

```
\header {
```

```
  copyright = "copyright"
  title = "title"
  subtitle = "subtitle"
  composer = "composer"
  arranger = "arranger"
  instrument = "instrument"
  meter = "meter"
  opus = "opus"
  piece = "piece"
  poet = "poet"
  texidoc = "All header fields with special meanings."
  copyright = "public domain"
  enteredby = "jcn"
  source = "urtext"
}

\layout {
  ragged-right = ##f
}

\score {
  \relative c'' { c1 | c | c | c }
}

\score {
   \relative c'' { c1 | c | c | c }
   \header {
     title = "localtitle"
     subtitle = "localsubtitle"
     composer = "localcomposer"
     arranger = "localarranger"
     instrument = "localinstrument"
     metre = "localmetre"
     opus = "localopus"
     piece = "localpiece"
     poet = "localpoet"
     copyright = "localcopyright"
   }
}
```

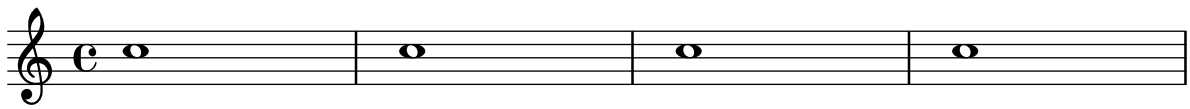
title

subtitle
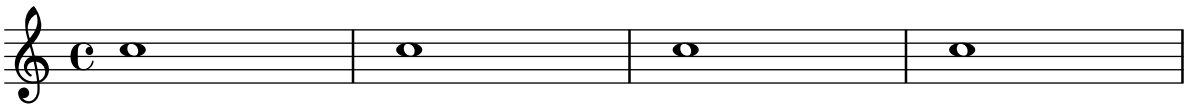
poet **instrument** composer

meter arranger

piece opus



localpiece localopus
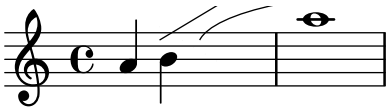


Intégration de PostScript dans un *markup*

Du code PostScript peut directement être intégré dans un bloc `\markup`.

```
% PostScript is a registered trademark of Adobe Systems Inc.

\relative c'' {
  a4-\markup { \postscript "3 4 moveto 5 3 rlineto stroke" }
  -\markup { \postscript "[ 0 1 ] 0 setdash 3 5 moveto 5 -3 rlineto stroke " }

  b4-\markup { \postscript "3 4 moveto 0 0 1 2 8 4 20 3.5 rcurveto stroke" }
  s2
  a'1
}
```



Mise en forme individuelle de syllabes

Le mode *markup* permet d'individualiser la mise en forme de certaines syllabes.

```
mel = \relative c'' { c4 c c c }
lyr = \lyricmode {
  Lyrics \markup { \italic can } \markup { \with-color #red contain }
  \markup { \fontsize #8 \bold Markup! }
}

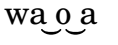
<<
  \new Voice = melody \mel
  \new Lyrics \lyricsto melody \lyr
>>
```



Séparation de syllabes par une liaison

Des mots monosyllabiques peuvent s'attacher par une liaison à l'aide d'un tilde.
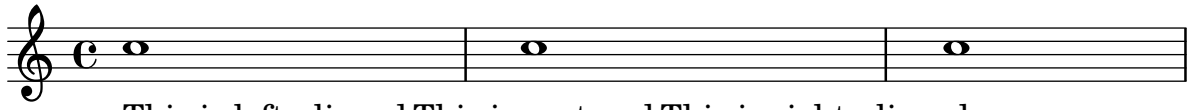
```
\lyrics {
  wa~o~a
}
```



Alignement des syllabes

L'alignement horizontal des paroles peut se gérer à l'aide de la propriété `self-alignment-X` de l'objet `LyricText`. Les valeurs `-1` ou `LEFT` produiront un alignement par la gauche, les valeurs `0` ou `CENTER` un alignement centré, et les valeurs `1` ou `RIGHT` un alignement par la droite.

```
\layout { ragged-right = ##f }
\relative c'' {
  c1
  c1
  c1
}
\addlyrics {
  \once \override LyricText.self-alignment-X = #LEFT
  "This is left-aligned"
  \once \override LyricText.self-alignment-X = #CENTER
  "This is centered"
  \once \override LyricText.self-alignment-X = #1
  "This is right-aligned"
}
```



This is left-aligned This is centered This is right-aligned

Liste de *markups*

Un texte susceptible de se répartir sur plusieurs page se saisit avec la commande `\markuplist`.

```
%% updated/modified by P.P.Schneider on Feb. 2014

#(set-default-paper-size "a6")

#(define-markup-list-command (paragraph layout props args) (markup-list?)
  (interpret-markup-list layout props
   (make-justified-lines-markup-list (cons (make-hspace-markup 2) args))))

% Candide, Voltaire
\markuplist {
  \override-lines #'(baseline-skip . 2.5) {
    \paragraph {
      Il y avait en Westphalie, dans le château de M. le baron de
      Thunder-ten-tronckh, un jeune garçon à qui la nature avait donné
      les mœurs les plus douces.  Sa physionomie annonçait son âme.
      Il avait le jugement assez droit, avec l'esprit le plus
      \concat { simple \hspace #.3 ; }
      c'est, je crois, pour cette raison qu'on le nommait Candide.  Les
      anciens domestiques de la maison soupçonnaient qu'il était fils
      de la sœur de monsieur le baron et d'un bon et honnête
      gentilhomme du voisinage, que cette demoiselle ne voulut jamais
      épouser parce qu'il n'avait pu prouver que soixante et onze
      quartiers, et que le reste de son arbre généalogique avait été
      perdu par l'injure du temps.
```

```
    }
    \vspace #.3
    \paragraph {
      Monsieur le baron était un des plus puissants seigneurs de la
      Westphalie, car son château avait une porte et des fenêtres.  Sa
      grande salle même était ornée d'une tapisserie.  Tous les chiens
      de ses basses-cours composaient une meute dans le
      \concat { besoin \hspace #.3 ; }
      ses palefreniers étaient ses
      \concat { piqueurs \hspace #.3 ; }
      le vicaire du village était
      son grand-aumônier.  Ils l'appelaient tous monseigneur, et ils
      riaient quand il faisait des contes.
    }
  }
}
```

Il y avait en Westphalie, dans le château de M. le baron de Thunder-ten-tronckh, un jeune garçon à qui la nature avait donné les mœurs les plus douces. Sa physionomie annonçait son âme. Il avait le jugement assez droit, avec l'esprit le plus simple ; c'est, je crois, pour cette raison qu'on le nommait Candide. Les anciens domestiques de la maison soupçonnaient qu'il était fils de la sœur de monsieur le baron et d'un bon et honnête gentilhomme du voisinage, que cette demoiselle ne voulut jamais épouser parce qu'il n'avait pu prouver que soixante et onze quartiers, et que le reste de son arbre généalogique avait été perdu par l'injure du temps.

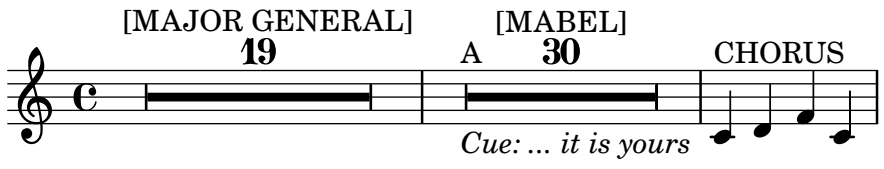
Monsieur le baron était un des plus puissants seigneurs de la Westphalie, car son château avait une porte et des fenêtres. Sa grande salle même était ornée d'une tapisserie. Tous les chiens de ses basses-cours composaient une meute dans le besoin ; ses palefreniers étaient ses piqueurs ; le vicaire du village était son grand-aumônier. Ils l'appelaient tous monseigneur, et ils riaient quand il faisait des contes.

Ajout de texte à un silence multimesure

Lorsque du texte est attaché à un silence multimesure, il sera centré dans la mesure, au-dessus ou en dessous de la portée. Afin d'étirer la mesure dans le cas où ce texte est relativement long, il suffit d'insérer un accord vide auquel on attache le texte en question, avant le silence multimesure.

Le texte attaché à un silence invisible sera aligné sur la gauche de là où serait positionnée la note dans la mesure. Cependant, si la taille de la mesure est déterminée par la longueur du texte, il apparaîtra comme centré.

```
\relative c' {
  \compressMMRests {
    \textLengthOn
    <>^\markup { [MAJOR GENERAL] }
    R1*19
    <>_\markup { \italic { Cue: ... it is yours } }
    <>^\markup { A }
    R1*30^\markup { [MABEL] }
    \textLengthOff
    c4^\markup { CHORUS } d f c
  }
}
```



De l'ubiquité des objets *markup*

Les objets textuels se saisissent soit en tant que simples chaînes de caractères bornées par des guillemets informatiques, soit dans des blocs `\markup` qui, eux, acceptent tout un panel de mise en forme avancée et améliorations graphiques.

En tant que tels, les blocs *markup* peuvent s'utiliser :

- pour tout objet `TextScript` (attaché aux notes par `-`, `^` ou `_`),
- pour tout `RehearsalMark` introduit par le mot-clé `\mark` ou tout autre objet similaire comme `MetronomeMark` introduit par `\tempo`,
- comme bloc de *markup* indépendant, placé au niveau supérieur en dehors de tout bloc `\score`,
- pour toute définition au sein du bloc `\header` (titre, sous-titre, compositeur. . .) ou dans certaines variables définies dans le bloc `\paper` telle que `evenHeaderMarkup` pour les numéros de page.

`\markup` peut aussi servir pour les paroles, les noms d'accord et les nuances. En fait, `\markup` peut servir à personnaliser l'apparence de pratiquement n'importe quel objet, comme l'illustre cet exemple qui recourt à différentes méthodes.

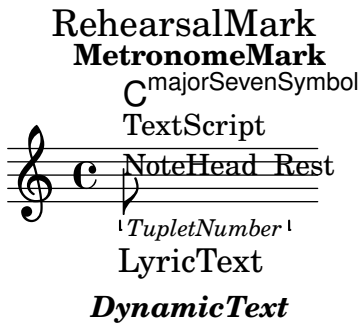
```
%% Thanks to Aaron Hill https://lists.gnu.org/archive/html/lilypond-user/2019-01/msg00437.ht

\paper {
  paper-width = 8\cm paper-height = 8\cm
}
\header {
  title = \markup "Header"
  tagline = \markup "(tagline)"
}
\markup "Top-level markup"
dyn = #(make-dynamic-script #{ \markup \text "DynamicText" #})
\score {
  <<
    \new ChordNames
    \with { majorSevenSymbol = \markup "majorSevenSymbol" }
    \chordmode { c1:maj7 }
    \new Staff {
      \tempo \markup "MetronomeMark"
      \mark \markup "RehearsalMark"
      \once \override TupletNumber.text = \markup "TupletNumber"
      \tuplet 3/2 {
        \once \override NoteHead.stencil = #ly:text-interface::print
        \once \override NoteHead.text = \markup \lower #0.5 "NoteHead"
        c''8^\markup "TextScript"
        \once \override Rest.stencil = #(lambda (grob)
          (grob-interpret-markup grob #{
            \markup  "Rest"
            #}))
        r4
      }
    }
    \new Lyrics \lyricmode { \markup "LyricText" 1 }
    \new Dynamics { s1\dyn }
  >>
```

```
}
```

Header

Top-level markup



Impression du numéro de version

L'insertion du résultat de `lilypond-version` dans des paroles permet d'ajouter à la partition ou au document généré par `lilypond-book` le numéro de la version de LilyPond utilisée. Une autre possibilité consiste à l'ajouter à une chaîne textuelle comme ci-dessous.

```
\score {
  \new Lyrics {
    \override Score.RehearsalMark.self-alignment-X = #LEFT
    \mark #(string-append "Processed with LilyPond version " (lilypond-version))
    s2
  }
}
```

Processed with LilyPond version 2.23.2

Piano et paroles entre les portées

Lorsque la mélodie est doublée au piano, cela ne nécessite pas forcément une portée spécifique. Les paroles peuvent s'insérer entre les deux portées de la partition pour piano.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

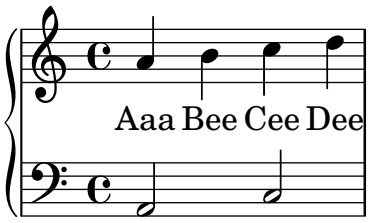
  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}
```

```
text = \lyricmode {
  Aaa Bee Cee Dee
}

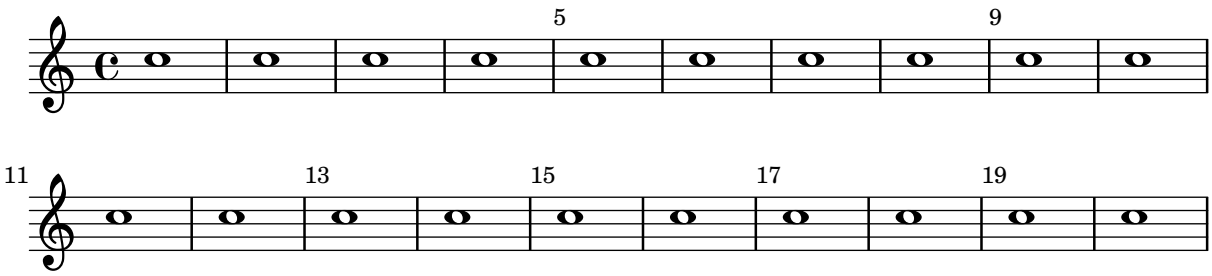
\score {
  \new PianoStaff <<
    \new Staff = upper { \new Voice = "singer" \upper }
    \new Lyrics \lyricsto "singer" \text
    \new Staff = lower { \lower }
  >>
  \layout { }
  \midi { }
}
```



Changement de la fréquence d'impression du numéro de mesure

La fonction de contexte `{set-bar-number-visibility}` permet de modifier la fréquence à laquelle les numéros de mesures s'impriment.

```
\relative c' {
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \context Score \applyContext #(set-bar-number-visibility 4)
  \repeat unfold 10 c'1
  \context Score \applyContext #(set-bar-number-visibility 2)
  \repeat unfold 10 c
}
```
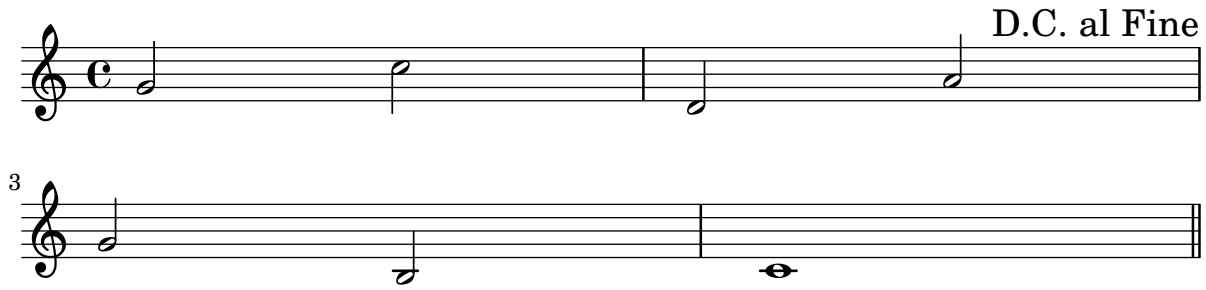


Indication d'un repère en fin de ligne

Les indications de repère peuvent s'imprimer à la fin d'une ligne plutôt qu'en tête de la suivante. L'alignement sur la barre de mesure devra alors s'effectuer par l'extrémité droite de l'indication.

```
\relative c'' {
  g2 c
  d,2 a'
  \once \override Score.RehearsalMark.break-visibility =
    #end-of-line-visible
  \once \override Score.RehearsalMark.self-alignment-X =
```

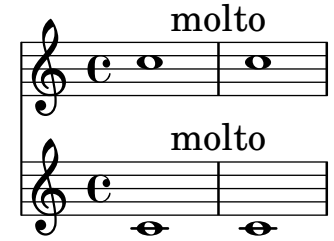
```
    #RIGHT
  \mark "D.C. al Fine"
  \break
  g2 b,
  c1 \bar "||"
}
```



Impression des indications sur toutes les portées d'un système

Bien que ces indications textuelles ne soient habituellement imprimées qu'au niveau de la portée supérieure d'un système, leur affectation peut être répercutée à chacune des portées.

```
\score {
  <<
    \new Staff { c''1 \mark "molto" c'' }
    \new Staff { c'1 \mark "molto" c' }
  >>
  \layout {
    \context {
      \Score
      \remove "Mark_engraver"
      \remove "Staff_collecting_engraver"
    }
    \context {
      \Staff
      \consists "Mark_engraver"
      \consists "Staff_collecting_engraver"
    }
  }
}
```

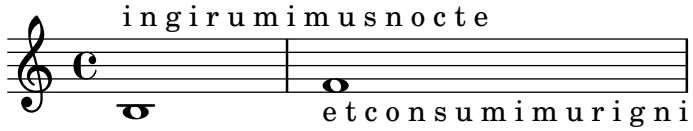


Impression de texte de droite à gauche

Du texte, inclus dans un objet *markup*, peut s'imprimer de droite à gauche, comme illustré ci-dessous.
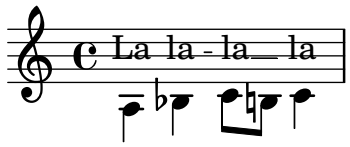
```
{
```

```
  b1^\markup {
    \line { i n g i r u m i m u s n o c t e }
  }
  f'_\markup {
    \override #'(text-direction . -1)
    \line { i n g i r u m i m u s n o c t e }
  }
}
```



Impression des paroles dans la portée

Des paroles peuvent venir s'inscrire dans la portée même. Ces paroles sont décalées par la dérogation `\override LyricText.extra-offset = #'(0 . dimension)` ; des commandes similaires s'occuperont des extenseurs et des tirets. Le décalage optimal ne peut s'obtenir qu'en procédant à tâtons.

```
<<
  \new Staff <<
    \new Voice = "voc" \relative c' { \stemDown a bes c8 b c4 }
  >>
  \new Lyrics \with {
    \override LyricText.extra-offset = #'(0 . 8.6)
    \override LyricExtender.extra-offset = #'(0 . 8.6)
    \override LyricHyphen.extra-offset = #'(0 . 8.6)
  } \lyricsto "voc" { La la -- la __ _ la }
>>
```



Bloc de texte indépendant sur deux colonnes

L'utilisation de la commande `\markup` permet de distribuer un bloc de texte indépendant sur plusieurs colonnes.

```
\markup {
  \fill-line {
    \hspace #1
    \column {
      \line { O sacrum convivium }
      \line { in quo Christus sumitur, }
      \line { recolitur memoria passionis ejus, }
      \line { mens impletur gratia, }
      \line { futurae gloriae nobis pignus datur. }
      \line { Amen. }
    }
```

```
    \hspace #2
    \column \italic {
      \line { O sacred feast }
      \line { in which Christ is received, }
      \line { the memory of His Passion is renewed, }
      \line { the mind is filled with grace, }
      \line { and a pledge of future glory is given to us. }
      \line { Amen. }
    }
    \hspace #1
  }
}
```

O sacrum convivium
in quo Christus sumitur,
recolitur memoria passionis ejus,
mens impletur gratia,
futurae gloriae nobis pignus datur.
Amen.
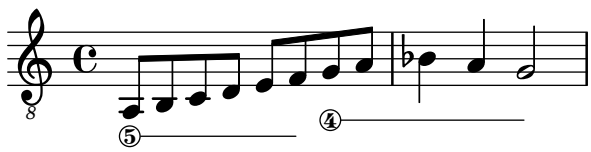
O sacred feast
in which Christ is received,
the memory of His Passion is renewed,
the mind is filled with grace,
and a pledge of future glory is given to us.
Amen.

Ligne de prolongation pour numéro de corde

Voici comment ajouter une ligne de prolongation à une indication de numéro de corde, afin de stipuler que les notes qui suivent doivent être jouées sur la corde en question.

```
stringNumberSpanner =
  #(define-music-function (StringNumber) (string?)
    #{
      \override TextSpanner.style = #'solid
      \override TextSpanner.font-size = #-5
      \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
      \override TextSpanner.bound-details.left.text =
        \markup { \circle \number $StringNumber }
    #})


\relative c {
  \clef "treble_8"
  \stringNumberSpanner "5"
  \textSpannerDown
  a8\startTextSpan
  b c d e f\stopTextSpan
  \stringNumberSpanner "4"
  g\startTextSpan a
  bes4 a g2\stopTextSpan
}
```



Encadrement sur trois côtés

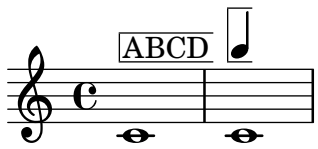
Voici comment construire une commande de *markup* chargée d'agrémenter du texte ou autre annotation, d'une bordure sur trois côtés.

```
% New command to add a three sided box, with sides north, west and south
% Based on the box-stencil command defined in scm/stencil.scm
% Note that ";;" is used to comment a line in Scheme
#(define-public (NWS-box-stencil stencil thickness padding)
   "Add a box around STENCIL, producing a new stencil."
   (let* ((x-ext (interval-widen (ly:stencil-extent stencil X) padding))
          (y-ext (interval-widen (ly:stencil-extent stencil Y) padding))
          (y-rule (make-filled-box-stencil (cons 0 thickness) y-ext))
          (x-rule (make-filled-box-stencil
                   (interval-widen x-ext thickness) (cons 0 thickness))))
     ;; (set! stencil (ly:stencil-combine-at-edge stencil X 1 y-rule padding))
     (set! stencil (ly:stencil-combine-at-edge stencil X LEFT y-rule padding))
     (set! stencil (ly:stencil-combine-at-edge stencil Y UP x-rule 0.0))
     (set! stencil (ly:stencil-combine-at-edge stencil Y DOWN x-rule 0.0))
     stencil))

% The corresponding markup command, based on the \box command defined
% in scm/define-markup-commands.scm
#(define-markup-command (NWS-box layout props arg) (markup?)
   #:properties ((thickness 0.1) (font-size 0) (box-padding 0.2))
   "Draw a box round @var{arg}.  Looks at @code{thickness},
@code{box-padding} and @code{font-size} properties to determine line
thickness and padding around the markup."
   (let ((pad (* (magstep font-size) box-padding))
         (m (interpret-markup layout props arg)))
     (NWS-box-stencil m thickness pad)))

% Test it:

\relative c' {
  c1^\markup { \NWS-box ABCD }
  c1^\markup { \NWS-box \note {4} #1.0 }
}
```



UTF-8

Différents éléments textuels peuvent s'ajouter à la notation, comme des titres ou des paroles, dès lors qu'ils sont encodés en UTF-8, grâce à un moteur de rendu reposant sur Pango. Selon les fontes disponibles devraient apparaître quatre lignes de texte : en bulgare (en cyrillique), en hébreux, en japonais et en portugais.

```
%{
You may have to install additional fonts.
```

```
Red Hat Fedora

  linux-libertine-fonts (Latin, Cyrillic, Hebrew)
  google-noto-serif-jp-fonts (Japanese)

Debian GNU/Linux, Ubuntu

  fonts-linuxlibertine (Latin, Cyrillic, Hebrew)
  fonts-noto-cjk (Japanese)
%}

% 'Linux Libertine' fonts also contain Cyrillic and Hebrew glyphs.
\paper {
  #(define fonts
    (set-global-fonts
     #:roman "Linux Libertine O, Noto Serif CJK JP, Noto Serif JP"
   ))
}

bulgarian = \lyricmode {
  Жълтата дюля беше щастлива, че пухът, който цъфна, замръзна като гьон.
}

hebrew = \lyricmode {
           .
}
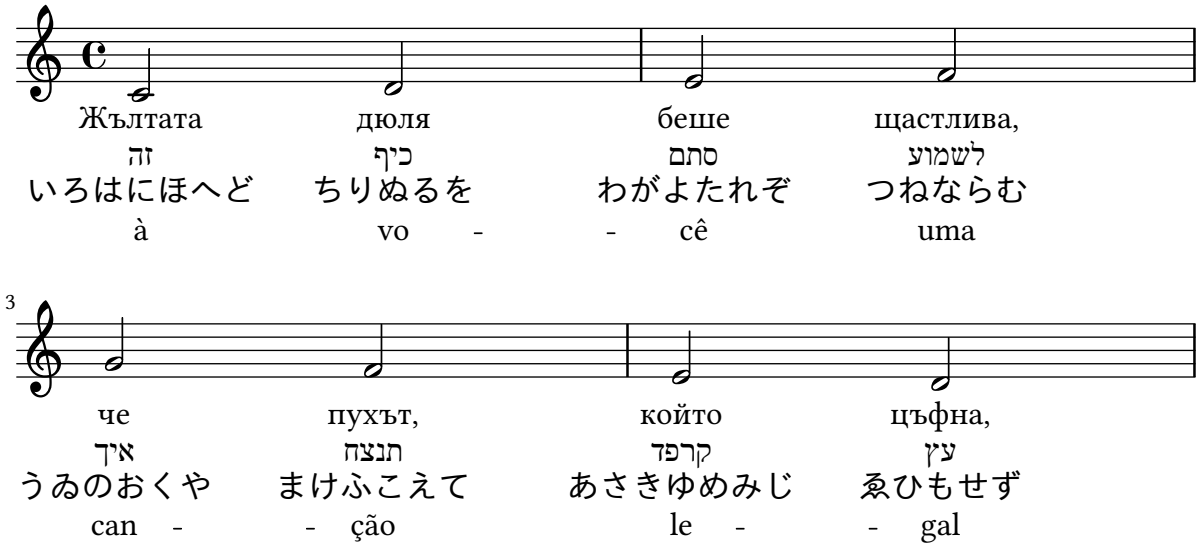
japanese = \lyricmode {




}

% "a nice song for you"
portuguese = \lyricmode {
  à vo -- cê uma can -- ção le -- gal
}

\relative c' {
  c2 d
  e2 f
  g2 f
  e2 d
}
\addlyrics { \bulgarian }
\addlyrics { \hebrew }
\addlyrics { \japanese }
\addlyrics { \portuguese }
```



Ensemble vocal avec alignement des paroles selon le contexte

Ce canevas ressemble beaucoup à celui pour chœur à quatre voix mixtes. La différence réside dans le fait que les paroles sont positionnées en ayant recours à `alignAboveContext` et `alignBelowContext`.

```
global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative c'' {
  c4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}
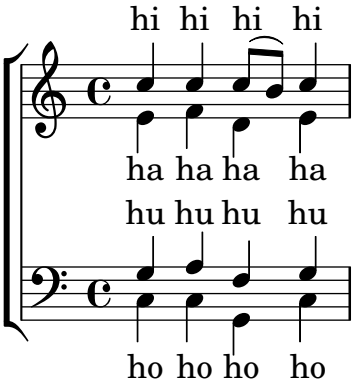
altoMusic = \relative c' {
  e4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

tenorMusic = \relative c' {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative c {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}
```

```
\score {
  \new ChoirStaff <<
    \new Staff = "women" <<
      \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
      \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = #"women" }
      \lyricsto "sopranos" \sopWords
    \new Lyrics \with { alignBelowContext = #"women" }
      \lyricsto "altos" \altoWords
    % we could remove the line about this with the line below, since
    % we want the alto lyrics to be below the alto Voice anyway.
    % \new Lyrics \lyricsto "altos" \altoWords

    \new Staff = "men" <<
      \clef bass
      \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
      \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = #"men" }
      \lyricsto "tenors" \tenorWords
    \new Lyrics \with { alignBelowContext = #"men" }
      \lyricsto "basses" \bassWords
    % again, we could replace the line above this with the line below.
    % \new Lyrics \lyricsto "basses" \bassWords
  >>
}
```



Indication de reprise avec texte grâce à `repeatCommands`
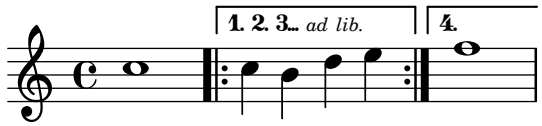
La commande `\repeat volta` permet d'indiquer facilement des reprises. Il est certains cas où l'adjonction d'un texte sous forme de `\markup` nécessite cependant de recourir à la propriété de contexte `repeatCommands`.

Dans la mesure où l'argument de `repeatCommands` est constitué d'une liste, le plus simple est de définir le texte dans une variable qui sera ensuite incorporée dans la liste en respectant la syntaxe Scheme `#(list (list 'volta texteIdentificateur))`. Les commandes de début et de fin de reprise peuvent s'ajouter séparément à la liste des éléments.

```
voltaAdLib = \markup { 1. 2. 3... \text \italic { ad lib. } }

\relative c'' {
```

```
  c1
  \set Score.repeatCommands = #(list (list 'volta voltaAdLib) 'start-repeat)
  c4 b d e
  \set Score.repeatCommands = #'((volta #f) (volta "4.") end-repeat)
  f1
  \set Score.repeatCommands = #'((volta #f))
}
```
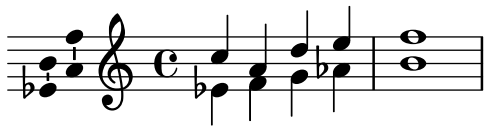


Vocal music

Section “Vocal music” dans *Manuel de notation*

Un ambitus par voix

L’`ambitus` peut être individualisé par voix. Il faut en pareil cas éviter qu’ils se chevauchent.

```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus.X-offset = #2.0
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```



Ajout d’indicateurs à une portée dédoublée après un saut de ligne

Dans cet extrait sont définies les commandes `\splitStaffBarLine`, `convUpStaffBarLine` et `convDownStaffBarLine`. Ces commandes ajoutent une double flèche après la dernière barre de mesure d’une portée, indiquant par là que ses différentes voix disposeront de leur propre portée à la ligne suivante.

```
#(define-markup-command (arrow-at-angle layout props angle-deg length fill)
   (number? number? boolean?)
   (let* (
           (PI-OVER-180 (/ (atan 1 1) 34))
           (degrees->radians (lambda (degrees) (* degrees PI-OVER-180)))
           (angle-rad (degrees->radians angle-deg))
           (target-x (* length (cos angle-rad)))
           (target-y (* length (sin angle-rad))))
     (interpret-markup layout props
       (markup
        #:translate (cons (/ target-x 2) (/ target-y 2))
        #:rotate angle-deg
        #:translate (cons (/ length -2) 0)
        #:concat (#:draw-line (cons length 0)
                   #:arrow-head X RIGHT fill)))))
```

```
splitStaffBarLineMarkup = \markup \with-dimensions #'(0 . 0) #'(0 . 0) {
  \combine
  \arrow-at-angle #45 #(sqrt 8) ##t
  \arrow-at-angle #-45 #(sqrt 8) ##t
}

splitStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob splitStaffBarLineMarkup)
      0))
  \break
}

convDownStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . -.13)\arrow-at-angle #-45 #(sqrt 8) ##t
        }#})
      0))
  \break
}

convUpStaffBarLine = {
  \once \override Staff.BarLine.stencil =
  #(lambda (grob)
     (ly:stencil-combine-at-edge
      (ly:bar-line::print grob)
      X RIGHT
      (grob-interpret-markup grob #{
        \markup\with-dimensions #'(0 . 0) #'(0 . 0) {
          \translate #'(0 . .14)\arrow-at-angle #45 #(sqrt 8) ##t
        }#})
      0))
  \break
}


\paper {
  ragged-right = ##t
  short-indent = 10\mm
}
```

```
separateSopranos = {
  \set Staff.instrumentName = "AI AII"
  \set Staff.shortInstrumentName = "AI AII"
  \splitStaffBarLine
  \change Staff = "up"
}
convSopranos = {
  \convDownStaffBarLine
  \change Staff = "shared"
  \set Staff.instrumentName = "S A"
  \set Staff.shortInstrumentName = "S A"
}

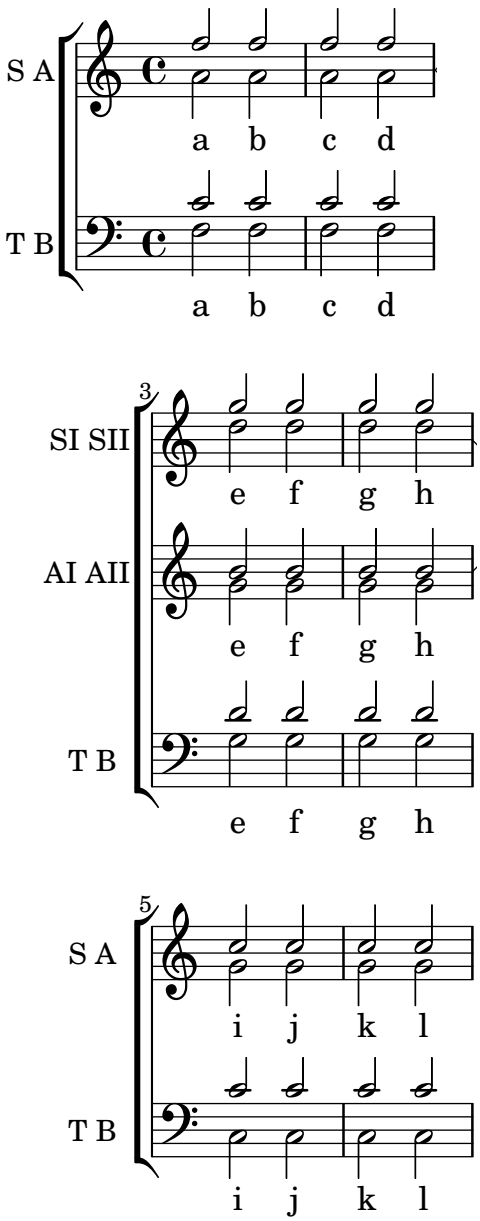
sI = {
  \voiceOne
  \repeat unfold 4 f''2
  \separateSopranos
  \repeat unfold 4 g''2
  \convSopranos
  \repeat unfold 4 c''2
}
sII = {
  s1*2
  \voiceTwo
  \change Staff = "up"
  \repeat unfold 4 d''2
}
aI = {
  \voiceTwo
  \repeat unfold 4 a'2
  \voiceOne
  \repeat unfold 4 b'2
  \convUpStaffBarLine
  \voiceTwo
  \repeat unfold 4 g'2
}
aII = {
  s1*2
  \voiceTwo
  \repeat unfold 4 g'2
}
ten = {
  \voiceOne
  \repeat unfold 4 c'2
  \repeat unfold 4 d'2
  \repeat unfold 4 c'2
}
bas = {
  \voiceTwo
  \repeat unfold 4 f2
  \repeat unfold 4 g2
```

```
  \repeat unfold 4 c2
}

\score {
  <<
    \new ChoirStaff <<
      \new Staff = up \with {
        instrumentName = "SI SII"
        shortInstrumentName = "SI SII"
      } {
        s1*4
      }

      \new Staff = shared \with {
        instrumentName = "S A"
        shortInstrumentName = "S A"
      } <<
        \new Voice = sopI \sI
        \new Voice = sopII \sII
        \new Voice = altI \aI
        \new Voice = altII \aII
      >>
      \new Lyrics \with {
        alignBelowContext = up
      }
      \lyricsto sopII { e f g h }
      \new Lyrics \lyricsto altI { a b c d e f g h i j k l }

      \new Staff = men \with {
        instrumentName = "T B"
        shortInstrumentName = "T B"
      } <<
        \clef F
        \new Voice = ten \ten
        \new Voice = bas \bas
      >>
      \new Lyrics \lyricsto bas { a b c d e f g h i j k l }
    >>
  >>
  \layout {
    \context {
      \Staff \RemoveEmptyStaves
      \override VerticalAxisGroup.remove-first = ##t
    }
  }
}
```



Indication de l'instrument cité dans l'accompagnement d'une partition pour chœur

Lorsque le nombre d'instruments cités dans la réduction pour piano se multiplie, vous pourriez avoir intérêt à créer votre propre fonction pour gérer ces repères. La fonction musicale `\cueWhile` prend quatre arguments : la musique d'où provient la citation, telle que définie par `\addQuote`, le nom qui sera mentionné en regard de cette citation, son positionnement – `UP` ou `DOWN` selon qu'il sera attribué à `\voiceOne` et placé au-dessus ou `\voiceTwo` et placé en dessous – et enfin la musique du piano qui interviendra en parallèle. Le nom de l'instrument en question viendra s'aligner sur la gauche de la citation. Bien que vous puissiez effectuer plusieurs citations, elle ne peuvent être simultanées.

```
cueWhile =
#(define-music-function
   (instrument name dir music)
   (string? string? ly:dir? ly:music?)
   #{
     \cueDuring $instrument #dir {
       \once \override TextScript.self-alignment-X = #RIGHT
       \once \override TextScript.direction = $dir
```

```
       <>-\markup { \tiny #name }
       $music
     }
   #})
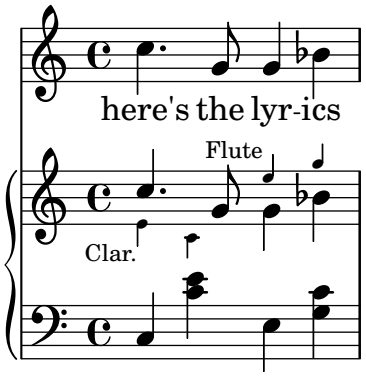
flute = \relative c'' {
  \transposition c'
  s4 s4 e g
}
\addQuote "flute" { \flute }

clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

singer = \relative c'' { c4. g8 g4 bes4 }
words = \lyricmode { here's the lyr -- ics }

pianoRH = \relative c'' {
  \transposition c'
  \cueWhile "clarinet" "Clar." #DOWN { c4. g8 }
  \cueWhile "flute" "Flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  <<
    \new Staff {
      \new Voice = "singer" {
        \singer
      }
    }
    \new Lyrics {
      \lyricsto "singer"
      \words
    }
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH
        }
      }
      \new Staff {
        \clef "bass"
        \pianoLH
      }
    >>
  >>
}
```
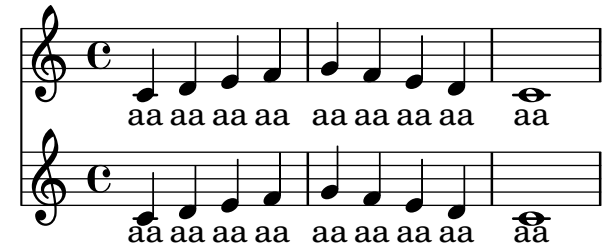


Ajustement de l'espacement vertical des paroles

Cet extrait illustre la manière de réduire l'espace entre la ligne de paroles et la portée.

```
% Default layout:
<<
  \new Staff \new Voice = melody \relative c' {
    c4 d e f
    g4 f e d
    c1
  }
  \new Lyrics \lyricsto melody { aa aa aa aa aa aa aa aa aa }

  \new Staff {
    \new Voice = melody \relative c' {
      c4 d e f
      g4 f e d
      c1
    }
  }
  % Reducing the minimum space below the staff and above the lyrics:
  \new Lyrics \with {
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing =
      #'((basic-distance . 1))
  }
  \lyricsto melody { aa aa aa aa aa aa aa aa aa }
>>
```
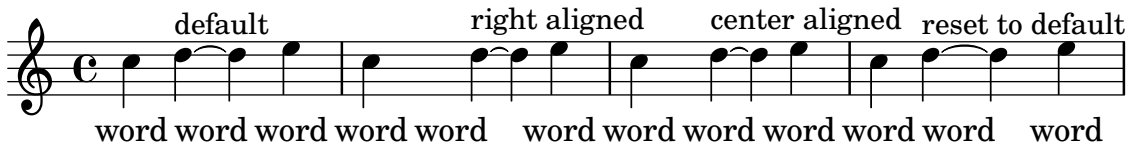


Alignement de la première syllabe d'un mélisme

Par défaut, une syllabe qui entame un mélisme est alignée sur la note par la gauche. Cet alignement peut se modifier à l'aide de la propriété `lyricMelismaAlignment`.

```
\score {
  <<
    \new Staff {
      \relative c''
```
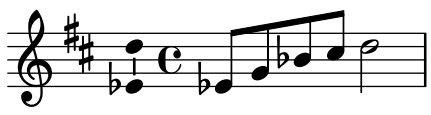
```
      \new Voice = "vocal" {
        c d~^\markup default  d e
        c d~^\markup "right aligned" d e
        c d~^\markup "center aligned" d e
        c d~^\markup "reset to default" d e
      }
    }
    \new Lyrics \lyricsto "vocal" {
      word word word
      \set lyricMelismaAlignment = #RIGHT
      word word word
      \set lyricMelismaAlignment = #CENTER
      word word word
      \unset lyricMelismaAlignment
      word word word
    }
  >>
}
```



Ambitus après l'armure

L'ambitus se place par défaut à gauche de la clef. La fonction `\ambitusAfter` permet cependant de modifier ce positionnement ; sa syntaxe est `\ambitusAfter` *`grob-interface`* – voir Section "Graphical Object Interfaces" dans *Référence des propriétés internes* pour une liste des valeurs de *`grob-interface`* possibles. L'un des cas d'usage est d'insérer l'ambitus entre l'armure et la métrique.

```
\new Staff \with {
  \consists Ambitus_engraver
} \relative {
  \ambitusAfter key-signature
  \key d \major
  es'8 g bes cis d2
}
```
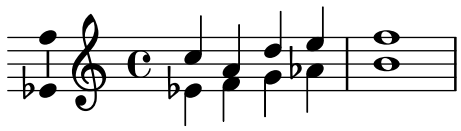


Ambitus sur plusieurs voix

Si plusieurs voix se trouvent sur une même portée, on peut attribuer le graveur `Ambitus_engraver` au contexte `Staff` afin d'obtenir l'ambitus sur toutes les voix cumulées, non d'une seule des voix actives.

```
\new Staff \with {
  \consists "Ambitus_engraver"
  }
<<
```

```
  \new Voice \relative c'' {
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```
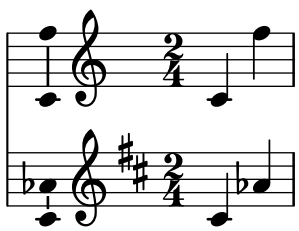


Ambitus

Un *ambitus* indique la tessiture, autrement dit les hauteurs extrêmes d'une voix.

Seules seront affichées les altérations non comprises dans l'armure. Les objets `AmbitusNoteHead` peuvent avoir des lignes supplémentaires.

```
\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

<<
  \new Staff {
    \relative c' {
      \time 2/4
      c4 f'
    }
  }
  \new Staff {
    \relative c' {
      \time  2/4
      \key d \major
      cis4 as'
    }
  }
>>
```



Exemples de notation ancienne – transcription moderne de musique grégorienne
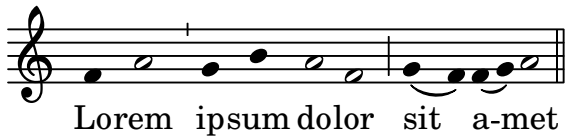
Voici comment vous pourriez transcrire du grégorien. Pour mémoire, il n'y a en grégorien ni découpage en mesure, ni hampe ; seules sont utilisées des têtes de note blanches ou noires, ainsi que des signes spécifiques permettant d'indiquer des silences de différentes durées.

```
\include "gregorian.ly"

chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g) a2 \finalis
}

verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met
}

\score {
  \new Staff <<
    \new Voice = "melody" \chant
    \new Lyrics = "one" \lyricsto melody \verba
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Bar_engraver"
      \hide Stem
    }
    \context {
      \Voice
      \override Stem.length = #0
    }
    \context {
      \Score
      barAlways = ##t
    }
  }
}
```



Psalmodie anglicane

Cet exemple illustre la manière de présenter un cantique tel qu'on le trouve dans l'église anglicane. Vous noterez comment sont ajoutés les couplets indépendamment de la musique. Dans le but de vous montrer plusieurs styles, comparez le code des deux couplets.

```
SopranoMusic = \relative g' {
```

```
  g1 | c2 b | a1 | \bar "||"
  a1 | d2 c | c b | c1 | \bar "||"
}

AltoMusic = \relative c' {
  e1 | g2 g | f1 |
  f1 | f2 e | d d | e1 |
}

TenorMusic = \relative a {
  c1 | c2 c | c1 |
  d1 | g,2 g | g g | g1 |
}

BassMusic =  \relative c {
  c1 | e2 e | f1 |
  d1 | b2 c | g' g | c,1 |
}

global = {
  \time 2/2
}

dot = \markup {
  \raise #0.7 \musicglyph "dots.dot"
}

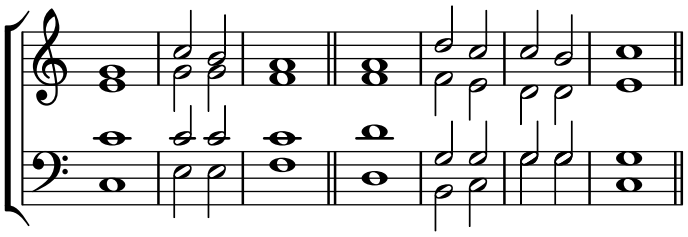
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}

% Use markup to center the chant on the page
\markup {
  \fill-line {
    \score {  % centered
      <<
        \new ChoirStaff <<
          \new Staff <<
            \global
            \clef "treble"
            \new Voice = "Soprano" <<
              \voiceOne
              \SopranoMusic
            >>
            \new Voice = "Alto" <<
              \voiceTwo
              \AltoMusic
            >>
          >>
          \new Staff <<
            \clef "bass"
            \global
```

```
            \new Voice = "Tenor" <<
              \voiceOne
              \TenorMusic
            >>
            \new Voice = "Bass" <<
              \voiceTwo
              \BassMusic
            >>
          >>
        >>
      >>
      \layout {
        \context {
          \Score
          \override SpacingSpanner.base-shortest-duration = #(ly:make-moment 1/2)
        }
        \context {
          \Staff
          \remove "Time_signature_engraver"
        }
      }
    }  % End score
  }
}  % End markup

\markup {
  \fill-line {
    \column {
      \left-align {
        \null \null \null
        \line {
          \fontsize #5 O
          \fontsize #3 come
          let us \bold sing | unto \dot the | Lord : let
        }
        \line {
          us heartily
          \concat { re \bold joice }
          in the | strength of | our
        }
        \line {
          sal | vation.
        }
        \null
        \line {
          \hspace #2.5 8. Today if ye will hear his voice *
        }
        \line {
          \concat { \bold hard en }
          \tick not your \tick hearts : as in the pro-
        }
        \line {
```

```
          vocation * and as in the \bold day of tempt- \tick
        }
        \line {
          -ation \tick in the \tick wilderness.
        }
      }
    }
  }
}
```



O come let us **sing** | unto • the | Lord : let us heartily re**joice** in the | strength of | our sal | vation.

8. Today if ye will hear his voice * **hard**en ' not your ' hearts : as in the provocation * and as in the **day** of tempt- ' -ation ' in the ' wilderness.

Agencement de paroles séparées sur une seule ligne

On peut vouloir positionner les paroles de différents intervenants sur une même ligne, notamment pour un dialogue par bribes. Cet extrait montre comment y parvenir avec l'instruction `\override VerticalAxisGroup.nonstaff-nonstaff-spacing.minimum-distance = ##f`.

```
\layout {
  \context {
    \Lyrics
    \override VerticalAxisGroup.nonstaff-nonstaff-spacing.minimum-distance = ##f
  }
}

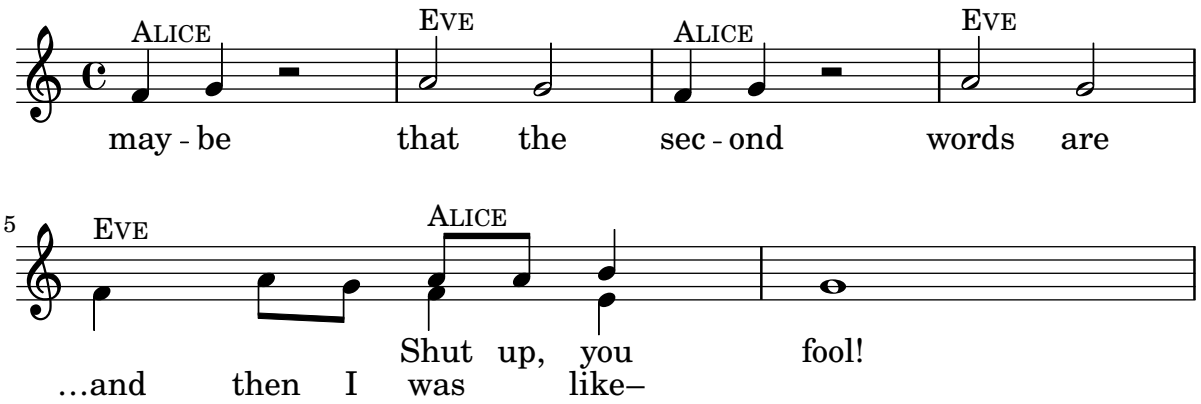
aliceSings = \markup { \smallCaps "Alice" }
eveSings = \markup { \smallCaps "Eve" }

<<
  \new Staff <<
    \new Voice = "alice" {
      f'4^\aliceSings g' r2 |
      s1 |
      f'4^\aliceSings g' r2 |
```

```
      s1 | \break
      % ...

      \voiceOne
      s2 a'8^\aliceSings a' b'4 |
      \oneVoice
      g'1
    }
    \new Voice = "eve" {
      s1 |
      a'2^\eveSings g' |
      s1 |
      a'2^\eveSings g'
      % ...

      \voiceTwo
      f'4^\eveSings a'8 g' f'4 e' |
      \oneVoice
      s1
    }
  >>
  \new Lyrics \lyricsto "alice" {
    may -- be
    sec -- ond
    % ...
    Shut up, you fool!
  }
  \new Lyrics \lyricsto "eve" {
    that the
    words are
    % ...
    ...and then I was like--
  }
>>
```



Changement de fontes des couplets

Chaque couplet peut disposer de sa propre fonte, y compris son numéro.

```
%{
You may have to install additional fonts.
```
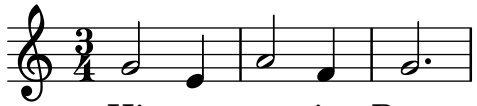
```
Red Hat Fedora

  dejavu-fonts-all

Debian GNU/Linux, Ubuntu

  fonts-dejavu-core
  fonts-dejavu-extra
%}

\relative c'' {
  \time 3/4
  g2 e4
  a2 f4
  g2.
}
\addlyrics {
  \set stanza = #"1. "
  Hi, my name is Bert.
}
\addlyrics {
  \override StanzaNumber.font-name = #"DejaVu Sans"
  \set stanza = #"2. "
  \override LyricText.font-family = #'typewriter
  Oh, ché -- ri, je t'aime
}
```



1. Hi, my name is Bert.
2. Oh, ché-ri, je t'aime

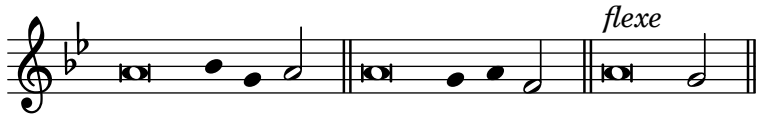
Notation pour psalmodie

Ce style de notation permet d'indiquer la mélodie d'une psalmodie lorsque les strophes sont de longueur inégale.

```
stemOff = \hide Staff.Stem
stemOn  = \undo \stemOff

\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \bar "||"
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \bar "||"
    \stemOff a'\breve^\markup { \italic flexe }
    \stemOn g'2 \bar "||"
```

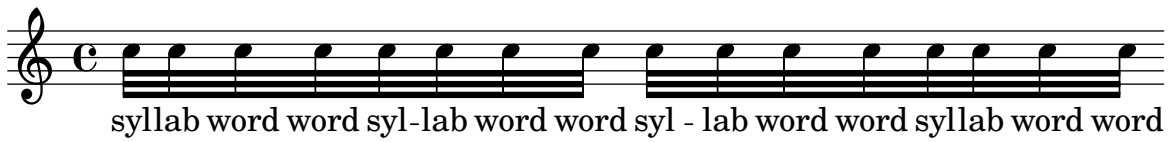
```
  }
}
```



Impression forcée de tirets entre syllabes

LilyPond n'imprimera de tiret entre deux syllabes que s'il juge qu'il y a suffisamment d'espace. Ce comportement peut être modifié grâce à la propriété `minimum-distance` de `LyricHyphen`.

```
\relative c'' {
  c32 c c c
  c32 c c c
  c32 c c c
  c32 c c c
}
\addlyrics {
  syl -- lab word word
  \override LyricHyphen.minimum-distance = #1.0
  syl -- lab word word
  \override LyricHyphen.minimum-distance = #2.0
  syl -- lab word word
  \revert LyricHyphen.minimum-distance
  syl -- lab word word
}
```
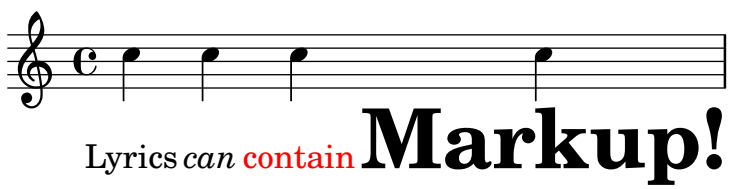


Mise en forme individuelle de syllabes

Le mode *markup* permet d'individualiser la mise en forme de certaines syllabes.

```
mel = \relative c'' { c4 c c c }
lyr = \lyricmode {
  Lyrics \markup { \italic can } \markup { \with-color #red contain }
  \markup { \fontsize #8 \bold Markup! }
}

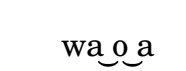
<<
  \new Voice = melody \mel
  \new Lyrics \lyricsto melody \lyr
>>
```



Séparation de syllables par une liaison

Des mots monosyllabiques peuvent s'attacher par une liaison à l'aide d'un tilde.

```
\lyrics {
  wa~o~a
}
```



Modèle pour cantique

Le code ci-dessous illustre la manière d'agencer un cantique liturgique dans lequel chaque ligne débute et se termine par une mesure incomplète. Vous noterez par ailleurs l'affichage des paroles indépendamment de la musique.

```
Timeline = {
  \time 4/4
  \tempo 4=96
  \partial 2
  s2 | s1 | s2 \breathe s2 | s1 | s2 \bar "||" \break
  s2 | s1 | s2 \breathe s2 | s1 | s2 \bar "||"
}

SopranoMusic = \relative g' {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

AltoMusic = \relative c' {
  d4 d | d d d d | d d d d | d d d d | d2
  d4 d | d d d d | d d d d | d d d d | d2
}

TenorMusic = \relative a {
  b4 b | b b b b | b b b b | b b b b | b2
  b4 b | b b b b | b b b b | b b b b | b2
}

BassMusic =  \relative g {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}
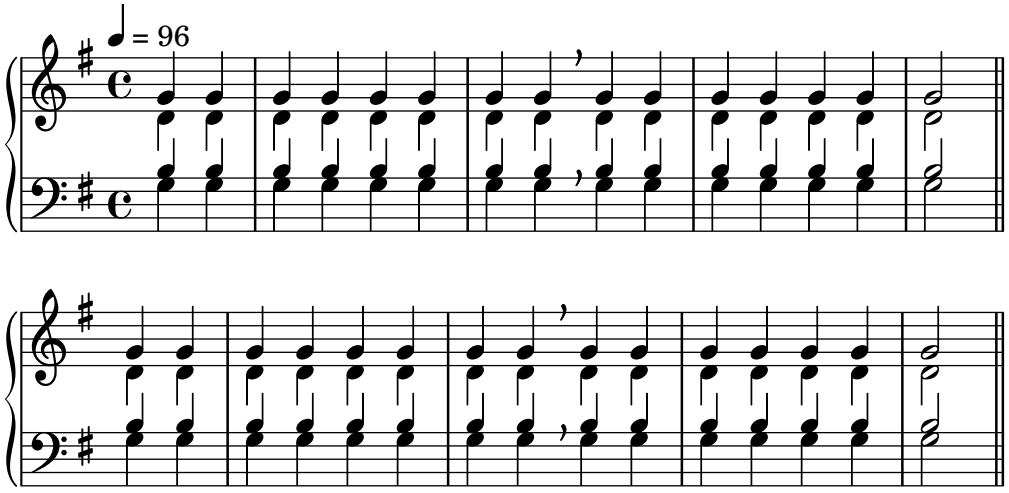
global = {
 \key g \major
}

\score {  % Start score
  <<
    \new PianoStaff <<  % Start pianostaff
      \new Staff <<  % Start Staff = RH
        \global
        \clef "treble"
```

```
        \new Voice = "Soprano" <<  % Start Voice = "Soprano"
          \Timeline
          \voiceOne
          \SopranoMusic
        >>  % End Voice = "Soprano"
        \new Voice = "Alto" <<  % Start Voice = "Alto"
          \Timeline
          \voiceTwo
          \AltoMusic
        >>  % End Voice = "Alto"
      >>  % End Staff = RH
      \new Staff <<  % Start Staff = LH
        \global
        \clef "bass"
        \new Voice = "Tenor" <<  % Start Voice = "Tenor"
          \Timeline
          \voiceOne
          \TenorMusic
        >>  % End Voice = "Tenor"
        \new Voice = "Bass" <<  % Start Voice = "Bass"
          \Timeline
          \voiceTwo
          \BassMusic
        >>  % End Voice = "Bass"
      >>  % End Staff = LH
    >>  % End pianostaff
  >>
}  % End score

\markup {
  \fill-line {
    ""
    {
      \column {
        \left-align {
          "This is line one of the first verse"
          "This is line two of the same"
          "And here's line three of the first verse"
          "And the last line of the same"
        }
      }
    }
    ""
  }
}

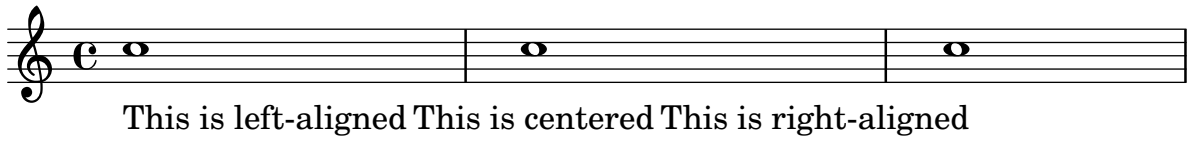
\paper {  % Start paper block
  indent = 0     % don't indent first system
  line-width = 130   % shorten line length to suit music
}  % End paper block
```



Alignement des syllabes

L'alignement horizontal des paroles peut se gérer à l'aide de la propriété `self-alignment-X` de l'objet `LyricText`. Les valeurs `-1` ou `LEFT` produiront un alignement par la gauche, les valeurs `0` ou `CENTER` un alignement centré, et les valeurs `1` ou `RIGHT` un alignement par la droite.

```
\layout { ragged-right = ##f }
\relative c'' {
  c1
  c1
  c1
}
\addlyrics {
  \once \override LyricText.self-alignment-X = #LEFT
  "This is left-aligned"
  \once \override LyricText.self-alignment-X = #CENTER
  "This is centered"
  \once \override LyricText.self-alignment-X = #1
  "This is right-aligned"
}
```



Marking notes of spoken parts with a cross on the stem (Sprechstimme)

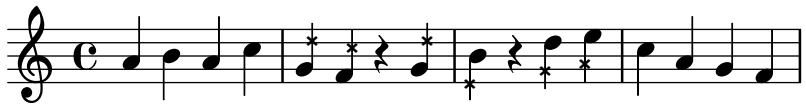
This example shows how to put crosses on stems. Mark the beginning of a spoken section with the `\speakOn` keyword, and end it with the `\speakOff` keyword.

```
speakOn = {
  \override Stem.stencil =
    #(lambda (grob)
       (let* ((x-parent (ly:grob-parent grob X))
```

```
             (is-rest? (ly:grob? (ly:grob-object x-parent 'rest))))
        (if is-rest?
            empty-stencil
            (ly:stencil-combine-at-edge
             (ly:stem::print grob)
             Y
             (- (ly:grob-property grob 'direction))
             (grob-interpret-markup grob
                                    (markup #:center-align #:fontsize -4
                                            #:musicglyph "noteheads.s2cross"))
             -2.3))))
}

speakOff = {
  \revert Stem.stencil
  \revert Flag.stencil
}

\score {
  \new Staff {
    \relative c'' {
      a4 b a c
      \speakOn
      g4 f r g
      b4 r d e
      \speakOff
      c4 a g f
    }
  }
}
```



Espacement des paroles selon les pratiques de la version 2.12

Le moteur d'espacement vertical des paroles a évolué avec la version 2.14. Celles-ci peuvent donc se retrouver positionnées différemment.

Le moteur adoptera les usages de la version 2.12 une fois que vous aurez réglé certaines propriétés des contextes `Lyric` et `Staff`.

```
global = {
  \key d \major
  \time 3/4
}

sopMusic = \relative c' {
  % VERSE ONE
  fis4 fis fis | \break
  fis4. e8 e4
}
```

```
altoMusic = \relative c' {
  % VERSE ONE
  d4 d d |
  d4. b8 b4 |
}

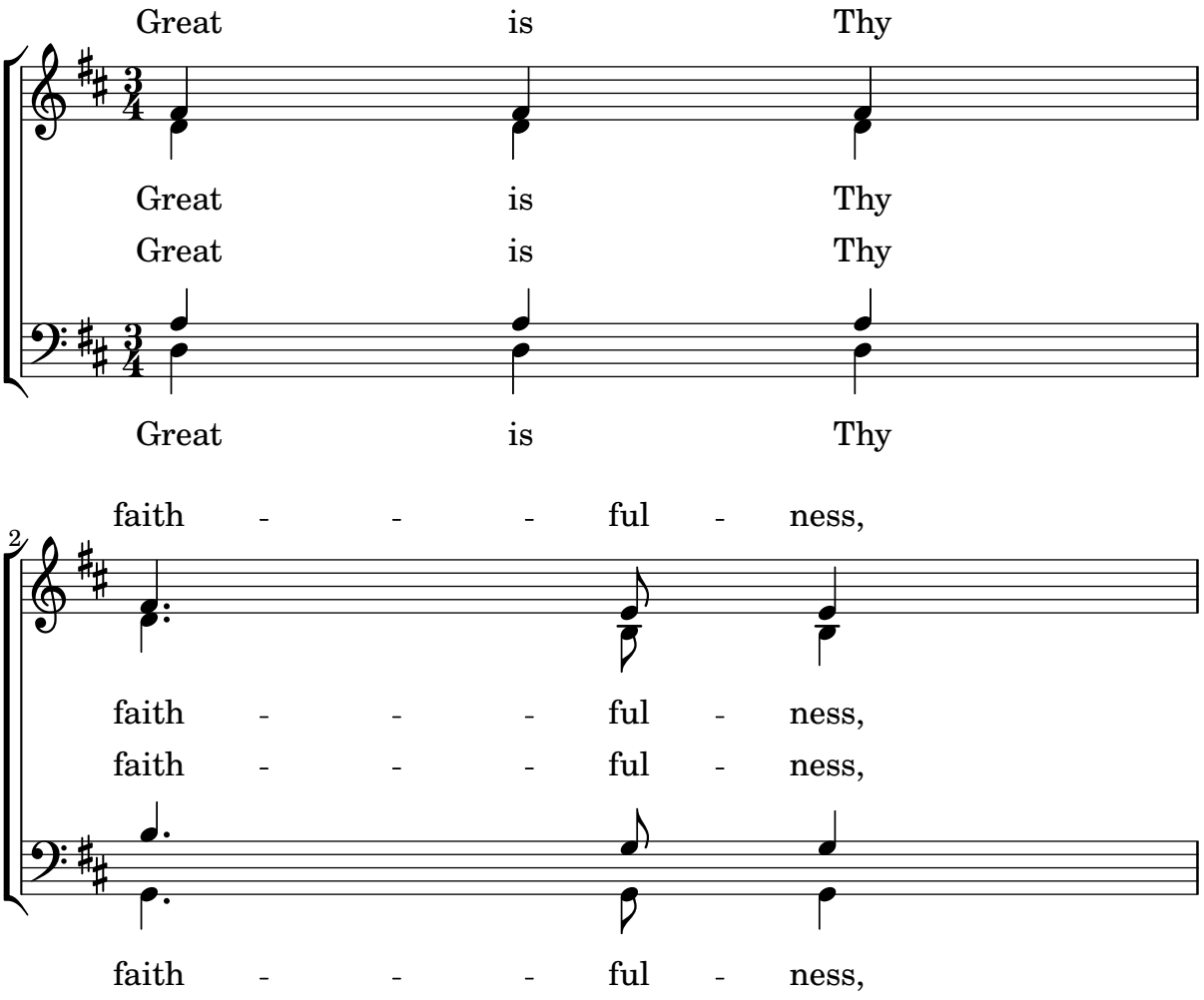
tenorMusic = \relative c' {
  a4 a a |
  b4. g8 g4 |
}

bassMusic = \relative c {
  d4 d d |
  g,4. g8 g4 |
}

words = \lyricmode {
  Great is Thy faith -- ful -- ness,
}

\score {
  \new ChoirStaff <<
    \new Lyrics = sopranos
    \new Staff = women <<
      \new Voice = "sopranos" {
        \voiceOne
        \global \sopMusic
      }
      \new Voice = "altos" {
        \voiceTwo
        \global \altoMusic
      }
    >>
    \new Lyrics = "altos"
    \new Lyrics = "tenors"
    \new Staff = men <<
      \clef bass
      \new Voice = "tenors" {
        \voiceOne
        \global \tenorMusic
      }
      \new Voice = "basses" {
        \voiceTwo  \global \bassMusic
      }
    >>
    \new Lyrics = basses
    \context Lyrics = sopranos \lyricsto sopranos \words
    \context Lyrics = altos \lyricsto altos \words
    \context Lyrics = tenors \lyricsto tenors \words
    \context Lyrics = basses \lyricsto basses \words
  >>
  \layout {
```

```
    \context {
      \Lyrics
      \override VerticalAxisGroup.staff-affinity = ##f
      \override VerticalAxisGroup.staff-staff-spacing =
        #'((basic-distance . 0)
           (minimum-distance . 2)
           (padding . 2))
    }
    \context {
      \Staff
      \override VerticalAxisGroup.staff-staff-spacing =
        #'((basic-distance . 0)
           (minimum-distance . 2)
           (padding . 2))
    }
  }
}
```



Orchestre, chœur et piano

Ce canevas illustre l'utilisation de contextes `StaffGroup` pour regrouper les instruments selon leur famille, imbriqués dans un `GrandStaff`, ainsi que le recours à la fonction `\transpose` pour les instruments transpositeurs. Dans tous les identificateurs, la musique est stockée en ut. Les

notes peuvent tout aussi bien être saisies en ut ou dans la tonalité particulière de l'instrument avant d'être transposées puis affectées à une variable.

```
#(set-global-staff-size 17)
\paper {
  indent = 3.0\cm  % add space for instrumentName
  short-indent = 1.5\cm  % add less space for shortInstrumentName
}

fluteMusic = \relative c' { \key g \major g'1 b }

% Pitches as written on a manuscript for Clarinet in A
% are transposed to concert pitch.

clarinetMusic = \transpose c' a
  \relative c'' { \key bes \major bes1 d }

trumpetMusic = \relative c { \key g \major g''1 b }

% Key signature is often omitted for horns

hornMusic = \transpose c' f
  \relative c { d'1 fis }

percussionMusic = \relative c { \key g \major g1 b }

sopranoMusic = \relative c'' { \key g \major g'1 b }

sopranoLyrics = \lyricmode { Lyr -- ics }

altoIMusic = \relative c' { \key g \major g'1 b }

altoIIMusic = \relative c' { \key g \major g'1 b }

altoILyrics =  \sopranoLyrics

altoIILyrics = \lyricmode { Ah -- ah }

tenorMusic = \relative c' { \clef "treble_8" \key g \major g1 b }

tenorLyrics = \sopranoLyrics

pianoRHMusic = \relative c { \key g \major g''1 b }

pianoLHMusic = \relative c { \clef bass \key g \major g1 b }

violinIMusic = \relative c' { \key g \major g'1 b }

violinIIMusic = \relative c' { \key g \major g'1 b }

violaMusic = \relative c { \clef alto \key g \major g'1 b }
```

```
celloMusic = \relative c { \clef bass \key g \major g1 b }

bassMusic = \relative c { \clef "bass_8" \key g \major g,1 b }

\score {
  <<
    \new StaffGroup = "StaffGroup_woodwinds" <<
      \new Staff = "Staff_flute" \with { instrumentName = "Flute" }
      \fluteMusic

      \new Staff = "Staff_clarinet" \with {
        instrumentName = \markup { \concat { "Clarinet in B" \flat } }
      }

      % Declare that written Middle C in the music
      % to follow sounds a concert B flat, for
      % output using sounded pitches such as MIDI.
      %\transposition bes

      % Print music for a B-flat clarinet
      \transpose bes c' \clarinetMusic
    >>

    \new StaffGroup = "StaffGroup_brass" <<
      \new Staff = "Staff_hornI" \with { instrumentName = "Horn in F" }
       % \transposition f
        \transpose f c' \hornMusic

      \new Staff = "Staff_trumpet" \with { instrumentName = "Trumpet in  C" }
      \trumpetMusic

    >>
    \new RhythmicStaff = "RhythmicStaff_percussion"
    \with { instrumentName = "Percussion" }
    <<
      \percussionMusic
    >>
    \new PianoStaff \with { instrumentName = "Piano" }
    <<
      \new Staff { \pianoRHMusic }
      \new Staff { \pianoLHMusic }
    >>
    \new ChoirStaff = "ChoirStaff_choir" <<
      \new Staff = "Staff_soprano" \with { instrumentName = "Soprano" }
      \new Voice = "soprano"
      \sopranoMusic

      \new Lyrics \lyricsto "soprano" { \sopranoLyrics }
      \new GrandStaff = "GrandStaff_altos"
      \with { \accepts Lyrics } <<
        \new Staff = "Staff_altoI"  \with { instrumentName = "Alto I" }
        \new Voice = "altoI"
```

```
        \altoIMusic

        \new Lyrics \lyricsto "altoI" { \altoILyrics }
        \new Staff = "Staff_altoII" \with { instrumentName = "Alto II" }
        \new Voice = "altoII"
        \altoIIMusic

        \new Lyrics \lyricsto "altoII" { \altoIILyrics }
      >>

      \new Staff = "Staff_tenor" \with { instrumentName = "Tenor" }
        \new Voice = "tenor"
        \tenorMusic

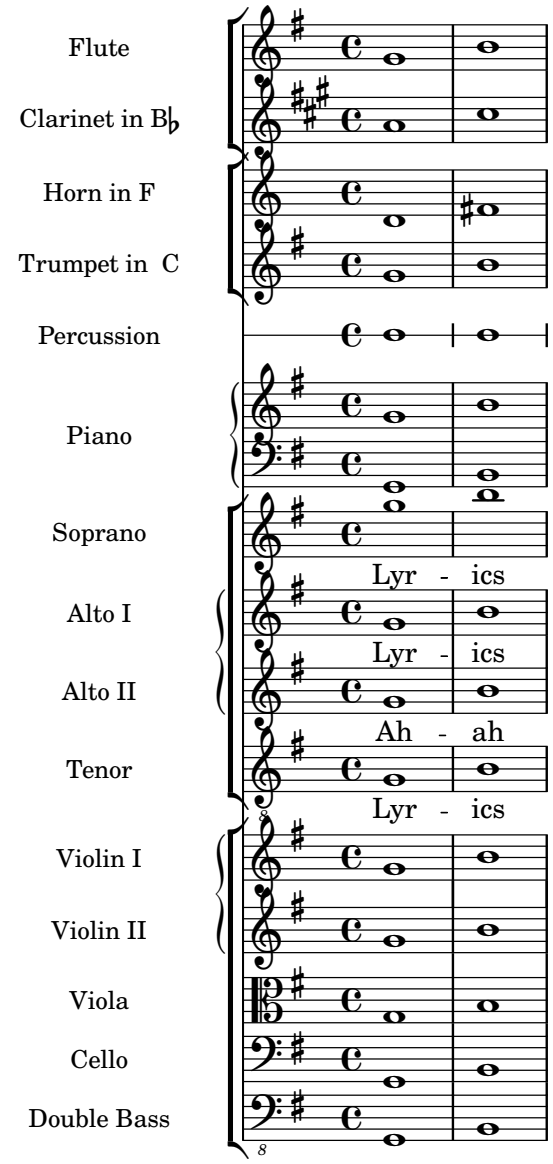
      \new Lyrics \lyricsto "tenor" { \tenorLyrics }
    >>
    \new StaffGroup = "StaffGroup_strings" <<
      \new GrandStaff = "GrandStaff_violins" <<
        \new Staff = "Staff_violinI" \with { instrumentName = "Violin I" }
        \violinIMusic

        \new Staff = "Staff_violinII" \with { instrumentName = "Violin II" }
        \violinIIMusic
      >>

      \new Staff = "Staff_viola" \with { instrumentName = "Viola" }
      \violaMusic

      \new Staff = "Staff_cello" \with { instrumentName = "Cello" }
      \celloMusic

      \new Staff = "Staff_bass" \with { instrumentName = "Double Bass" }
      \bassMusic
    >>
  >>
  \layout { }
}
```



Piano, mélodie et paroles

Il s'agit du format classique pour le chant : une portée pour la mélodie et les paroles au-dessus de l'accompagnement au piano.

```
melody = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

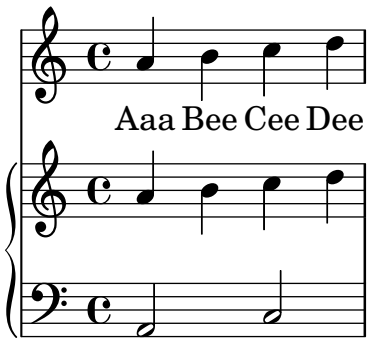
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4
```

```
  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

\score {
  <<
    \new Voice = "mel" { \autoBeamOff \melody }
    \new Lyrics \lyricsto mel \text
    \new PianoStaff <<
      \new Staff = "upper" \upper
      \new Staff = "lower" \lower
    >>
  >>
  \layout {
    \context { \Staff \RemoveEmptyStaves }
  }
  \midi { }
}
```
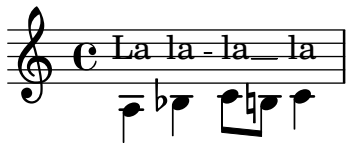


Impression des paroles dans la portée

Des paroles peuvent venir s'inscrire dans la portée même. Ces paroles sont décalées par la dérogation `\override LyricText.extra-offset = #'(0 . dimension)` ; des commandes similaires s'occuperont des extenseurs et des tirets. Le décalage optimal ne peut s'obtenir qu'en procédant à tâtons.

```
<<
  \new Staff <<
    \new Voice = "voc" \relative c' { \stemDown a bes c8 b c4 }
  >>
  \new Lyrics \with {
    \override LyricText.extra-offset = #'(0 . 8.6)
    \override LyricExtender.extra-offset = #'(0 . 8.6)
    \override LyricHyphen.extra-offset = #'(0 . 8.6)
```

```
  } \lyricsto "voc" { La la -- la __ _ la }
>>
```
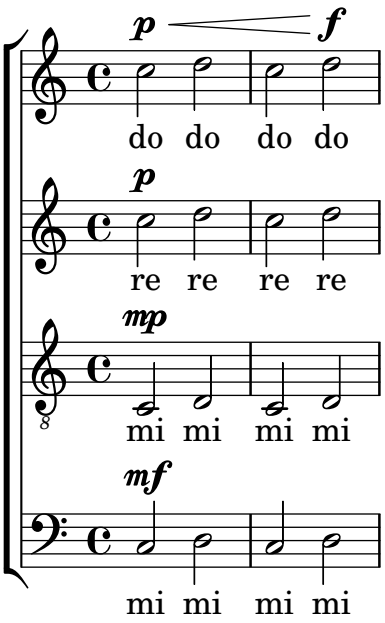


Modèle pour chœur SATB, sur quatre portées

Modèle pour chœur à quatre voix mixtes, chaque pupitre ayant sa propre portée.

```
global = {
  \key c \major
  \time 4/4
  \dynamicUp
}
sopranonotes = \relative c'' {
  c2 \p \< d c d \f
}
sopranowords = \lyricmode { do do do do }
altonotes = \relative c'' {
  c2\p d c d
}
altowords = \lyricmode { re re re re }
tenornotes = {
  \clef "G_8"
  c2\mp d c d
}
tenorwords = \lyricmode { mi mi mi mi }
bassnotes = {
  \clef bass
  c2\mf d c d
}
basswords = \lyricmode { mi mi mi mi }

\score {
  \new ChoirStaff <<
    \new Staff <<
      \new Voice = "soprano" <<
        \global
        \sopranonotes
      >>
      \new Lyrics \lyricsto "soprano" \sopranowords
    >>
    \new Staff <<
      \new Voice = "alto" <<
        \global
        \altonotes
      >>
      \new Lyrics \lyricsto "alto" \altowords
    >>
```

```
    \new Staff <<
      \new Voice = "tenor" <<
        \global
        \tenornotes
      >>
      \new Lyrics \lyricsto "tenor" \tenorwords
    >>
    \new Staff <<
      \new Voice = "bass" <<
        \global
        \bassnotes
      >>
      \new Lyrics \lyricsto "bass" \basswords
    >>
  >>
}
```



Paroles, musique et accords

Ce canevas comporte tous les éléments d'une chanson : la mélodie, les paroles, les accords.

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

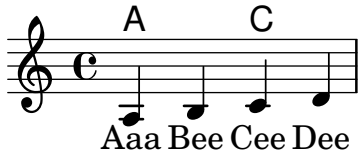
  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

harmonies = \chordmode {
```

```
  a2 c
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Voice = "one" { \autoBeamOff \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}
```



Paroles, musique, accords et diagrammes de fret

Ce canevas comporte, en plus de la mélodie, des paroles et des accords, les diagrammes de fret.

```
verseI = \lyricmode {
  \set stanza = #"1."
  This is the first verse
}

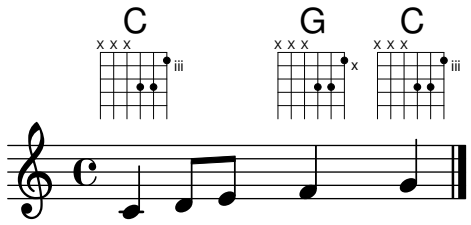
verseII = \lyricmode {
  \set stanza = #"2."
  This is the second verse.
}

theChords = \chordmode {
  % insert chords for chordnames and fretboards here
  c2 g4 c
}

staffMelody = \relative c' {
   \key c \major
   \clef treble
   % Type notes for melody here
   c4 d8 e f4 g
   \bar "|."
}

\score {
  <<
    \context ChordNames { \theChords }
    \context FretBoards { \theChords }
```

```
    \new Staff {
      \context Voice = "voiceMelody" { \staffMelody }
    }
    \new Lyrics = "lyricsI" {
      \lyricsto "voiceMelody" \verseI
    }
    \new Lyrics = "lyricsII" {
      \lyricsto "voiceMelody" \verseII
    }
  >>
  \layout { }
  \midi { }
}
```



Portée unique et paroles

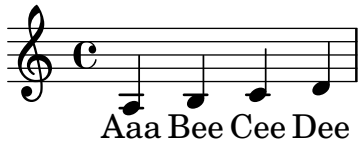
Ce canevas comporte une simple ligne mélodique agrémentée de paroles. Recopiez-le, ajoutez-y d'autres notes et paroles. Les ligatures automatiques sont ici désactivées comme il est d'usage en matière de musique vocale. Pour activer la fonction de ligature automatique, modifiez ou commentez la ligne en question.

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}
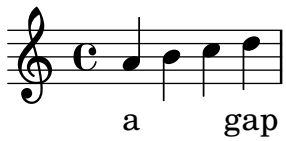
\score{
  <<
    \new Voice = "one" {
      \autoBeamOff
      \melody
    }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}
```



Sauts de notes en mode paroles (2)

Sachant qu'un `s` au sein d'un bloc `\lyricmode` est interprété non comme de l'espace mais comme une syllabe, l'utilisation de paires de guillemets informatiques (`""`) ou un caractère souligné simple (`_`) permet de « sauter » une note.

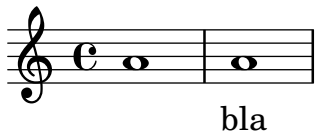
```
<<
  \relative c'' { a4 b c d }
  \new Lyrics \lyricmode { a4 "" _ gap }
>>
```



Sauts de notes en mode paroles

La syntaxe `s` qui permet de « faire un saut dans le temps » n'est disponible qu'en mode notes et en mode accords. Dans les autres situations, comme en mode paroles par exemple, la commande `\skip` produit les mêmes effets.

```
<<
  \relative c'' { a1 | a }
  \new Lyrics \lyricmode { \skip 1 bla1 }
>>
```



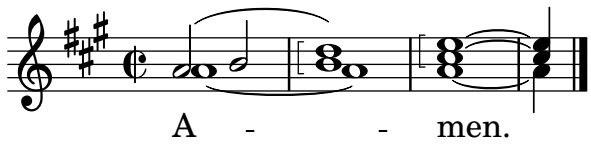
Utilisation d'un `arpeggioBracket` pour rendre les divisions plus évidentes

Un crochet d'arpège (`arpeggioBracket`) permet de mettre en évidence les divisions d'un pupitre en l'absence de hampe, comme on le voit régulièrement dans les partitions pour chœur.

```
\include "english.ly"

\score {
  \relative c'' {
    \key a \major
    \time 2/2
    <<
      \new Voice = "upper"
      <<
        { \voiceOne \arpeggioBracket
          a2( b2
          <b d>1\arpeggio)
          <cs e>\arpeggio ~
          <cs e>4
        }
```

```
        \addlyrics { \lyricmode { A -- men. } }
      >>
      \new Voice = "lower"
      { \voiceTwo
        a1 ~
        a
        a ~
        a4 \bar "|."
      }
    >>
  }
  \layout { ragged-right = ##t }
}
```



Recours aux balises pour produire une partition ancienne et moderne à partir de la même source

Grâce aux balises (*tags*), il est possible d'utiliser une même source pour produire une partition de musique mensurale et moderne. Dans cet exemple est créée la fonction `menrest` qui permettra de positionner les silence comme dans la version originale, tout en respectant leur position sur une portée standard. Les balises permettent d'adapter les barres en fin de pièce ; elles peuvent aussi gérer d'autres différenciations selon les besoins, comme des « mesures de silence » (R1, R\breve, etc.) en notation moderne mais des silences normaux (r1, r\breve, etc.) en notation ancienne. L'action de convertir de la musique mensurale en version moderne est communément appelée « transcription ».

```
menrest = #(define-music-function (note)
  (ly:music?)
#{
    \tag #'mens $(make-music 'RestEvent note)
    \tag #'mod $(make-music 'RestEvent note 'pitch '())
#})

MensStyle = {
  \autoBeamOff
  \override NoteHead.style = #'petrucci
  \override Score.BarNumber.transparent = ##t
  \override Stem.neutral-direction = #up
}

finalis = {
  \once \override BreathingSign.stencil = #ly:breathing-sign::finalis
  \once \override BreathingSign.Y-offset = #0
  \once \override BreathingSign.minimum-X-extent = #'(-1.0 . 0.0)
  \once \override BreathingSign.minimum-Y-extent = #'(-2.5 . 2.5)
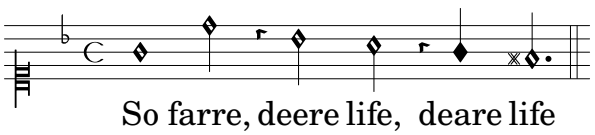
  \breathe
}
```
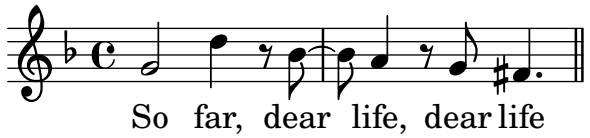
```
Music = \relative c'' {
  \set Score.tempoHideNote = ##t
  \key f \major
  \time 4/4
  g1 d'2 \menrest bes4 bes2 a2 r4 g4 fis2.
  \tag #'mens { \finalis }
  \tag #'mod { \bar "||" }
}

MenLyr = \lyricmode { So farre, deere life, deare life }
ModLyr = \lyricmode { So far, dear life, dear life }

\score {
  \keepWithTag #'mens {
    <<
      \new MensuralStaff
      {
        \new MensuralVoice = Cantus
          \clef "mensural-c1" \MensStyle \Music
      }
      \new Lyrics \lyricsto Cantus \MenLyr
    >>
  }
}

\score {
  \keepWithTag #'mod {
    \new ChoirStaff <<
      \new Staff
      {
        \new Voice = Sop \with {
          \remove "Note_heads_engraver"
          \consists "Completion_heads_engraver"
          \remove "Rest_engraver"
          \consists "Completion_rest_engraver" }
        {
          \shiftDurations #1 #0 { \autoBeamOff \Music }
        }
      }
      \new Lyrics \lyricsto Sop \ModLyr
    >>
  }
}
```



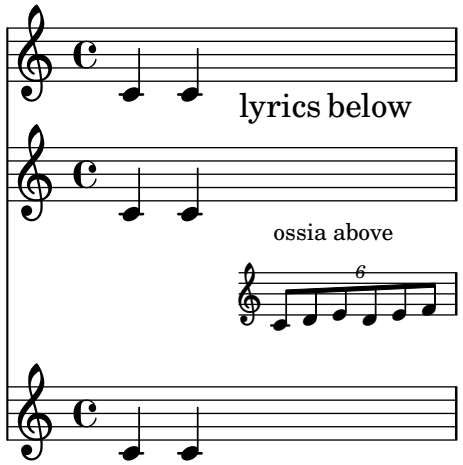


Positionnement d'une ossia et des paroles

Cet exemple illustre la manière de positionner une portée d'ossia et des paroles à l'aide des propriétés de contexte `alignBelowContext` et `alignAboveContext`.

```
\paper {
  ragged-right = ##t
}

\relative c' <<
  \new Staff = "1" { c4 c s2 }
  \new Staff = "2" { c4 c s2 }
  \new Staff = "3" { c4 c s2 }
  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = #"1"
        lyrics4 below
      }
      \new Staff \with {
        alignAboveContext = #"3"
        fontSize = #-2
        \override StaffSymbol.staff-space = #(magstep -2)
        \remove "Time_signature_engraver"
      } {
        \tuplet 6/4 {
          \override TextScript.padding = #3
          c8[^"ossia above" d e d e f]
        }
      }
    >>
  }
>>
```



Centrage vertical de paroles communes

Il arrive, dans certaines partitions vocales, que quelques paroles soient communes à tous les couplets. Cette partie commune peut peut se présenter verticalement centrée, comme dans l'exemple suivant.

```
dropLyrics = {
  \override LyricText.extra-offset = #'(0 . -4.5)
  \override LyricHyphen.extra-offset = #'(0 . -4.5)
  \override LyricExtender.extra-offset = #'(0 . -4.5)
  \override StanzaNumber.extra-offset = #'(0 . -4.5)
}

raiseLyrics = {
  \revert LyricText.extra-offset
  \revert LyricHyphen.extra-offset
  \revert LyricExtender.extra-offset
  \revert StanzaNumber.extra-offset
}

skipFour = \repeat unfold 4 { \skip 8 }

lyricsA = \lyricmode {
  The first verse has
  \dropLyrics
  \set stanza = #"   All:"
  the com -- mon __ words
  \raiseLyrics
  used in all four.
}
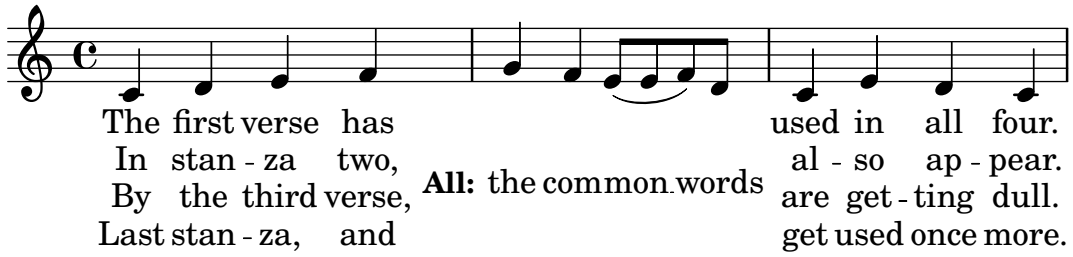
lyricsB = \lyricmode { In stan -- za two,   \skipFour al -- so ap -- pear. }

lyricsC = \lyricmode { By the third verse,  \skipFour are get -- ting dull. }

lyricsD = \lyricmode { Last stan -- za, and \skipFour get used once more. }

melody = \relative c' {
  c4 d e f |
  g f e8( e f) d |
  c4 e d c |
}

\score {
  <<
    \new Voice = m \melody
    \new Lyrics \lyricsto m \lyricsA
    \new Lyrics \lyricsto m \lyricsB
    \new Lyrics \lyricsto m \lyricsC
    \new Lyrics \lyricsto m \lyricsD
  >>
}
```



Ensemble vocal avec réduction pour piano

Ce canevas ajoute une réduction pour piano à une partition standard pour chœur à quatre voix mixtes. Ceci illustre l'un des avantages de LilyPond : une expression musicale peut être réutilisée sans effort. Toute modification apportée à l'une des voix, mettons `tenorMusique`, sera automatiquement reportée dans la réduction pour piano.

```
\paper {
  top-system-spacing.basic-distance = #10
  score-system-spacing.basic-distance = #20
  system-system-spacing.basic-distance = #20
  last-bottom-spacing.basic-distance = #10
}

global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative {
  c''4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

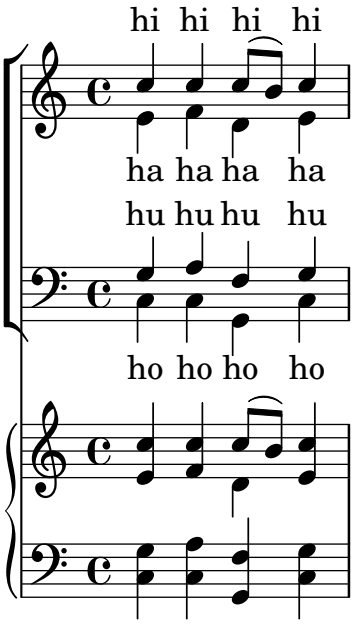
altoMusic = \relative {
  e'4 f d e
}
altoWords =\lyricmode {
  ha ha ha ha
}

tenorMusic = \relative {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}
```

```
\score {
  <<
    \new ChoirStaff <<
      \new Lyrics = "sopranos" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "women" <<
        \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
        \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
      >>
      \new Lyrics = "altos"
      \new Lyrics = "tenors" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }

      \new Staff = "men" <<
        \clef bass
        \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
        \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
      >>
      \new Lyrics = "basses"
      \context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
      \context Lyrics = "altos" \lyricsto "altos" \altoWords
      \context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
      \context Lyrics = "basses" \lyricsto "basses" \bassWords
    >>
    \new PianoStaff <<
      \new Staff <<
        \set Staff.printPartCombineTexts = ##f
        \partCombine
        << \global \sopMusic >>
        << \global \altoMusic >>
      >>
      \new Staff <<
        \clef bass
        \set Staff.printPartCombineTexts = ##f
        \partCombine
        << \global \tenorMusic >>
        << \global \bassMusic >>
      >>
    >>
  >>
}
```



Ensemble vocal avec alignement des paroles selon le contexte

Ce canevas ressemble beaucoup à celui pour chœur à quatre voix mixtes. La différence réside dans le fait que les paroles sont positionnées en ayant recours à `alignAboveContext` et `alignBelowContext`.

```
global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative c'' {
  c4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

altoMusic = \relative c' {
  e4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}
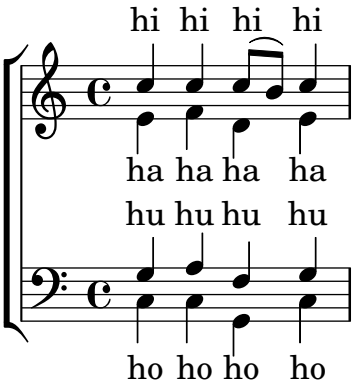
tenorMusic = \relative c' {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative c {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
```

```
}

\score {
  \new ChoirStaff <<
    \new Staff = "women" <<
      \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
      \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = #"women" }
      \lyricsto "sopranos" \sopWords
    \new Lyrics \with { alignBelowContext = #"women" }
      \lyricsto "altos" \altoWords
    % we could remove the line about this with the line below, since
    % we want the alto lyrics to be below the alto Voice anyway.
    % \new Lyrics \lyricsto "altos" \altoWords

    \new Staff = "men" <<
      \clef bass
      \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
      \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = #"men" }
      \lyricsto "tenors" \tenorWords
    \new Lyrics \with { alignBelowContext = #"men" }
      \lyricsto "basses" \bassWords
    % again, we could replace the line above this with the line below.
    % \new Lyrics \lyricsto "basses" \bassWords
  >>
}
```



Ensemble vocal avec couplet et refrain

Ce canevas illustre la manière d'agencer une œuvre vocale où le couplet est chanté en solo et le refrain à deux voix. Vous noterez le recours aux silences invisibles dans la variable `\global` : ils permettent de positionner les changements de métrique et autres éléments communs à toutes les parties, ce pour l'intégralité du morceau.

```
global = {
  \key g \major

  % verse
```

```
  \time 3/4
  s2.*2
  \break

  % refrain
  \time 2/4
  s2*2
  \bar "|."
}

SoloNotes = \relative g' {
  \clef "treble"

  % verse
  g4 g g |
  b4 b b |

  % refrain
  R2*2 |
}

SoloLyrics = \lyricmode {
  One two three |
  four five six |
}

SopranoNotes = \relative c'' {
  \clef "treble"

  % verse
  R2.*2 |

  % refrain
  c4 c |
  g4 g |
}

SopranoLyrics = \lyricmode {
  la la |
  la la |
}

BassNotes = \relative c {
  \clef "bass"

  % verse
  R2.*2 |

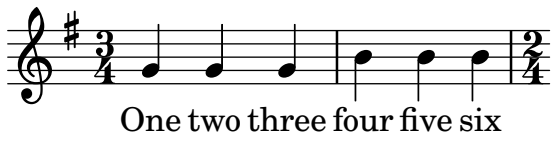
  % refrain
  c4 e |
  d4 d |
}
```

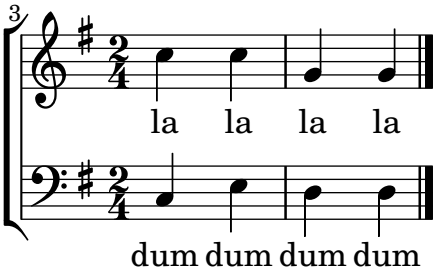
```
BassLyrics = \lyricmode {
  dum dum |
  dum dum |
}

\score {
  <<
    \new Voice = "SoloVoice" << \global \SoloNotes >>
    \new Lyrics \lyricsto "SoloVoice" \SoloLyrics

    \new ChoirStaff <<
      \new Voice = "SopranoVoice" << \global \SopranoNotes >>
      \new Lyrics \lyricsto "SopranoVoice" \SopranoLyrics

      \new Voice = "BassVoice" << \global \BassNotes >>
      \new Lyrics \lyricsto "BassVoice" \BassLyrics
    >>
  >>
  \layout {
    ragged-right = ##t
    \context { \Staff
      % these lines prevent empty staves from being printed
      \RemoveEmptyStaves
      \override VerticalAxisGroup.remove-first = ##t
    }
  }
}
```





Ensemble vocal (simple)

Ce fichier constitue un canevas standard de partition pour chœur à quatre voix mixtes. Lorsque les ensembles s'étoffent, il est judicieux de recourir à une section spécifique incluse dans chacune des parties, tout particulièrement pour gérer la métrique et la tonalité qui, la plupart du temps, sont communes à tous les pupitres. Comme il est d'usage pour les hymnes, les quatre voix sont réparties sur deux portées.

```
\paper {
  top-system-spacing.basic-distance = #10
```

```
  score-system-spacing.basic-distance = #20
  system-system-spacing.basic-distance = #20
  last-bottom-spacing.basic-distance = #10
}

global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative {
  c''4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}
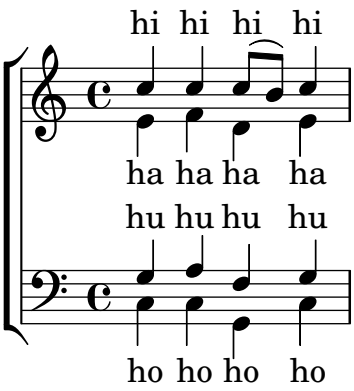
altoMusic = \relative {
  e'4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

tenorMusic = \relative {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  \new ChoirStaff <<
    \new Lyrics = "sopranos" \with {
      % this is needed for lyrics above a staff
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff = "women" <<
      \new Voice = "sopranos" {
        \voiceOne
        << \global \sopMusic >>
      }
      \new Voice = "altos" {
        \voiceTwo
        << \global \altoMusic >>
```

```
      }
    >>
    \new Lyrics = "altos"
    \new Lyrics = "tenors" \with {
      % this is needed for lyrics above a staff
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff = "men" <<
      \clef bass
      \new Voice = "tenors" {
        \voiceOne
        << \global \tenorMusic >>
      }
      \new Voice = "basses" {
        \voiceTwo << \global \bassMusic >>
      }
    >>
    \new Lyrics = "basses"
    \context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
    \context Lyrics = "altos" \lyricsto "altos" \altoWords
    \context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
    \context Lyrics = "basses" \lyricsto "basses" \bassWords
  >>
}
```



Chords

Section “Chord notation” dans *Manuel de notation*

Ajout d’une basse chiffrée au-dessus ou au-dessous des notes
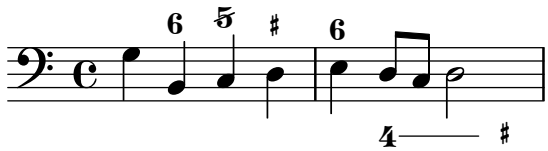
Une ligne de basse chiffrée peut se positionner au-dessus ou en dessous d’une partie de basse, grâce à la propriété `BassFigureAlignmentPositioning.direction` attachée à un contexte `Staff`. Les possibilités sont `#UP` (ou `#1`) pour au-dessus, `#CENTER` (ou `#0`) pour centrée, et `#DOWN` (ou `#-1`) pour en dessous.

Cette propriété peut être sujette à modification au cours de la partition. Il vous faudra insérer un `\once \override` si ce changement devait être temporaire.

```
bass = {
  \clef bass
  g4 b, c d
  e d8 c d2
}

continuo = \figuremode {
  <_>4 <6>4 <5/>4
  \override Staff.BassFigureAlignmentPositioning.direction = #UP
  %\bassFigureStaffAlignmentUp
  < _+ >4 <6>
  \set Staff.useBassFigureExtenders = ##t
  \override Staff.BassFigureAlignmentPositioning.direction = #DOWN
  %\bassFigureStaffAlignmentDown
  <4>4. <4>8 <_+>4
}

\score {
  <<
    \new Staff = bassStaff \bass
    \context Staff = bassStaff \continuo
  >>
}
```



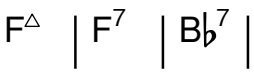
Chiffrages et barres de mesure

L’ajout du graveur `Bar_engraver` à un contexte `ChordNames` permet d’imprimer les barres de mesure entre les chiffrages.

```
\new ChordNames \with {
  \override BarLine.bar-extent = #'(-2 . 2)
  \consists "Bar_engraver"
}

\chordmode {
```

```
  f1:maj7 f:7 bes:7
}
```



Notation d'accords barrés pour guitare (avec extension textuelle)

Voici comment imprimer un « accord barré » – il suffit de dé-commenter la ligne appropriée.

La syntaxe est la suivante : `\bbarre #"fret_number" note(s)`

La fonction Scheme ici présentée peut faire l'objet d'un fichier de définition personnalisée à inclure dans vos propres fichiers sources.

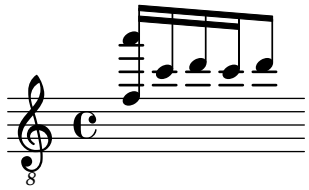
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% %%%%%%  Cut here ----- Start 'bbarred.ly'

%% C with slash -------------------------------
cWithSlash = \markup {
  \combine \roman C \translate #'(0.6 . -0.4) \draw-line #'(0 . 2.0)
}
%% Span ----------------------------------
%% Syntax: \bbarre #"text" { notes } - text = any number of box
bbarre =
#(define-music-function (barre location str music) (string? ly:music?)
   (let ((elts (extract-named-music music 'rhythmic-event)))
     (if (pair? elts)
         (let ((first-element (first elts))
               (last-element (last elts)))
           (set! (ly:music-property first-element 'articulations)
                 (cons (make-music 'TextSpanEvent 'span-direction -1)
                       (ly:music-property first-element 'articulations)))
           (set! (ly:music-property last-element 'articulations)
                 (cons (make-music 'TextSpanEvent 'span-direction 1)
                       (ly:music-property last-element 'articulations))))))
   #{
       \once \override TextSpanner.font-size = #-2
       \once \override TextSpanner.font-shape = #'upright
       \once \override TextSpanner.staff-padding = #3
       \once \override TextSpanner.style = #'line
       \once \override TextSpanner.to-barline = ##f
       \once \override TextSpanner.bound-details =
            #`((left
                (text . ,#{ \markup { \draw-line #'( 0 . -.5) } #})
                (Y . 0)
                (padding . 0.25)
                (attach-dir . -2))
               (right
                (text . ,#{ \markup { \cWithSlash #str } #})
                (Y . 0)
                (padding . 0.25)
                (attach-dir . 2)))
%% uncomment this line for make full barred
```

```
      % \once \override TextSpanner.bound-details.left.text = \markup{"B" #str}
      $music
   #})

%% %%%%%%  Cut here ----- End 'bbarred.ly'
%% Copy and change the last line for full barred. Rename in 'fbarred.ly'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Syntaxe: \bbarre #"text" { notes } - text = any number of box
\relative c'{ \clef "G_8" \stemUp \bbarre #"III" { <f a'>16[  c' d c d8] } }
```



Personnalisation du séparateur d'accords

Le séparateur de termes d'un chiffrage peut adopter n'importe quelle forme à l'aide d'un *markup*.

```
\chords {
  c:7sus4
  \set chordNameSeparator
    = \markup { \typewriter | }
  c:7sus4
}
```

C7 sus4 C7 | sus4

Adoption de la dénomination germanique ou semi-germanique des accords
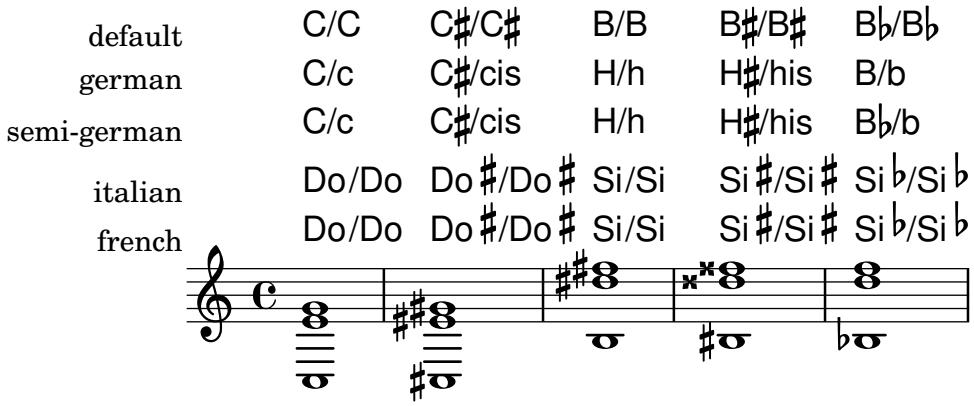
LilyPond adpote par défaut la dénomination anglaise pour le nom et chiffrage des accords. L'instruction `\germanChords` permet d'afficher `H` et `B` en lieu et place de `B` et `Bes`. L'instruction `\semiGermanChords` leur substituera `H` et `Bb`.

```
scm = \chordmode {
  c1/c | cis/cis
  b1/b | bis/bis | bes/bes
}


\layout {
  ragged-right = ##t
  \context {
    \ChordNames
    \consists "Instrument_name_engraver"
  }
}

<<
  \new ChordNames {
```

```
    \set instrumentName = #"default"
    \scm
  }
  \new ChordNames {
    \set instrumentName = #"german"
    \germanChords \scm
  }
  \new ChordNames {
    \set instrumentName = #"semi-german"
    \semiGermanChords \scm
  }
  \new ChordNames {
    \set instrumentName = #"italian"
    \italianChords \scm
  }
  \new ChordNames {
    \set instrumentName = #"french"
    \frenchChords \scm
  }
  \context Voice { \scm }
>>
```



Emplacement des altération en basse continue

On peut choisir d'imprimer les altérations et signes plus aussi bien avant qu'après les chiffres, en réglant les propriétés `figuredBassAlterationDirection` et `figuredBassPlusDirection`.

```
\figures {
  <6\+> <5+> <6 4-> r
  \set figuredBassAlterationDirection = #RIGHT
  <6\+> <5+> <6 4-> r
  \set figuredBassPlusDirection = #RIGHT
  <6\+> <5+> <6 4-> r
  \set figuredBassAlterationDirection = #LEFT
  <6\+> <5+> <6 4-> r
}
```



Modèles de chiffrage d'accords

Il est possible de créer votre propre modèle de chiffrages en réglant la propriété `chordNameExceptions`.

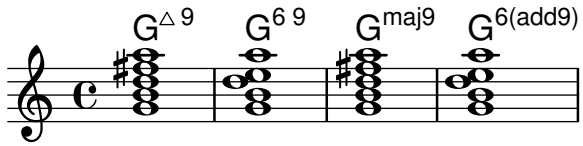
```
% modify maj9 and 6(add9)
% Exception music is chords with markups
chExceptionMusic = {
  <c e g b d'>1-\markup { \super "maj9" }
  <c e g a d'>1-\markup { \super "6(add9)" }
}

% Convert music to list and prepend to existing exceptions.
chExceptions = #(append
  (sequential-music-to-chord-exceptions chExceptionMusic #t)
  ignatzekExceptions)

theMusic = \chordmode {
  g1:maj9 g1:6.9
  \set chordNameExceptions = #chExceptions
  g1:maj9 g1:6.9
}

\layout {
  ragged-right = ##t
}

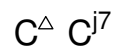
<<
   \new ChordNames \theMusic
   \new Voice \theMusic
>>
```



Chiffrage d'un maj7

La représentation d'un accord de septième majeure se gère par le `majorSevenSymbol`.

```
\chords {
  c:7+
  \set majorSevenSymbol = \markup { j7 }
  c:7+
}
```



Noms d'accord alternatifs

Les noms d'accord sont générés à partir d'une liste de hauteurs. Les fonctions qui construisent ces noms peuvent être personnalisées.

Voici les accords suivant la nomenclature d'Ignatzek (pp. 17-18, 1995) qu'utilise LilyPond par défaut depuis la version 1.7.20, comparée à une alternative jazz et à la notation d'Harald Banter (1987). Une police plus petite est utilisée pour cette dernière dans la mesure où elle tend à être plus verbeuse.

Ceci est un reflet du mécanisme utilisé à l'origine par LilyPond, jusque dans les version 1.7. N'ayant pas fait l'objet d'une maintenance suivie, certaines fonctionnalités ont été perdues – notamment au niveau des listes d'exception – et des bogues ont fait leur apparition.

```
%%%% Legacy chord naming functions (formerly in scm/chord-generic-names.scm)
%%%% Copyright (C) 2003--2015 Jan Nieuwenhuizen <janneke@gnu.org>

#(set-global-staff-size 19.7)

#(define-public (banter-chordnames pitches bass inversion context)
  (old_chord->markup 'banter pitches bass inversion context))

#(define-public (jazz-chordnames pitches bass inversion context)
  (old_chord->markup 'jazz pitches bass inversion context))

#(define (define-translator-property symbol type? description)
  (if (not (and (symbol? symbol)
    (procedure? type?)
    (string? description)))
      (ly:error "error in call of define-translator-property"))
  (if (not (equal? (object-property symbol 'translation-doc) #f))
      (ly:error (_ "symbol ~S redefined") symbol))

  (set-object-property! symbol 'translation-type? type?)
  (set-object-property! symbol 'translation-doc description)
  symbol)

#(for-each
  (lambda (x)
    (apply define-translator-property x))
  `((chordNameExceptionsFull ,list? "An alist of full chord
exceptions.  Contains @code{(@var{chord} . @var{markup})} entries.")
    (chordNameExceptionsPartial ,list? "An alist of partial chord
exceptions.  Contains @code{(@var{chord} . (@var{prefix-markup}
@var{suffix-markup}))} entries.")))

#(define-public (old_chord->markup
                style pitches bass inversion context)
  "Entry point for @code{Chord_name_engraver}.
@var{pitches}, @var{bass}, and @var{inversion} are lily pitches."
  (define (default-note-namer pitch)
    (note-name->markup pitch #f))

  (define (markup-or-empty-markup markup)
    "Return MARKUP if markup, else empty-markup"
    (if (markup? markup) markup empty-markup))

  (define (accidental->markup alteration)
```

```
  "Return accidental markup for ALTERATION."
  (if (= alteration 0)
      (make-line-markup (list empty-markup))
      (conditional-kern-before
       (alteration->text-accidental-markup alteration)
       (= alteration FLAT) 0.094725)))

(define (list-minus a b)
  "Return list of elements in A that are not in B."
  (lset-difference eq? a b))

(define (markup-join markups sep)
  "Return line-markup of MARKUPS, joining them with markup SEP"
  (if (pair? markups)
      (make-line-markup (list-insert-separator markups sep))
      empty-markup))

(define (conditional-kern-before markup bool amount)
  "Add AMOUNT of space before MARKUP if BOOL is true."
  (if bool
      (make-line-markup
       (list (make-hspace-markup amount)
         markup))
      markup))

(define (step-nr pitch)
  (let* ((pitch-nr (+ (* 7 (ly:pitch-octave pitch))
                      (ly:pitch-notename pitch)))
         (root-nr (+ (* 7 (ly:pitch-octave (car pitches)))
                     (ly:pitch-notename (car pitches)))))
    (+ 1 (- pitch-nr root-nr))))

(define (next-third pitch)
  (ly:pitch-transpose pitch
                      (ly:make-pitch 0 2 (if (or (= (step-nr pitch) 3)
                                                 (= (step-nr pitch) 5))
                                             FLAT 0))))

(define (step-alteration pitch)
  (let* ((diff (ly:pitch-diff (ly:make-pitch 0 0 0) (car pitches)))
         (normalized-pitch (ly:pitch-transpose pitch diff))
         (alteration (ly:pitch-alteration normalized-pitch)))
    (if (= (step-nr pitch) 7) (+ alteration SEMI-TONE) alteration)))

(define (pitch-unalter pitch)
  (let ((alteration (step-alteration pitch)))
    (if (= alteration 0)
        pitch
        (ly:make-pitch (ly:pitch-octave pitch) (ly:pitch-notename pitch)
                       (- (ly:pitch-alteration pitch) alteration)))))

(define (step-even-or-altered? pitch)
```

```
  (let ((nr (step-nr pitch)))
    (if (!= (modulo nr 2) 0)
        (!= (step-alteration pitch) 0)
        #t)))

(define (step->markup-plusminus pitch)
  (let ((alt (step-alteration pitch)))
    (make-line-markup
     (list
      (number->string (step-nr pitch))
      (cond
       ((= alt DOUBLE-FLAT) "--")
       ((= alt FLAT) "-")
       ((= alt NATURAL) "")
       ((= alt SHARP) "+")
       ((= alt DOUBLE-SHARP) "++"))))))

(define (step->markup-accidental pitch)
  (make-line-markup
   (list (accidental->markup (step-alteration pitch))
         (make-simple-markup (number->string (step-nr pitch))))))

(define (step->markup-ignatzek pitch)
  (make-line-markup
   (if (and (= (step-nr pitch) 7)
            (= (step-alteration pitch) 1))
       (list (ly:context-property context 'majorSevenSymbol))
       (list (accidental->markup (step-alteration pitch))
             (make-simple-markup (number->string (step-nr pitch)))))))

;; tja, kennok
(define (make-sub->markup step->markup)
  (lambda (pitch)
    (make-line-markup (list (make-simple-markup "no")
                            (step->markup pitch)))))

(define (step-based-sub->markup step->markup pitch)
  (make-line-markup (list (make-simple-markup "no") (step->markup pitch))))

(define (get-full-list pitch)
  (if (<= (step-nr pitch) (step-nr (last pitches)))
      (cons pitch (get-full-list (next-third pitch)))
      '()))

(define (get-consecutive nr pitches)
  (if (pair? pitches)
      (let* ((pitch-nr (step-nr (car pitches)))
             (next-nr (if (!= (modulo pitch-nr 2) 0) (+ pitch-nr 2) nr)))
        (if (<= pitch-nr nr)
            (cons (car pitches) (get-consecutive next-nr (cdr pitches)))
            '()))
      '()))
```

```
;;; FIXME -- exceptions no longer work. -vv

(define (full-match exceptions)
  (if (pair? exceptions)
      (let* ((e (car exceptions))
             (e-pitches (car e)))
        (if (equal? e-pitches pitches)
            e
            (full-match (cdr exceptions))))
      #f))

(define (partial-match exceptions)
  (if (pair? exceptions)
      (let* ((e (car exceptions))
             (e-pitches (car e)))
        (if (equal? e-pitches (take pitches (length e-pitches)))
            e
            (partial-match (cdr exceptions))))
      #f))

;; FIXME: exceptions don't work anyway.
(if #f (begin
         (write-me "pitches: " pitches)))
(let* ((full-exceptions
        (ly:context-property context 'chordNameExceptionsFull))
       (full-exception (full-match full-exceptions))
       (full-markup (if full-exception (cadr full-exception) '()))
       (partial-exceptions
        (ly:context-property context 'chordNameExceptionsPartial))
       (partial-exception (partial-match partial-exceptions))
       (partial-pitches (if partial-exception (car partial-exception) '()))
       (partial-markup-prefix
        (if partial-exception (markup-or-empty-markup
                               (cadr partial-exception)) empty-markup))
       (partial-markup-suffix
        (if (and partial-exception (pair? (cddr partial-exception)))
            (markup-or-empty-markup (caddr partial-exception)) empty-markup))
       (root (car pitches))
       (full (get-full-list root))
       ;; kludge alert: replace partial matched lower part of all with
       ;; 'normal' pitches from full
       ;; (all pitches)
       (all (append (take full (length partial-pitches))
                    (drop pitches (length partial-pitches))))

       (highest (last all))
       (missing (list-minus full (map pitch-unalter all)))
       (consecutive (get-consecutive 1 all))
       (rest (list-minus all consecutive))
       (altered (filter step-even-or-altered? all))
       (cons-alt (filter step-even-or-altered? consecutive))
```

```
      (base (list-minus consecutive altered)))


    (if #f (begin
             (write-me "full:" full)
             ;; (write-me "partial-pitches:" partial-pitches)
             (write-me "full-markup:" full-markup)
             (write-me "partial-markup-perfix:" partial-markup-prefix)
             (write-me "partial-markup-suffix:" partial-markup-suffix)
             (write-me "all:" all)
             (write-me "altered:" altered)
             (write-me "missing:" missing)
             (write-me "consecutive:" consecutive)
             (write-me "rest:" rest)
             (write-me "base:" base)))

    (case style
      ((banter)
       ;;    root
       ;;    + steps:altered + (highest all -- if not altered)
       ;;    + subs:missing

       (let* ((root->markup default-note-namer)
              (step->markup step->markup-plusminus)
              (sub->markup (lambda (x)
                             (step-based-sub->markup step->markup x)))
              (sep (make-simple-markup "/")))

         (if
          (pair? full-markup)
          (make-line-markup (list (root->markup root) full-markup))

          (make-line-markup
           (list
            (root->markup root)
            partial-markup-prefix
            (make-super-markup
             (markup-join
              (append
               (map step->markup
                    (append altered
                            (if (and (> (step-nr highest) 5)
                                     (not
                                      (step-even-or-altered? highest)))
                                (list highest) '())))
               (list partial-markup-suffix)
               (map sub->markup missing))
              sep)))))))


      ((jazz)
       ;;    root
```

```
        ;;    + steps:(highest base) + cons-alt
        ;;    + 'add'
        ;;    + steps:rest
        (let* ((root->markup default-note-namer)
               (step->markup step->markup-ignatzek)
               (sep (make-simple-markup " "))
               (add-prefix (make-simple-markup " add")))

          (if
           (pair? full-markup)
           (make-line-markup (list (root->markup root) full-markup))

           (make-line-markup
            (list
             (root->markup root)
             partial-markup-prefix
             (make-super-markup
              (make-line-markup
               (list

                ;; kludge alert: omit <= 5
                ;;(markup-join (map step->markup
                ;;                        (cons (last base) cons-alt)) sep)

                ;; This fixes:
                ;;  c     C5       -> C
                ;;  c:2   C5 2     -> C2
                ;;  c:3-  Cm5      -> Cm
                ;;  c:6.9 C5 6add9 -> C6 add 9 (add?)
                ;;  ch = \chords { c c:2 c:3- c:6.9^7 }
                (markup-join (map step->markup
                                  (let ((tb (last base)))
                                    (if (> (step-nr tb) 5)
                                        (cons tb cons-alt)
                                        cons-alt))) sep)

                (if (pair? rest)
                    add-prefix
                    empty-markup)
                (markup-join (map step->markup rest) sep)
                partial-markup-suffix))))))))

      (else empty-markup))))


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
%%%% Here begins the actual snippet:

chs = \transpose c' c' {
  <c e g>1
  <c es g>  % m = minor triad
```

```
  <c e gis>
  <c es ges> \break
  <c e g bes>
  <c es g bes>
  <c e g b>  % triangle = maj
  <c es ges beses>
  <c es ges b> \break
  <c e gis bes>
  <c es g b>
  <c e gis b>
  <c es ges bes> \break
  <c e g a>  % 6 = major triad with added sixth
  <c es g a>  % m6 = minor triad with added sixth
  <c e g bes d'>
  <c es g bes d'> \break
  <c es g bes d' f' a' >
  <c es g bes d' f' >
  <c es ges bes d' >
  <c e g bes des' > \break
  <c e g bes dis'>
  <c e g bes d' f'>
  <c e g bes d' fis'>
  <c e g bes d' f' a'> \break
  <c e g bes d' fis' as'>
  <c e gis bes dis'>
  <c e g bes dis' fis'>
  <c e g bes d' f' as'> \break
  <c e g bes des' f' as'>
  <c e g bes d' fis'>
  <c e g b d'>
  <c e g bes d' f' as'> \break
  <c e g bes des' f' as'>
  <c e g bes des' f' a'>
  <c e g b d'>
  <c e g b d' f' a'> \break
  <c e g b d' fis'>
  <c e g bes des' f ' a'>
  <c f g>
  <c f g bes> \break
  <c f g bes d'>
  <c e g d'>  % add9
  <c es g f'>
  <c e g b fis'>  % Lydian
  <c e g bes des' ees' fis' aes'>  % altered chord
}


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% alternate Jazz notation

efullmusicJazzAlt = {
  <c e gis>1-\markup { "+" }
```

```
  <c e g b>-\markup {
    \normal-size-super
    % \override #'(font-family . math) "N"
    \override #'(font-family . math) "M"
  }
  %%c:3.5.7 = \markup { \override #'(font-family . math) "M" }
  %%c:3.5.7 = \markup { \normal-size-super "maj7" }

  <c es ges>-\markup { \super "o" }  % should be $\circ$ ?
  <c es ges bes>-\markup { \super \combine "o" "/" }
  <c es ges beses>-\markup { \super  "o7" }
}

efullJazzAlt = #(sequential-music-to-chord-exceptions efullmusicJazzAlt #f)

epartialmusicJazzAlt = {
  <c d>1-\markup { \normal-size-super "2" }
  <c es>-\markup { "m" }
  <c f>-\markup { \normal-size-super "sus4" }
  <c g>-\markup { \normal-size-super "5" }
  %% TODO, partial exceptions
  <c es f>-\markup { "m" }-\markup { \normal-size-super "sus4" }
  <c d es>-\markup { "m" }-\markup { \normal-size-super "sus2" }
}

epartialJazzAlt = #(sequential-music-to-chord-exceptions epartialmusicJazzAlt #f)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

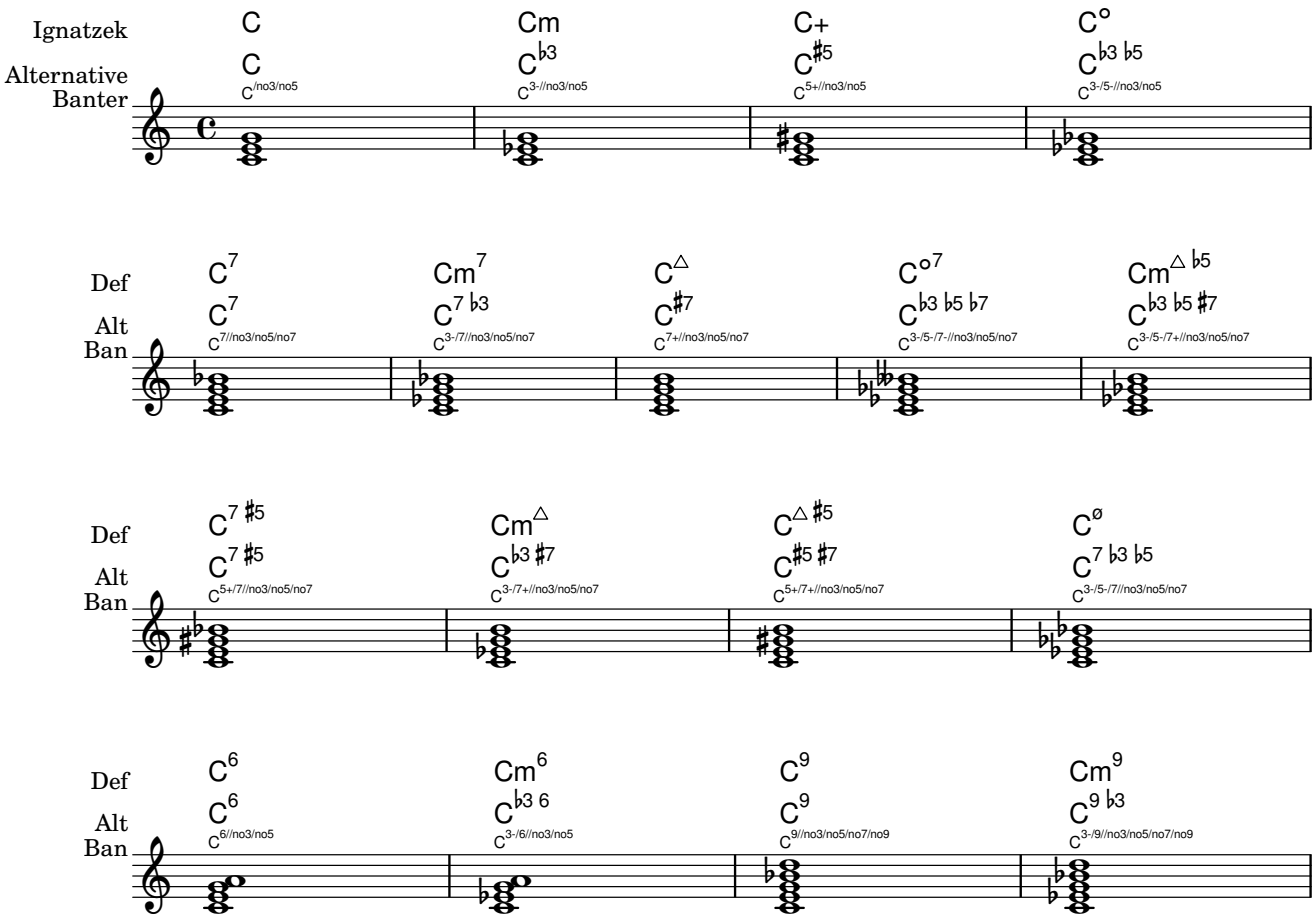
\score {
  <<
    \new ChordNames {
      %% Already set by default:
      %\set chordNameFunction = #ignatzek-chord-names
      \set instrumentName = "Ignatzek"
      \set shortInstrumentName = "Def"
      \chs
    }

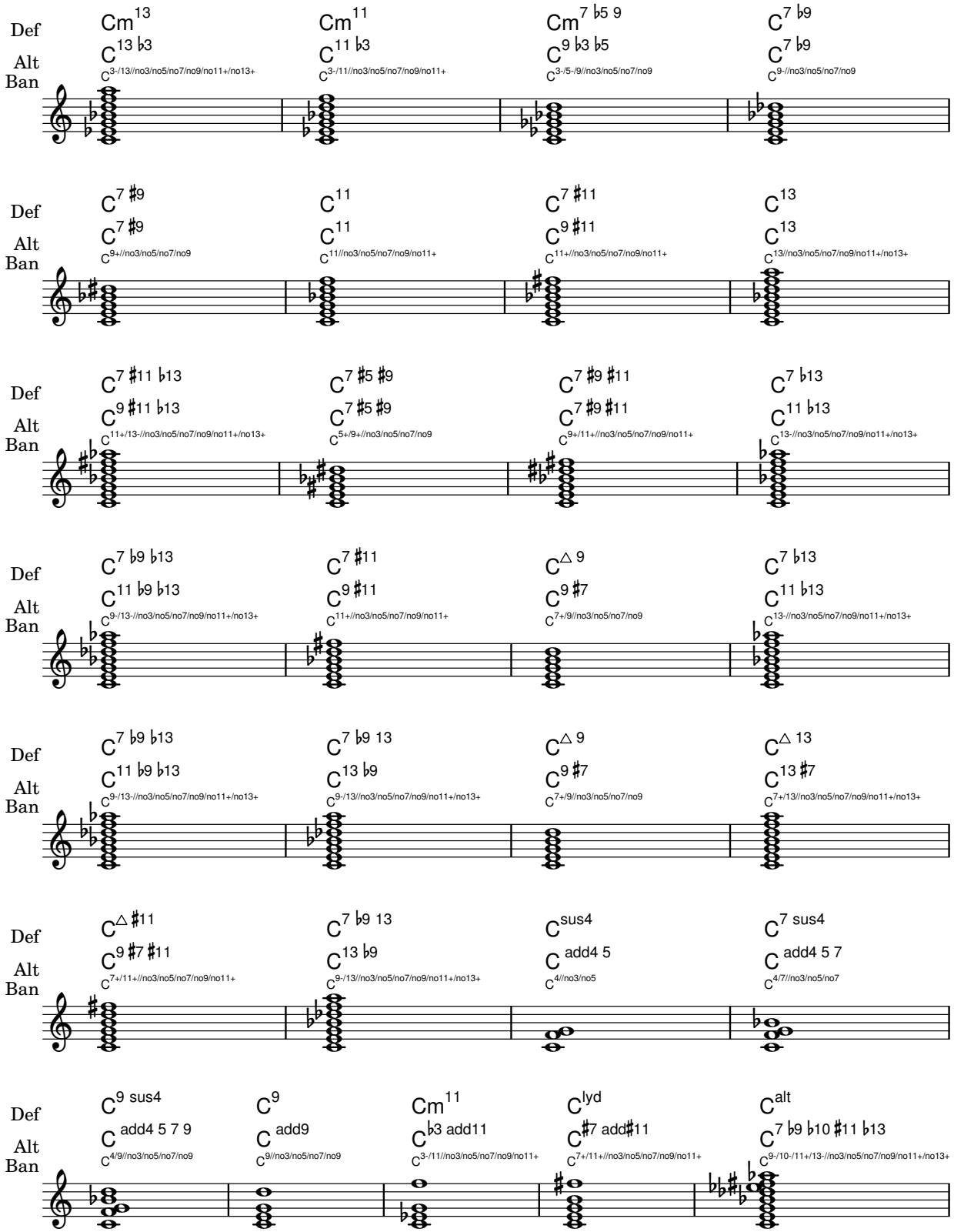
    \new ChordNames {
      \set chordNameFunction = #jazz-chordnames
      \set majorSevenSymbol = \whiteTriangleMarkup
      \set chordNameSeparator = "/"
      \set chordNameExceptionsFull = \efullJazzAlt
      \set chordNameExceptionsPartial = \epartialJazzAlt
      \set instrumentName = "Alternative"
      \set shortInstrumentName = "Alt"
      \chs
    }

    %% This is the Banter (1987) style.  It gives exceedingly
    %% verbose (wide) names, making the output file take up to 4 pages.
```

```
    \new ChordNames {
      \set chordNameFunction = #banter-chordnames
      \override ChordName.font-size = #-3
      \set instrumentName = "Banter"
      \set shortInstrumentName = "Ban"
      \chs
    }

  \new Staff \transpose c c' { \chs }
  >>
  \layout {
    #(layout-set-staff-size 16)
    system-system-spacing.basic-distance = #0
    \context {
      \ChordNames
      \consists "Instrument_name_engraver"
    }
    \context {
      \Score
      \remove "Bar_number_engraver"
    }
  }
}
```





Accords avec doigté espacé pour `FretBoards` et `TabVoice`

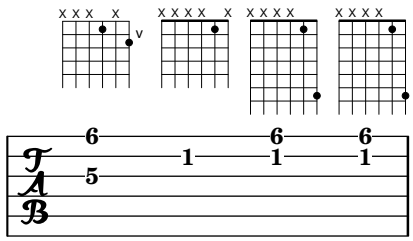
Il peut arriver que le doigté d'un accord soit assez étendu. Sauf mention contraire, la propriété de contexte `maximumFretStretch` est cependant fixée à `4`, ce qui peut générer un avertissement « Pas de corde pour la hauteur… » et la note est omise. On peut régler `maximumFretStretch` sur une valeur appropriée ou assigner explicitement leur numéro de corde à toutes les notes d'un accord.

```
%% The code below will print two warnings, which may be omitted by uncommenting:
```

```
%#(for-each (lambda (x) (ly:expect-warning "No string for pitch")) (iota 2))

mus = {
  <c' bes'>
  <c'\2 bes'>
  \set maximumFretStretch = 5
  <c' bes'>
  <c'\2 bes'\1>
}

<<
  \new FretBoards \mus
  \new TabVoice \mus
>>
```

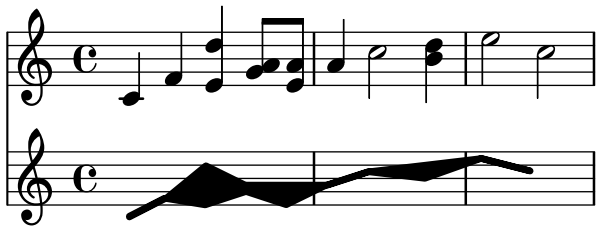


Clusters

Un cluster indique à l'instrumentiste qu'il doit jouer toute une suite de sons.

```
fragment = \relative c' {
  c4 f <e d'>4
  <g a>8 <e a> a4 c2 <d b>4
  e2 c
}

<<
  \new Staff \fragment
  \new Staff \makeClusters \fragment
>>
```

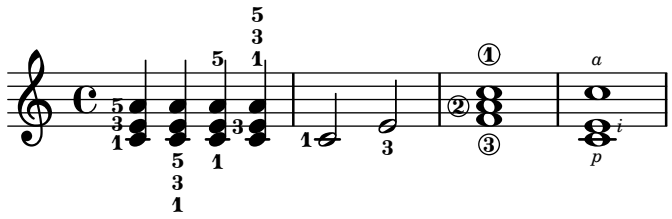


Contrôler la position des doigtés dans un accord

Le positionnement des doigtés peut être contrôlé de manière très précise. Afin que l'orientation soit prise en compte, il est nécessaire d'utiliser une syntaxe d'accord `< >`, même s'il ne s'agit que d'une seule note. Le positionnement des numéros de corde et doigtés main droite se règle de manière analogue.

```
\relative c' {
  \set fingeringOrientations = #'(left)
```

```
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger #1 e\rightHandFinger #2 c'\rightHandFinger #4 >
}
```

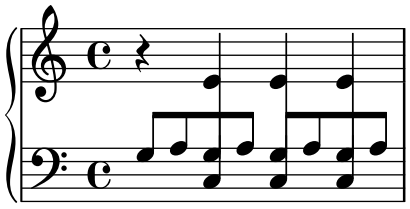


Accord distribué et problème de hampe - solution

Il est parfois préférable d'utiliser les hampes de la portée supérieure pour créer des accords distribués, afin d'éviter tout risque de collision au niveau des ligatures automatiques. Dans l'exemple suivant, le fait de partir des hampes de la portée inférieure aurait rendu nécessaire l'adaptation du détecteur de collision des ligatures, par une clause `\override Staff.Beam.collision-voice-only = ##t`, afin qu'il ne tienne pas compte des collisions entre portées.

```
\new PianoStaff <<
  \new Staff = up
    \relative c' {
      <<
        { r4
          \override Stem.cross-staff = ##t
          \override Stem.length = #19 % this is in half-spaces,
              % so it makes stems 9.5 staffspaces long
          \override Stem.Y-offset = #-6 % stems are normally lengthened
              % upwards, so here we must lower the stem by the amount
              % equal to the lengthening - in this case (19 - 7) / 2
              % (7 is default stem length)
          e e e }
        { s4
          \change Staff = "bottom"
          \override NoteColumn.ignore-collision = ##t
          c, c c
        }
      >>
    }
```

```
  \new Staff = bottom
    \relative c' {
      \clef bass
      \voiceOne
      g8 a g a g a g a
    }
>>
```
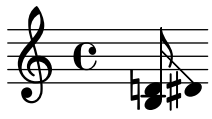


Impression d'accords complexes

Voici comment obtenir l'impression d'un accord au sein duquel une même note est jouée deux fois avec des altérations différentes.

```
fixA = {
  \once \override Stem.length = #11
}

fixB = {
  \once \override NoteHead.X-offset = #1.7
  \once \override Stem.length = #7
  \once \override Stem.rotation = #'(45 0 0)
  \once \override Stem.extra-offset = #'(-0.1 . -0.2)
  \once \override Flag.style = #'no-flag
  \once \override Accidental.extra-offset = #'(4 . -.1)
}


\relative c' {
  << { \fixA <b d!>8 } \\ { \voiceThree \fixB dis } >> s
}
```

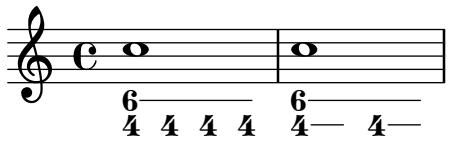


Interruption manuelle des prolongations de certains chiffrages

La basse chiffrée utilise fréquemment des lignes pour indiquer la prolongation jusqu'à un certain point. LilyPond est alors dispendieux en ce sens qu'il place le plus de prolongateurs possible. L'interruption d'un prolongateur particulier s'obtient en affectant d'un `\!` le chiffre qui doit être réimprimé ; ceci interrompra de fait la ligne de prolongation juste avant ce chiffre.

```
bassfigures = \figuremode {
  \set useBassFigureExtenders = ##t
  <6 4>4 <6 4\!> <6 4\!> <6 4\!> |  <6\! 4\!>  <6 4> <6 4\!> <6 4>
}
```

```
<<
  \new Staff \relative c'' { c1 c1 }
  \new FiguredBass \bassfigures
>>
```

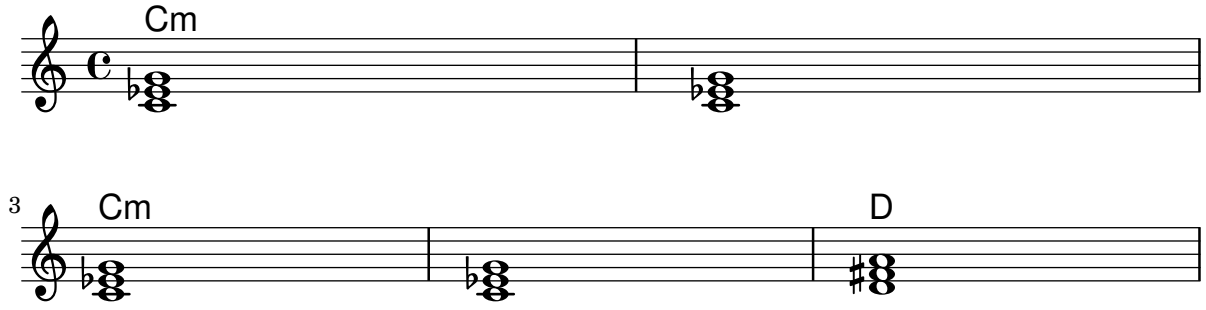


Impression des accords si changement

Vous pouvez faire ressortir les chiffrages d'accords s'ils ne sont imprimés qu'aux changements d'accord ou en début de ligne.

```
harmonies = \chordmode {
  c1:m c:m \break c:m c:m d
}

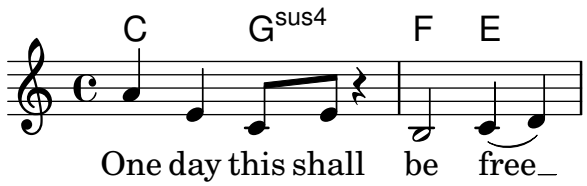
<<
  \new ChordNames {
    \set chordChanges = ##t
    \harmonies
  }
  \new Staff {
    \relative c' { \harmonies }
  }
>>
```



Chanson simple

Assembler des noms d'accords, une mélodie et des paroles permet d'obtenir la partition d'une chanson :

```
<<
  \chords { c2 g:sus4 f e }
  \new Staff \relative c'' {
    a4 e c8 e r4
    b2 c4( d)
  }
  \addlyrics { One day this shall be free __ }
>>
```



Paroles, musique et accords

Ce canevas comporte tous les éléments d'une chanson : la mélodie, les paroles, les accords.

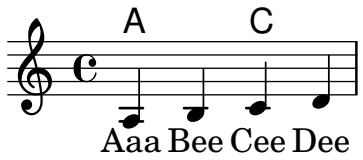
```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

harmonies = \chordmode {
  a2 c
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Voice = "one" { \autoBeamOff \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}
```



Paroles, musique, accords et diagrammes de fret

Ce canevas comporte, en plus de la mélodie, des paroles et des accords, les diagrammes de fret.

```
verseI = \lyricmode {
  \set stanza = #"1."
  This is the first verse
}

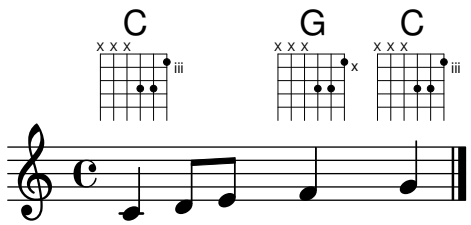
verseII = \lyricmode {
  \set stanza = #"2."
  This is the second verse.
```

```
}

theChords = \chordmode {
  % insert chords for chordnames and fretboards here
  c2 g4 c
}

staffMelody = \relative c' {
   \key c \major
   \clef treble
   % Type notes for melody here
   c4 d8 e f4 g
   \bar "|."
}

\score {
  <<
    \context ChordNames { \theChords }
    \context FretBoards { \theChords }
    \new Staff {
      \context Voice = "voiceMelody" { \staffMelody }
    }
    \new Lyrics = "lyricsI" {
      \lyricsto "voiceMelody" \verseI
    }
    \new Lyrics = "lyricsII" {
      \lyricsto "voiceMelody" \verseII
    }
  >>
  \layout { }
  \midi { }
}
```



1. This is the first verse
2. This is the second verse.

Mélodie simple et accords

Vous avez besoin de la partition d'une mélodie avec les accords ? N'allez pas plus loin !
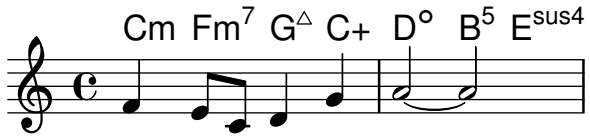
```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  f4 e8[ c] d4 g
  a2 ~ a
```

```
}

harmonies = \chordmode {
  c4:m f:min7 g:maj c:aug
  d2:dim b4:5 e:sus
}

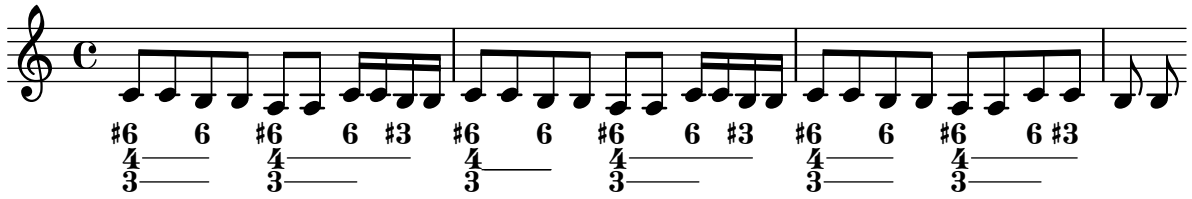
\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Staff \melody
  >>
  \layout{ }
  \midi { }
}
```



Prolongateur commun de basse figurée

L'activation de la propriété `useBassFigureExtenders` permet d'afficher des lignes de prolongation pour les chiffres qui se répètent. Deux chiffres prolongés sur la même durée se verront affublés d'un unique prolongateur, verticalement centré entre eux, dès lors que la propriété `figuredBassCenterContinuations` aura elle aussi été activée.

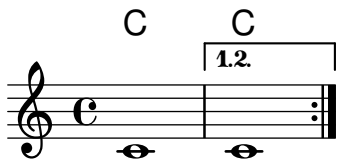
```
<<
  \relative c' {
    c8 c b b a a c16 c b b
    c8 c b b a a c16 c b b
    c8 c b b a a c c b b
  }
  \figures {
    \set useBassFigureExtenders = ##t
    <6+ 4 3>4 <6 4 3>8 r
    <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
    \set figuredBassCenterContinuations = ##t
    <6+ 4 3>4 <6 4 3>8 r
    <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
    \set figuredBassCenterContinuations = ##f
    <6+ 4 3>4 <6 4 3>8 r
    <6+ 4 3>4 <6 4 3>8 <4 3+>8
  }
>>
```



Crochet de reprise sous les chiffrages d'accord

L'ajout du `Volta_engraver` à la bonne portée permet d'imprimer les crochets de reprise entre les chiffrages et la portée.

```
\score {
  <<
    \chords {
      c1
      c1
    }
    \new Staff \with {
      \consists "Volta_engraver"
    }
    {
      \repeat volta 2 { c'1 }
      \alternative { c' }
    }
  >>
  \layout {
    \context {
      \Score
      \remove "Volta_engraver"
    }
  }
}
```



Keyboards

Section “Keyboard and other multi-staff instruments” dans *Manuel de notation*

Symboles de chant pour accordéon

Cet extrait de code est rendu obsolète par l'apparition de commandes de *markup* prédéfinies. Il reste néanmoins utile à titre d'exemple sur la manière de combiner des symboles : le positionnement de tels symboles, ajoutés sous forme de *markups*, s'ajuste à l'aide des arguments de `\translate-scaled`. La fonction `\translate-scaled` est ici préférée à `\translate` parce qu'elle permet au positionnement des parties symboliques de s'adapter aux changements de `font-size`.

```
discant = \markup {
  \musicglyph "accordion.discant"
}
dot = \markup {
  \musicglyph "accordion.dot"
}

\layout { ragged-right = ##t }

% 16 voets register
accBasson = ^\markup {
  \combine
  \discant
  \translate-scaled #'(0 . 0.5) \dot
}

% een korig 8 en 16 voets register
accBandon = ^\markup {
  \combine
    \discant
    \combine
      \translate-scaled #'(0 . 0.5) \dot
      \translate-scaled #'(0 . 1.5) \dot
}

accVCello = ^\markup {
  \combine
    \discant
    \combine
      \translate-scaled #'(0 . 0.5) \dot
      \combine
        \translate-scaled #'(0 . 1.5) \dot
        \translate-scaled #'(1 . 1.5) \dot
}

% 4-8-16 voets register
accHarmon = ^\markup {
  \combine
    \discant
    \combine
```

```
      \translate-scaled #'(0 . 0.5) \dot
      \combine
        \translate-scaled #'(0 . 1.5) \dot
        \translate-scaled #'(0 . 2.5) \dot
}

accTrombon = ^\markup {
  \combine
    \discant
    \combine
      \translate-scaled #'(0 . 0.5) \dot
      \combine
        \translate-scaled #'(0 . 1.5) \dot
        \combine
          \translate-scaled #'(1 . 1.5) \dot
          \translate-scaled #'(-1 . 1.5) \dot
}

% eenkorig 4 en 16 voets register
accOrgan = ^\markup {
  \combine
    \discant
    \combine
      \translate-scaled #'(0 . 0.5) \dot
      \translate-scaled #'(0 . 2.5) \dot
}

accMaster = ^\markup {
  \combine
    \discant
    \combine
      \translate-scaled #'(0 . 0.5) \dot
      \combine
        \translate-scaled #'(0 . 1.5) \dot
        \combine
          \translate-scaled #'(1 . 1.5) \dot
          \combine
            \translate-scaled #'(-1 . 1.5) \dot
            \translate-scaled #'(0 . 2.5) \dot
}

accAccord = ^\markup {
  \combine
    \discant
    \combine
      \translate-scaled #'(0 . 1.5) \dot
      \combine
        \translate-scaled #'(1 . 1.5) \dot
        \combine
          \translate-scaled #'(-1 . 1.5) \dot
          \translate-scaled #'(0 . 2.5) \dot
}
```

```
accMusette = ^\markup {
  \combine
    \discant
    \combine
      \translate-scaled #'(0 . 1.5) \dot
      \combine
        \translate-scaled #'(1 . 1.5) \dot
        \translate-scaled #'(-1 . 1.5) \dot
}

accCeleste = ^\markup {
  \combine
    \discant
    \combine
      \translate-scaled #'(0 . 1.5) \dot
      \translate-scaled #'(-1 . 1.5) \dot
}

accOboe = ^\markup {
  \combine
    \discant
    \combine
      \translate-scaled #'(0 . 1.5) \dot
      \translate-scaled #'(0 . 2.5) \dot
}

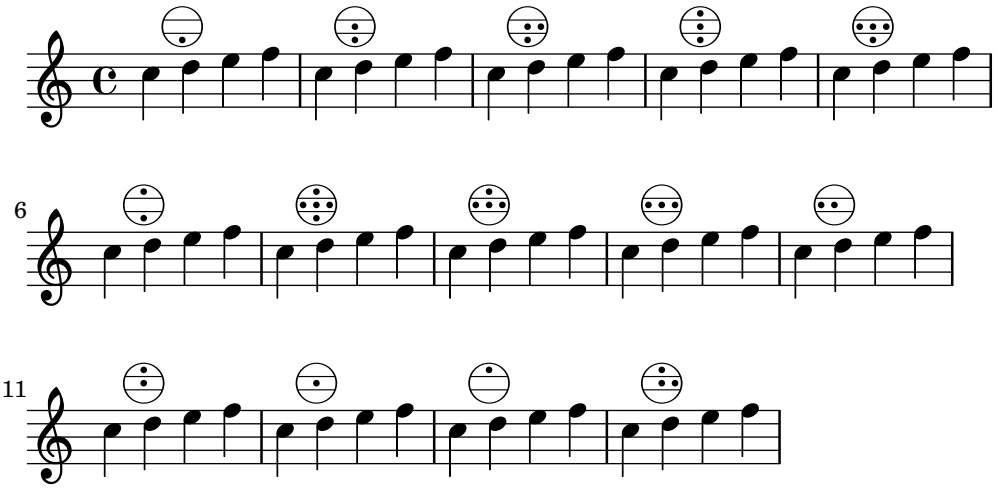
accClarin = ^\markup {
  \combine
    \discant
    \translate-scaled #'(0 . 1.5) \dot
}

accPiccolo = ^\markup {
    \combine
       \discant
       \translate-scaled #'(0 . 2.5) \dot
}

accViolin = ^\markup {
  \combine
    \discant
    \combine
      \translate-scaled #'(0 . 1.5) \dot
      \combine
        \translate-scaled #'(1 . 1.5) \dot
        \translate-scaled #'(0 . 2.5) \dot
}

\relative c'' {
  c4 d\accBasson e f
  c4 d\accBandon e f
```

```
  c4 d\accVCello e f
  c4 d\accHarmon e f
  c4 d\accTrombon e f
  \break
  c4 d\accOrgan e f
  c4 d\accMaster e f
  c4 d\accAccord e f
  c4 d\accMusette e f
  c4 d\accCeleste e f
  \break
  c4 d\accOboe e f
  c4 d\accClarin e f
  c4 d\accPiccolo e f
  c4 d\accViolin e f
}
```



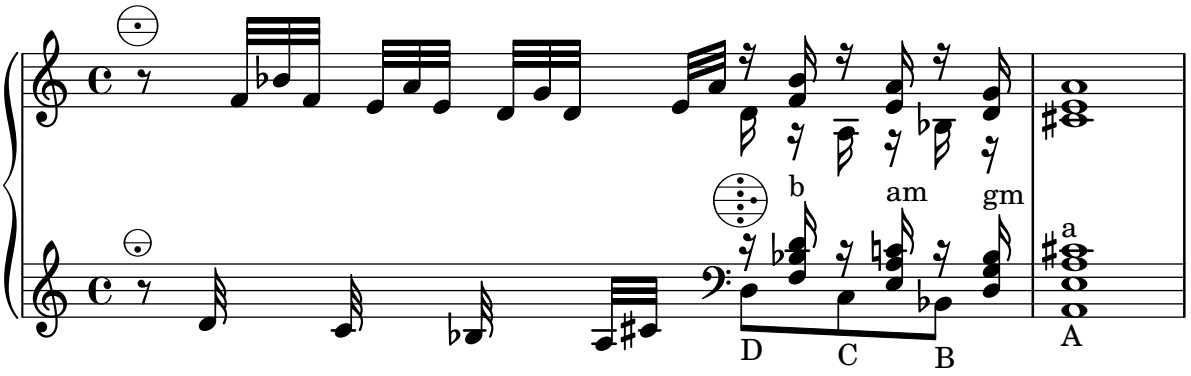
Symboles de registre pour accordéon

Les symboles spécifiques aux registres d'accordéon sont disponible tant sous forme de `\markup` que d'événements musicaux indépendants – un changement de registre a tendance à intervenir entre des événements musicaux réels. Les registres de basse ne sont pas très standardisés. Les différentes commandes disponibles sont regroupées à l'annexe Section "Registres d'accordéon" dans *Manuel de notation*.

```
      \\
      { d r a r bes r }
    >> |
    <cis e a>1
  }

  \new Staff \relative {
    \clef treble
    \freeBass "1"
    r8 d'32 s16. c32 s16. bes32 s16. a32[ cis] s16
    \clef bass \stdBass "Master"
    <<
      { r16 <f, bes d>^"b" r <e a c>^"am" r <d g bes>^"gm" |
      <e a cis>1^"a" }
```

```
      \\
      { d8_"D" c_"C" bes_"B" | a1_"A" }
    >>
  }
>>
```
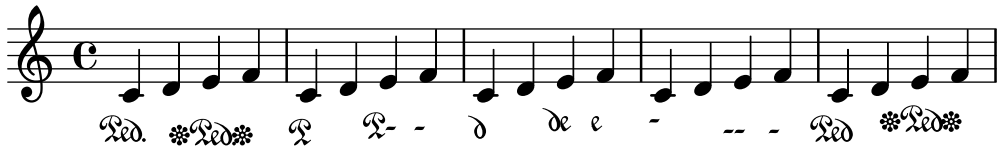


Modification du texte des indications de pédale

`Staff.pedalSustainStrings` permet de définir le texte affiché pour les instructions de pédale. Les seuls caractères autorisés sont les glyphes particuliers de pédale, comme vous pouvez le constater ici.

```
sustainNotes = { c4\sustainOn d e\sustainOff\sustainOn f\sustainOff }

\relative c' {
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("P" "P-" "-")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("d" "de" "e")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("M" "M-" "-")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("Ped" "*Ped" "*")
  \sustainNotes
}
```



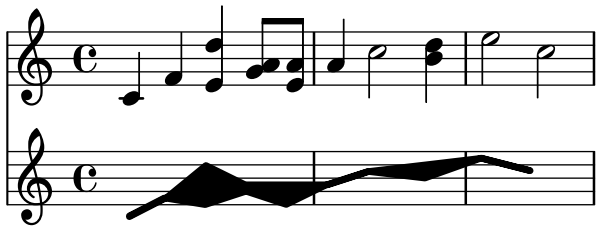
Clusters

Un cluster indique à l'instrumentiste qu'il doit jouer toute une suite de sons.

```
fragment = \relative c' {
  c4 f <e d'>4
  <g a>8 <e a> a4 c2 <d b>4
  e2 c
}

<<
```

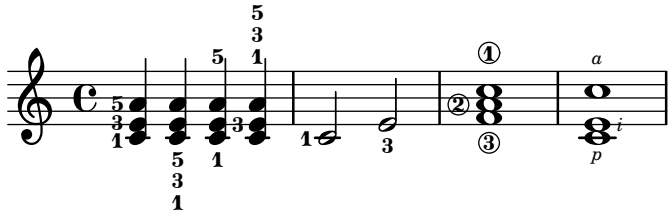
```
  \new Staff \fragment
  \new Staff \makeClusters \fragment
>>
```



Contrôler la position des doigtés dans un accord

Le positionnement des doigtés peut être contrôlé de manière très précise. Afin que l'orientation soit prise en compte, il est nécessaire d'utiliser une syntaxe d'accord `< >`, même s'il ne s'agit que d'une seule note. Le positionnement des numéros de corde et doigtés main droite se règle de manière analogue.

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger #1 e\rightHandFinger #2 c'\rightHandFinger #4 >
}
```
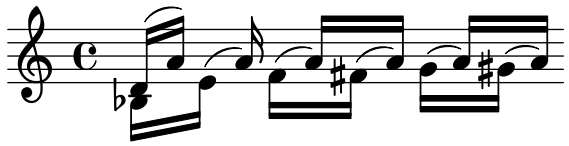


Création d'une liaison entre plusieurs voix

Dans certaines situations, il peut être nécessaire de lier des notes appartenant à des voix différentes. La solution consiste à ajouter une note invisible dans l'une des voix, à l'aide de la commande `\hideNotes`.

Voici par exemple la mesure 235 de la chaconne de la seconde partita pour violon solo, BWV 1004, de Bach.

```
\relative c' {
```

```
  <<
    {
      d16( a') s a s a[ s a] s a[ s a]
    }
    \\
    {
      \slurUp
      bes,16[ s e](
      \hideNotes a)
      \unHideNotes f[(
      \hideNotes a)
      \unHideNotes fis](
      \hideNotes a)
      \unHideNotes g[(
      \hideNotes a)
      \unHideNotes gis](
      \hideNotes a)
    }
  >>
}
```

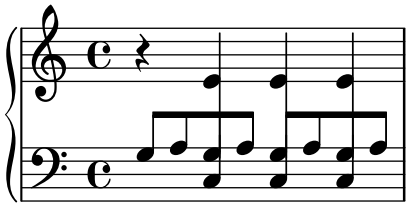


Accord distribué et problème de hampe - solution

Il est parfois préférable d'utiliser les hampes de la portée supérieure pour créer des accords distribués, afin d'éviter tout risque de collision au niveau des ligatures automatiques. Dans l'exemple suivant, le fait de partir des hampes de la portée inférieure aurait rendu nécessaire l'adaptation du détecteur de collision des ligatures, par une clause `\override Staff.Beam.collision-voice-only = ##t`, afin qu'il ne tienne pas compte des collisions entre portées.

```
\new PianoStaff <<
  \new Staff = up
    \relative c' {
      <<
        { r4
          \override Stem.cross-staff = ##t
          \override Stem.length = #19 % this is in half-spaces,
              % so it makes stems 9.5 staffspaces long
          \override Stem.Y-offset = #-6 % stems are normally lengthened
              % upwards, so here we must lower the stem by the amount
              % equal to the lengthening - in this case (19 - 7) / 2
              % (7 is default stem length)
          e e e }
        { s4
          \change Staff = "bottom"
          \override NoteColumn.ignore-collision = ##t
          c, c c
        }
```
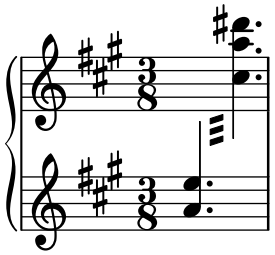
```
      >>
    }
  \new Staff = bottom
    \relative c' {
      \clef bass
      \voiceOne
      g8 a g a g a g a
    }
>>
```



Trémolo et changement de portée

Dans la mesure où `\repeat tremolo` requiert deux arguments musicaux pour un trémolo d'accords, la note ou l'accord de la portée opposée doit être encadré par des accolades et se voir adjoindre la commande `\change Staff`.
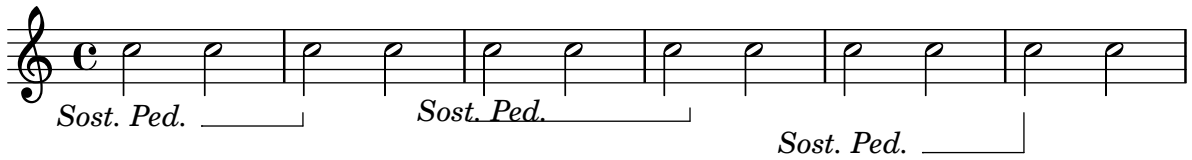
```
\new PianoStaff <<
  \new Staff = "up" \relative c'' {
    \key a \major
    \time 3/8
    s4.
  }
  \new Staff = "down" \relative c'' {
    \key a \major
    \time 3/8
    \voiceOne
    \repeat tremolo 6 {
      <a e'>32
      {
        \change Staff = "up"
        \voiceTwo
        <cis a' dis>32
      }
    }
  }
>>
```



Affinage des indications de pédale

Les crochets d'indication de pédale peuvent se modifier de différentes manières.

```
\paper { ragged-right = ##f }
\relative c'' {
  c2\sostenutoOn c
  c2\sostenutoOff c
  \once \override Staff.PianoPedalBracket.shorten-pair = #'(-7 . -2)
  c2\sostenutoOn c
  c2\sostenutoOff c
  \once \override Staff.PianoPedalBracket.edge-height = #'(0 . 3)
  c2\sostenutoOn c
  c2\sostenutoOff c
}
```



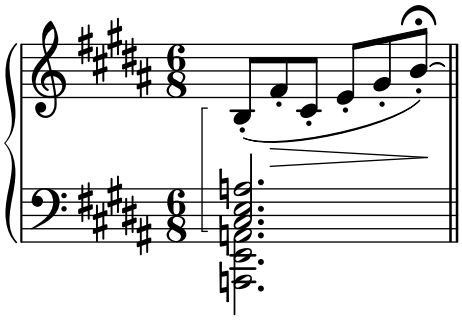
Indication d'un accord à cheval sur deux portées par un crochet d'arpège

Un crochet d'arpège peut indiquer que des notes réparties sur deux portées différentes doivent être jouées par la même main. Le contexte `PianoStaff` doit accepter ces arpèges « distribués », et les indications d'arpège du contexte `PianoStaff` adopter une allure de crochet.

(Debussy, Les collines d'Anacapri, mesure 65)

```
\new PianoStaff <<
  \set PianoStaff.connectArpeggios = ##t
  \override PianoStaff.Arpeggio.stencil =
    #ly:arpeggio::brew-chord-bracket
  \new Staff {
    \relative c' {
      \key b \major
      \time 6/8
      b8-.(\arpeggio fis'-.\> cis-.
        e-. gis-. b-.)\!\fermata^\laissezVibrer \bar "||"
    }
  }
  \new Staff {
    \relative c' {
      \clef bass
      \key b \major
      <<
        {
          <a e cis>2.\arpeggio
        }
        \\
        {
          <a, e a,>2.
        }
      >>
```

```
    }
  }
>>
```



Symboles de jazz

Bien que compliqué de prime abord, voici un canevas tout à fait indiqué pour les ensembles de jazz. Vous noterez que tous les instruments sont notés en ut (`\key c \major`), la tonalité de concert. Les notes seront automatiquement transposées dès lors qu'elles seront inscrites dans une section `\transpose`.

```
\header {
  title = "Song"
  subtitle = "(tune)"
  composer = "Me"
  meter = "moderato"
  piece = "Swing"
  tagline = \markup {
    \column {
      "LilyPond example file by Amelie Zapf,"
      "Berlin 07/07/2003"
    }
  }
}
% To make the example display in the documentation
\paper {
  paper-width = 130
}
%#(set-global-staff-size 16)
\include "english.ly"

%%%%%%%%%%% Some macros %%%%%%%%%%%%%%%%%%

sl = {
  \override NoteHead.style = #'slash
  \hide Stem
}
nsl = {
  \revert NoteHead.style
  \undo \hide Stem
}
crOn = \override NoteHead.style = #'cross
crOff = \revert NoteHead.style
```

```
%% insert chord name style stuff here.

jazzChords = { }

%%%%%%%%%%%% Keys'n'thangs %%%%%%%%%%%%%%%%

global = { \time 4/4 }

Key = { \key c \major }

% ############ Horns ############

% ------ Trumpet ------
trpt = \transpose c d \relative c'' {
  \Key
  c1 | c | c |
}
trpHarmony = \transpose c' d {
  \jazzChords
}
trumpet = {
  \global
  \clef treble
  <<
    \trpt
  >>
}

% ------ Alto Saxophone ------
alto = \transpose c a \relative c' {
  \Key
  c1 | c | c |
}
altoHarmony = \transpose c' a {
  \jazzChords
}
altoSax = {
  \global
  \clef treble
  <<
    \alto
  >>
}

% ------ Baritone Saxophone ------
bari = \transpose c a' \relative c {
  \Key
  c1
  c1
  \sl
  d4^"Solo" d d d
```

```
  \nsl
}
bariHarmony = \transpose c' a \chordmode {
  \jazzChords s1 s d2:maj e:m7
}
bariSax = {
  \global
  \clef treble
  <<
    \bari
  >>
}

% ------ Trombone ------
tbone = \relative c {
  \Key
  c1 | c | c
}
tboneHarmony = \chordmode {
  \jazzChords
}
trombone = {
  \global
  \clef bass
  <<
    \tbone
  >>
}

% ############ Rhythm Section ##############

% ------ Guitar ------
gtr = \relative c'' {
  \Key
  c1
  \sl
  b4 b b b
  \nsl
  c1
}
gtrHarmony = \chordmode {
  \jazzChords
  s1 c2:min7+ d2:maj9
}
guitar = {
  \global
  \clef treble
  <<
    \gtr
  >>
}
```

```
%% ------ Piano ------
rhUpper = \relative c'' {
  \voiceOne
  \Key
  c1 | c | c
}
rhLower = \relative c' {
  \voiceTwo
  \Key
  e1 | e | e
}

lhUpper = \relative c' {
  \voiceOne
  \Key
  g1 | g | g
}
lhLower = \relative c {
  \voiceTwo
  \Key
  c1 | c | c
}

PianoRH = {
  \clef treble
  \global
  <<
    \new Voice = "one" \rhUpper
    \new Voice = "two" \rhLower
  >>
}
PianoLH = {
  \clef bass
  \global
  <<
    \new Voice = "one" \lhUpper
    \new Voice = "two" \lhLower
  >>
}

piano = {
  <<
    \new Staff = "upper" \PianoRH
    \new Staff = "lower" \PianoLH
  >>
}

% ------ Bass Guitar ------
Bass = \relative c {
  \Key
  c1 | c | c
}
```

```
bass = {
  \global
  \clef bass
  <<
    \Bass
  >>
}

% ------ Drums ------
up = \drummode {
  \voiceOne
  hh4 <hh sn> hh <hh sn>
  hh4 <hh sn> hh <hh sn>
  hh4 <hh sn> hh <hh sn>
}
down = \drummode {
  \voiceTwo
  bd4 s bd s
  bd4 s bd s
  bd4 s bd s
}

drumContents = {
  \global
  <<
    \new DrumVoice \up
    \new DrumVoice \down
  >>
}

%%%%%%%% It All Goes Together Here %%%%%%%%%%%%%%%%%%%%

\score {
  <<
    \new StaffGroup = "horns" <<
      \new Staff = "trumpet" \with { instrumentName = "Trumpet" }
      \trumpet
      \new Staff = "altosax" \with { instrumentName = "Alto Sax" }
      \altoSax
      \new ChordNames = "barichords" \with { instrumentName = "Trumpet" }
      \bariHarmony
      \new Staff = "barisax" \with { instrumentName = "Bari Sax" }
      \bariSax
      \new Staff = "trombone" \with { instrumentName = "Trombone" }
      \trombone
    >>

    \new StaffGroup = "rhythm" <<
      \new ChordNames = "chords" \gtrHarmony
      \new Staff = "guitar" \with { instrumentName = "Guitar" }
      \guitar
      \new PianoStaff = "piano" \with {
```

```
        instrumentName = "Piano"
        midiInstrument = "acoustic grand"
      }
      \piano
      \new Staff = "bass" \with { instrumentName = "Bass" }
      \bass
      \new DrumStaff \with { instrumentName = "Drums" }
      \drumContents
    >>
  >>
  \layout {
    \context { \Staff \RemoveEmptyStaves }
    \context {
      \Score
      \override BarNumber.padding = #3
      \override RehearsalMark.padding = #2
      skipBars = ##t
    }
  }
  \midi { }
}
```

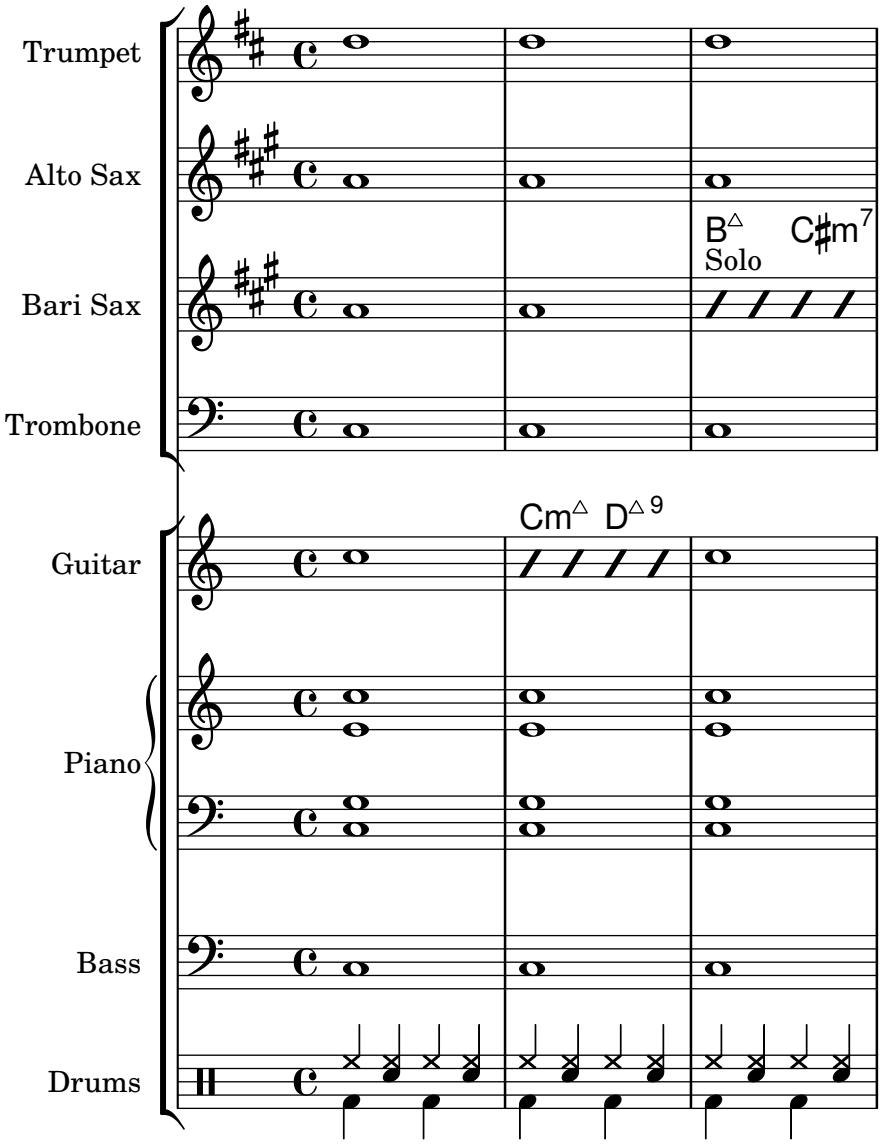
Song

(tune)

Me

moderato

Swing
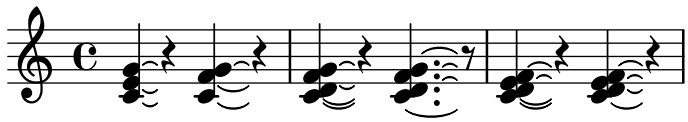


Liaison « Laissez vibrer »

Les indications de « laisser vibrer » ont une taille fixe. Leur formatage est accessible au travers de la propriété `tie-configuration`.

```
\relative c' {
  <c e g>4\laissezVibrer r <c f g>\laissezVibrer r
  <c d f g>4\laissezVibrer r <c d f g>4.\laissezVibrer r8

  <c d e f>4\laissezVibrer r
  \override LaissezVibrerTieColumn.tie-configuration
     = #`((-7 . ,DOWN)
          (-5 . ,DOWN)
          (-3 . ,UP)
          (-1 . ,UP))
  <c d e f>4\laissezVibrer r
}
```



Piano – cannevas simple

Voici une simple partition pour piano avec quelques notes.

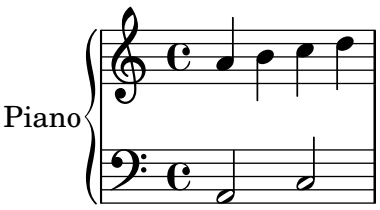
```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

\score {
  \new PianoStaff \with { instrumentName = "Piano" }
  <<
    \new Staff = "upper" \upper
    \new Staff = "lower" \lower
  >>
  \layout { }
  \midi { }
}
```



Piano et paroles entre les portées

Lorsque la mélodie est doublée au piano, cela ne nécessite pas forcément une portée spécifique. Les paroles peuvent s'insérer entre les deux portées de la partition pour piano.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

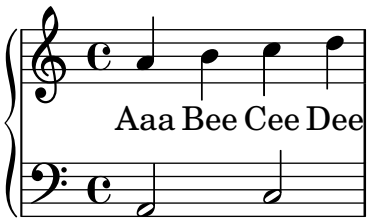
  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4
```

```
  a2 c
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

\score {
  \new PianoStaff <<
    \new Staff = upper { \new Voice = "singer" \upper }
    \new Lyrics \lyricsto "singer" \text
    \new Staff = lower { \lower }
  >>
  \layout { }
  \midi { }
}
```



Piano, mélodie et paroles

Il s'agit du format classique pour le chant : une portée pour la mélodie et les paroles au-dessus de l'accompagnement au piano.

```
melody = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

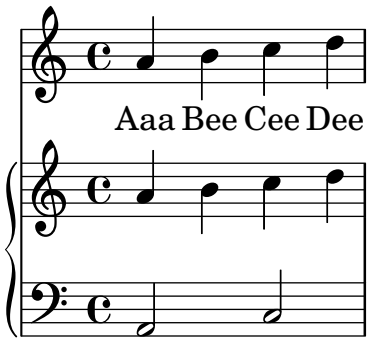
  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
```

```
  \time 4/4

  a2 c
}

\score {
  <<
    \new Voice = "mel" { \autoBeamOff \melody }
    \new Lyrics \lyricsto mel \text
    \new PianoStaff <<
      \new Staff = "upper" \upper
      \new Staff = "lower" \lower
    >>
  >>
  \layout {
    \context { \Staff \RemoveEmptyStaves }
  }
  \midi { }
}
```



Retrait de l'accolade à la première ligne d'une pièce pour piano

Dans cet extrait est supprimée la première accolade d'un `PianoStaff` ou d'un `GrandStaff`.

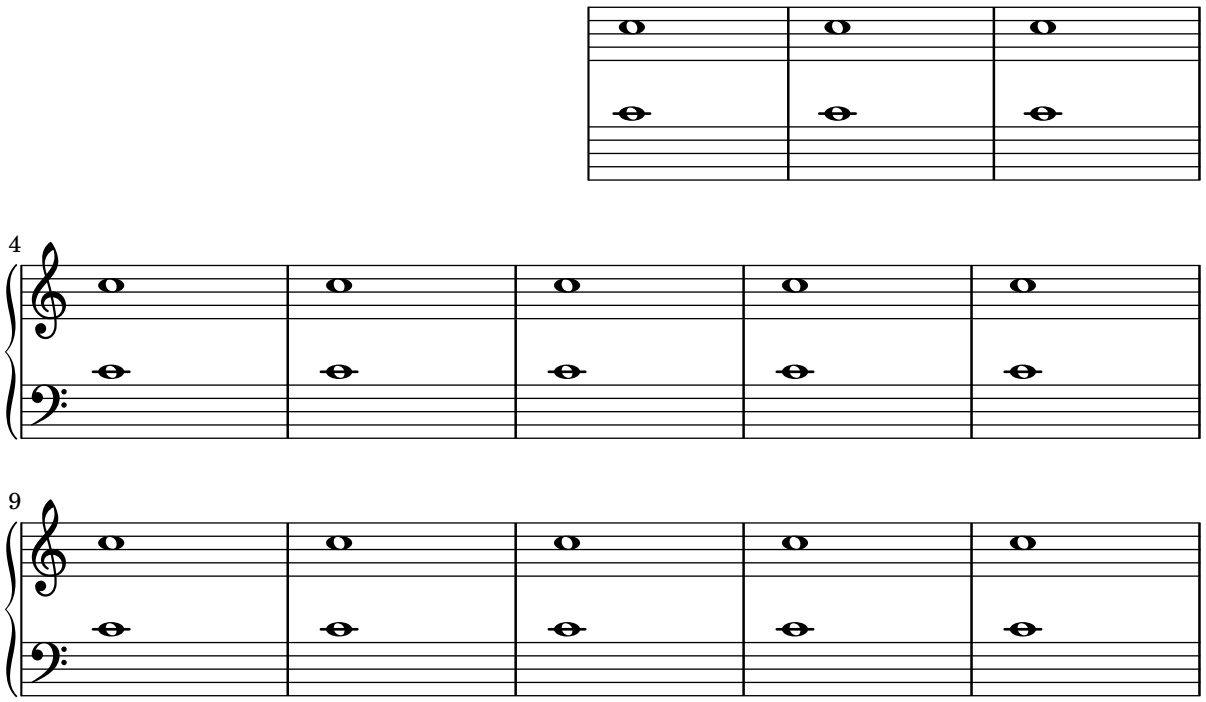
Ceci peut s'avérer utile pour couper et coller l'image générée dans de la musique préexistante.

Est utilisée la fonction `\alterBroken`.

```
someMusic =  {
  \once \override Staff.Clef.stencil = ##f
  \once \override Staff.TimeSignature.stencil = ##f
  \repeat unfold 3 c1 \break
  \repeat unfold 5 c1 \break
  \repeat unfold 5 c1
}

\score {
  \new PianoStaff
  <<
    \new Staff = "right" \relative c'' \someMusic
    \new Staff = "left" \relative c' { \clef F \someMusic }
  >>
  \layout {
    indent=75
```

```
    \context {
      \PianoStaff
      \alterBroken transparent #'(#t) SystemStartBrace
    }
  }
}
```



Utilisation d'autoChange avec plus d'une voix

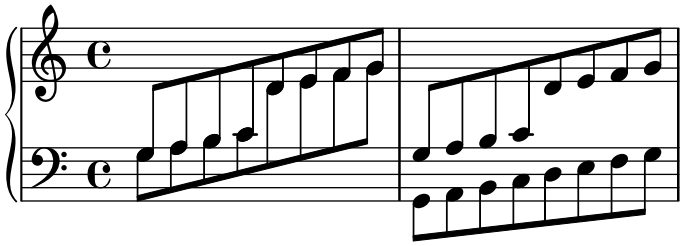
L'instruction `\autoChange` est opérationnelle y compris en présence de voix multiples.

```
\score
{
  \new PianoStaff
  <<
    \new Staff = "up" {
      <<
        \set Timing.beamExceptions = #'()
        \set Timing.beatStructure = #'(4)
        \new Voice {
          \voiceOne
          \autoChange
          \relative c' {
            g8 a b c d e f g
            g,8 a b c d e f g
          }
        }

        \new Voice {
          \voiceTwo
          \autoChange
          \relative c' {
```

```
            g8 a b c d e f g
            g,,8 a b c d e f g
          }
        }
      >>
    }

    \new Staff = "down" {
      \clef bass
    }
  >>
}
```



Ensemble vocal avec réduction pour piano

Ce canevas ajoute une réduction pour piano à une partition standard pour chœur à quatre voix mixtes. Ceci illustre l'un des avantages de LilyPond : une expression musicale peut être réutilisée sans effort. Toute modification apportée à l'une des voix, mettons `tenorMusique`, sera automatiquement reportée dans la réduction pour piano.

```
\paper {
  top-system-spacing.basic-distance = #10
  score-system-spacing.basic-distance = #20
  system-system-spacing.basic-distance = #20
  last-bottom-spacing.basic-distance = #10
}

global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative {
  c''4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

altoMusic = \relative {
  e'4 f d e
}
altoWords =\lyricmode {
  ha ha ha ha
}
```

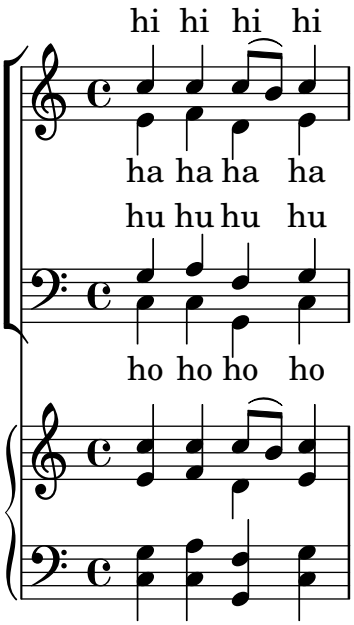
```
tenorMusic = \relative {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  <<
    \new ChoirStaff <<
      \new Lyrics = "sopranos" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }
      \new Staff = "women" <<
        \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
        \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
      >>
      \new Lyrics = "altos"
      \new Lyrics = "tenors" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }

      \new Staff = "men" <<
        \clef bass
        \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
        \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
      >>
      \new Lyrics = "basses"
      \context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
      \context Lyrics = "altos" \lyricsto "altos" \altoWords
      \context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
      \context Lyrics = "basses" \lyricsto "basses" \bassWords
    >>
    \new PianoStaff <<
      \new Staff <<
        \set Staff.printPartCombineTexts = ##f
        \partCombine
        << \global \sopMusic >>
        << \global \altoMusic >>
      >>
      \new Staff <<
        \clef bass
```

```
        \set Staff.printPartCombineTexts = ##f
        \partCombine
        << \global \tenorMusic >>
        << \global \bassMusic >>
      >>
    >>
  >>
}
```



Percussion

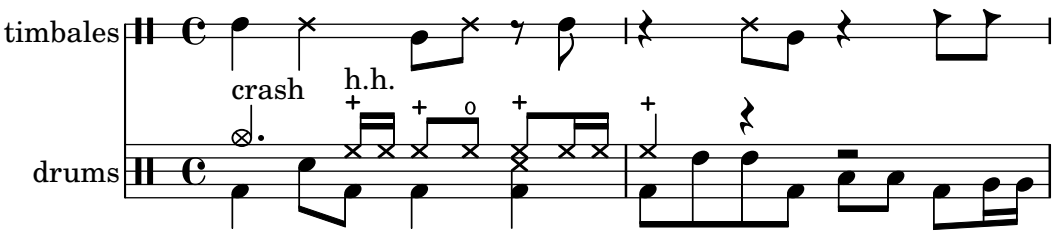
Section “Percussion” dans *Manuel de notation*

Ajout de parties de batterie

Grâce à la puissance des outils préconfigurés tels que la fonction `\drummode` et le contexte `DrumStaff`, la saisie de parties pour percussions est extrêmement simplifiée : chaque composant d'une batterie trouve sa place sur une portée dédiée (avec une clef particulière) et les têtes de note sont spécifiques à chaque élément. Il est également possible d'affecter un symbole particulier à chaque élément, tout comme de restreindre le nombre de lignes de la portée.

```
drh = \drummode {
        cymc4.^"crash" hhc16^"h.h." hh hhc8 hho hhc8 hh16 hh
        hhc4 r4 r2
      }
drl = \drummode {
        bd4 sn8 bd bd4 << bd ss >>
        bd8 tommh tommh bd toml toml bd tomfh16 tomfh
      }
timb = \drummode {
         timh4 ssh timl8 ssh r timh r4
         ssh8 timl r4 cb8 cb
       }

\score {
  <<
    \new DrumStaff \with {
      instrumentName = "timbales"
      drumStyleTable = #timbales-style
      \override StaffSymbol.line-count = #2
      \override BarLine.bar-extent = #'(-1 . 1)
    }
    <<
      \timb
    >>
    \new DrumStaff \with { instrumentName = "drums" }
    <<
      \new DrumVoice { \stemUp \drh }
      \new DrumVoice { \stemDown \drl }
    >>
  >>
  \layout { }
  \midi { \tempo 4 = 120 }
}
```



Personnalisation de batterie pour les sorties imprimable et MIDI

Des noms de « hauteurs » de batterie personnalisée peuvent s'utiliser à la fois pour la sortie imprimable et le fichier MIDI généré. Ces variables doivent être définies en tant que `drumPitchNames`, `drumStyleTable` et `midiDrumPitches`, comme indiqué ci-dessous. En résumé, cet extrait

– définit quelques noms de « hauteur »,
– définit comment ils seront rendus,
– indique à LilyPond de les utiliser pour la mise en forme,
– assigne les hauteurs aux noms,
– indique à LilyPond de les utiliser pour le rendu MIDI.

```
%% This snippet tries to amend
%% NR 2.5.1 Common notation for percussion - Custom percussion staves
%% http://lilypond.org/doc/v2.18/Documentation/notation/common-notation-for-percussion#custo

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% To use custom drum-pitch-names for your score and midi you need to follow
%% this route:
%%
%%%%%%%%%%%%
%% LAYOUT:
%%%%%%%%%%%%
%%
%% (1) Define a name and put it in `drumPitchNames'
%%     This can be done at toplevel with
%%         drumPitchNames.my-name  = #'my-name
%%     It's possible to add an alias as well.
%% (2) Define how it should be printed
%%     Therefore put them into a toplevel-list, where each entry should look:
%%         (my-name
%%           note-head-style-or-default
%%             articulation-string-or-#f
%%               staff-position)
%%     Example:
%%         #(define my-style
%%           '(
%%             (my-name default "tenuto" -1)
%%             ; ...
%%             ))
%% (3) Tell LilyPond to use this custom-definitions, with
%%         drumStyleTable = #(alist->hash-table my-style)
%%     in a \layout or \with
%%
%%     Now we're done for layout, here a short, but complete example:
%%         \new DrumStaff
%%           \with { drumStyleTable = #(alist->hash-table my-style) }
%%           \drummode { my-name }
%%
%%%%%%%%%%%%
%% MIDI:
%%%%%%%%%%%%
%%
```

```
%% (1) Again at toplvel, assign a pitch to your custom-note-name
%%         midiDrumPitches.my-name = ges
%%     Note that you have to use the name, which is in drumPitchNames, no alias
%% (2) Tell LilyPond to use this pitch(es), with
%%         drumPitchTable = #(alist->hash-table midiDrumPitches)
%%
%%     Example:
%%         \score {
%%            \new DrumStaff
%%             \with {
%%               drumStyleTable = #(alist->hash-table my-style)
%%               drumPitchTable = #(alist->hash-table midiDrumPitches)
%%             }
%%             \drummode { my-name4 }
%%           \layout {}
%%           \midi {}
%%         }
%%
%%%%%%%%%%%%
%% TESTING
%%%%%%%%%%%%
%%
%% To test whether all is fine, run the following sequence in terminal:
%%         lilypond my-file.ly
%%         midi2ly my-file.midi
%%         gedit my-file-midi.ly
%%
%% Which will do:
%% 1. create pdf and midi
%% 2. transform the midi back to a .ly-file
%%    (note: midi2ly is not always good in correctly identifying enharmonic pitches)
%% 3. open this file in gedit (or use another editor)
%% Now watch what you've got.
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% FULL EXAMPLE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
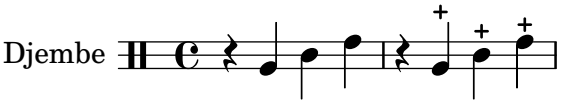
drumPitchNames.dbass      = #'dbass
drumPitchNames.dba        = #'dbass  % 'db is in use already
drumPitchNames.dbassmute  = #'dbassmute
drumPitchNames.dbm        = #'dbassmute
drumPitchNames.do         = #'dopen
drumPitchNames.dopenmute  = #'dopenmute
drumPitchNames.dom        = #'dopenmute
drumPitchNames.dslap      = #'dslap
drumPitchNames.ds         = #'dslap
drumPitchNames.dslapmute  = #'dslapmute
drumPitchNames.dsm        = #'dslapmute
```

```
#(define djembe
  '((dbass      default  #f         -2)
    (dbassmute  default  "stopped"  -2)
    (dopen      default  #f          0)
    (dopenmute  default  "stopped"   0)
    (dslap      default  #f          2)
    (dslapmute  default  "stopped"   2)))

midiDrumPitches.dbass = g
midiDrumPitches.dbassmute = fis
midiDrumPitches.dopen =  a
midiDrumPitches.dopenmute = gis
midiDrumPitches.dslap =  b
midiDrumPitches.dslapmute = ais

one = \drummode { r4 dba4 do ds r dbm dom dsm }

\score {
   \new DrumStaff
    \with {
      \override StaffSymbol.line-count =  #3
      instrumentName = #"Djembe "
      drumStyleTable = #(alist->hash-table djembe)
      drumPitchTable = #(alist->hash-table midiDrumPitches)
    }
    \one
  \layout {}
  \midi {}
}
```



Indication personnalisée d'une polymétrie complexe

Bien que la métrique complexe ne soit pas l'élément primordial de cet exemple, elle permet d'indiquer la pulsation de cette pièce qui, par ailleurs, constitue le canevas d'une chanson traditionnelle des Balkans.
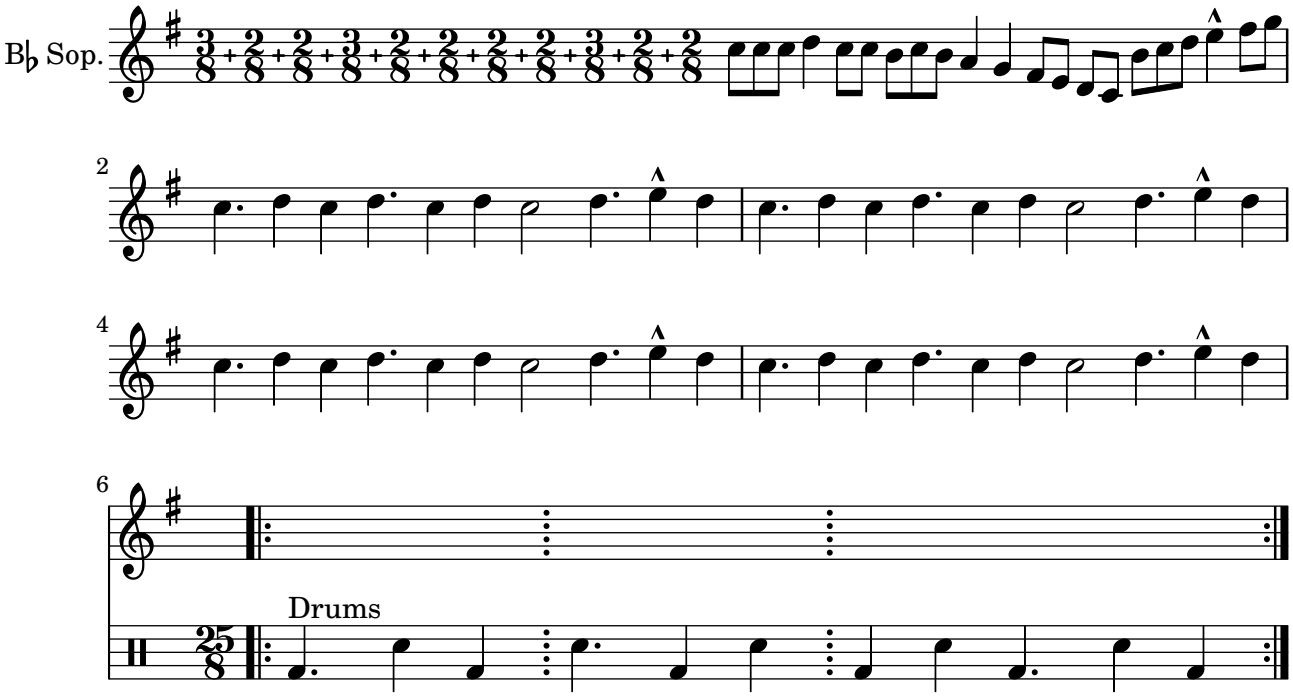
```
melody = \relative c'' {
  \key g \major
  \compoundMeter #'((3 8) (2 8) (2 8) (3 8) (2 8) (2 8)
                    (2 8) (2 8) (3 8) (2 8) (2 8))
  c8 c c d4 c8 c b c b a4 g fis8 e d c b' c d e4-^ fis8 g \break
  c,4. d4 c4 d4. c4 d c2 d4. e4-^ d4
  c4. d4 c4 d4. c4 d c2 d4. e4-^ d4 \break
  c4. d4 c4 d4. c4 d c2 d4. e4-^ d4
  c4. d4 c4 d4. c4 d c2 d4. e4-^ d4 \break
}

drum = \new DrumStaff \drummode {
  \bar ".|:" bd4.^\markup { Drums } sn4 bd \bar ";" sn4.
```

```
  bd4 sn \bar ";" bd sn bd4. sn4 bd \bar ":|."
}

  \new Staff \with {
    instrumentName = \markup { \concat { "B" \flat " Sop." } }
  }

{
  \melody
  \drum
}
```



Symboles de jazz

Bien que compliqué de prime abord, voici un canevas tout à fait indiqué pour les ensembles de jazz. Vous noterez que tous les instruments sont notés en ut (`\key c \major`), la tonalité de concert. Les notes seront automatiquement transposées dès lors qu'elles seront inscrites dans une section `\transpose`.

```
\header {
  title = "Song"
  subtitle = "(tune)"
  composer = "Me"
  meter = "moderato"
  piece = "Swing"
  tagline = \markup {
    \column {
      "LilyPond example file by Amelie Zapf,"
      "Berlin 07/07/2003"
    }
  }
}
```

```
% To make the example display in the documentation
\paper {
  paper-width = 130
}
%#(set-global-staff-size 16)
\include "english.ly"

%%%%%%%%%%% Some macros %%%%%%%%%%%%%%%%%%

sl = {
  \override NoteHead.style = #'slash
  \hide Stem
}
nsl = {
  \revert NoteHead.style
  \undo \hide Stem
}
crOn = \override NoteHead.style = #'cross
crOff = \revert NoteHead.style

%% insert chord name style stuff here.

jazzChords = { }

%%%%%%%%%%% Keys'n'thangs %%%%%%%%%%%%%%%%

global = { \time 4/4 }

Key = { \key c \major }

% ############ Horns ############

% ------ Trumpet ------
trpt = \transpose c d \relative c'' {
  \Key
  c1 | c | c |
}
trpHarmony = \transpose c' d {
  \jazzChords
}
trumpet = {
  \global
  \clef treble
  <<
    \trpt
  >>
}

% ------ Alto Saxophone ------
alto = \transpose c a \relative c' {
  \Key
  c1 | c | c |
```

```
}
altoHarmony = \transpose c' a {
  \jazzChords
}
altoSax = {
  \global
  \clef treble
  <<
    \alto
  >>
}

% ------ Baritone Saxophone ------
bari = \transpose c a' \relative c {
  \Key
  c1
  c1
  \sl
  d4^"Solo" d d d
  \nsl
}
bariHarmony = \transpose c' a \chordmode {
  \jazzChords s1 s d2:maj e:m7
}
bariSax = {
  \global
  \clef treble
  <<
    \bari
  >>
}

% ------ Trombone ------
tbone = \relative c {
  \Key
  c1 | c | c
}
tboneHarmony = \chordmode {
  \jazzChords
}
trombone = {
  \global
  \clef bass
  <<
    \tbone
  >>
}

% ############ Rhythm Section ##############

% ------ Guitar ------
gtr = \relative c'' {
```

```
  \Key
  c1
  \sl
  b4 b b b
  \nsl
  c1
}
gtrHarmony = \chordmode {
  \jazzChords
  s1 c2:min7+ d2:maj9
}
guitar = {
  \global
  \clef treble
  <<
    \gtr
  >>
}

%% ------ Piano ------
rhUpper = \relative c'' {
  \voiceOne
  \Key
  c1 | c | c
}
rhLower = \relative c' {
  \voiceTwo
  \Key
  e1 | e | e
}

lhUpper = \relative c' {
  \voiceOne
  \Key
  g1 | g | g
}
lhLower = \relative c {
  \voiceTwo
  \Key
  c1 | c | c
}

PianoRH = {
  \clef treble
  \global
  <<
    \new Voice = "one" \rhUpper
    \new Voice = "two" \rhLower
  >>
}
PianoLH = {
  \clef bass
```

```
  \global
  <<
    \new Voice = "one" \lhUpper
    \new Voice = "two" \lhLower
  >>
}

piano = {
  <<
    \new Staff = "upper" \PianoRH
    \new Staff = "lower" \PianoLH
  >>
}

% ------ Bass Guitar ------
Bass = \relative c {
  \Key
  c1 | c | c
}
bass = {
  \global
  \clef bass
  <<
    \Bass
  >>
}

% ------ Drums ------
up = \drummode {
  \voiceOne
  hh4 <hh sn> hh <hh sn>
  hh4 <hh sn> hh <hh sn>
  hh4 <hh sn> hh <hh sn>
}
down = \drummode {
  \voiceTwo
  bd4 s bd s
  bd4 s bd s
  bd4 s bd s
}

drumContents = {
  \global
  <<
    \new DrumVoice \up
    \new DrumVoice \down
  >>
}

%%%%%%%% It All Goes Together Here %%%%%%%%%%%%%%%%%%%%

\score {
```

```
  <<
    \new StaffGroup = "horns" <<
      \new Staff = "trumpet" \with { instrumentName = "Trumpet" }
      \trumpet
      \new Staff = "altosax" \with { instrumentName = "Alto Sax" }
      \altoSax
      \new ChordNames = "barichords" \with { instrumentName = "Trumpet" }
      \bariHarmony
      \new Staff = "barisax" \with { instrumentName = "Bari Sax" }
      \bariSax
      \new Staff = "trombone" \with { instrumentName = "Trombone" }
      \trombone
    >>

    \new StaffGroup = "rhythm" <<
      \new ChordNames = "chords" \gtrHarmony
      \new Staff = "guitar" \with { instrumentName = "Guitar" }
      \guitar
      \new PianoStaff = "piano" \with {
        instrumentName = "Piano"
        midiInstrument = "acoustic grand"
      }
      \piano
      \new Staff = "bass" \with { instrumentName = "Bass" }
      \bass
      \new DrumStaff \with { instrumentName = "Drums" }
      \drumContents
    >>
  >>
  \layout {
    \context { \Staff \RemoveEmptyStaves }
    \context {
      \Score
      \override BarNumber.padding = #3
      \override RehearsalMark.padding = #2
      skipBars = ##t
    }
  }
  \midi { }
}
```

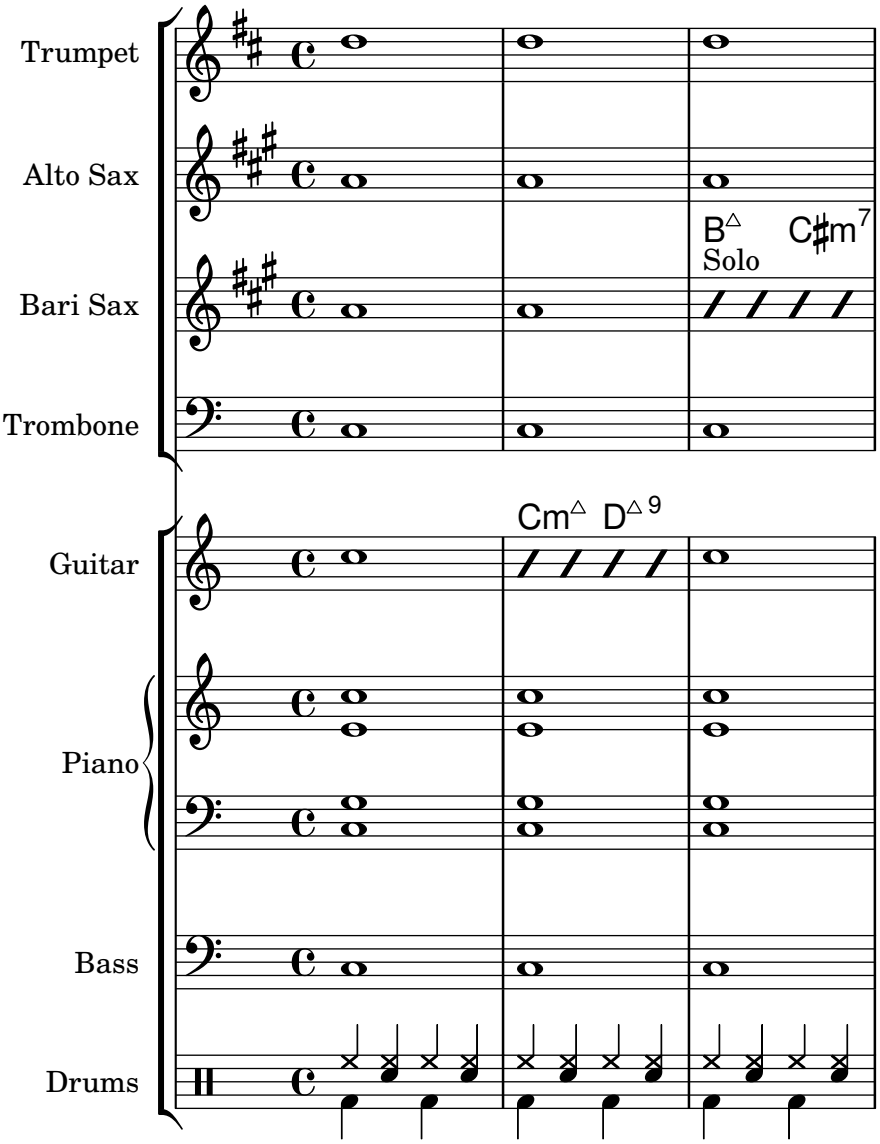
Song

(tune)

Me

moderato

Swing



Baguettes pour percussion

La représentation graphique des instruments à percussion n'est pas prise en charge nativement. De tels symboles peuvent cependant être inclus, soit à l'aide de fichiers EPS indépendants, soit à l'aide d'un *markup* contenant le code PostScript correspondant comme dans cet exemple.

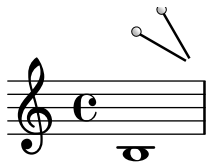
```
stick = \markup {
  \with-dimensions #'(0 . 5) #'(0 . 5)
  \postscript "
    0 6 translate
    0.8 -0.8 scale
    0 0 0 setrgbcolor
    [] 0 setdash
    1 setlinewidth
    0 setlinejoin
    0 setlinecap
    gsave [1 0 0 1 0 0] concat
    gsave [1 0 0 1 -3.5406095 -199.29342] concat
    gsave
    0 0 0 setrgbcolor
    newpath
    7.1434065 200.94354 moveto
```

```
7.2109628 200.90454 7.2785188 200.86554 7.3460747 200.82654 curveto
8.2056347 202.31535 9.0651946 203.80414 9.9247546 205.29295 curveto
9.8571989 205.33195 9.7896429 205.37095 9.7220864 205.40996 curveto
8.8625264 203.92115 8.0029664 202.43233 7.1434065 200.94354 curveto
closepath
eofill
grestore
gsave
0 0 0 setrgbcolor
newpath
4.9646672 203.10444 moveto
5.0036707 203.03688 5.0426744 202.96933 5.0816777 202.90176 curveto
6.5704792 203.76133 8.0592809 204.6209 9.5480824 205.48045 curveto
9.5090791 205.54801 9.4700754 205.61556 9.4310717 205.68311 curveto
7.94227 204.82356 6.4534687 203.96399 4.9646672 203.10444 curveto
closepath
eofill
grestore
gsave
<<
/ShadingType 3
/ColorSpace /DeviceRGB
/Coords [113.13708 207.87465 0 113.13708 207.87465 16.162441]
/Extend [true true]
/Domain [0 1]
/Function <<
/FunctionType 3
/Functions
[
<<
/FunctionType 2
/Domain [0 1]
/C0 [1 1 1]
/C1 [0.72941178 0.72941178 0.72941178]
/N 1
>>
]
/Domain [0 1]
/Bounds [ ]
/Encode [ 0 1 ]
>>
>>
newpath
7.6422017 200.76488 moveto
7.6505696 201.02554 7.3905363 201.24867 7.1341335 201.20075 curveto
6.8759501 201.16916 6.6949602 200.87978 6.7801462 200.63381 curveto
6.8480773 200.39155 7.1438307 200.25377 7.3728389 200.35861 curveto
7.5332399 200.42458 7.6444521 200.59122 7.6422017 200.76488 curveto
closepath
clip
gsave [
  0.052859054 0.063089841 -0.020912282 0.017521108 5.7334261 189.76443
```

```
] concat
shfill
grestore
grestore
0 0 0 setrgbcolor
[] 0 setdash
0.027282091 setlinewidth
0 setlinejoin
0 setlinecap
newpath
7.6422017 200.76488 moveto
7.6505696 201.02554 7.3905363 201.24867 7.1341335 201.20075 curveto
6.8759501 201.16916 6.6949602 200.87978 6.7801462 200.63381 curveto
6.8480773 200.39155 7.1438307 200.25377 7.3728389 200.35861 curveto
7.5332399 200.42458 7.6444521 200.59122 7.6422017 200.76488 curveto
closepath
stroke
gsave
<<
/ShadingType 3
/ColorSpace /DeviceRGB
/Coords [113.13708 207.87465 0 113.13708 207.87465 16.162441]
/Extend [true true]
/Domain [0 1]
/Function <<
/FunctionType 3
/Functions
[
<<
/FunctionType 2
/Domain [0 1]
/C0 [1 1 1]
/C1 [0.72941178 0.72941178 0.72941178]
/N 1
>>
]
/Domain [0 1]
/Bounds [ ]
/Encode [ 0 1 ]
>>
>>
newpath
5.2721217 202.83181 moveto
5.2804896 203.09247 5.0204563 203.3156 4.7640539 203.26768 curveto
4.5058701 203.23609 4.3248803 202.94671 4.4100662 202.70074 curveto
4.4779975 202.45848 4.7737511 202.3207 5.0027593 202.42554 curveto
5.1631598 202.49149 5.2743721 202.65813 5.2721217 202.83181 curveto
closepath
clip
gsave [
  0.052859054 0.063089841 -0.020912282 0.017521108 3.363346 191.83136
] concat
```

```
    shfill
    grestore
    grestore
    0 0 0 setrgbcolor
    [] 0 setdash
    0.027282091 setlinewidth
    0 setlinejoin
    0 setlinecap
    newpath
    5.2721217 202.83181 moveto
    5.2804896 203.09247 5.0204563 203.3156 4.7640539 203.26768 curveto
    4.5058701 203.23609 4.3248803 202.94671 4.4100662 202.70074 curveto
    4.4779975 202.45848 4.7737511 202.3207 5.0027593 202.42554 curveto
    5.1631598 202.49149 5.2743721 202.65813 5.2721217 202.83181 curveto
    closepath
    stroke
    grestore
    grestore
  "
}

\score {
  b1^\stick
}
```



Exemple pour percussions

Ce court exemple est tiré de « L'histoire du Soldat » de Stravinsky.

```
#(define mydrums '((bassdrum   default #t  4)
                   (snare      default #t -4)
                   (tambourine default #t  0)))

global = {
  \time 3/8 s4.
  \time 2/4 s2*2
  \time 3/8 s4.
  \time 2/4 s2
}

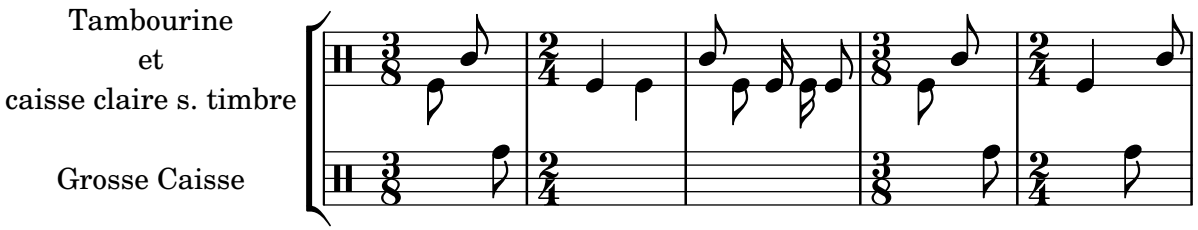
drumsA = {
  \context DrumVoice <<
    { \global }
    { \drummode {
        \autoBeamOff
        \stemDown sn8 \stemUp tamb s8 |
        sn4 \stemDown sn4 |
```

```
        \stemUp tamb8 \stemDown sn8 \stemUp sn16 \stemDown sn \stemUp sn8 |
        \stemDown sn8 \stemUp tamb s8 |
        \stemUp sn4 s8 \stemUp tamb
      }
    }
  >>
}

drumsB = {
  \drummode {
    s4 bd8 s2*2 s4 bd8 s4 bd8 s8
  }
}

\layout {
  indent = #40
}

\score {
  \new StaffGroup <<
    \new DrumStaff \with {
      instrumentName = \markup {
        \center-column {
          "Tambourine"
          "et"
          "caisse claire s. timbre"
        }
      }
    drumStyleTable = #(alist->hash-table mydrums)
  }
  \drumsA
  \new DrumStaff \with {
    instrumentName = #"Grosse Caisse"
    drumStyleTable = #(alist->hash-table mydrums)
  }
  \drumsB
  >>
}
```



Impression de musique aux métriques différentes

Bien qu'ayant des métriques bien différentes, les deux parties ci-dessous présentées restent synchrones.

Les barres de mesure ne peuvent plus être gérées au niveau du contexte `Score` ; les `Default_barline_engraver` et `Timing_translator` doivent être déplacés du contexte `Score` au contexte `Staff` afin de permettre des barres de mesure individualisées.

Le `Bar_number_engraver` devra lui aussi être déplacé, puisqu'il dépend de propriétés attachées au `Timing_translator`, afin de numéroter les mesures. L'utilisation d'un bloc `\with` dans la portée concernée permettra un affichage des numéros de mesure.

```
\paper {
  indent = #0
  ragged-right = ##t
}

global = { \time 3/4 { s2.*3 } \bar "" \break { s2.*3 } }

\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Default_bar_line_engraver"
    \remove "Bar_number_engraver"
    \override SpacingSpanner.uniform-stretching = ##t
    \override SpacingSpanner.strict-note-spacing = ##t
    proportionalNotationDuration = #(ly:make-moment 1/64)
  }
  \context {
    \Staff
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  }
  \context {
    \Voice
    \remove "Forbid_line_break_engraver"
    tupletFullLength = ##t
  }
}

Bassklarinette = \new Staff \with {
  \consists "Bar_number_engraver"
  barNumberVisibility = #(every-nth-bar-number-visible 2)
  \override BarNumber.break-visibility = #end-of-line-invisible
} <<
  \global {
    \bar "|"
    \clef treble
    \time 3/8
    d''4.

    \bar "|"
    \time 3/4
    r8 des''2( c''8)

    \bar "|"
```

```
    \time 7/8
    r4. ees''2 ~

    \bar "|"
    \time 2/4
    \tupletUp
    \tuplet 3/2 { ees''4 r4 d''4 ~ }

    \bar "|"
    \time 3/8
    \tupletUp
    \tuplet 4/3 { d''4 r4 }

    \bar "|"
    \time 2/4
    e''2

    \bar "|"
    \time 3/8
    es''4.

    \bar "|"
    \time 3/4
    r8 d''2 r8
    \bar "|"
  }
>>

Perkussion = \new StaffGroup <<
  \new Staff <<
    \global {
      \bar "|"
      \clef percussion
      \time 3/4
      r4 c'2 ~

      \bar "|"
      c'2.

      \bar "|"
      R2.

      \bar "|"
      r2 g'4 ~

      \bar "|"
      g'2. ~

      \bar "|"
      g'2.
    }
  >>
```

```
  \new Staff <<
    \global {
      \bar "|"
      \clef percussion
      \time 3/4
      R2.

      \bar "|"
      g'2. ~

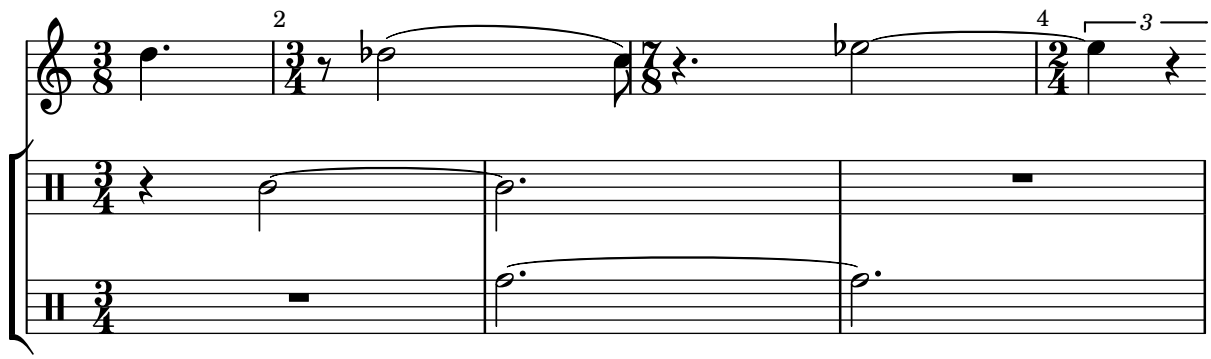
      \bar "|"
      g'2.

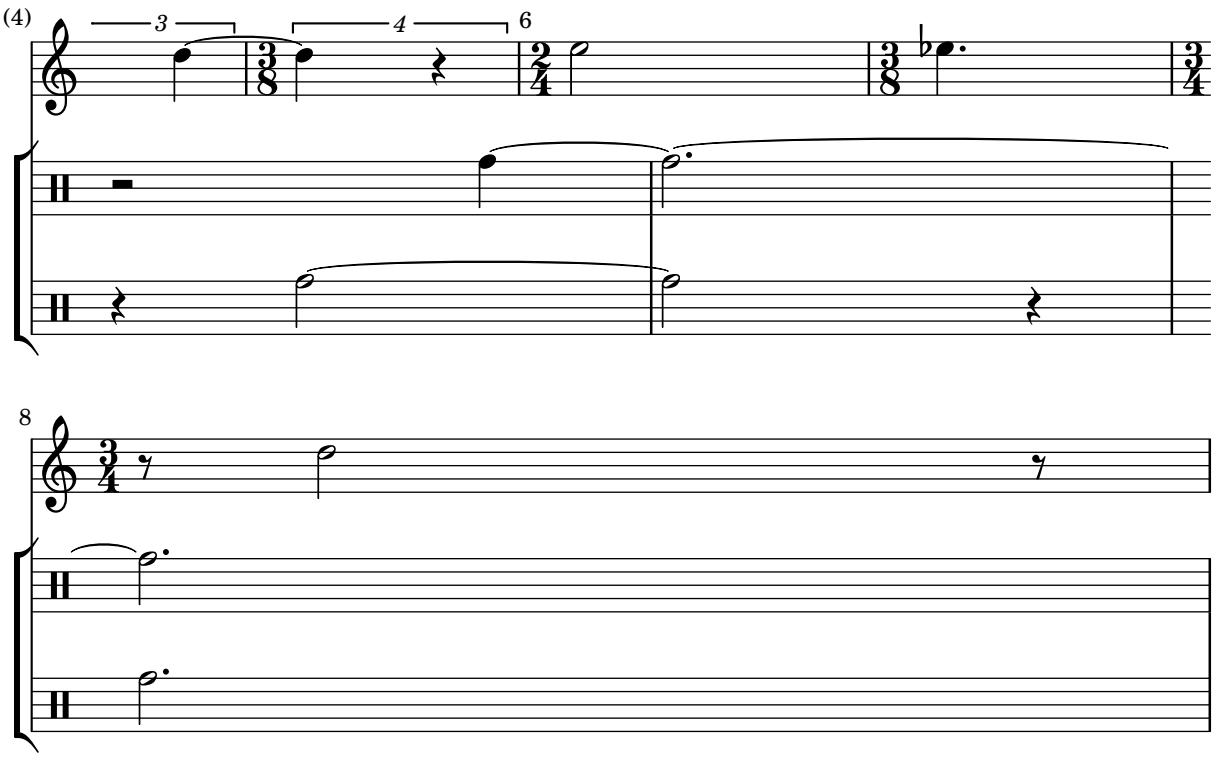
      \bar "|"
      r4 g'2 ~

      \bar "|"
      g'2 r4

      \bar "|"
      g'2.
    }
  >>
>>

\score {
  <<
    \Bassklarinette
    \Perkussion
  >>
}
```





Exemple pour tam-tam

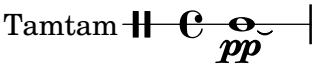
Un simple exemple pour tam-tam, saisi « tt ».

```
#(define mydrums '((tamtam default #t 0)))

\new DrumStaff \with { instrumentName = #"Tamtam" }

\drummode {
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \override Staff.StaffSymbol.line-positions = #'( 0 )
  \override Staff.BarLine.bar-extent = #'(-1.5 . 1.5)

  tt 1 \pp \laissezVibrer
}
```
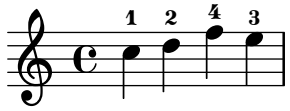


Fretted strings

Section “Fretted string instruments” dans *Manuel de notation*

Ajout de doigtés à une partition

Les instructions de doigtés se saisissent selon une syntaxe très simple.

```
\relative c'' {
  c4-1 d-2 f-4 e-3
}
```

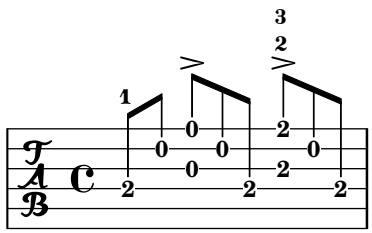


Ajout de doigtés à des tablatures

L’ajout de doigtés à des tablatures s’obtient en conjuguant des `\markup` et des `\finger`.

```
one = \markup { \finger 1 }
two = \markup { \finger 2 }
threeTwo = \markup {
  \override #'(baseline-skip . 2)
  \column {
    \finger 3
    \finger 2
  }
}
threeFour = \markup {
  \override #'(baseline-skip . 2)
  \column {
    \finger 3
    \finger 4
  }
}

\score {
  \new TabStaff {
    \tabFullNotation
    \stemUp
    e8\4^\one b\2 <g\3 e'\1>^>[ b\2 e\4]
    <a\3 fis'\1>^>^\threeTwo[ b\2 e\4]
  }
}
```



Ajout de *markups* à une tablature

Par défaut, les *markups* n'apparaissent pas dans une tablature.

Il suffit, pour les voir apparaître, d'une simple commande `\revert TabStaff.TextScript.stencil`.

```
%% http://lsr.di.unimi.it/LSR/Item?id=919
% by P.P.Schneider on June 2014

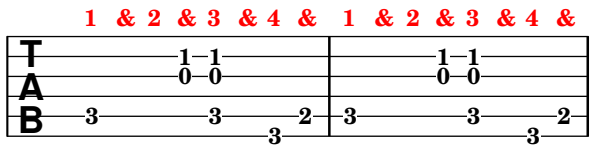
high  = { r4 r8 <g c'> q r8 r4 }

low = { c4 r4 c8 r8 g,8 b, }

pulse = { s8^"1" s^"&" s^"2" s^"&" s^"3" s^"&" s^"4" s^"&" }

\score {
 \new TabStaff {
   \repeat unfold 2 << \high \\ \low \\ \pulse >>
  }
  \layout {
    \context {
      \TabStaff
      \clef moderntab
      \revert TextScript.stencil
      \override TextScript.font-series = #'bold
      \override TextScript.font-size = #-2
      \override TextScript.color = #red

    }
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1/8)
    }
  }
}
```
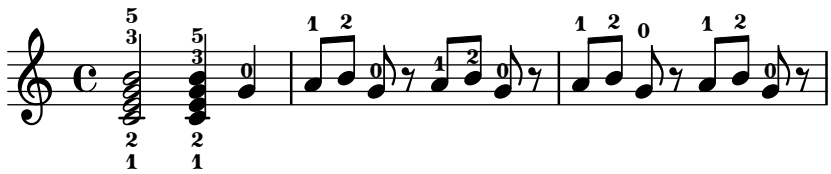


Impression des doigtés à l'intérieur de la portée

L'empilement des indications de doigté se fait par défaut à l'extérieur de la portée. Il est néanmoins possible d'annuler ce comportement. Une attention particulière doit toutefois être portée dans les cas où doigté et hampe vont dans la même direction : les indications de doigté n'évitent les hampe qu'en présence de ligature. Ce réglage peut s'adapter pour éviter toutes les hampes ou aucune d'elles. L'exemple suivant illustre ces deux options, ainsi que la manière de revenir au comportement par défaut.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 g'-0
```

```
  a8[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##f
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##t
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = #only-if-beamed
  a[-1 b]-2 g-0 r
}
```

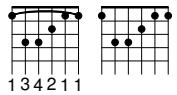


Barrés et diagrammes de fret automatiques

L'utilisation de diagrammes de fret automatiques permet d'imprimer un indication de barré dès lors qu'un même doigt s'applique à plusieurs cordes.

Lorsqu'aucune indication de doigt n'est fournie dans l'accord qui sera rendu sous forme de diagramme, aucune indication de barré n'apparaîtra puisqu'il n'y a aucun moyen d'identifier où les barrés devraient venir se positionner.

```
\new FretBoards {
  <f,-1 c-3 f-4 a-2 c'-1 f'-1>1
  <f, c f a c' f'>1
}
```



Notation d'accords barrés pour guitare (avec extension textuelle)

Voici comment imprimer un « accord barré » – il suffit de dé-commenter la ligne appropriée.

La syntaxe est la suivante : `\bbarre #"fret_number" note(s)`

La fonction Scheme ici présentée peut faire l'objet d'un fichier de définition personnalisée à inclure dans vos propres fichiers sources.

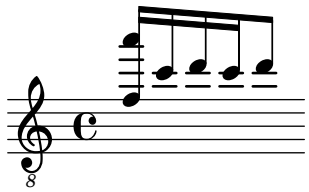
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% %%%%%%  Cut here ----- Start 'bbarred.ly'

%% C with slash -------------------------------
cWithSlash = \markup {
  \combine \roman C \translate #'(0.6 . -0.4) \draw-line #'(0 . 2.0)
}
%% Span ----------------------------------
%% Syntax: \bbarre #"text" { notes } - text = any number of box
bbarre =
#(define-music-function (barre location str music) (string? ly:music?)
   (let ((elts (extract-named-music music 'rhythmic-event)))
     (if (pair? elts)
         (let ((first-element (first elts))
```

```
              (last-element (last elts)))
            (set! (ly:music-property first-element 'articulations)
                  (cons (make-music 'TextSpanEvent 'span-direction -1)
                        (ly:music-property first-element 'articulations)))
            (set! (ly:music-property last-element 'articulations)
                  (cons (make-music 'TextSpanEvent 'span-direction 1)
                        (ly:music-property last-element 'articulations))))))
   #{
        \once \override TextSpanner.font-size = #-2
        \once \override TextSpanner.font-shape = #'upright
        \once \override TextSpanner.staff-padding = #3
        \once \override TextSpanner.style = #'line
        \once \override TextSpanner.to-barline = ##f
        \once \override TextSpanner.bound-details =
             #`((left
                 (text . ,#{ \markup { \draw-line #'( 0 . -.5) } #})
                 (Y . 0)
                 (padding . 0.25)
                 (attach-dir . -2))
                (right
                 (text . ,#{ \markup { \cWithSlash #str } #})
                 (Y . 0)
                 (padding . 0.25)
                 (attach-dir . 2)))
%% uncomment this line for make full barred
      % \once \override TextSpanner.bound-details.left.text = \markup{"B" #str}
      $music
   #})

%% %%%%%%  Cut here ----- End 'bbarred.ly'
%% Copy and change the last line for full barred. Rename in 'fbarred.ly'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Syntaxe: \bbarre #"text" { notes } - text = any number of box
\relative c'{ \clef "G_8" \stemUp \bbarre #"III" { <f a'>16[  c' d c d8] } }
```



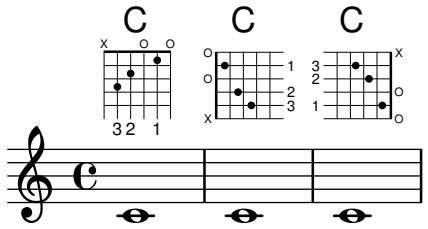
Orientation des diagrammes de fret

Les diagrammes de fret peuvent s'orienter de trois manières différentes. Ils s'aligneront par défaut sur la corde du haut ou le sommet du fret.

```
\include "predefined-guitar-fretboards.ly"

<<
  \chords {
    c1
```

```
    c1
    c1
  }
  \new FretBoards {
    \chordmode {
      c1
      \override FretBoard.fret-diagram-details.orientation =
        #'landscape
      c1
      \override FretBoard.fret-diagram-details.orientation =
        #'opposing-landscape
      c1
    }
  }
  \new Voice {
    c'1
    c'1
    c'
  }
>>
```



Glissando d'accords et tablature

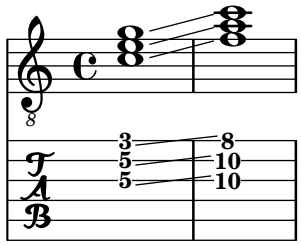
Un glissando sur des accords s'indique dans un `TabStaff` de la même manière que dans un `Staff`, à ceci près que nous aurons besoin des numéros de corde afin de déterminer correctement les frets d'arrivée.

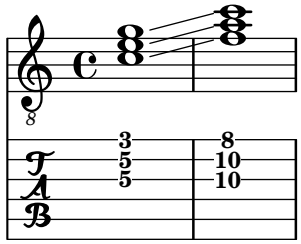
```
myMusic = \relative c' {
  <c e g>1 \glissando <f a c>
}

\score {
  <<
    \new Staff {
      \clef "treble_8"
      \myMusic
    }
    \new TabStaff \myMusic
  >>
}

\score {
  <<
    \new Staff {
      \clef "treble_8"
      \myMusic
```

```
    }
    \new TabStaff \with { \override Glissando.style = #'none } {
      \myMusic
    }
  >>
}
```





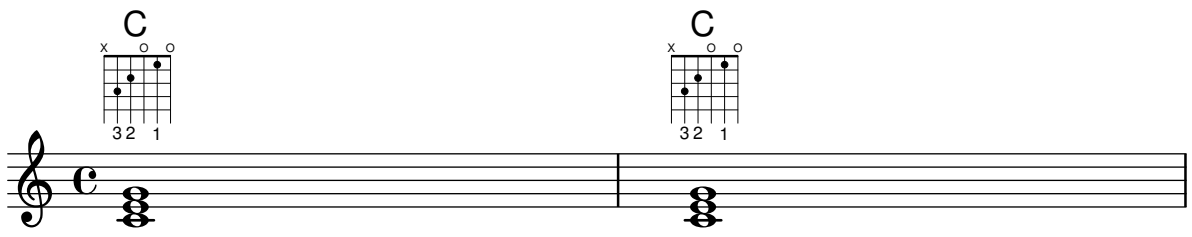
Changement d'accord et diagramme de fret

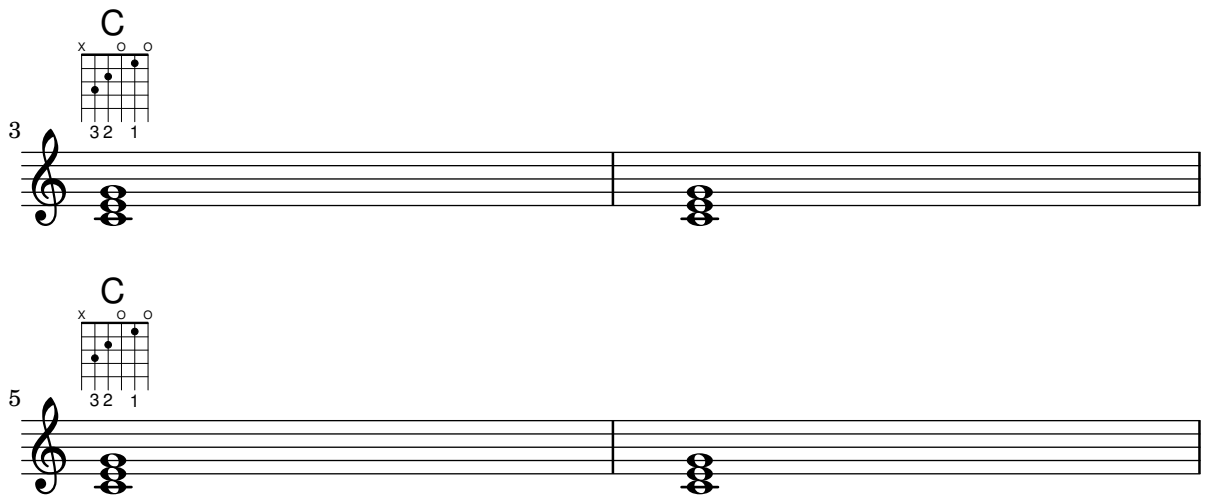
Vous pouvez opter pour n'imprimer les diagrammes de fret qu'à l'occasion d'un changement d'accord ou de saut de ligne.

```
\include "predefined-guitar-fretboards.ly"

myChords = \chordmode {
  c1 c1 \break
  \set chordChanges = ##t
  c1 c1 \break
  c1 c1
}

<<
  \new ChordNames { \myChords }
  \new FretBoards { \myChords }
  \new Staff { \myChords }
>>
```





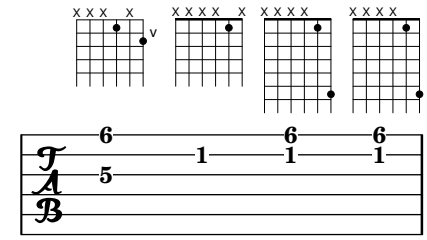
Accords avec doigté espacé pour FretBoards et TabVoice

Il peut arriver que le doigté d'un accord soit assez étendu. Sauf mention contraire, la propriété de contexte `maximumFretStretch` est cependant fixée à 4, ce qui peut générer un avertissement « Pas de corde pour la hauteur... » et la note est omise. On peut régler `maximumFretStretch` sur une valeur appropriée ou assigner explicitement leur numéro de corde à toutes les notes d'un accord.

```
%% The code below will print two warnings, which may be omitted by uncommenting:
%#(for-each (lambda (x) (ly:expect-warning "No string for pitch")) (iota 2))

mus = {
  <c' bes'>
  <c'\2 bes'>
  \set maximumFretStretch = 5
  <c' bes'>
  <c'\2 bes'\1>
}

<<
  \new FretBoards \mus
  \new TabVoice \mus
>>
```

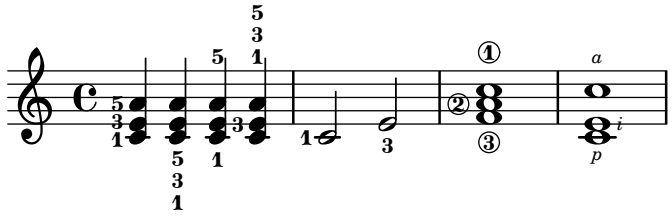


Contrôler la position des doigtés dans un accord

Le positionnement des doigtés peut être contrôlé de manière très précise. Afin que l'orientation soit prise en compte, il est nécessaire d'utiliser une syntaxe d'accord `< >`, même s'il ne s'agit que d'une seule note. Le positionnement des numéros de corde et doigtés main droite se règle de manière analogue.

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
```

```
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
  \set stringNumberOrientations = #'(up left down)
  <f\3 a\2 c\1>1
  \set strokeFingerOrientations = #'(down right up)
  <c\rightHandFinger #1 e\rightHandFinger #2 c'\rightHandFinger #4 >
}
```



Personnalisation de diagrammes de fret

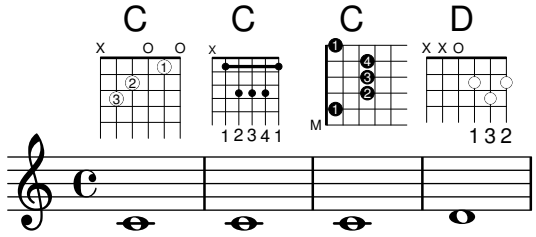
Les propriétés d'un diagramme de fret sont définies par les `fret-diagram-details`. En matière de diagramme de fret, les adaptations s'appliquent à l'objet `FretBoards.FretBoard`. Un `FretBoards` est comparable à un `Voice` : il s'agit d'un contexte du plus bas niveau, et il n'est donc pas primordial de l'instancier de manière explicite pour adapter ses propriétés.

```
\include "predefined-guitar-fretboards.ly"
\storePredefinedDiagram #default-fret-table \chordmode { c' }
                        #guitar-tuning
                        #"x;1-1-(;3-2;3-3;3-4;1-1-);"

% shorthand
oo = #(define-music-function
       (grob-path value)
       (list? scheme?)
       #{ \once \override $grob-path = #value #})

<<
  \new ChordNames {
    \chordmode { c1 | c | c | d }
  }
  \new FretBoards {
    % Set global properties of fret diagram
    \override FretBoards.FretBoard.size = #'1.2
    \override FretBoard.fret-diagram-details.finger-code = #'in-dot
    \override FretBoard.fret-diagram-details.dot-color = #'white
    \chordmode {
      c
      \oo FretBoard.size #'1.0
```

```
      \oo FretBoard.fret-diagram-details.barre-type #'straight
      \oo FretBoard.fret-diagram-details.dot-color #'black
      \oo FretBoard.fret-diagram-details.finger-code #'below-string
      c'
      \oo FretBoard.fret-diagram-details.barre-type #'none
      \oo FretBoard.fret-diagram-details.number-type #'arabic
      \oo FretBoard.fret-diagram-details.orientation #'landscape
      \oo FretBoard.fret-diagram-details.mute-string #"M"
      \oo FretBoard.fret-diagram-details.label-dir #LEFT
      \oo FretBoard.fret-diagram-details.dot-color #'black
      c'
      \oo FretBoard.fret-diagram-details.finger-code #'below-string
      \oo FretBoard.fret-diagram-details.dot-radius #0.35
      \oo FretBoard.fret-diagram-details.dot-position #0.5
      \oo FretBoard.fret-diagram-details.fret-count #3
      d
    }
  }
  \new Voice {
    c'1 | c' | c' | d'
  }
>>
```



Personnalisation des diagrammes de fret

Les propriétés d'un diagramme de fret sont modifiables grâce au `fret-diagram-details`. Lorsqu'ils sont générés sous forme de `\markup`, rien n'empêche de modifier les diagrammes en jouant sur les réglages de l'objet `Voice.TextScript` ou bien directement sur le *markup*.

```
<<
  \chords { c1 | c | c | d }

  \new Voice = "mel" {
    \textLengthOn
    % Set global properties of fret diagram
    \override TextScript.size = #'1.2
    \override TextScript.fret-diagram-details.finger-code = #'in-dot
    \override TextScript.fret-diagram-details.dot-color = #'white

    %% C major for guitar, no barre, using defaults
       % terse style
    c'1^\markup { \fret-diagram-terse "x;3-3;2-2;o;1-1;o;" }

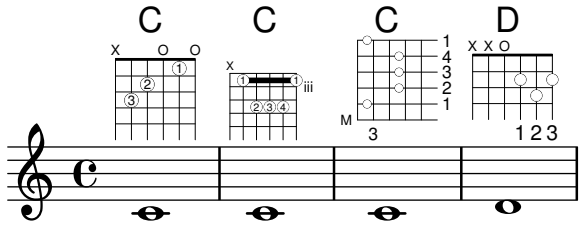
    %% C major for guitar, barred on third fret
       % verbose style
```

```
   % size 1.0
   % roman fret label, finger labels below string, straight barre
c'1^\markup {
  % standard size
  \override #'(size . 1.0) {
    \override #'(fret-diagram-details . (
                 (number-type . roman-lower)
                 (finger-code . in-dot)
                 (barre-type . straight))) {
      \fret-diagram-verbose #'((mute 6)
                               (place-fret 5 3 1)
                               (place-fret 4 5 2)
                               (place-fret 3 5 3)
                               (place-fret 2 5 4)
                               (place-fret 1 3 1)
                               (barre 5 1 3))
    }
  }
}

%% C major for guitar, barred on third fret
   % verbose style
   % landscape orientation, arabic numbers, M for mute string
   % no barre, fret label down or left, small mute label font
c'1^\markup {
  \override #'(fret-diagram-details . (
               (finger-code . below-string)
               (number-type . arabic)
               (label-dir . -1)
               (mute-string . "M")
               (orientation . landscape)
               (barre-type . none)
               (xo-font-magnification . 0.4)
               (xo-padding . 0.3))) {
    \fret-diagram-verbose #'((mute 6)
                             (place-fret 5 3 1)
                             (place-fret 4 5 2)
                             (place-fret 3 5 3)
                             (place-fret 2 5 4)
                             (place-fret 1 3 1)
                             (barre 5 1 3))
  }
}

%% simple D chord
   % terse style
   % larger dots, centered dots, fewer frets
   % label below string
d'1^\markup {
  \override #'(fret-diagram-details . (
               (finger-code . below-string)
               (dot-radius . 0.35)
```

```
                    (dot-position . 0.5)
                    (fret-count . 3))) {
          \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
        }
      }
    }
>>
```



Création de diagrammes de fret prédéfinis pour d'autres instruments

La liste des diagrammes standards prédéfinis pour la guitare peut être augmentée d'autres définitions spécifiques à d'autres instruments. Voici comment définir un nouvel accordage ainsi que quelques diagrammes prédéfinis pour le « cuatro vénézuélien ».

Cet exemple illustre aussi la manière d'ajouter des doigtés aux accords ; ils serviront de référence pour la boucle d'accord et seront indiqués dans les diagrammes et le `TabStaff`, mais pas dans la musique.

Ces diagrammes ne peuvent pas être transposés, dans la mesure où ils contiennent des informations sur les cordes. Ceci est amené à évoluer.

```
% add FretBoards for the Cuatro
%   Note: This section could be put into a separate file
%      predefined-cuatro-fretboards.ly
%      and \included into each of your compositions

cuatroTuning = #`(,(ly:make-pitch 0 6 0)
                  ,(ly:make-pitch 1 3 SHARP)
                  ,(ly:make-pitch 1 1 0)
                  ,(ly:make-pitch 0 5 0))

dSix = { <a\4 b\1 d\3 fis\2> }
dMajor = { <a\4 d\1 d\3 fis \2> }
aMajSeven = { <a\4 cis\1 e\3 g\2> }
dMajSeven = { <a\4 c\1 d\3 fis\2> }
gMajor = { <b\4 b\1 d\3 g\2> }

\storePredefinedDiagram #default-fret-table \dSix
                        #cuatroTuning
                        #"o;o;o;o;"
\storePredefinedDiagram #default-fret-table \dMajor
                        #cuatroTuning
                        #"o;o;o;3-3;"
\storePredefinedDiagram #default-fret-table \aMajSeven
                        #cuatroTuning
                        #"o;2-2;1-1;2-3;"
```

```
\storePredefinedDiagram #default-fret-table \dMajSeven
                        #cuatroTuning
                        #"o;o;o;1-1;"
\storePredefinedDiagram #default-fret-table \gMajor
                        #cuatroTuning
                        #"2-2;o;1-1;o;"

% end of potential include file /predefined-cuatro-fretboards.ly


#(set-global-staff-size 16)

primerosNames = \chordmode {
  d:6 d a:maj7 d:maj7
  g
}
primeros = {
  \dSix \dMajor \aMajSeven \dMajSeven
  \gMajor
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \primerosNames
    }

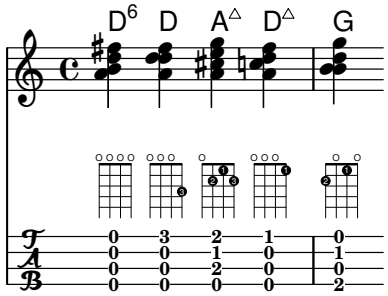
    \new Staff {
      \new Voice \with {
        \remove "New_fingering_engraver"
      }
      \relative c'' {
        \primeros
      }
    }

    \new FretBoards {
      \set Staff.stringTunings = #cuatroTuning
%      \override FretBoard
%        #'(fret-diagram-details string-count) = 4
      \override FretBoard.fret-diagram-details.finger-code = #'in-dot
      \primeros
    }

    \new TabStaff \relative c'' {
      \set TabStaff.stringTunings = #cuatroTuning
      \primeros
    }

  >>
```

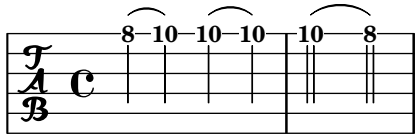
```
  \layout {
    \context {
      \Score
      \override SpacingSpanner.base-shortest-duration =
        #(ly:make-moment 1 16)
    }
  }
  \midi { }
}
```



Simulation d'un *hammer* en tablature

En mode tablature, un *hammer*, ou lié ascendant, peut se simuler à l'aide d'une liaison.

```
\score {
  \new TabStaff {
    \relative c'' {
      \tabFullNotation
      c4( d) d( d)
      d2( c)
    }
  }
}
```

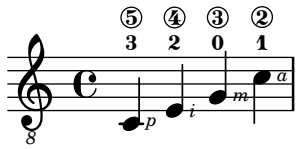


Doigtés, indications de corde, et doigtés main droite

L'exemple suivant illustre comment combiner des doigtés pour la main gauche, des indications de corde et des doigtés pour la main droite.

```
#(define RH rightHandFinger)

\relative c {
  \clef "treble_8"
  <c-3\5\RH #1 >4
  <e-2\4\RH #2 >4
  <g-0\3\RH #3 >4
  <c-1\2\RH #4 >4
}
```



Notation de flamenco

La guitare flamenco fait appel à des éléments de notation particuliers :

– le *golpe* : coup frappé sur la caisse de résonance avec l'ongle de l'annulaire,

– une flèche pour indiquer le sens des butés (*strokes*),

– les doigtés : « p » pouce, « i » index, « m » majeur, « a » annulaire et « x » auriculaire,

– les *rasgueados* sur trois ou quatre doigts : grattage des cordes en étendant les doigts rapidement les uns après les autres dans un mouvement continu, se terminant sur l'index,

– les *abanicos* : séries de butés du pouce, index et auriculaire. Il existe aussi un *abanico 2* pour lequel l'index et l'annulaire remplacent l'auriculaire.

– *alza púa* : jeu rapide du pouce.

La plupart de ces symboles utilisent une flèche en plus des doigtés. Dans le cas d'un *abanico*, les têtes de note ne sont imprimées que pour le premier accord.

Le début du code ci-dessous répertorie le paramétrage de ces différents symboles, que vous pouvez copier dans un fichier `flamenco.ly` pour inclusion dans vos propres compositions.

```
%%%%%%%  Cut here ----- Start 'flamenco.ly'

% Text indicators :
abanico = ^\markup\small { \italic Abanico }
rasgueado = ^\markup\small { \italic Ras. }
alzapua = ^\markup\small { \italic Alzapua }

% Finger stroke symbols :
strokeUp = \markup\combine\override #'(thickness . 1.3) \draw-line #'(0 . 2)\raise #2 \arrow
strokeDown = \markup\combine\arrow-head #Y #DOWN ##f \override #'(thickness . 1.3) \draw-lin

% Golpe symbol :
golpe = \markup {
  \filled-box #'(0 . 1) #'(0 . 1) #0
  \hspace #-1.6
  \with-color #white
  \filled-box #'(0.15 . 0.85) #'(0.15 . 0.85) #0
}

% Strokes, fingers and golpe command :
RHp = \rightHandFinger #1
RHi = \rightHandFinger #2
RHm = \rightHandFinger #3
RHa = \rightHandFinger #4
RHx = \rightHandFinger #5
RHu = \rightHandFinger \strokeUp
RHd = \rightHandFinger \strokeDown
RHg = \rightHandFinger \golpe

% Just handy :)
tupletOff = {
  \once \omit TupletNumber
  \once \omit TupletBracket
```

```
}

tupletsOff = {
  \omit TupletNumber
  \override TupletBracket.bracket-visibility = #'if-no-beam
}

tupletsOn = {
  \override TupletBracket.bracket-visibility = #'default
  \undo \omit TupletNumber
}

headsOff = {
  \hide TabNoteHead
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}

headsOn = {
  \override TabNoteHead.transparent = ##f
  \override NoteHead.transparent = ##f
  \override NoteHead.no-ledgers = ##f
}

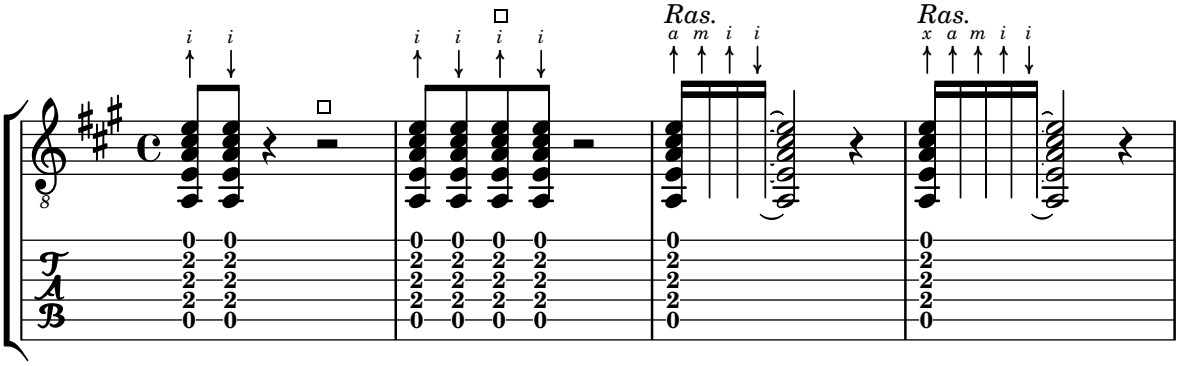
%%%%%%  Cut here ----- End 'flamenco.ly'
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

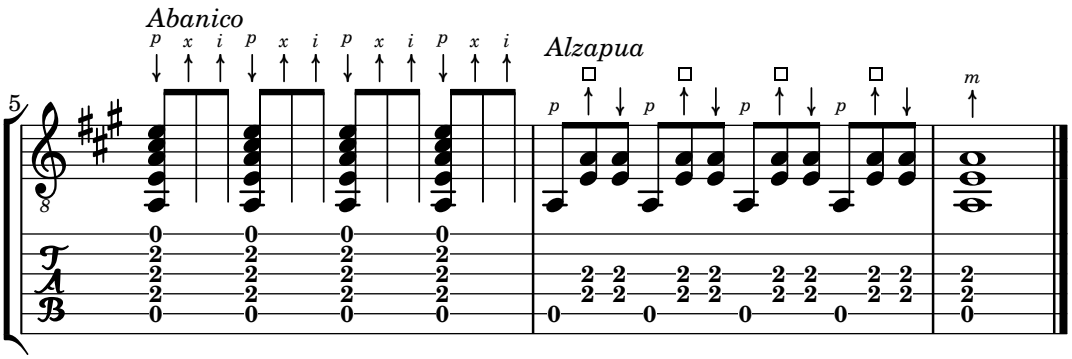
part = \relative c' {
  \set strokeFingerOrientations = #'(up)
  \key a\major
  <a, e' a cis e\RHu\RHi>8
  <a e' a cis e\RHd\RHi>8
  r4
  r2^\markup\golpe
  <a e' a cis e\RHu\RHi>8
  <a e' a cis e\RHd\RHi>8
  <a e' a cis e\RHu\RHi\RHg>8
  <a e' a cis e\RHd\RHi>8
  r2
  <a e' a cis e\RHu\RHa>16\rasgueado
  \headsOff
  <a e' a cis e\RHu\RHm>
  <a e' a cis e\RHu\RHi>
  <a e' a cis e\RHd\RHi>~
  \headsOn
  <a e' a cis e>2
  r4
  \tupletOff
  \tuplet 5/4 {
    <a e' a cis e\RHu\RHx>16\rasgueado
    \headsOff
    <a e' a cis e\RHu\RHa>
```

```
    <a e' a cis e\RHu\RHm>
    <a e' a cis e\RHu\RHi>
    <a e' a cis e\RHd\RHi>~
    \headsOn
  }
  <a e' a cis e>2
  r4
  \tupletsOff
  \tuplet 3/2 {
    <a e' a cis e\RHd\RHp>8\abanico
    \headsOff
    <a e' a cis e\RHu\RHx>
    <a e' a cis e\RHu\RHi>
    \headsOn
  }
  \tuplet 3/2 {
    <a e' a cis e\RHd\RHp>8
    \headsOff
    <a e' a cis e\RHu\RHx>
    <a e' a cis e\RHu\RHi>
    \headsOn
  }
  \tuplet 3/2 {
    <a e' a cis e\RHd\RHp>8
    \headsOff
    <a e' a cis e\RHu\RHx>
    <a e' a cis e\RHu\RHi>
    \headsOn
  }
  \tuplet 3/2 {
    <a e' a cis e\RHd\RHp>8
    \headsOff
    <a e' a cis e\RHu\RHx>
    <a e' a cis e\RHu\RHi>
    \headsOn
  }
  \tupletsOff
  \override Beam.positions = #'(2 . 2)
  \tuplet 3/2 {
    a8\RHp\alzapua
    <e' a\RHu\RHg>
    <e a\RHd>
  }
  \tuplet 3/2 {
    a,8\RHp
    <e' a\RHu\RHg>
    <e a\RHd>
  }
  \tuplet 3/2 {
    a,8\RHp
    <e' a\RHu\RHg>
    <e a\RHd>
```

```
  }
  \tuplet 3/2 {
    a,8\RHp
    <e' a\RHu\RHg>
    <e a\RHd>
  }
  \tupletsOn
  <a, e' a\RHu\RHm>1
  \bar "|."
}

\score {
  \new StaffGroup <<
    \context Staff = "part" <<
      \clef "G_8"
      {
        \part
      }
    >>
    \context TabStaff {
      \part
    }
  >>
  \layout {
    ragged-right = ##t
  }
}
```





Construction et développement de diagrammes de fret

Voici différentes manières d'obtenir et de personnaliser des diagrammes de fret :

```
<<
  \chords {
    a2 a
    \repeat unfold 3 {
      c c c d d
    }
  }

  \new Voice = "mel" {
    \textLengthOn
    % Set global properties of fret diagram
    \override TextScript.size = #1.2
    \override TextScript.fret-diagram-details.finger-code = #'below-string
    \override TextScript.fret-diagram-details.dot-color = #'black

    %% A chord for ukulele
    a'2^\markup {
      \override #'(fret-diagram-details . (
                  (string-count . 4)
                  (dot-color . white)
                  (finger-code . in-dot))) {
        \fret-diagram "4-2-2;3-1-1;2-o;1-o;"
      }
    }

    %% A chord for ukulele, with formatting defined in definition string
    %  1.2 * size, 4 strings, 4 frets, fingerings below string
    %  dot radius .35 of fret spacing, dot position 0.55 of fret spacing
    a'2^\markup {
      \override #'(fret-diagram-details . (
                  (dot-color . white)
                  (open-string . "o"))) {
        \fret-diagram "s:1.2;w:4;h:3;f:2;d:0.35;p:0.55;4-2-2;3-1-1;2-o;1-o;"
      }
    }

      %% These chords will be in normal orientation

    %% C major for guitar, barred on third fret
    %  verbose style
    %  roman fret label, finger labels below string, straight barre
    c'2^\markup {
      % 110% of default size
      \override #'(size . 1.1) {
        \override #'(fret-diagram-details . (
                    (number-type . roman-lower)
                    (finger-code . below-string)
                    (barre-type . straight))) {
          \fret-diagram-verbose #'((mute 6)
```

```
                                (place-fret 5 3 1)
                                (place-fret 4 5 2)
                                (place-fret 3 5 3)
                                (place-fret 2 5 4)
                                (place-fret 1 3 1)
                                (barre 5 1 3))
    }
  }
}

%% C major for guitar, barred on third fret
%% Double barre used to test barre function
%  verbose style
c'2^\markup {
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                (number-type . arabic)
                (dot-label-font-mag . 0.9)
                (finger-code . in-dot)
                (fret-label-font-mag . 0.6)
                (fret-label-vertical-offset . 0)
                (label-dir . -1)
                (mute-string . "M")
                (xo-font-magnification . 0.4)
                (xo-padding . 0.3))) {
      \fret-diagram-verbose #'((mute 6)
                               (place-fret 5 3 1)
                               (place-fret 4 5 2)
                               (place-fret 3 5 3)
                               (place-fret 2 5 4)
                               (place-fret 1 3 1)
                               (barre 4 2 5)
                               (barre 5 1 3))
    }
  }
}

%% C major for guitar, with capo on third fret
%  verbose style
c'2^\markup {
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                (number-type . roman-upper)
                (dot-label-font-mag . 0.9)
                (finger-code . none)
                (fret-label-vertical-offset . 0.5)
                (xo-font-magnification . 0.4)
                (xo-padding . 0.3))) {
      \fret-diagram-verbose #'((mute 6)
                               (capo 3)
```

```
                                (open 5)
                                (place-fret 4 5 1)
                                (place-fret 3 5 2)
                                (place-fret 2 5 3)
                                (open 1))
    }
  }
}

%% simple D chord
d'2^\markup {
  \override #'(fret-diagram-details . (
               (finger-code . below-string)
               (dot-radius . 0.35)
               (string-thickness-factor . 0.3)
               (dot-position . 0.5)
               (fret-count . 3))) {
    \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
  }
}

%% simple D chord, large top fret thickness
d'2^\markup  {
  \override #'(fret-diagram-details . (
               (finger-code . below-string)
               (dot-radius . 0.35)
               (dot-position . 0.5)
               (top-fret-thickness . 7)
               (fret-count . 3))) {
    \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
  }
}

  % These chords will be in landscape orientation
\override TextScript.fret-diagram-details.orientation = #'landscape

%% C major for guitar, barred on third fret
%  verbose style
%  roman fret label, finger labels below string, straight barre
c'2^\markup {
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                 (number-type . roman-lower)
                 (finger-code . below-string)
                 (barre-type . straight))) {
      \fret-diagram-verbose #'((mute 6)
                               (place-fret 5 3 1)
                               (place-fret 4 5 2)
                               (place-fret 3 5 3)
                               (place-fret 2 5 4)
                               (place-fret 1 3 1)
```

```
                                (barre 5 1 3))
    }
  }
}

%% C major for guitar, barred on third fret
%% Double barre used to test barre function
%  verbose style
c'2^\markup {
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                (number-type . arabic)
                (dot-label-font-mag . 0.9)
                (finger-code . in-dot)
                (fret-label-font-mag . 0.6)
                (fret-label-vertical-offset . 0)
                (label-dir . -1)
                (mute-string . "M")
                (xo-font-magnification . 0.4)
                (xo-padding . 0.3))) {
      \fret-diagram-verbose #'((mute 6)
                               (place-fret 5 3 1)
                               (place-fret 4 5 2)
                               (place-fret 3 5 3)
                               (place-fret 2 5 4)
                               (place-fret 1 3 1)
                               (barre 4 2 5)
                               (barre 5 1 3))
    }
  }
}

%% C major for guitar, with capo on third fret
%  verbose style
c'2^\markup {
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                (number-type . roman-upper)
                (dot-label-font-mag . 0.9)
                (finger-code . none)
                (fret-label-vertical-offset . 0.5)
                (xo-font-magnification . 0.4)
                (xo-padding . 0.3))) {
      \fret-diagram-verbose #'((mute 6)
                               (capo 3)
                               (open 5)
                               (place-fret 4 5 1)
                               (place-fret 3 5 2)
                               (place-fret 2 5 3)
                               (open 1))
```

```
    }
  }
}

%% simple D chord
d'2^\markup {
  \override #'(fret-diagram-details . (
               (finger-code . below-string)
               (dot-radius . 0.35)
               (dot-position . 0.5)
               (fret-count . 3))) {
    \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
  }
}

%% simple D chord, large top fret thickness
d'2^\markup {
  \override #'(fret-diagram-details . (
               (finger-code . below-string)
               (dot-radius . 0.35)
               (dot-position . 0.5)
               (top-fret-thickness . 7)
               (fret-count . 3))) {
    \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
  }
}

  % These chords will be in opposing-landscape orientation
\override TextScript.fret-diagram-details.orientation = #'opposing-landscape

%% C major for guitar, barred on third fret
%  verbose style
%  roman fret label, finger labels below string, straight barre
c'2^\markup {
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                 (number-type . roman-lower)
                 (finger-code . below-string)
                 (barre-type . straight))) {
      \fret-diagram-verbose #'((mute 6)
                               (place-fret 5 3 1)
                               (place-fret 4 5 2)
                               (place-fret 3 5 3)
                               (place-fret 2 5 4)
                               (place-fret 1 3 1)
                               (barre 5 1 3))
    }
  }
}

%% C major for guitar, barred on third fret
```

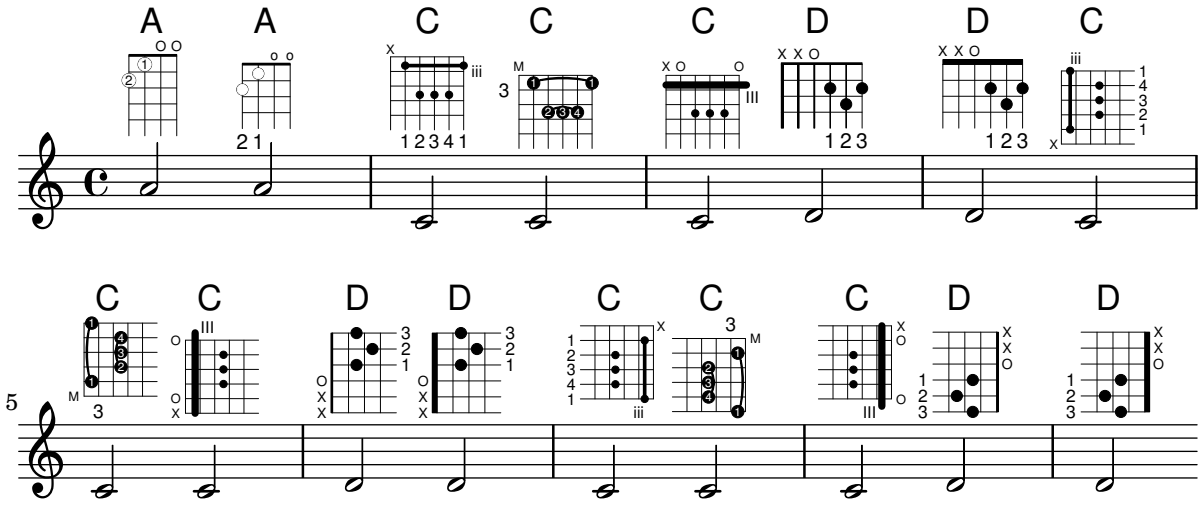
```
%% Double barre used to test barre function
%  verbose style
c'2^\markup {
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                 (number-type . arabic)
                 (dot-label-font-mag . 0.9)
                 (finger-code . in-dot)
                 (fret-label-font-mag . 0.6)
                 (fret-label-vertical-offset . 0)
                 (label-dir . -1)
                 (mute-string . "M")
                 (xo-font-magnification . 0.4)
                 (xo-padding . 0.3))) {
      \fret-diagram-verbose #'((mute 6)
                               (place-fret 5 3 1)
                               (place-fret 4 5 2)
                               (place-fret 3 5 3)
                               (place-fret 2 5 4)
                               (place-fret 1 3 1)
                               (barre 4 2 5)
                               (barre 5 1 3))
    }
  }
}

%% C major for guitar, with capo on third fret
%  verbose style
c'2^\markup {
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                 (number-type . roman-upper)
                 (dot-label-font-mag . 0.9)
                 (finger-code . none)
                 (fret-label-vertical-offset . 0.5)
                 (xo-font-magnification . 0.4)
                 (xo-padding . 0.3))) {
      \fret-diagram-verbose #'((mute 6)
                               (capo 3)
                               (open 5)
                               (place-fret 4 5 1)
                               (place-fret 3 5 2)
                               (place-fret 2 5 3)
                               (open 1))
    }
  }
}

%% simple D chord
d'2^\markup {
```

```
      \override #'(fret-diagram-details . (
                   (finger-code . below-string)
                   (dot-radius . 0.35)
                   (dot-position . 0.5)
                   (fret-count . 3))) {
        \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
      }
    }

    %% simple D chord, large top fret thickness
    d'2^\markup {
      \override #'(fret-diagram-details . (
                   (finger-code . below-string)
                   (dot-radius . 0.35)
                   (dot-position . 0.5)
                   (top-fret-thickness . 7)
                   (fret-count . 3))) {
        \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
      }
    }
  }
>>
```



Diagrammes de fret alternatifs

Vous pouvez tout à fait créer des tables de diagrammes de fret supplémentaires, notamment dans l'optique d'un fret alternatif pour un accord donné.

Avant de pouvoir utiliser un diagramme alternatif, vous devrez avoir alimenté une table à cet effet. Les différents diagrammes seront ajoutés à cette table.

Il peut aussi bien s'agir d'une table vide, que de la recopie d'une table existante.

La table servant de base pour les diagrammes prédéfinis est sélectionnée par la propriété `\predefinedDiagramTable`.

```
\include "predefined-guitar-fretboards.ly"

% Make a blank new fretboard table
#(define custom-fretboard-table-one
```

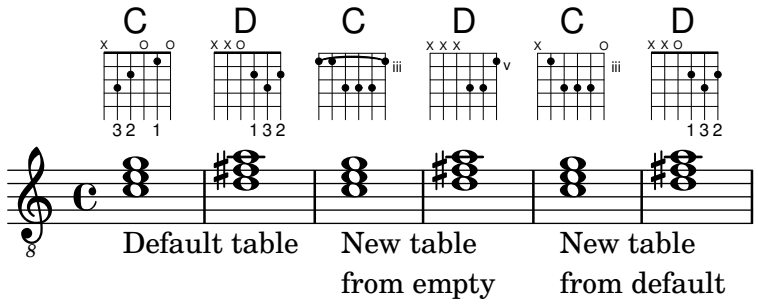
```
   (make-fretboard-table))

% Make a new fretboard table as a copy of default-fret-table
#(define custom-fretboard-table-two
   (make-fretboard-table default-fret-table))

% Add a chord to custom-fretboard-table-one
\storePredefinedDiagram #custom-fretboard-table-one
                        \chordmode {c}
                        #guitar-tuning
                        "3-(;3;5;5;5;3-);"

% Add a chord to custom-fretboard-table-two
\storePredefinedDiagram #custom-fretboard-table-two
                        \chordmode {c}
                        #guitar-tuning
                        "x;3;5;5;5;o;"

<<
  \chords {
    c1 | d1 |
    c1 | d1 |
    c1 | d1 |
  }
  \new FretBoards {
    \chordmode {
      \set predefinedDiagramTable = #default-fret-table
      c1 | d1 |
      \set predefinedDiagramTable = #custom-fretboard-table-one
      c1 | d1 |
      \set predefinedDiagramTable = #custom-fretboard-table-two
      c1 | d1 |
    }
  }
  \new Staff {
    \clef "treble_8"
    <<
      \chordmode {
        c1 | d1 |
        c1 | d1 |
        c1 | d1 |
      }
      {
        s1_\markup "Default table" | s1 |
        s1_\markup \column {"New table" "from empty"} | s1 |
        s1_\markup \column {"New table" "from default"} | s1 |
      }
    >>
  }
>>
```



Harmoniques et tablature

Harmoniques et tablature (harmoniques artificielles).

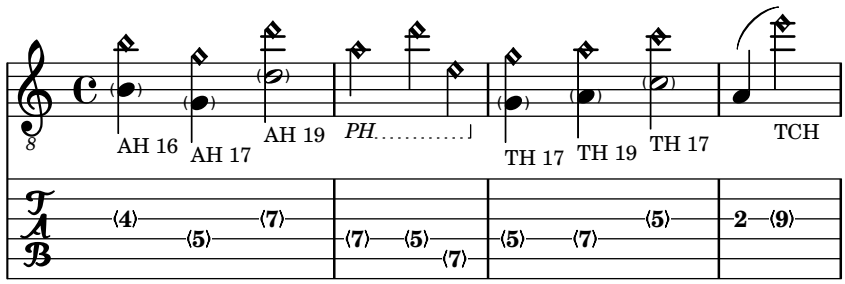
```
pinchedHarmonics = {
   \textSpannerDown
   \override TextSpanner.bound-details.left.text =
      \markup {\halign #-0.5 \teeny "PH" }
      \override TextSpanner.style =
         #'dashed-line
   \override TextSpanner.dash-period = #0.6
   \override TextSpanner.bound-details.right.attach-dir = #1
   \override TextSpanner.bound-details.right.text =
      \markup { \draw-line #'(0 . 1) }
   \override TextSpanner.bound-details.right.padding = #-0.5
}

harmonics = {
  %artificial harmonics (AH)
  \textLengthOn
  <\parenthesize b b'\harmonic>4_\markup { \teeny "AH 16" }
  <\parenthesize g g'\harmonic>4_\markup { \teeny "AH 17" }
  <\parenthesize d' d''\harmonic>2_\markup { \teeny "AH 19" }
  %pinched harmonics (PH)
  \pinchedHarmonics
  <a'\harmonic>2\startTextSpan
  <d''\harmonic>4
  <e'\harmonic>4\stopTextSpan
  %tapped harmonics (TH)
  <\parenthesize g\4 g'\harmonic>4_\markup { \teeny "TH 17" }
  <\parenthesize a\4 a'\harmonic>4_\markup { \teeny "TH 19" }
  <\parenthesize c'\3 c''\harmonic>2_\markup { \teeny "TH 17" }
  %touch harmonics (TCH)
  a4( <e''\harmonic>2. )_\markup { \teeny "TCH" }
}

frettedStrings = {
  %artificial harmonics (AH)
  \harmonicByFret #4 g4\3
  \harmonicByFret #5 d4\4
  \harmonicByFret #7 g2\3
  %pinched harmonics (PH)
  \harmonicByFret #7 d2\4
  \harmonicByFret #5 d4\4
  \harmonicByFret #7 a4\5
  %tapped harmonics (TH)
```

```
  \harmonicByFret #5 d4\4
  \harmonicByFret #7 d4\4
  \harmonicByFret #5 g2\3
  %touch harmonics (TCH)
  a4 \harmonicByFret #9 g2.\3
}

\score {
  <<
    \new Staff
    \with { \omit StringNumber } {
      \new Voice {
        \clef "treble_8"
        \harmonics
      }
    }
    \new TabStaff {
      \new TabVoice {
        \frettedStrings
      }
    }
  >>
}
```



Indication d'un glissé de guitare

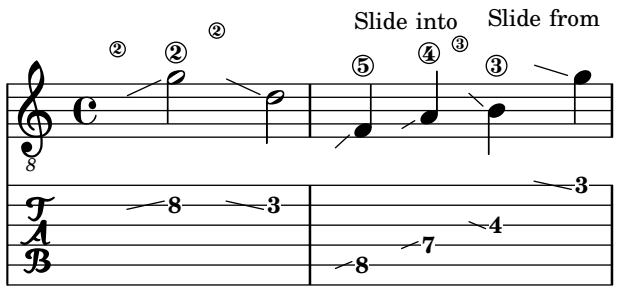
Contrairement au glissando, un « slide » peut aller d'un point non précisé du manche jusqu'à un fret spécifique. Ceci peut s'indiquer à l'aide d'une note d'ornement masquée précédant la note effectivement jouée, comme dans l'exemple suivant.

```
%% Hide fret number: useful to draw slide into/from a casual point of
%% the fretboard.
hideFretNumber = {
  \once \hide TabNoteHead
  \once \hide NoteHead
  \once \hide Stem
  \once \override NoteHead.no-ledgers = ##t
  \once \override Glissando.bound-details.left.padding = #0.3
}

music= \relative c' {
  \grace { \hideFretNumber d8\2 \glissando s2 } g2\2
  \grace { \hideFretNumber g8\2 \glissando s2 } d2 |
```

```
  \grace { \hideFretNumber c,8 \glissando s } f4\5^\markup \tiny { Slide into }
  \grace { \hideFretNumber f8 \glissando s } a4\4
  \grace { \hideFretNumber e'8\3 \glissando s } b4\3^\markup \tiny { Slide from }
  \grace { \hideFretNumber b'8 \glissando s2 } g4 |
}

\score {
  <<
    \new Staff {
      \clef "G_8"
      \music
    }
    \new TabStaff {
      \music
    }
  >>
}
```

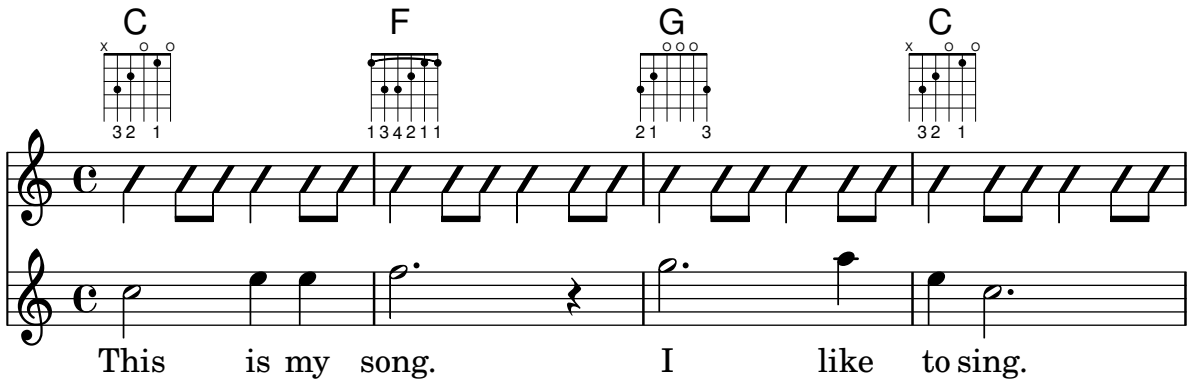


Rythmique et guitare

En matière de notation pour guitare, il arrive que soient indiqués les « coups de gratte » en plus de la mélodie, grilles d'accords et diagrammes de tablature.

```
\include "predefined-guitar-fretboards.ly"
<<
  \new ChordNames {
    \chordmode {
      c1 | f | g | c
    }
  }
  \new FretBoards {
    \chordmode {
      c1 | f | g | c
    }
  }
  \new Voice \with {
    \consists "Pitch_squash_engraver"
  } {
    \relative c'' {
      \improvisationOn
      c4 c8 c c4 c8 c
      f4 f8 f f4 f8 f
      g4 g8 g g4 g8 g
      c4 c8 c c4 c8 c
```

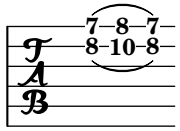
```
    }
  }
  \new Voice = "melody" {
    \relative c'' {
      c2 e4 e4
      f2. r4
      g2. a4
      e4 c2.
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
      This is my song.
      I like to sing.
    }
  }
>>
```



Hammer on et pull off avec accords

Dans le cadre de notes en accord, les *hammer-on* et *pull-off* sont indiqués par un arc simple. Vous obtiendrez néanmoins un arc double en réglant la propriété `doubleSlurs` sur `#t`.

```
\new TabStaff {
  \relative c' {
    % chord hammer-on and pull-off
    \set doubleSlurs = ##t
    <g' b>8( <a c> <g b>)
  }
}
```

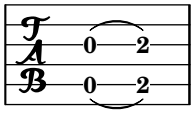


Hammer on et pull off gérés par les voix

L'arc des *hammer-on* et *pull-off* est ascendant dans les voix une et trois, et descendant dans les voix deux et quatre.

```
\new TabStaff {
```

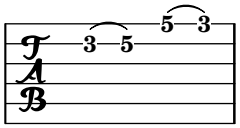
```
  \relative c' {
    << { \voiceOne g2( a) }
    \\ { \voiceTwo a,( b) }
    >> \oneVoice
  }
}
```



Hammer on et pull off

Hammer-on et *pull-off* peuvent s'indiquer par des liaisons.

```
\new TabStaff {
  \relative c' {
    d4( e\2)
    a( g)
  }
}
```



Repositionnement d'un diagramme de fret

Différents moyens permettent de repositionner un diagramme de fret pour éviter des collisions ou le placer entre deux notes :

– La modification des valeurs de `padding` ou de `extra-offset` comme pour le second diagramme ;

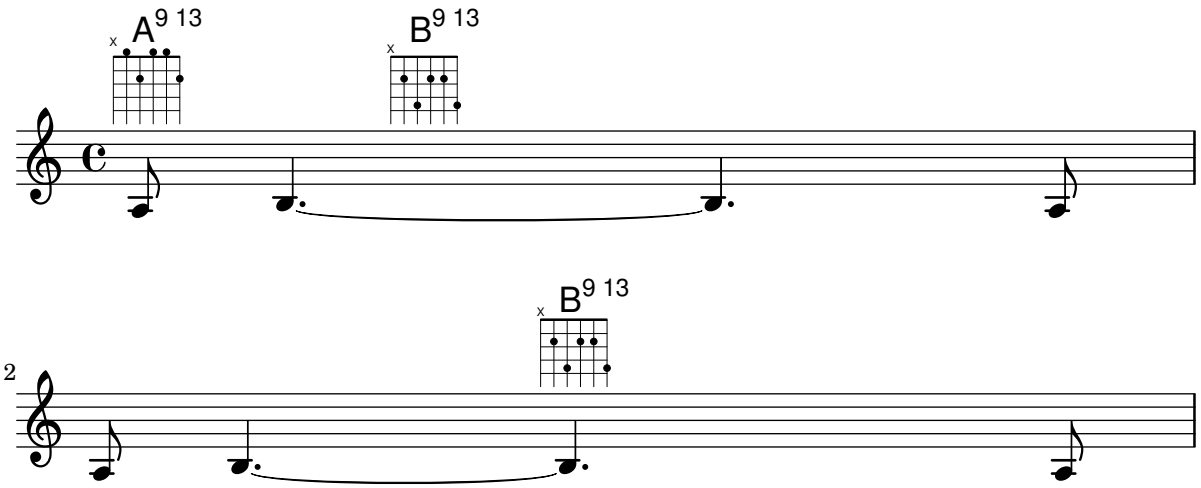
– L'adjonction d'une voix invisible dans laquelle les diagrammes sont attachés à des notes invisibles comme pour le troisième diagramme.

Lorsque le diagramme doit correspondre à une position rythmique dans la mesure, comme au troisième temps de la deuxième mesure, la seconde méthode est plus appropriée puisque le diagramme sera aligné sur le temps.

```
harmonies = \chordmode
{
  a8:13
% THE FOLLOWING IS THE COMMAND TO MOVE THE CHORD NAME
  \once \override ChordNames.ChordName.extra-offset = #'(10 . 0)
  b8:13 s2.
% THIS LINE IS THE SECOND METHOD
    s4 s4  b4:13
}

\score
{
  <<
    \new ChordNames \harmonies
```

```
    \new Staff
    {a8^\markup { \fret-diagram  "6-x;5-0;4-2;3-0;2-0;1-2;"  }
% THE FOLLOWING IS THE COMMAND TO MOVE THE FRET DIAGRAM
     \once \override TextScript.extra-offset = #'(10 . 0)
     b4.~^\markup { \fret-diagram  "6-x;5-2;4-4;3-2;2-2;1-4;"  } b4. a8\break
% HERE IS THE SECOND METHOD
     <<
       { a8 b4.~ b4. a8}
       { s4 s4 s4^\markup { \fret-diagram  "6-x;5-2;4-4;3-2;2-2;1-4;"  }
       }
     >>
   }
  >>
}
```



Symboles de jazz

Bien que compliqué de prime abord, voici un canevas tout à fait indiqué pour les ensembles de jazz. Vous noterez que tous les instruments sont notés en ut (`\key c \major`), la tonalité de concert. Les notes seront automatiquement transposées dès lors qu'elles seront inscrites dans une section `\transpose`.

```
\header {
  title = "Song"
  subtitle = "(tune)"
  composer = "Me"
  meter = "moderato"
  piece = "Swing"
  tagline = \markup {
    \column {
      "LilyPond example file by Amelie Zapf,"
      "Berlin 07/07/2003"
    }
  }
}
% To make the example display in the documentation
\paper {
  paper-width = 130
```

```
}
%#(set-global-staff-size 16)
\include "english.ly"

%%%%%%%%%%% Some macros %%%%%%%%%%%%%%%%%

sl = {
  \override NoteHead.style = #'slash
  \hide Stem
}
nsl = {
  \revert NoteHead.style
  \undo \hide Stem
}
crOn = \override NoteHead.style = #'cross
crOff = \revert NoteHead.style

%% insert chord name style stuff here.

jazzChords = { }

%%%%%%%%%%% Keys'n'thangs %%%%%%%%%%%%%%%%

global = { \time 4/4 }

Key = { \key c \major }

% ############ Horns ############

% ------ Trumpet ------
trpt = \transpose c d \relative c'' {
  \Key
  c1 | c | c |
}
trpHarmony = \transpose c' d {
  \jazzChords
}
trumpet = {
  \global
  \clef treble
  <<
    \trpt
  >>
}

% ------ Alto Saxophone ------
alto = \transpose c a \relative c' {
  \Key
  c1 | c | c |
}
altoHarmony = \transpose c' a {
  \jazzChords
```

```
}
altoSax = {
  \global
  \clef treble
  <<
    \alto
  >>
}

% ------ Baritone Saxophone ------
bari = \transpose c a' \relative c {
  \Key
  c1
  c1
  \sl
  d4^"Solo" d d d
  \nsl
}
bariHarmony = \transpose c' a \chordmode {
  \jazzChords s1 s d2:maj e:m7
}
bariSax = {
  \global
  \clef treble
  <<
    \bari
  >>
}

% ------ Trombone ------
tbone = \relative c {
  \Key
  c1 | c | c
}
tboneHarmony = \chordmode {
  \jazzChords
}
trombone = {
  \global
  \clef bass
  <<
    \tbone
  >>
}

% ############ Rhythm Section ##############

% ------ Guitar ------
gtr = \relative c'' {
  \Key
  c1
  \sl
```

```
  b4 b b b
  \nsl
  c1
}
gtrHarmony = \chordmode {
  \jazzChords
  s1 c2:min7+ d2:maj9
}
guitar = {
  \global
  \clef treble
  <<
    \gtr
  >>
}

%% ------ Piano ------
rhUpper = \relative c'' {
  \voiceOne
  \Key
  c1 | c | c
}
rhLower = \relative c' {
  \voiceTwo
  \Key
  e1 | e | e
}

lhUpper = \relative c' {
  \voiceOne
  \Key
  g1 | g | g
}
lhLower = \relative c {
  \voiceTwo
  \Key
  c1 | c | c
}

PianoRH = {
  \clef treble
  \global
  <<
    \new Voice = "one" \rhUpper
    \new Voice = "two" \rhLower
  >>
}
PianoLH = {
  \clef bass
  \global
  <<
    \new Voice = "one" \lhUpper
```

```
    \new Voice = "two" \lhLower
  >>
}

piano = {
  <<
    \new Staff = "upper" \PianoRH
    \new Staff = "lower" \PianoLH
  >>
}

% ------ Bass Guitar ------
Bass = \relative c {
  \Key
  c1 | c | c
}
bass = {
  \global
  \clef bass
  <<
    \Bass
  >>
}

% ------ Drums ------
up = \drummode {
  \voiceOne
  hh4 <hh sn> hh <hh sn>
  hh4 <hh sn> hh <hh sn>
  hh4 <hh sn> hh <hh sn>
}
down = \drummode {
  \voiceTwo
  bd4 s bd s
  bd4 s bd s
  bd4 s bd s
}

drumContents = {
  \global
  <<
    \new DrumVoice \up
    \new DrumVoice \down
  >>
}

%%%%%%%% It All Goes Together Here %%%%%%%%%%%%%%%%%%%%%

\score {
  <<
    \new StaffGroup = "horns" <<
      \new Staff = "trumpet" \with { instrumentName = "Trumpet" }
```

```
      \trumpet
      \new Staff = "altosax" \with { instrumentName = "Alto Sax" }
      \altoSax
      \new ChordNames = "barichords" \with { instrumentName = "Trumpet" }
      \bariHarmony
      \new Staff = "barisax" \with { instrumentName = "Bari Sax" }
      \bariSax
      \new Staff = "trombone" \with { instrumentName = "Trombone" }
      \trombone
    >>

    \new StaffGroup = "rhythm" <<
      \new ChordNames = "chords" \gtrHarmony
      \new Staff = "guitar" \with { instrumentName = "Guitar" }
      \guitar
      \new PianoStaff = "piano" \with {
        instrumentName = "Piano"
        midiInstrument = "acoustic grand"
      }
      \piano
      \new Staff = "bass" \with { instrumentName = "Bass" }
      \bass
      \new DrumStaff \with { instrumentName = "Drums" }
      \drumContents
    >>
  >>
  \layout {
    \context { \Staff \RemoveEmptyStaves }
    \context {
      \Score
      \override BarNumber.padding = #3
      \override RehearsalMark.padding = #2
      skipBars = ##t
    }
  }
  \midi { }
}
```

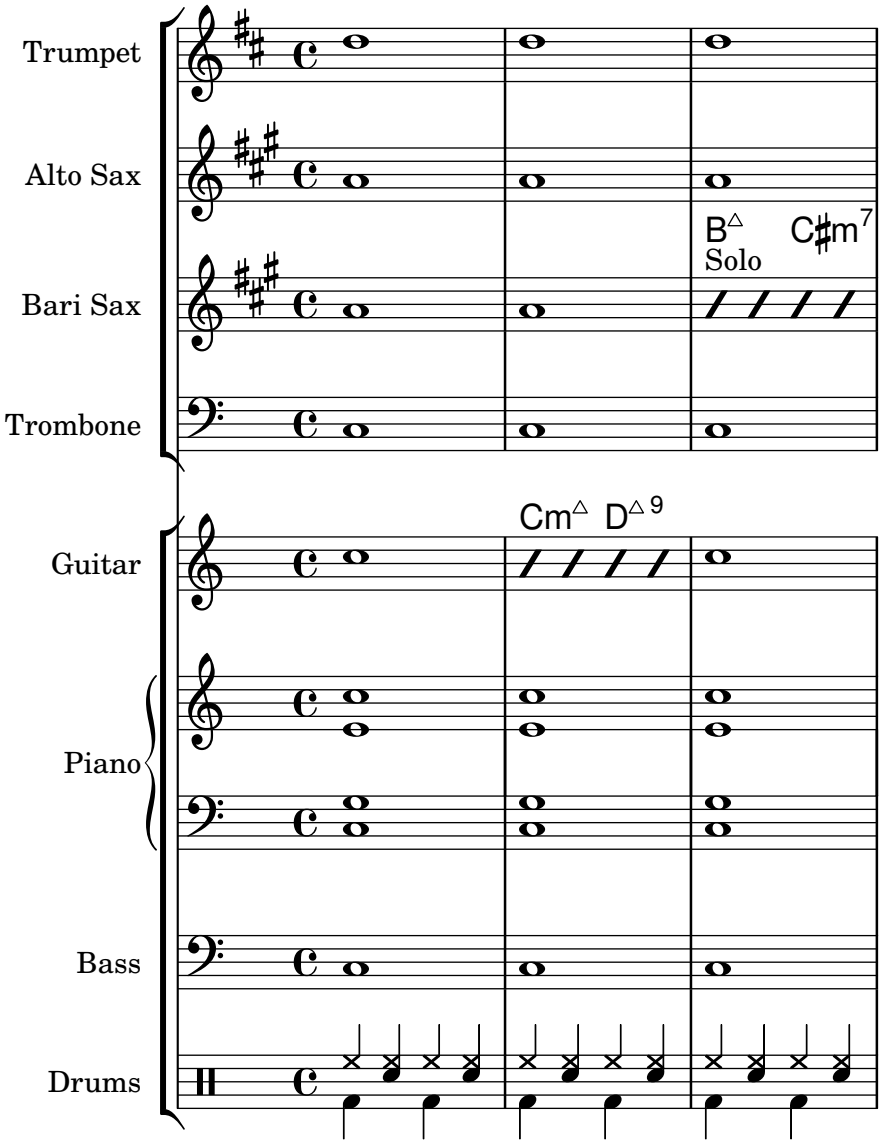
Song

(tune)

Me

moderato

Swing
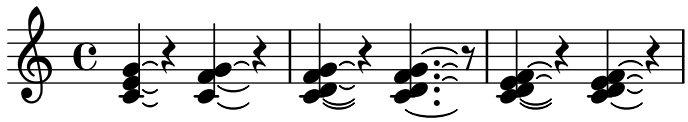


Liaison « Laissez vibrer »

Les indications de « laisser vibrer » ont une taille fixe. Leur formatage est accessible au travers de la propriété `tie-configuration`.

```
\relative c' {
  <c e g>4\laissezVibrer r <c f g>\laissezVibrer r
  <c d f g>4\laissezVibrer r <c d f g>4.\laissezVibrer r8

  <c d e f>4\laissezVibrer r
  \override LaissezVibrerTieColumn.tie-configuration
     = #`((-7 . ,DOWN)
          (-5 . ,DOWN)
          (-3 . ,UP)
          (-1 . ,UP))
  <c d e f>4\laissezVibrer r
}
```



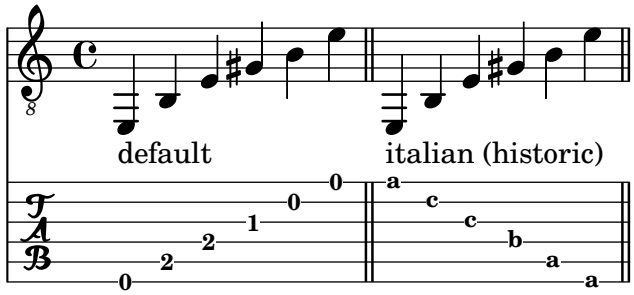
Impression de la corde aiguë d'un `TabStaff` en bas

Il est d'usage, en matière de tablature, d'imprimer la première corde en haut. Il est toutefois possible de la positionner en bas grâce à une modification de la propriété de contexte `stringOneTopmost`. Définir ce réglage au sein d'un bloc `\layout` l'appliquera à l'intégralité du contexte.

```
%
%\layout {
%  \context {
%    \Score
%    stringOneTopmost = ##f
%  }
%  \context {
%    \TabStaff
%    tablatureFormat = #fret-letter-tablature-format
%  }
%}

m = {
  \cadenzaOn
  e, b, e gis! b e'
  \bar "||"
}

<<
  \new Staff { \clef "G_8" <>_"default" \m <>_"italian (historic)"\m }
  \new TabStaff
  {
    \m
    \set Score.stringOneTopmost = ##f
    \set TabStaff.tablatureFormat = #fret-letter-tablature-format
    \m
  }
>>
```

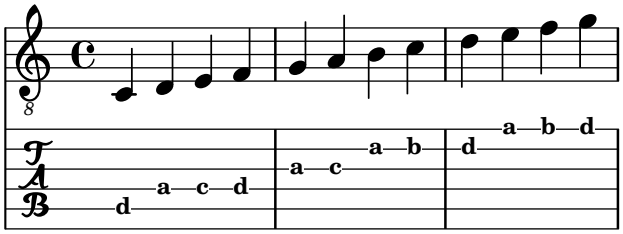


Tablature en lettres

Une tablature peut comporter des lettres en lieu et place des chiffres.

```
music = \relative c {
  c4 d e f
  g4 a b c
  d4 e f g
}
```

```
<<
  \new Staff {
    \clef "G_8"
    \music
  }
  \new TabStaff \with {
    tablatureFormat = #fret-letter-tablature-format
  }
  {
    \music
  }
>>
```



Harmoniques sur corde à vide en tablature

Voici comment obtenir des harmoniques sur corde à vide (harmoniques naturelles) dans une tablature.

```
openStringHarmonics = {
  \textSpannerDown
  \override TextSpanner.staff-padding = #3
  \override TextSpanner.dash-fraction = #0.3
  \override TextSpanner.dash-period = #1

  %first harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "1st harm. "
  \harmonicByFret #12 e,2\6\startTextSpan
  \harmonicByRatio #1/2 e,\6\stopTextSpan

  %second harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "2nd harm. "
  \harmonicByFret #7 e,\6\startTextSpan
  \harmonicByRatio #1/3 e,\6
  \harmonicByFret #19 e,\6
  \harmonicByRatio #2/3 e,\6\stopTextSpan
  %\harmonicByFret #19 < e,\6 a,\5 d\4 >
  %\harmonicByRatio #2/3 < e,\6 a,\5 d\4 >

  %third harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "3rd harm. "
  \harmonicByFret #5 e,\6\startTextSpan
```

```
  \harmonicByRatio #1/4 e,\6
  \harmonicByFret #24 e,\6
  \harmonicByRatio #3/4 e,\6\stopTextSpan
  \break

  %fourth harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "4th harm. "
  \harmonicByFret #4 e,\6\startTextSpan
  \harmonicByRatio #1/5 e,\6
  \harmonicByFret #9 e,\6
  \harmonicByRatio #2/5 e,\6
  \harmonicByFret #16 e,\6
  \harmonicByRatio #3/5 e,\6\stopTextSpan

  %fifth harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "5th harm. "
  \harmonicByFret #3 e,\6\startTextSpan
  \harmonicByRatio #1/6 e,\6\stopTextSpan
  \break

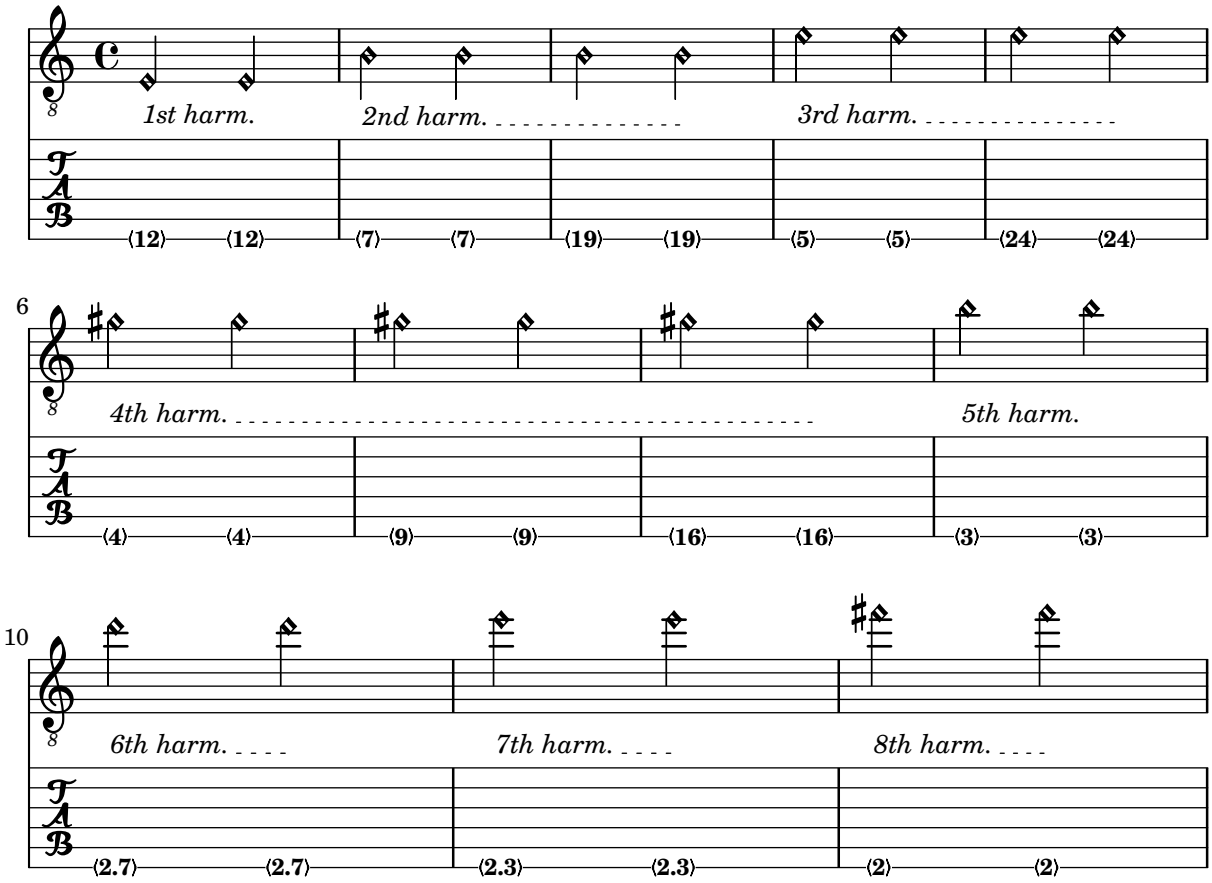
  %sixth harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "6th harm. "
  \harmonicByFret #2.7 e,\6\startTextSpan
  \harmonicByRatio #1/7 e,\6\stopTextSpan

  %seventh harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "7th harm. "
  \harmonicByFret #2.3 e,\6\startTextSpan
  \harmonicByRatio #1/8 e,\6\stopTextSpan

  %eighth harmonic
  \override TextSpanner.bound-details.left.text =
    \markup\small "8th harm. "
  \harmonicByFret #2 e,\6\startTextSpan
  \harmonicByRatio #1/9 e,\6\stopTextSpan
}

\score {
  <<
    \new Staff
    \with { \omit StringNumber } {
      \new Voice {
        \clef "treble_8"
        \openStringHarmonics
      }
    }
    \new TabStaff {
      \new TabVoice {
```

```
        \openStringHarmonics
      }
    }
  >>
}
```



Positionnement des doigtés main droite

Le positionnement des doigtés main droite, grâce à une propriété spécifique, peut se contrôler finement, comme l'indique l'exemple suivant.

```
#(define RH rightHandFinger)

\relative c {
  \clef "treble_8"

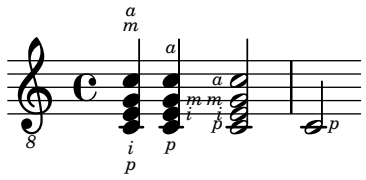
  \set strokeFingerOrientations = #'(up down)
  <c\RH #1 e\RH #2 g\RH #3 c\RH #4 >4

  \set strokeFingerOrientations = #'(up right down)
  <c\RH #1 e\RH #2 g\RH #3 c\RH #4 >4

  \set strokeFingerOrientations = #'(left)
  <c\RH #1 e\RH #2 g\RH #3 c\RH #4 >2

  \set strokeFingerOrientations = #'(right)
  c\RH #1
```

```
}
```



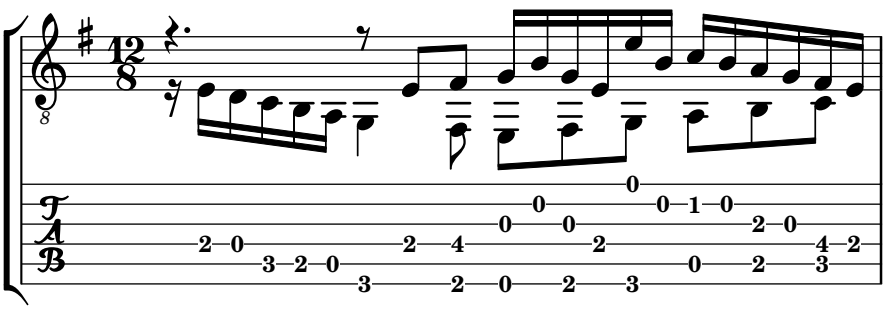
Polyphonie en mode tablature

Une section polyphonique s'obtient dans un `TabStaff` de la même manière que dans une portée normale.

```
upper = \relative c' {
  \time 12/8
  \key e \minor
  \voiceOne
  r4. r8 e, fis g16 b g e e' b c b a g fis e
}

lower = \relative c {
  \key e \minor
  \voiceTwo
  r16 e d c b a g4 fis8 e fis g a b c
}

\score {
  <<
    \new StaffGroup = "tab with traditional" <<
      \new Staff = "guitar traditional" <<
        \clef "treble_8"
        \new Voice = "upper" \upper
        \new Voice = "lower" \lower
      >>
      \new TabStaff = "guitar tab" <<
        \new TabVoice = "upper" \upper
        \new TabVoice = "lower" \lower
      >>
    >>
  >>
}
```

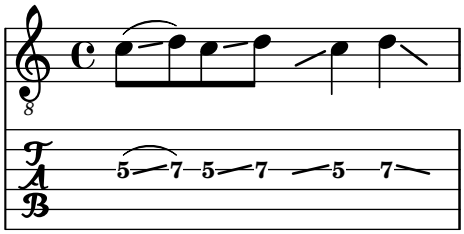


Glissando et tablature

Un glissando s'indique dans un `TabStaff` tout comme dans un `Staff`.

```
slides = {
  c'8\3(\glissando d'8\3)
  c'8\3\glissando d'8\3
  \hideNotes
  \grace { g16\glissando }
  \unHideNotes
  c'4\3
  \afterGrace d'4\3\glissando {
  \stemDown \hideNotes
  g16 }
  \unHideNotes
}

\score {
  <<
    \new Staff { \clef "treble_8" \slides }
    \new TabStaff { \slides }
  >>
  \layout {
    \context {
      \Score
      \override Glissando.minimum-length = #4
      \override Glissando.springs-and-rods =
                          #ly:spanner::set-spacing-rods
      \override Glissando.thickness = #2
      \omit StringNumber
      % or:
      %\override StringNumber.stencil = ##f
    }
  }
}
```

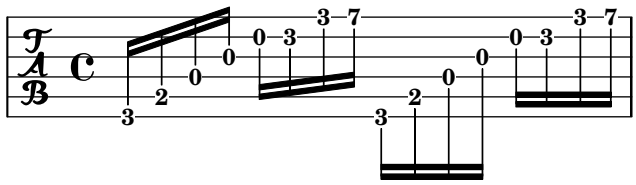


Hampes et ligatures en mode tablature

La direction des hampes se gère dans les tablatures tout comme en notation traditionnelle. Les ligatures peuvent être mises à l'horizontale comme le montre cet exemple.

```
\new TabStaff {
  \relative c {
    \tabFullNotation
    g16 b d g b d g b
    \stemDown
```

```
    \override Beam.concaveness = #10000
    g,,16 b d g b d g b
  }
}
```

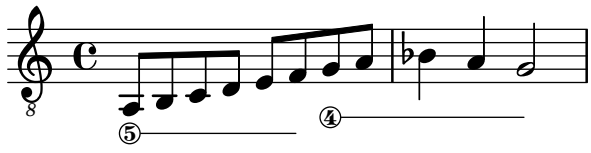


Ligne de prolongation pour numéro de corde

Voici comment ajouter une ligne de prolongation à une indication de numéro de corde, afin de stipuler que les notes qui suivent doivent être jouées sur la corde en question.

```
stringNumberSpanner =
  #(define-music-function (StringNumber) (string?)
    #{
      \override TextSpanner.style = #'solid
      \override TextSpanner.font-size = #-5
      \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
      \override TextSpanner.bound-details.left.text =
        \markup { \circle \number $StringNumber }
    #})


\relative c {
  \clef "treble_8"
  \stringNumberSpanner "5"
  \textSpannerDown
  a8\startTextSpan
  b c d e f\stopTextSpan
  \stringNumberSpanner "4"
  g\startTextSpan a
  bes4 a g2\stopTextSpan
}
```



Unfretted strings
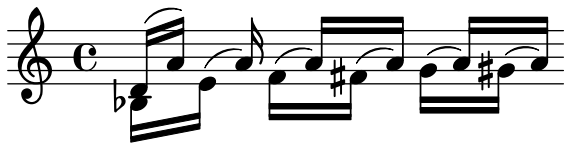
Section “Unfretted string instruments” dans *Manuel de notation*

Création d’une liaison entre plusieurs voix

Dans certaines situations, il peut être nécessaire de lier des notes appartenant à des voix différentes. La solution consiste à ajouter une note invisible dans l’une des voix, à l’aide de la commande `\hideNotes`.

Voici par exemple la mesure 235 de la chaconne de la seconde partita pour violon solo, BWV 1004, de Bach.
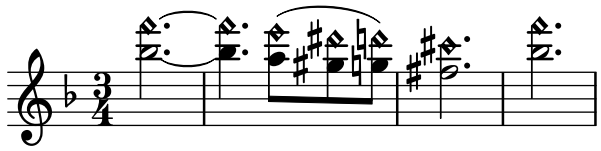
```
\relative c' {
  <<
    {
      d16( a') s a s a[ s a] s a[ s a]
    }
    \\
    {
      \slurUp
      bes,16[ s e](
      \hideNotes a)
      \unHideNotes f[(
      \hideNotes a)
      \unHideNotes fis](
      \hideNotes a)
      \unHideNotes g[(
      \hideNotes a)
      \unHideNotes gis](
      \hideNotes a)
    }
  >>
}
```



Harmoniques pointées

Les notes harmoniques artificielles, obtenues grâce à `\harmonic`, ne sont pas pointées. Ce comportement peut être modifié en activant la propriété de contexte `harmonicDots`.

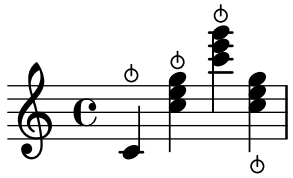
```
\relative c''' {
  \time 3/4
  \key f \major
  \set harmonicDots = ##t
  <bes f'\harmonic>2. ~
  <bes f'\harmonic>4. <a e'\harmonic>8( <gis dis'\harmonic> <g d'\harmonic>)
  <fis cis'\harmonic>2.
  <bes f'\harmonic>2.
}
```



Slap ou pizzicato Bartok

Un *slap*, aussi appelé *pizzicato à la Bartók*, est un pizzicato où, au lieu de tirer la corde de côté comme à l'accoutumée, « on la pince en la soulevant plus fortement, et à la verticale, en la lâchant violemment. Celle-ci frappe alors la touche, et produit à la fois la note et le son percussif » (Wikipedia). Il s'indique par un cercle flanqué d'un trait vertical.

```
\relative c' {
  c4\snappizzicato
  <c' e g>4\snappizzicato
  <c' e g>4^\snappizzicato
  <c, e g>4_\snappizzicato
}
```



Quatuor à cordes (conducteur)

Voici un canevas pour quatuor à cordes. Notez l'utilisation de la variable `\global` pour traiter la métrique et la tonalité.

```
global= {
  \time 4/4
  \key c \major
}

violinOne = \new Voice \relative c'' {
  c2 d
  e1
  \bar "|."
}

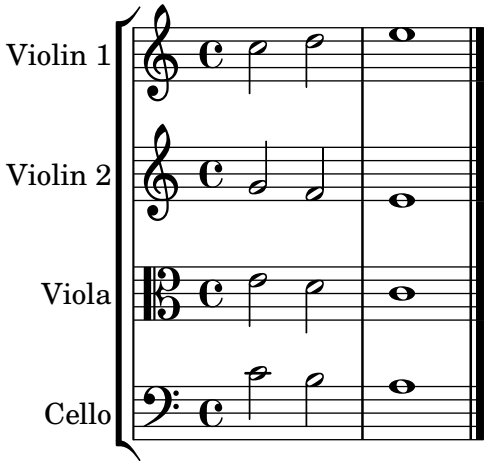
violinTwo = \new Voice \relative c'' {
  g2 f
  e1
  \bar "|."
}

viola = \new Voice \relative c' {
  \clef alto
  e2 d
  c1
  \bar "|."
}

cello = \new Voice \relative c' {
  \clef bass
```

```
  c2 b
  a1
  \bar "|."
}

\score {
  \new StaffGroup <<
    \new Staff \with { instrumentName = "Violin 1" }
    << \global \violinOne >>
    \new Staff \with { instrumentName = "Violin 2" }
    << \global \violinTwo >>
    \new Staff \with { instrumentName = "Viola" }
    << \global \viola >>
    \new Staff \with { instrumentName = "Cello" }
    << \global \cello >>
  >>
  \layout { }
  \midi { }
}
```



Quatuor à cordes, avec parties séparées

Grâce à ce canevas, vous pouvez obtenir une partition d'excellente facture pour quatuor à cordes mais aussi, si le besoin s'en faisait sentir, une partie séparée par instrument. Par ailleurs, cet exemple illustre l'utilisation de la fonction `\tag` dans le but d'extraire des parties séparées.

Il vous faudra découper ce canevas en plusieurs fichiers séparés ; leur nom respectif est indiqué en commentaire : `piece.ly` comporte tout ce qui a trait à la musique, les autres fichiers – `score.ly`, `vn1.ly`, `vn2.ly`, `vla.ly` et `vlc.ly` – vous permettront d'obtenir les parties selon le pupitre.

N'oubliez pas de supprimer les commentaires superflus des fichiers individualisés !

```
%%%%% piece.ly
%%%%% (This is the global definitions file)

global= {
  \time 4/4
  \key c \major
}
```

```
Violinone = \new Voice {
  \relative c'' {
    c2 d e1
    \bar "|."
  }
}


Violintwo = \new Voice {
  \relative c'' {
    g2 f e1
    \bar "|."
  }
}


Viola = \new Voice {
  \relative c' {
    \clef alto
    e2 d c1
    \bar "|."
  }
}


Cello = \new Voice {
  \relative c' {
    \clef bass
    c2 b a1
    \bar "|."
  }
}


music = {
  <<
    \tag #'score \tag #'vn1
    \new Staff \with { instrumentName = "Violin 1" }
    << \global \Violinone >>

    \tag #'score \tag #'vn2
    \new Staff \with { instrumentName = "Violin 2" }
    << \global \Violintwo>>

    \tag #'score \tag #'vla
    \new Staff \with { instrumentName = "Viola" }
    << \global \Viola>>

    \tag #'score \tag #'vlc
    \new Staff \with { instrumentName = "Cello" }
```

```
    << \global \Cello >>
  >>
}

% These are the other files you need to save on your computer

% score.ly
% (This is the main file)

% uncomment the line below when using a separate file
%\include "piece.ly"

#(set-global-staff-size 14)

\score {
  \new StaffGroup \keepWithTag #'score \music
  \layout { }
  \midi { }
}


%{ Uncomment this block when using separate files

% vn1.ly
% (This is the Violin 1 part file)

\include "piece.ly"
\score {
  \keepWithTag #'vn1 \music
  \layout { }
}


% vn2.ly
% (This is the Violin 2 part file)

\include "piece.ly"
\score {
  \keepWithTag #'vn2 \music
  \layout { }
}

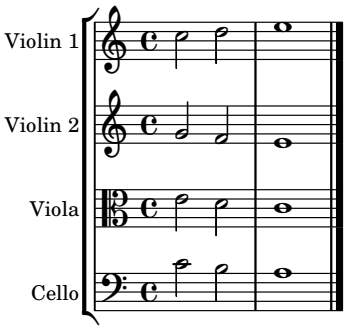

% vla.ly
% (This is the Viola part file)

\include "piece.ly"
\score {
  \keepWithTag #'vla \music
  \layout { }
}
```

```
% vlc.ly
% (This is the Cello part file)

\include "piece.ly"
\score {
  \keepWithTag #'vlc \music
  \layout { }
}

%}
```



Winds

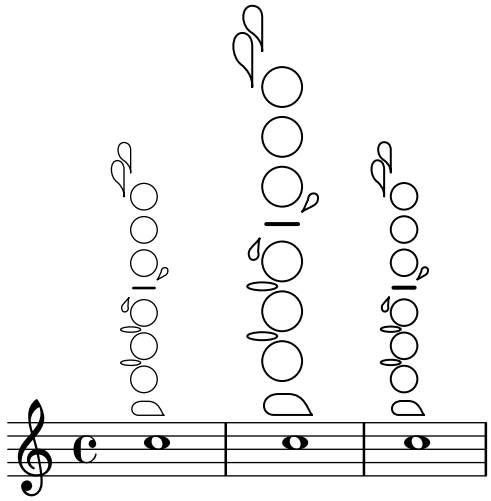
Section “Wind instruments” dans *Manuel de notation*

Modification de la taille d’un diagramme pour bois

La taille et l’épaisseur des diagrammes de doigté pour bois est modifiable à souhait.

```
\relative c'' {
  \textLengthOn
  c1^\markup
    \woodwind-diagram
      #'piccolo
      #'()

  c^\markup
    \override #'(size . 1.5) {
      \woodwind-diagram
        #'piccolo
        #'()
    }
  c^\markup
    \override #'(thickness . 0.15) {
      \woodwind-diagram
        #'piccolo
        #'()
    }
}
```



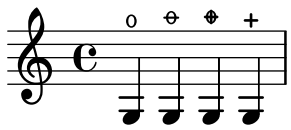
Symboles de doigtés pour instruments à vent

Des symboles spécifiques peuvent être obtenus en combinant les glyphes disponibles, ce qui est tout à fait indiqué en matière d’instrument à vent.

```
centermarkup = {
  \once \override TextScript.self-alignment-X = #CENTER
  \once \override TextScript.X-offset =#(lambda (g)
  (+ (ly:self-alignment-interface::centered-on-x-parent g)
     (ly:self-alignment-interface::x-aligned-on-self g)))
```

```
}

\score {
  \relative c'{
    g\open
    \once \override TextScript.staff-padding = #-1.0
    \centermarkup
    g^\markup {
      \combine
        \musicglyph "scripts.open"
        \musicglyph "scripts.tenuto"
    }
    \centermarkup
    g^\markup {
      \combine
        \musicglyph "scripts.open"
        \musicglyph "scripts.stopped"
    }
    g\stopped
  }
}
```
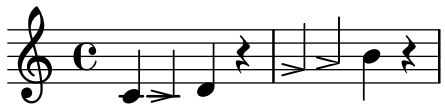


Coup de langue à la flûte

L'indication d'une technique particulière d'articulation, telle le « coup de langue » des flûtistes, s'obtient en remplaçant la tête de note par un glyphe approprié.

```
slap =
#(define-music-function (music) (ly:music?)
#{
  \temporary \override NoteHead.stencil =
  #(lambda (grob)
     (grob-interpret-markup grob
      (markup #:musicglyph "scripts.sforzato")))
  \temporary \override NoteHead.stem-attachment =
  #(lambda (grob)
     (let* ((thickness (ly:staff-symbol-line-thickness grob))
            (stem (ly:grob-object grob 'stem))
            (dir (ly:grob-property stem 'direction UP)))
       (cons 1 (+ (if (= dir DOWN)
                      0.5
                      0)
                  (/ thickness 2)))))
  #music
  \revert NoteHead.stencil
  \revert NoteHead.stem-attachment
#})
```

```
\relative c' {
  c4 \slap c d r
  \slap { g4 a } b r
}
```

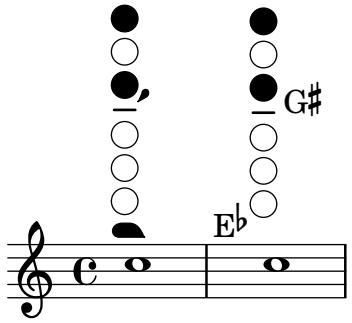


Ajout de texte à un diagramme de doigté

Dans certains cas, vous pouvez opter pour l'affichage textuel d'une clé située à côté d'un trou plutôt que sa représentation graphique.

```
\relative c'' {
  \textLengthOn
  c1^\markup
    \woodwind-diagram
      #'piccolo
      #'((cc . (one three))
         (lh . (gis))
         (rh . (ees)))

  c^\markup
    \override #'(graphical . #f) {
      \woodwind-diagram
        #'piccolo
        #'((cc . (one three))
           (lh . (gis))
           (rh . (ees)))
    }
}
```



Doigtés pour flûte à bec

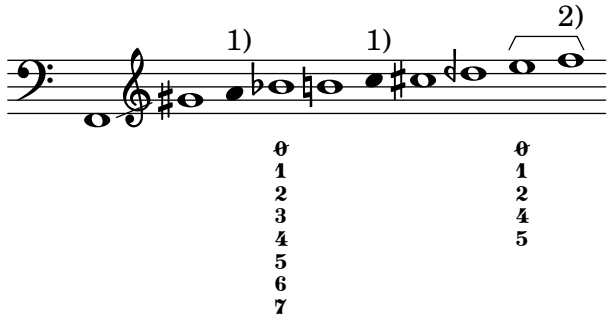
Cet exemple illustre la manière de créer et afficher des indications de doigté pour instrument à vent.

```
% range chart for paetzold contrabass recorder

centermarkup = {
  \once \override TextScript.self-alignment-X = #CENTER
  \once \override TextScript.X-offset = #(lambda (g)
    (+ (ly:self-alignment-interface::centered-on-x-parent g)
```

```
      (ly:self-alignment-interface::x-aligned-on-self g)))
}

\score {
  \new Staff \with {
    \remove "Time_signature_engraver"
    \omit Stem
    \omit Flag
    \consists "Horizontal_bracket_engraver"
  }
  {
    \clef bass
    \set Score.timing = ##f
    f,1*1/4 \glissando
    \clef violin
    gis'1*1/4
    \stemDown a'4^\markup {1)}
    \centermarkup
    \once \override TextScript.padding = #2
    bes'1*1/4_\markup {\override #'(baseline-skip . 1.7) \column
      { \fontsize #-5 \slashed-digit #0 \finger 1 \finger 2
        \finger 3 \finger 4 \finger 5 \finger 6 \finger 7} }
    b'1*1/4
    c''4^\markup {1)}
    \centermarkup
    \once \override TextScript.padding = #2
    cis''1*1/4
    deh''1*1/4
    \centermarkup
    \once \override TextScript.padding = #2
    \once \override Staff.HorizontalBracket.direction = #UP
    e''1*1/4_\markup {\override #'(baseline-skip . 1.7) \column
      { \fontsize #-5 \slashed-digit #0 \finger 1 \finger 2
        \finger 4 \finger 5} }\startGroup
    f''1*1/4^\markup {2)}\stopGroup
  }
}
```
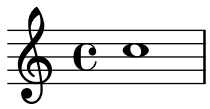


Liste des différents diagrammes de doigtés pour bois

Le code suivant permet d'obtenir une liste de toutes les possibilités en matière de doigtés pour bois, tels qu'ils sont définis dans le fichier `scm/define-woodwind-diagrams.scm`. Cette liste sera

produite dans le fichier de journalisation, mais pas sous forme de musique. Pour un affichage en console, supprimez la partie `(current-error-port)` des commandes.

```
#(print-keys-verbose 'piccolo (current-error-port))
#(print-keys-verbose 'flute (current-error-port))
#(print-keys-verbose 'flute-b-extension (current-error-port))
#(print-keys-verbose 'tin-whistle (current-error-port))
#(print-keys-verbose 'oboe (current-error-port))
#(print-keys-verbose 'clarinet (current-error-port))
#(print-keys-verbose 'bass-clarinet (current-error-port))
#(print-keys-verbose 'low-bass-clarinet (current-error-port))
#(print-keys-verbose 'saxophone (current-error-port))
#(print-keys-verbose 'soprano-saxophone (current-error-port))
#(print-keys-verbose 'alto-saxophone (current-error-port))
#(print-keys-verbose 'tenor-saxophone (current-error-port))
#(print-keys-verbose 'baritone-saxophone (current-error-port))
#(print-keys-verbose 'bassoon (current-error-port))
#(print-keys-verbose 'contrabassoon (current-error-port))

\score {c''1}
```



Liste des diagrammes de doigtés pour bois

Voici les différents instruments à vent de la section des bois pour lesquels LilyPond peut, à ce jour, afficher des doigtés.

```
\layout {
  indent = 0
}

\relative c' {
  \textLengthOn
  c1^
  \markup {
    \center-column {
      'tin-whistle
      " "
       \woodwind-diagram
                  #'tin-whistle
                  #'()
    }
  }

  c1^
  \markup {
    \center-column {
      'piccolo
      " "
       \woodwind-diagram
                  #'piccolo
```

```
                #'()
  }
}

c1^
\markup {
  \center-column {
     'flute
     " "
     \woodwind-diagram
        #'flute
        #'()
  }
}
c1^\markup {
  \center-column {
    'oboe
    " "
    \woodwind-diagram
      #'oboe
      #'()
  }
}

c1^\markup {
  \center-column {
    'clarinet
    " "
    \woodwind-diagram
      #'clarinet
      #'()
  }
}

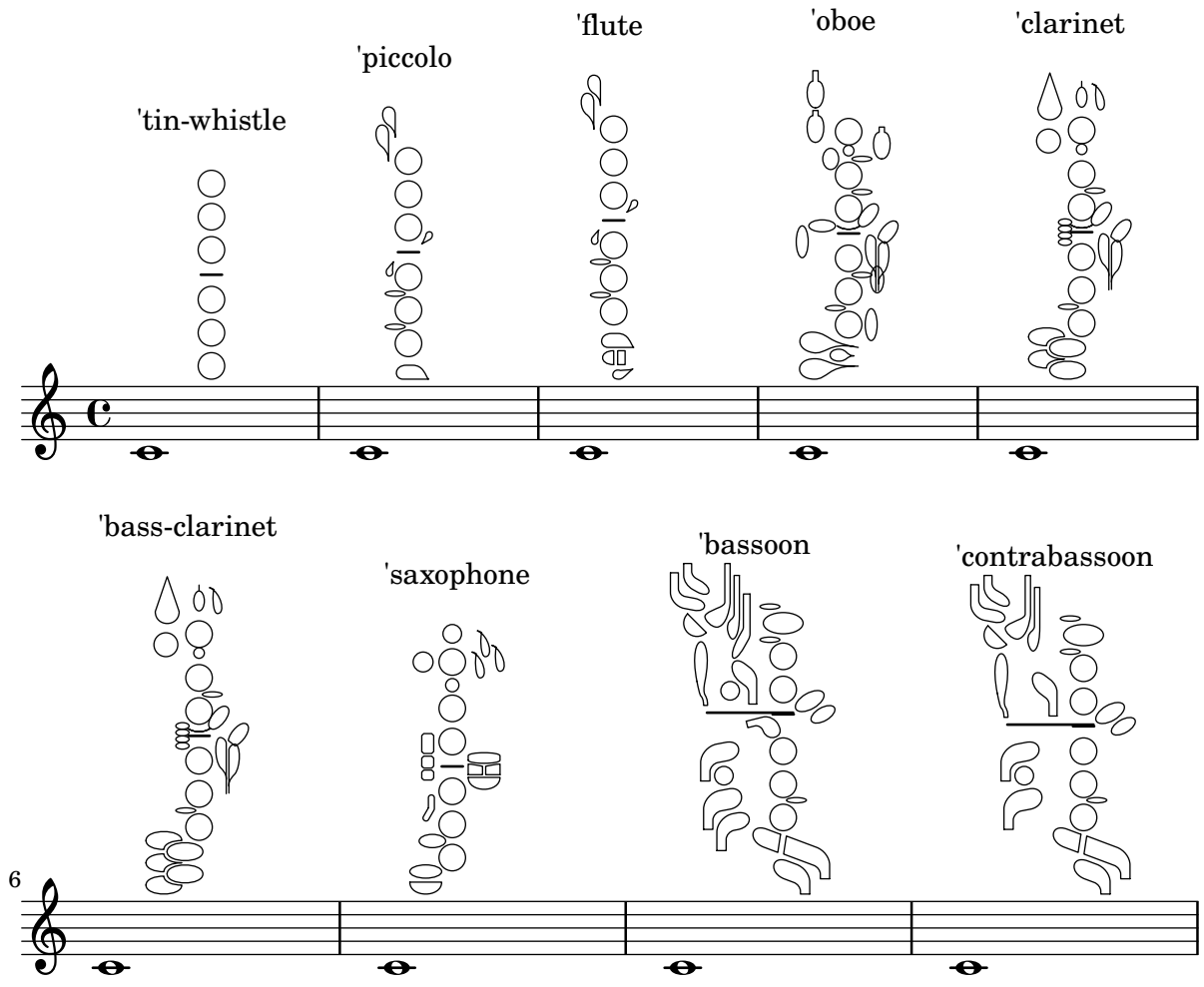
c1^\markup {
  \center-column {
    'bass-clarinet
    " "
    \woodwind-diagram
      #'bass-clarinet
      #'()
  }
}

c1^\markup {
  \center-column {
    'saxophone
    " "
    \woodwind-diagram
      #'saxophone
      #'()
  }
```

```
  }

  c1^\markup {
    \center-column {
      'bassoon
      " "
      \woodwind-diagram
        #'bassoon
        #'()
    }
  }

  c1^\markup {
    \center-column {
      'contrabassoon
      " "
      \woodwind-diagram
        #'contrabassoon
        #'()
    }
  }
}
```



Ancient notation

Section “Ancient notation” dans *Manuel de notation*

Ajout d’une basse chiffrée au-dessus ou au-dessous des notes
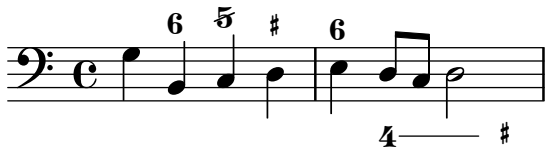
Une ligne de basse chiffrée peut se positionner au-dessus ou en dessous d’une partie de basse, grâce à la propriété `BassFigureAlignmentPositioning.direction` attachée à un contexte `Staff`. Les possibilités sont `#UP` (ou `#1`) pour au-dessus, `#CENTER` (ou `#0`) pour centrée, et `#DOWN` (ou `#-1`) pour en dessous.

Cette propriété peut être sujette à modification au cours de la partition. Il vous faudra insérer un `\once \override` si ce changement devait être temporaire.

```
bass = {
  \clef bass
  g4 b, c d
  e d8 c d2
}

continuo = \figuremode {
  <_>4 <6>4 <5/>4
  \override Staff.BassFigureAlignmentPositioning.direction = #UP
  %\bassFigureStaffAlignmentUp
  < _+ >4 <6>
  \set Staff.useBassFigureExtenders = ##t
  \override Staff.BassFigureAlignmentPositioning.direction = #DOWN
  %\bassFigureStaffAlignmentDown
  <4>4. <4>8 <_+>4
}

\score {
  <<
    \new Staff = bassStaff \bass
    \context Staff = bassStaff \continuo
  >>
}
```



Gravure de musique ancienne

Voici comment graver la plupart sinon tous les symboles que LilyPond prend en charge en matière de musique ancienne.

```
upperStaff = \new VaticanaStaff = "upperStaff" <<
  \context VaticanaVoice <<
    \transpose c c {

      \override NoteHead.style = #'vaticana.punctum
      \key es \major
```

```
\clef "vaticana-fa2"
c1 des e f ges

\override NoteHead.style = #'vaticana.inclinatum
a! b ces'
\bar "|"

\override NoteHead.style = #'vaticana.quilisma
b! des'! ges! fes!
\breathe
\clef "vaticana-fa1"
\override NoteHead.style = #'vaticana.plica
es d
\override NoteHead.style = #'vaticana.reverse.plica
c d
\bar "|"

\override NoteHead.style = #'vaticana.punctum.cavum
es f
\override NoteHead.style = #'vaticana.lpes
g as
\override NoteHead.style = #'vaticana.upes
bes as
\override NoteHead.style = #'vaticana.vupes
g f
\override NoteHead.style = #'vaticana.linea.punctum
\once \override Staff.BarLine.bar-extent = #'(-1 . 1) \bar "|"

es d
\override NoteHead.style = #'vaticana.epiphonus
c d
\override NoteHead.style = #'vaticana.cephalicus
es f

\override Staff.KeySignature.glyph-name-alist =
  #alteration-medicaea-glyph-name-alist
\override Staff.Accidental.glyph-name-alist =
  #alteration-medicaea-glyph-name-alist
\override Staff.Custos.style = #'medicaea
\override NoteHead.style = #'medicaea.punctum
\clef "medicaea-fa2"
ces des
\bar "|"

e! f! ges
\clef "medicaea-do2"
\override NoteHead.style = #'medicaea.inclinatum
a! b! ces'
\override NoteHead.style = #'medicaea.virga
b! a!
\bar "|"
```

```
      ges fes
      \clef "medicaea-fa1"
      \override NoteHead.style = #'medicaea.rvirga
      e des ces

      \override Staff.KeySignature.glyph-name-alist =
        #alteration-hufnagel-glyph-name-alist
      \override Staff.Accidental.glyph-name-alist =
        #alteration-hufnagel-glyph-name-alist
      \override Staff.Custos.style = #'hufnagel
      \override NoteHead.style = #'hufnagel.punctum
      \clef "hufnagel-fa2"
      ces des es
      \bar "|"

      fes ges
      \clef "hufnagel-do2"
      \override NoteHead.style = #'hufnagel.lpes
      as! bes! ces'
      \override NoteHead.style = #'hufnagel.virga
      bes! as!
      \bar "|"

      ges! fes!
      \clef "hufnagel-do-fa"
      \override NoteHead.style = #'hufnagel.punctum
      es! des ces des! es! fes!
      \bar "||"

      s32*1
    }
  >>
>>

lowerStaff = \new MensuralStaff = "lowerStaff" <<
  \context MensuralVoice <<
    \transpose c c {

      \key a \major
      cis'1 d'\breve gis'\breve e'\breve \[ e'\longa fis'\longa \]
      \set Staff.forceClef = ##t
      \clef "neomensural-c2"
      cis1
      \bar "|"

      \[ g\breve dis''\longa \]
      b\breve \[ a\longa d\longa \]
      \clef "petrucci-c2"

      fis1 ces1
      \clef "petrucci-c2"
      r\longa
```

```
      \set Staff.forceClef = ##t
      \clef "mensural-c2"
      r\breve
      \bar "|"

      r2
      \clef "mensural-g"
      r4 r8 r16 r16
      \override NoteHead.style = #'mensural
      \override Rest.style = #'mensural
      \clef "petrucci-f"
      c8 b, c16 b, c32 b, c64 b, c64 b,
      d8 e d16 e d32 e d64 e d64 e
      r\longa
      \set Staff.forceClef = ##t
      \clef "petrucci-f"
      r\breve
      \bar "|"

      r\breve
      \clef "mensural-f"
      r2 r4 r8 r16 r16

      \set Staff.forceClef = ##t
      \clef "mensural-f"
      e\breve f g a1
      \clef "mensural-g"

      \[ bes'!\longa a'!\longa c''!\longa \]
      e'1 d' c' d' \bar "|"
      \bar "|"

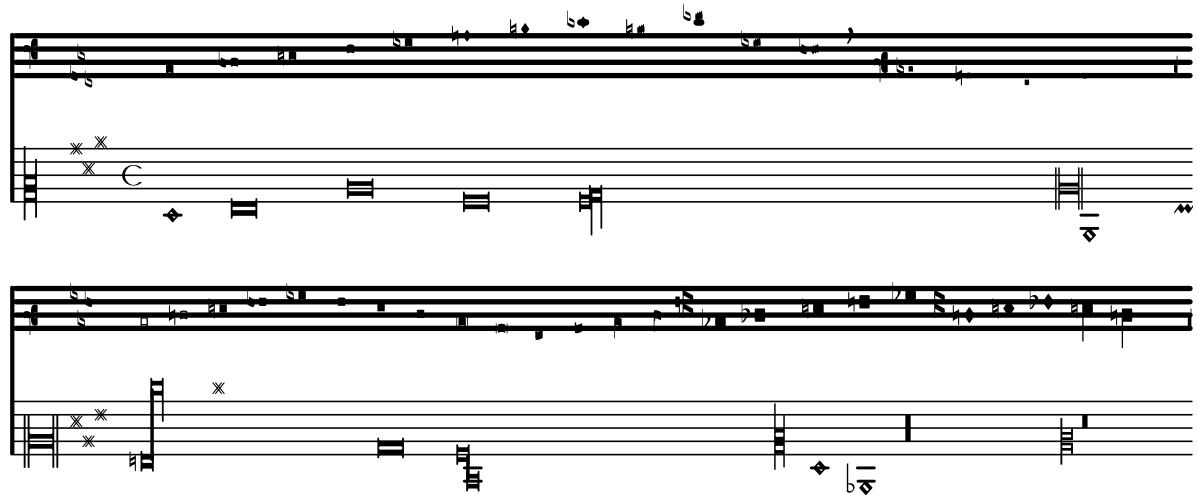
      bes'!\longa fis'!1 as'!1 ges'!\longa % lig
      \set Staff.forceClef = ##t
      \clef "mensural-g"
      e'2 d' c' \bar "|"
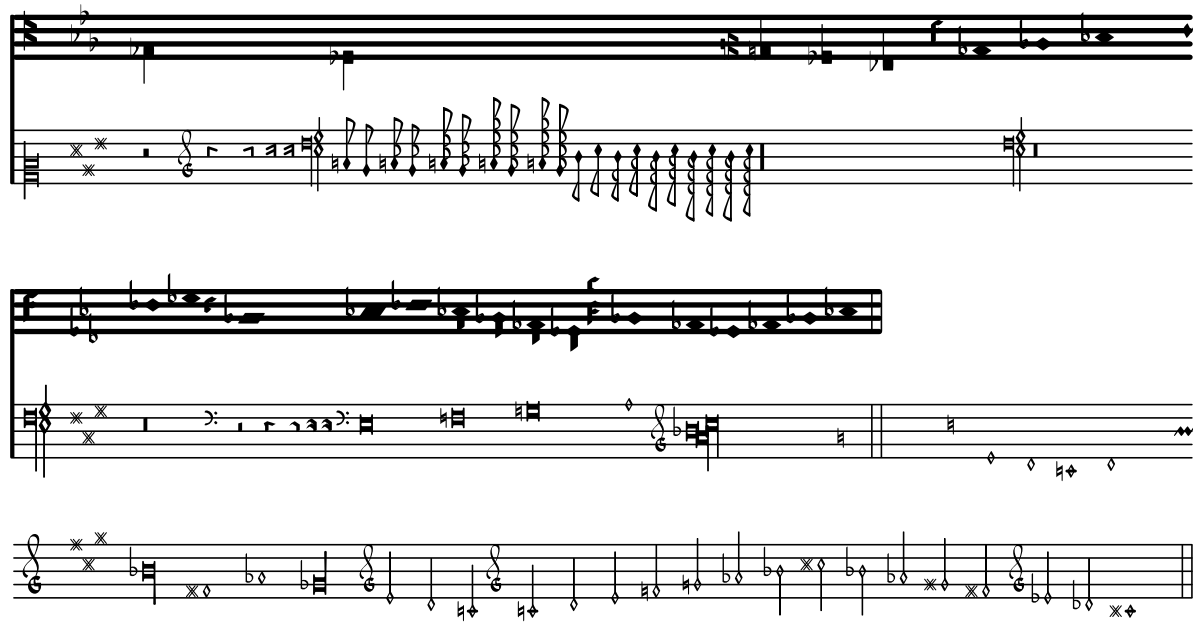
      \set Staff.forceClef = ##t
      \clef "petrucci-g"
      c'2 d' e' f'
      \clef "petrucci-g"
      g' as'! bes'! cis''!
      bes'! as'! gis'! fis'!
      \set Staff.forceClef = ##t
      \clef "mensural-g"
      es'! des'! cis'!1 \bar "||"
    }
  >>
>>

\paper {
  line-thickness = #(/ staff-space 5.0)
```

```
}

\score {
  <<
    \upperStaff
    \lowerStaff
  >>
  \layout {
    indent = 0.0
    \context {
      \Score
      timing = ##f
    }
    \context {
      \MensuralVoice
      \override NoteHead.style = #'neomensural
      \override Rest.style = #'neomensural
      \override Flag.style = #'mensural
      \override Stem.thickness = #1.0
    }
    \context {
      \MensuralStaff
      \revert  BarLine.transparent
      \override KeySignature.glyph-name-alist =
        #alteration-mensural-glyph-name-alist
      clefGlyph = #"clefs.petrucci.c2"
    }
    \context {
      \VaticanaStaff
      \revert  BarLine.transparent
      \override StaffSymbol.thickness = #2.0
      \override KeySignature.glyph-name-alist =
        #alteration-vaticana-glyph-name-alist
      \override Custos.neutral-position = #4
    }
  }
}
```





Exemples de notation ancienne – transcription moderne de musique grégorienne

Voici comment vous pourriez transcrire du grégorien. Pour mémoire, il n'y a en grégorien ni découpage en mesure, ni hampe ; seules sont utilisées des têtes de note blanches ou noires, ainsi que des signes spécifiques permettant d'indiquer des silences de différentes durées.
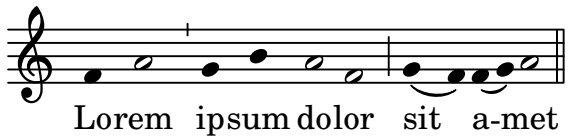
```
\include "gregorian.ly"

chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g) a2 \finalis
}

verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met
}

\score {
  \new Staff <<
    \new Voice = "melody" \chant
    \new Lyrics = "one" \lyricsto melody \verba
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Bar_engraver"
      \hide Stem
    }
    \context {
      \Voice
      \override Stem.length = #0
    }
    \context {
```

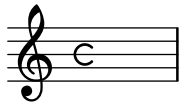
```
      \Score
      barAlways = ##t
    }
  }
}
```



Métrique ancienne

La métrique peut s'imprimer dans un style ancien.

```
{
  \override Staff.TimeSignature.style = #'neomensural
  s1
}
```

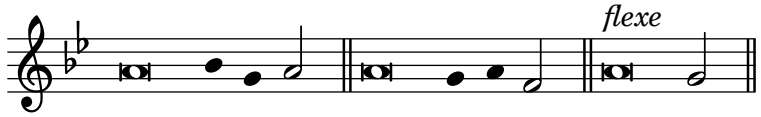


Notation pour psalmodie

Ce style de notation permet d'indiquer la mélodie d'une psalmodie lorsque les strophes sont de longueur inégale.

```
stemOff = \hide Staff.Stem
stemOn  = \undo \stemOff

\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \bar "||"
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \bar "||"
    \stemOff a'\breve^\markup { \italic flexe }
    \stemOn g'2 \bar "||"
  }
}
```



Guidons

Les guidons peuvent adopter différents styles.

```
\layout { ragged-right = ##t }

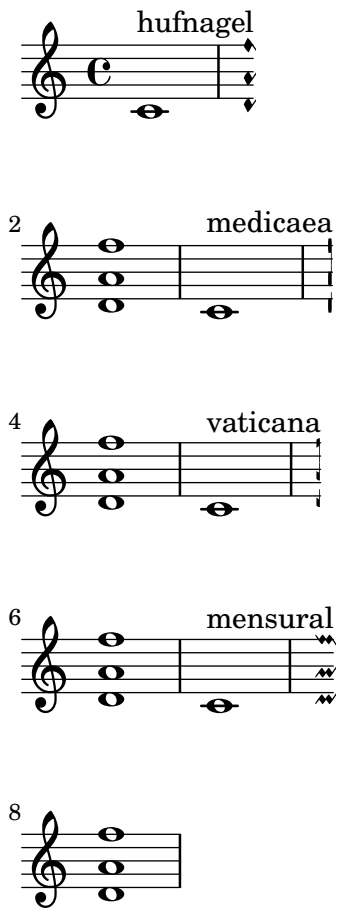
\new Staff \with { \consists "Custos_engraver" } \relative c' {
  \override Staff.Custos.neutral-position = #4

  \override Staff.Custos.style = #'hufnagel
  c1^"hufnagel" \break
  <d a' f'>1

  \override Staff.Custos.style = #'medicaea
  c1^"medicaea" \break
  <d a' f'>1

  \override Staff.Custos.style = #'vaticana
  c1^"vaticana" \break
  <d a' f'>1

  \override Staff.Custos.style = #'mensural
  c1^"mensural" \break
  <d a' f'>1
}
```



Incipit

Dans le cadre de la transcription de musique mensurale, un « incipit » en début de pièce permet d'indiquer tonalité et tempo originaux. Les musiciens sont maintenant habitués aux barres de mesure, ce qui n'était pas le cas à l'époque. Dans une forme de compromis, les barres de mesures sont souvent imprimées entre les portées, mise en forme que l'on appelle « lignes de mensuration ».

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% A short excerpt from the Jubilate Deo by Orlande de Lassus
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

global = {
  \set Score.skipBars = ##t
  \key g \major
  \time 4/4

  % the actual music
  \skip 1*8

  % let finis bar go through all staves
  \override Staff.BarLine.transparent = ##f

  % finis bar
  \bar "|."
}

discantusIncipit = {
  \clef "neomensural-c1"
  \key f \major
  \time 2/2
  c''1.
}

discantusNotes = {
  \transpose c' c'' {
    \clef "treble"
    d'2. d'4 |
    b e' d'2 |
    c'4 e'4.( d'8 c' b |
    a4) b a2 |
    b4.( c'8 d'4) c'4 |
    \once \hide NoteHead
    c'1 |
    b\breve |
  }
}

discantusLyrics = \lyricmode {
  Ju -- bi -- la -- te De -- o,
  om -- nis ter -- ra, __ om-
  "..."
  -us.
```

```
}

altusIncipit = {
  \clef "neomensural-c3"
  \key f \major
  \time 2/2
  r1 f'1.
}

altusNotes = {
  \transpose c' c'' {
    \clef "treble"
    r2 g2. e4 fis g |
    a2 g4 e |
    fis g4.( fis16 e fis4) |
    g1 |
    \once \hide NoteHead
    g1 |
    g\breve |
  }
}

altusLyrics = \lyricmode {
  Ju -- bi -- la -- te
  De -- o, om -- nis ter -- ra,
  "..."
  -us.
}

tenorIncipit = {
  \clef "neomensural-c4"
  \key f \major
  \time 2/2
  r\longa
  r\breve
  r1 c'1.
}

tenorNotes = {
  \transpose c' c' {
    \clef "treble_8"
    R1 |
    R1 |
    R1 |
    % two measures
    r2 d'2. d'4 b e' |
    \once \hide NoteHead
    e'1 |
    d'\breve |
  }
}
```

```
tenorLyrics = \lyricmode {
  Ju -- bi -- la -- te
  "..."
  -us.
}
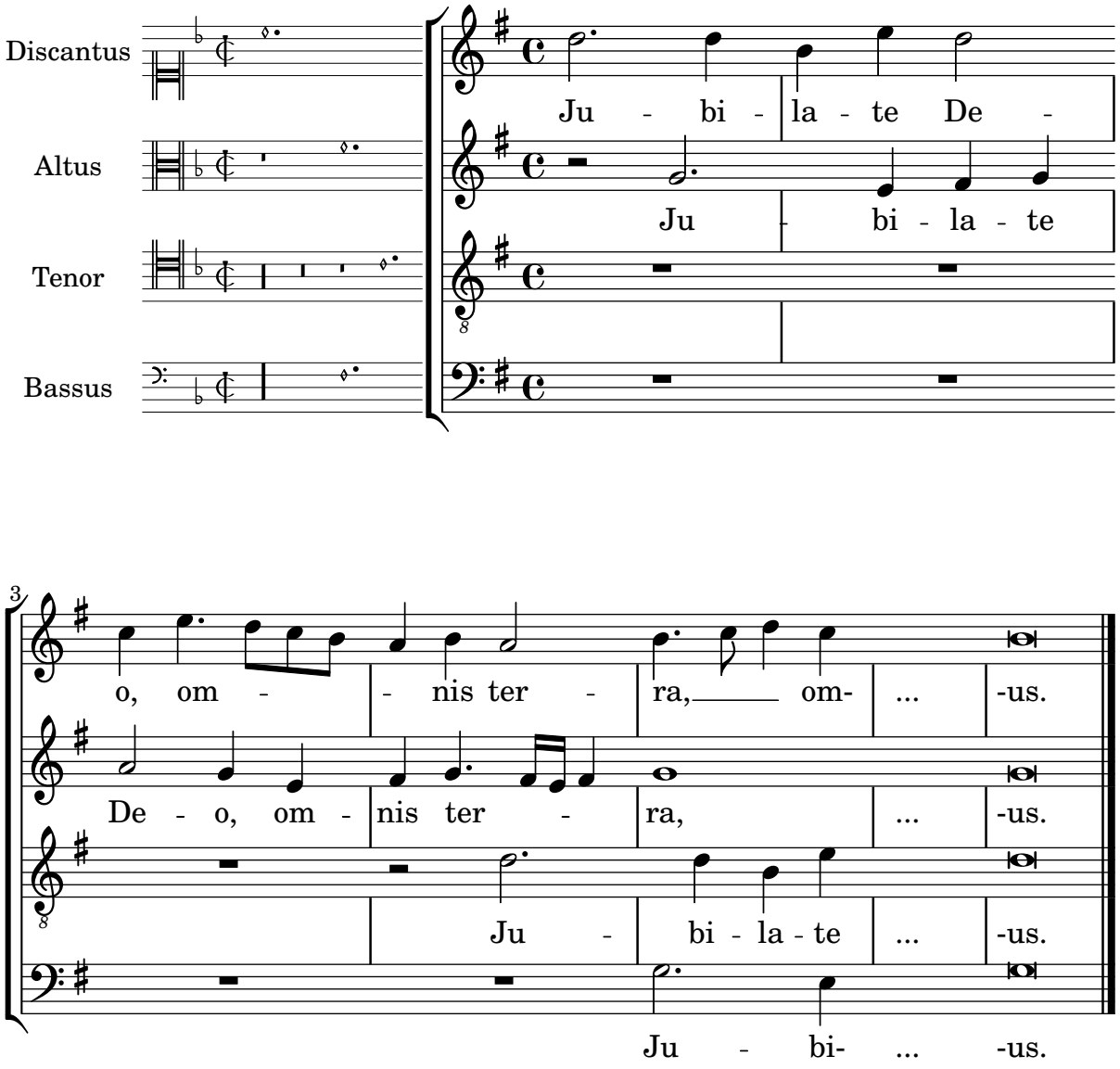
bassusIncipit = {
  \clef "mensural-f"
  \key f \major
  \time 2/2
  r\maxima
  f1.
}

bassusNotes = {
  \transpose c' c' {
    \clef "bass"
    R1 |
    R1 |
    R1 |
    R1 |
    g2. e4 |
    \once \hide NoteHead
    e1 |
    g\breve |
  }
}

bassusLyrics = \lyricmode {
  Ju -- bi-
  "..."
  -us.
}

\score {
  <<
    \new StaffGroup = choirStaff <<
      \new Voice = "discantusNotes" <<
        \set Staff.instrumentName = "Discantus"
        \incipit \discantusIncipit
        \global
        \discantusNotes
      >>
      \new Lyrics \lyricsto discantusNotes { \discantusLyrics }
      \new Voice = "altusNotes" <<
        \set Staff.instrumentName = "Altus"
        \global
        \incipit \altusIncipit
        \altusNotes
      >>
      \new Lyrics \lyricsto altusNotes { \altusLyrics }
      \new Voice = "tenorNotes" <<
```

```
        \set Staff.instrumentName = "Tenor"
        \global
        \incipit \tenorIncipit
        \tenorNotes
      >>
      \new Lyrics \lyricsto tenorNotes { \tenorLyrics }
      \new Voice = "bassusNotes" <<
        \set Staff.instrumentName = "Bassus"
        \global
        \incipit \bassusIncipit
        \bassusNotes
      >>
      \new Lyrics \lyricsto bassusNotes { \bassusLyrics }
    >>
  >>
  \layout {
    \context {
      \Score
      %% no bar lines in staves or lyrics
      \hide BarLine
    }
    %% the next two instructions keep the lyrics between the bar lines
    \context {
      \Lyrics
      \consists "Bar_engraver"
      \consists "Separating_line_group_engraver"
    }
    \context {
      \Voice
      %% no slurs
      \hide Slur
      %% Comment in the below "\remove" command to allow line
      %% breaking also at those bar lines where a note overlaps
      %% into the next measure.  The command is commented out in this
      %% short example score, but especially for large scores, you
      %% will typically yield better line breaking and thus improve
      %% overall spacing if you comment in the following command.
      %%\remove "Forbid_line_break_engraver"
    }
    indent = 6\cm
    incipit-width = 4\cm
  }
}
```

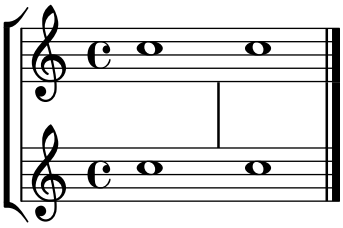


Présentation à l'ancienne (barres de mesure entre les portées)

En musique mensurale, les barres de mesure ne traversent pas les portées. Pour obtenir ce résultat avec un `StaffGroup` plutôt qu'en utilisant un `ChoirStaff`, il faudra masquer les portions de barre qui recouvrent les portées à l'aide d'un `\hide`.

```
global = {
  \hide Staff.BarLine
  s1 s
  % the final bar line is not interrupted
  \undo \hide Staff.BarLine
  \bar "|."
}

\new StaffGroup \relative c'' {
  <<
    \new Staff { << \global { c1 c } >> }
    \new Staff { << \global { c c } >> }
  >>
}
```



Styles de silences

Les silences peuvent être gravés selon différents styles.

```
\new Staff \relative c {
  \omit Score.TimeSignature
  \cadenzaOn

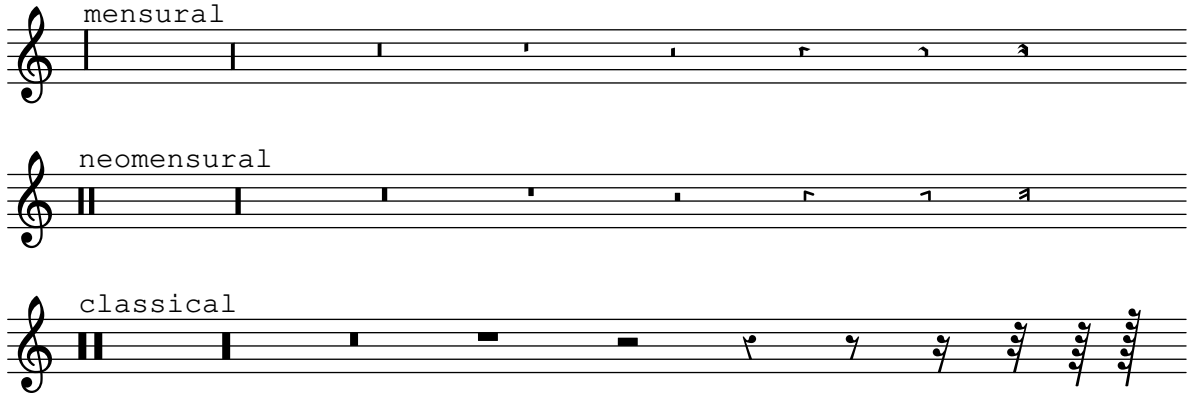
  \override Staff.Rest.style = #'mensural
  r\maxima^\markup \typewriter { mensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""
  \break

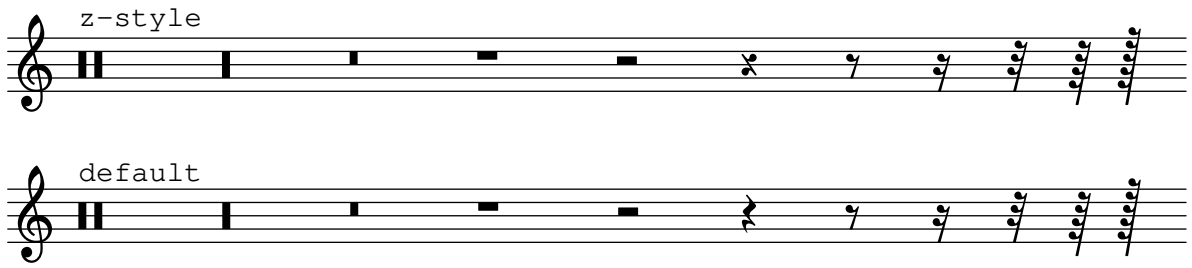
  \override Staff.Rest.style = #'neomensural
  r\maxima^\markup \typewriter { neomensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'classical
  r\maxima^\markup \typewriter { classical }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'z
  r\maxima^\markup \typewriter { z-style }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'default
  r\maxima^\markup \typewriter { default }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
}
```





Recours aux balises pour produire une partition ancienne et moderne à partir de la même source

Grâce aux balises (*tags*), il est possible d'utiliser une même source pour produire une partition de musique mensurale et moderne. Dans cet exemple est créée la fonction `menrest` qui permettra de positionner les silence comme dans la version originale, tout en respectant leur position sur une portée standard. Les balises permettent d'adapter les barres en fin de pièce ; elles peuvent aussi gérer d'autres différenciations selon les besoins, comme des « mesures de silence » (R1, R\breve, etc.) en notation moderne mais des silences normaux (r1, r\breve, etc.) en notation ancienne. L'action de convertir de la musique mensurale en version moderne est communément appelée « transcription ».

```
menrest = #(define-music-function (note)
  (ly:music?)
#{
    \tag #'mens $(make-music 'RestEvent note)
    \tag #'mod $(make-music 'RestEvent note 'pitch '())
#})

MensStyle = {
  \autoBeamOff
  \override NoteHead.style = #'petrucci
  \override Score.BarNumber.transparent = ##t
  \override Stem.neutral-direction = #up
}

finalis = {
  \once \override BreathingSign.stencil = #ly:breathing-sign::finalis
  \once \override BreathingSign.Y-offset = #0
  \once \override BreathingSign.minimum-X-extent = #'(-1.0 . 0.0)
  \once \override BreathingSign.minimum-Y-extent = #'(-2.5 . 2.5)

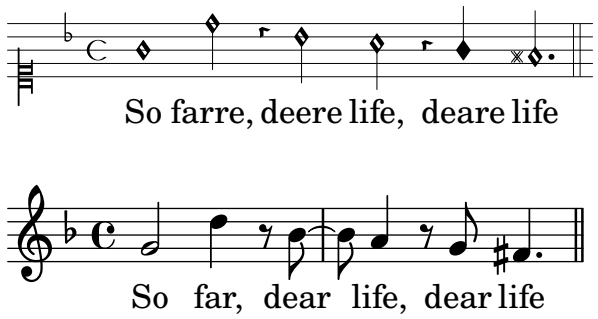
  \breathe
}

Music = \relative c'' {
  \set Score.tempoHideNote = ##t
  \key f \major
  \time 4/4
  g1 d'2 \menrest bes4 bes2 a2 r4 g4 fis2.
  \tag #'mens { \finalis }
  \tag #'mod { \bar "||" }
}

MenLyr = \lyricmode { So farre, deere life, deare life }
ModLyr = \lyricmode { So far, dear life, dear life }
```

```
\score {
  \keepWithTag #'mens {
    <<
      \new MensuralStaff
      {
        \new MensuralVoice = Cantus
          \clef "mensural-c1" \MensStyle \Music
      }
      \new Lyrics \lyricsto Cantus \MenLyr
    >>
  }
}

\score {
  \keepWithTag #'mod {
    \new ChoirStaff <<
      \new Staff
      {
        \new Voice = Sop \with {
          \remove "Note_heads_engraver"
          \consists "Completion_heads_engraver"
          \remove "Rest_engraver"
          \consists "Completion_rest_engraver" }
        {
          \shiftDurations #1 #0 { \autoBeamOff \Music }
        }
      }
      \new Lyrics \lyricsto Sop \ModLyr
    >>
  }
}
```
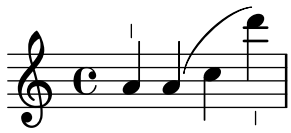


Articulation baroque en forme de coche

On trouve régulièrement, en musique baroque, cette courte ligne verticale. Sa signification peut varier, mais elle indique le plus souvent une note plus « appuyée ». Voici comment générer ce signe particulier.

```
upline =
\tweak stencil
  #(lambda (grob)
    (grob-interpret-markup grob #{ \markup \draw-line #'(0 . 1) #}))
  \stopped
```

```
\relative c' {
  a'4^\upline a( c d')_\upline
}
```



World music

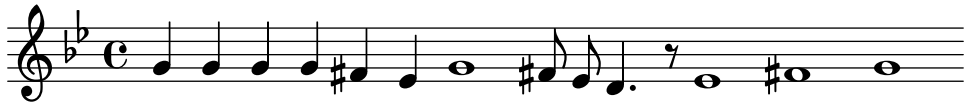
Section “World music” dans *Manuel de notation*

Improvisation en musique arabe

Lorsque les improvisations ou *taqasim* sont temporairement libres, la métrique peut ne pas apparaître, auquel cas on utilisera un `\cadenzaOn`. Les altérations accidentelles devront alors être répétées en raison de l’absence de barre de mesure. Voici comment pourrait débuter une improvisation de *hijaz*.

```
\include "arabic.ly"

\relative sol' {
  \key re \kurd
  \accidentalStyle forget
  \cadenzaOn
  sol4 sol sol sol fad mib sol1 fad8 mib re4. r8 mib1 fad sol
}
```
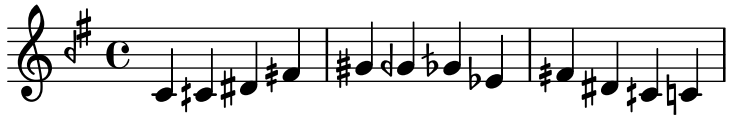


Exemple de musique « Makam »

Le « makam » est une forme de mélodie turque qui utilise des altérations d’un neuvième de ton.

Consultez le fichier d’initialisation `ly/makam.ly` pour plus de détails sur les hauteurs et altérations utilisées (voir le chapitre 4.6.3 - Autres sources d’information du manuel d’initiation pour le localiser).

```
% Initialize makam settings
\include "makam.ly"

\relative c' {
  \set Staff.keyAlterations = #`((6 . ,(- KOMA)) (3 . ,BAKIYE))
  c4 cc db fk
  gbm4 gfc gfb efk
  fk4 db cc c
}
```



Armures inhabituelles

La commande `\key` détermine la propriété `keyAlterations` d’un contexte `Staff`. Des armures inhabituelles peuvent être spécifiées en modifiant directement cette propriété.
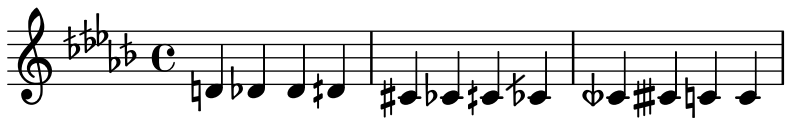
Il s’agit en l’occurrence de définir une liste :

```
\set Staff.keyAlterations =
  #`(((octave . pas) . altération) ((octave . pas) . altération)
  @dots{})
```

dans laquelle, et pour chaque élément, `octave` spécifie l'octave (0 pour celle allant du do médium au si supérieur), `pas` la note dans cette octave (0 pour do et 6 pour si), et `altération` sera `,SHARP` ou `,FLAT` ou `,DOUBLE-SHARP`, etc. (attention à la virgule en préfixe).

Une formulation abrégée – `(pas . altération)` – signifie que l'altération de l'élément en question sera valide quelle que soit l'octave. En ce qui concerne les gammes microtonales dans lesquelles un « dièse » n'est pas d'un centième, `altération` se réfère à un deux-centième de ton entier.

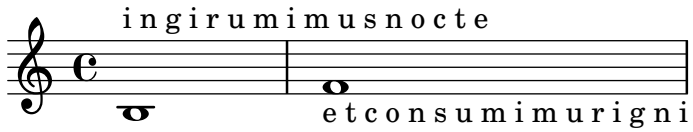
```
\include "arabic.ly"
\relative do' {
  \set Staff.keyAlterations = #`((0 . ,SEMI-FLAT)
                                 (1 . ,SEMI-FLAT)
                                 (2 . ,FLAT)
                                 (5 . ,FLAT)
                                 (6 . ,SEMI-FLAT))
%\set Staff.extraNatural = ##f
  re reb \dwn reb resd
  dod dob dosd \dwn dob |
  dobsb dodsd do do |
}
```



Impression de texte de droite à gauche

Du texte, inclus dans un objet *markup*, peut s'imprimer de droite à gauche, comme illustré ci-dessous.

```
{
  b1^\markup {
    \line { i n g i r u m i m u s n o c t e }
  }
  f'_\markup {
    \override #'(text-direction . -1)
    \line { i n g i r u m i m u s n o c t e }
  }
}
```



Exemple de makam turc

Ce canevas utilise le début d'un *saz semai* turc bien connu du répertoire aux fins d'illustrer certains éléments de la notation musicale turque.

```
% Initialize makam settings
\include "turkish-makam.ly"

\header {
```

```
    title = "Hüseyni Saz Semaisi"
    composer = "Lavtacı Andon"
}

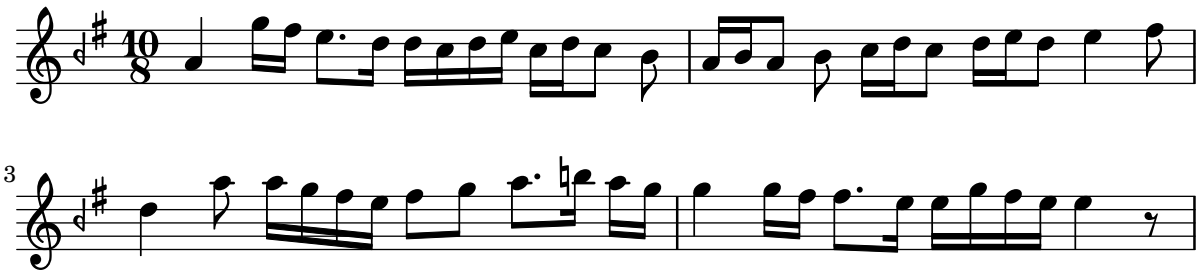
\relative {
  \set Staff.extraNatural = ##f
  \set Staff.autoBeaming = ##f

  \key a \huseyni
  \time 10/8

  a'4 g'16 [fb] e8. [d16] d [c d e] c [d c8] bfc |
  a16 [bfc a8] bfc c16 [d c8] d16 [e d8] e4 fb8 |
  d4 a'8 a16 [g fb e] fb8 [g] a8. [b16] a16 [g] |
  g4 g16 [fb] fb8. [e16] e [g fb e] e4 r8 |
}
```

Hüseyni Saz Semaisi

Lavtacı Andon



Contexts and engravers

Section “Changing defaults” dans *Manuel de notation*

Section “Contexts and engravers” dans *Manuel d’initiation*

Ajout d’une basse chiffrée au-dessus ou au-dessous des notes
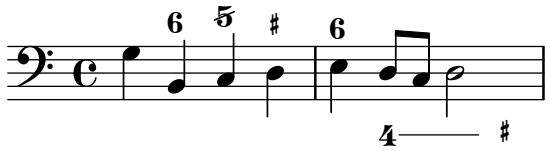
Une ligne de basse chiffrée peut se positionner au-dessus ou en dessous d’une partie de basse, grâce à la propriété `BassFigureAlignmentPositioning.direction` attachée à un contexte `Staff`. Les possibilités sont `#UP` (ou `#1`) pour au-dessus, `#CENTER` (ou `#0`) pour centrée, et `#DOWN` (ou `#-1`) pour en dessous.

Cette propriété peut être sujette à modification au cours de la partition. Il vous faudra insérer un `\once \override` si ce changement devait être temporaire.

```
bass = {
  \clef bass
  g4 b, c d
  e d8 c d2
}

continuo = \figuremode {
  <_>4 <6>4 <5/>4
  \override Staff.BassFigureAlignmentPositioning.direction = #UP
  %\bassFigureStaffAlignmentUp
  < _+ >4 <6>
  \set Staff.useBassFigureExtenders = ##t
  \override Staff.BassFigureAlignmentPositioning.direction = #DOWN
  %\bassFigureStaffAlignmentDown
  <4>4. <4>8 <_+>4
}

\score {
  <<
    \new Staff = bassStaff \bass
    \context Staff = bassStaff \continuo
  >>
}
```

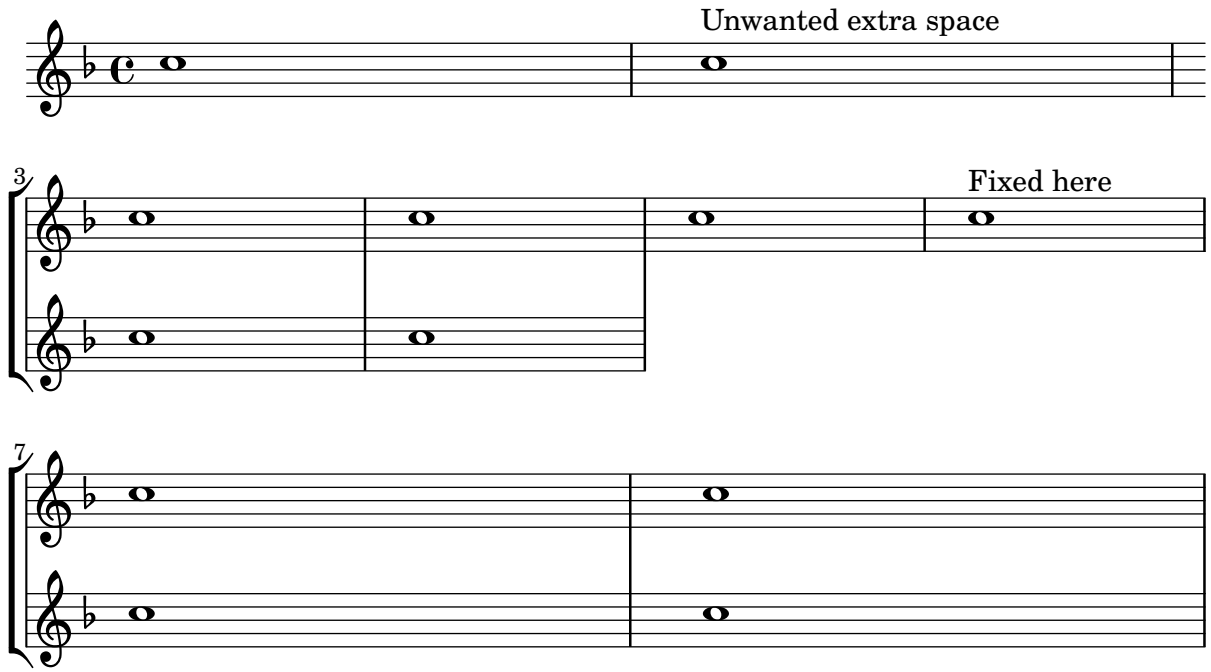


Ajout d’une portée supplémentaire après un saut de ligne

Lorsqu’une nouvelle portée vient s’ajouter après un saut de ligne, LilyPond préserve un espace juste avant le saut de ligne – pour un éventuel changement d’armure qui, quoi qu’il en soit, ne sera pas imprimé. L’astuce consiste alors, comme indiqué dans l’exemple suivant, à ajuster `Staff.explicitKeySignatureVisibility`.

```
\score {
  \new StaffGroup \relative c'' {
    \new Staff
```

```
    \key f \major
    c1 c^"Unwanted extra space" \break
    << { c1 | c }
       \new Staff {
         \key f \major
         \once \omit Staff.TimeSignature
         c1 | c
       }
    >>
    c1 | c^"Fixed here" \break
    << { c1 | c }
       \new Staff {
         \once \set Staff.explicitKeySignatureVisibility = #end-of-line-invisible
         \key f \major
         \once \omit Staff.TimeSignature
         c1 | c
       }
    >>
  }
}
```

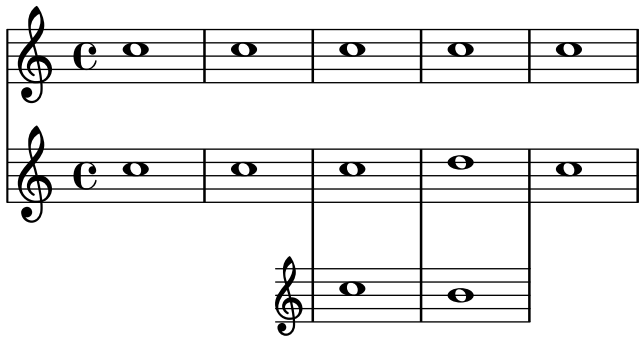


Ajout d'une portée supplémentaire

Vous pouvez ajouter une nouvelle portée, éventuellement de manière temporaire, après le début d'un morceau.

```
\score {
  <<
    \new Staff \relative c'' {
      c1 | c | c | c | c
    }
    \new StaffGroup \relative c'' {
      \new Staff {
```

```
        c1 | c
        <<
          {
            c1 | d
          }
          \new Staff {
            \once \omit Staff.TimeSignature
            c1 | b
          }
        >>
        c1
      }
    }
  >>
}
```
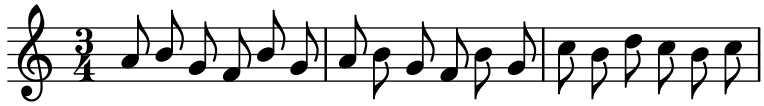


Changement automatique de l'orientation de hampe de la note médiane selon la mélodie

Afin de suivre la ligne mélodique, LilyPond peut inverser l'orientation de hampe de la note médiane, dès lors qu'aura été ajouté au contexte de voix le graveur `Melody_engraver`.

La propriété de contexte `suspendMelodyDecisions` permet, si besoin, de désactiver temporairement ce comportement.

```
\relative c'' {
  \time 3/4
  a8 b g f b g |
  \set suspendMelodyDecisions = ##t
  a  b g f b g |
  \unset suspendMelodyDecisions
  c  b d c b c |
}

\layout {
  \context {
    \Voice
    \consists "Melody_engraver"
    \autoBeamOff
  }
}
```



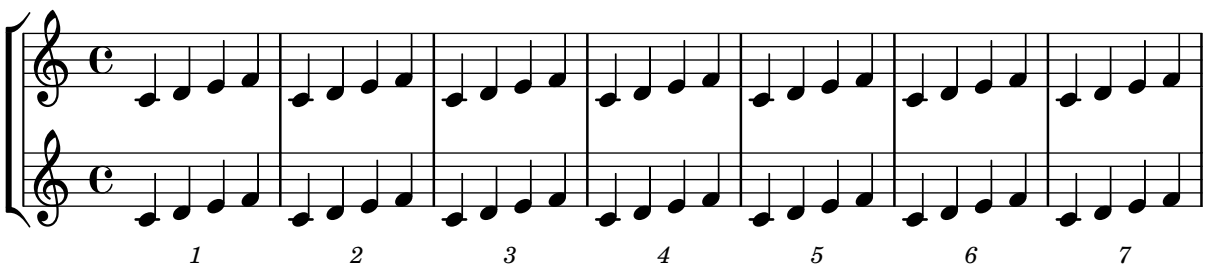
Numéros de mesure centrés

Les partitions pour grands ensembles disposent souvent de la numérotation des mesures centrée en dessous du système. L'extrait ci-dessous indique comment le `Measure_counter_engraver` permet de répondre à cette pratique. Le graveur est ici ajouté à un contexte `Dynamics`.

```
\layout {
  \context {
    \Dynamics
    \consists #Measure_counter_engraver
    \override MeasureCounter.direction = #DOWN
    \override MeasureCounter.font-encoding = #'latin1
    \override MeasureCounter.font-shape = #'italic
    % to control the distance of the Dynamics context from the staff:
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing.padding = #2
  }
  \context {
    \Score
    \remove "Bar_number_engraver"
  }
}

pattern = \repeat unfold 7 { c'4 d' e' f' }

\new StaffGroup <<
  \new Staff {
    \pattern
  }
  \new Staff {
    \pattern
  }
  \new Dynamics {
    \startMeasureCount
    s1*7
    \stopMeasureCount
  }
>>
```
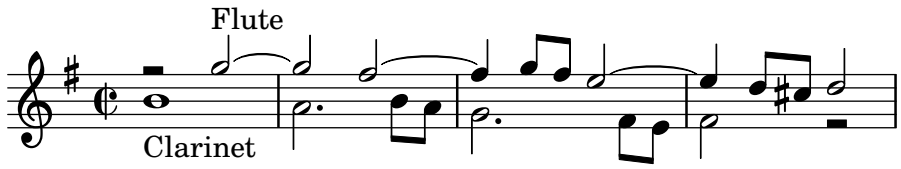


Affectation d'un canal MIDI par voix

Lorsque LilyPond génère un fichier MIDI, chaque portée sera par défaut affectée à un canal, quel que soit le nombre de voix qu'elle contient. Ceci permet d'éviter de se retrouver à court de canaux, sachant qu'il n'y en a que seize de disponibles par piste.

Le fait de déplacer le `Staff_performer` dans le contexte `Voice` permet d'affecter à chaque voix d'une même portée un canal MIDI spécifique. Dans l'exemple suivant, la même portée donnera lieu à deux canaux MIDI différents, chacun étant affecté de son propre `midiInstrument`.

```
\score {
  \new Staff <<
    \new Voice \relative c''' {
      \set midiInstrument = #"flute"
      \voiceOne
      \key g \major
      \time 2/2
      r2 g-"Flute" ~
      g fis ~
      fis4 g8 fis e2 ~
      e4 d8 cis d2
    }
    \new Voice \relative c'' {
      \set midiInstrument = #"clarinet"
      \voiceTwo
      b1-"Clarinet"
      a2. b8 a
      g2. fis8 e
      fis2 r
    }
  >>
  \layout { }
  \midi {
    \context {
      \Staff
      \remove "Staff_performer"
    }
    \context {
      \Voice
      \consists "Staff_performer"
    }
    \tempo 2 = 72
  }
}
```

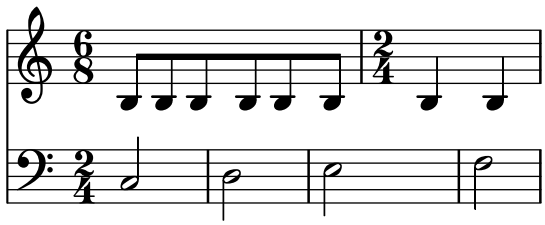


Modification de la métrique au sein d'un fragment polymétrique à l'aide de `\scaleDurations`

La propriété `measureLength`, conjointement avec `measurePosition`, détermine l'endroit où une barre de mesure est requise. L'utilisation d'un `\scaleDurations` a l'inconvénient que l'échelonnement des durées rend compliqués les changements de métrique. Il est nécessaire en pareil cas de définir manuellement `measureLength`, avec un appel à `ly:make-moment`. Le second argument devra être identique au deuxième argument de `\scaleDurations`.

```
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Default_bar_line_engraver"
  }
  \context {
    \Staff
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  }
}

<<
  \new Staff {
    \scaleDurations 8/5 {
      \time 6/8
      \set Timing.measureLength = #(ly:make-moment 6/5)
      b8 b b b b b
      \time 2/4
      \set Timing.measureLength = #(ly:make-moment 4/5)
      b4 b
    }
  }
  \new Staff {
    \clef bass
    \time 2/4
    c2 d e f
  }
>>
```



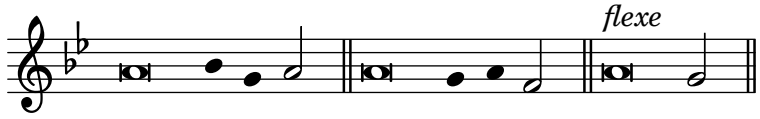
Notation pour psalmodie

Ce style de notation permet d'indiquer la mélodie d'une psalmodie lorsque les strophes sont de longueur inégale.

```
stemOff = \hide Staff.Stem
stemOn  = \undo \stemOff

\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \bar "||"
```

```
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \bar "||"
    \stemOff a'\breve^\markup { \italic flexe }
    \stemOn g'2 \bar "||"
  }
}
```



Impression de papier à musique

Rien n'est plus simple, pour obtenir une feuille blanche avec des portées, que de générer des mesures vides, de supprimer le `Bar_number_engraver` du contexte `Score`, et les `Time_signature_engraver`, `Clef_engraver` et `Bar_engraver` du contexte `Staff`.

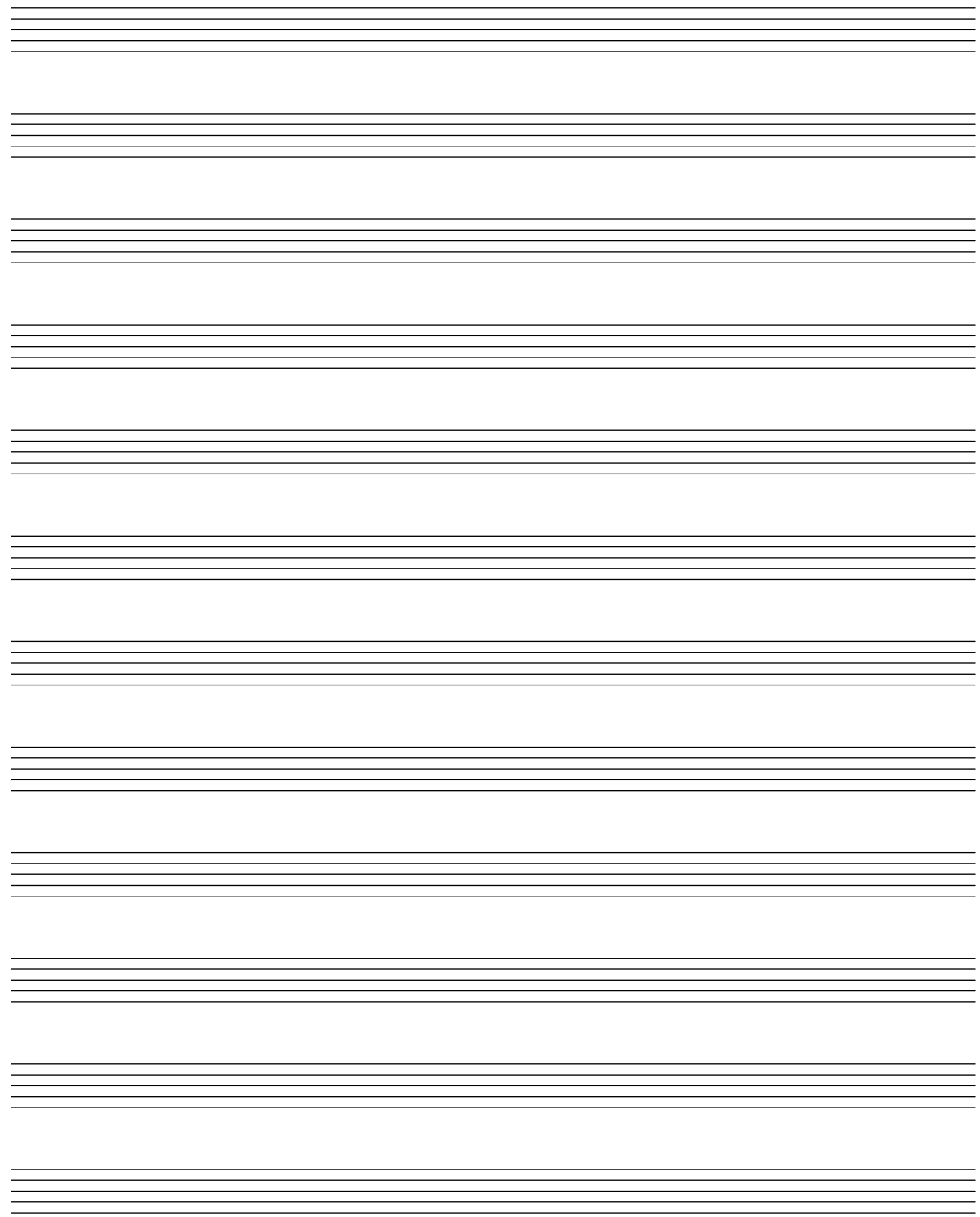
```
#(set-global-staff-size 20)

\score {
  {
    \repeat unfold 12 { s1 \break }
  }
  \layout {
    indent = 0\in
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Clef_engraver"
      \remove "Bar_engraver"
    }
    \context {
      \Score
      \remove "Bar_number_engraver"
    }
  }
}

% uncomment these lines for "letter" size
%{
\paper {
  #(set-paper-size "letter")
  ragged-last-bottom = ##f
  line-width = 7.5\in
  left-margin = 0.5\in
  bottom-margin = 0.25\in
  top-margin = 0.25\in
}
%}

% uncomment these lines for "A4" size
%{
\paper {
```

```
  #(set-paper-size "a4")
  ragged-last-bottom = ##f
  line-width = 180
  left-margin = 15
  bottom-margin = 10
  top-margin = 10
}
%}
```



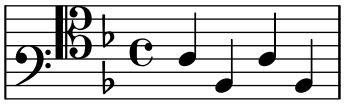
Création d'armures personnalisées

LilyPond accepte les armures personnalisées. Dans cet exemple, il s'agit d'une tonalité de ré mineur à laquelle on rajoute des bémols.

```
\new Staff \with {
  \override StaffSymbol.line-count = #8
  \override KeySignature.flat-positions = #'((-7 . 6))
  \override KeyCancellation.flat-positions = #'((-7 . 6))
  % presumably sharps are also printed in both octaves
  \override KeySignature.sharp-positions = #'((-6 . 7))
  \override KeyCancellation.sharp-positions = #'((-6 . 7))

  \override Clef.stencil = #
  (lambda (grob)(grob-interpret-markup grob
  #{ \markup\combine
    \musicglyph "clefs.C"
    \translate #'(-3 . -2)
    \musicglyph "clefs.F"
   #}))
    clefPosition = #3
    middleCPosition = #3
    middleCClefPosition = #3
}

{
  \key d\minor
  f bes, f bes,
}
```



Hampes interportées
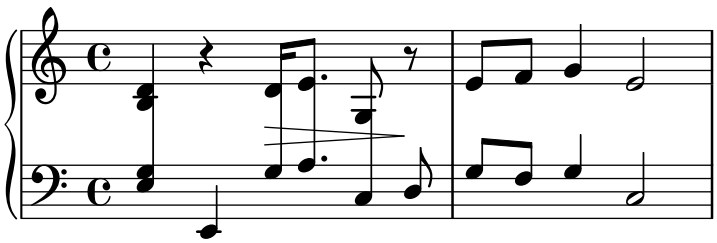
L'exemple ci-dessous illustre l'utilisation du `Span_stem_engraver` et de la commande `\crossStaff` afin de connecter des hampes entre les portées.

Nul n'est besoin de spécifier la taille des hampes ; le graveur calcule automatiquement la distance relative des têtes de note avec les portées.

```
\layout {
  \context {
    \PianoStaff
    \consists #Span_stem_engraver
  }
}

{
  \new PianoStaff <<
    \new Staff {
      <b d'>4 r d'16\> e'8. g8 r\!
      e'8 f' g'4 e'2
    }
```

```
    \new Staff {
      \clef bass
      \voiceOne
      \autoBeamOff
      \crossStaff { <e g>4 e, g16 a8. c8} d
      \autoBeamOn
      g8 f g4 c2
    }
  >>
}
```



Définition d'un graveur en Scheme : graveur d'ambitus

Cet exemple démontre comment définir son propre graveur de tessiture à base de code Scheme.

Il s'agit d'une réécriture en Scheme du code contenu dans le fichier `lily/ambitus-engraver.cc`.

```
#(use-modules (oop goops))

%%%
%%% Grob utilities
%%%
%%% These are literal rewrites of some C++ methods used by the ambitus engraver.

#(define (ly:separation-item::add-conditional-item grob grob-item)
   "Add @var{grob-item} to the array of conditional elements of @var{grob}.
Rewrite of @code{Separation_item::add_conditional_item} from @file{lily/separation-item.cc}
   (ly:pointer-group-interface::add-grob grob 'conditional-elements grob-item))

#(define (ly:accidental-placement::accidental-pitch accidental-grob)
   "Get the pitch from the grob cause of @var{accidental-grob}.
Rewrite of @code{accidental_pitch} from @file{lily/accidental-placement.cc}."
   (ly:event-property (ly:grob-property (ly:grob-parent accidental-grob Y) 'cause)
                      'pitch))

#(define (ly:accidental-placement::add-accidental grob accidental-grob)
   "Add @var{accidental-grob}, an @code{Accidental} grob, to the
list of the accidental grobs of @var{grob}, an @code{AccidentalPlacement}
grob.
Rewrite of @code{Accidental_placement::add_accidental} from @file{lily/accidental-placement
   (let ((pitch (ly:accidental-placement::accidental-pitch accidental-grob)))
     (set! (ly:grob-parent accidental-grob X) grob)
     (let* ((accidentals (ly:grob-object grob 'accidental-grobs))
            (handle (assq (ly:pitch-notename pitch) accidentals))
            (entry (if handle (cdr handle) '())))
```

```
       (set! (ly:grob-object grob 'accidental-grobs)
             (assq-set! accidentals
                        (ly:pitch-notename pitch)
                        (cons accidental-grob entry))))))

%%%
%%% Ambitus data structure
%%%

%%% The <ambitus> class holds the various grobs that are created
%%% to print an ambitus:
%%% - ambitus-group: the grob that groups all the components of an ambitus
%%% (Ambitus grob);
%%% - ambitus-line: the vertical line between the upper and lower ambitus
%%% notes (AmbitusLine grob);
%%% - ambitus-up-note and ambitus-down-note: the note head and accidental
%%% for the lower and upper note of the ambitus (see <ambitus-note> class
%%% below).
%%% The other slots define the key and clef context of the engraver:
%%% - start-c0: position of middle c at the beginning of the piece.  It
%%% is used to place the ambitus notes according to their pitch;
%%% - start-key-sig: the key signature at the beginning of the piece.  It
%%% is used to determine if accidentals shall be printed next to ambitus
%%% notes.

#(define-class <ambitus> ()
   (ambitus-group #:accessor ambitus-group)
   (ambitus-line #:accessor ambitus-line)
   (ambitus-up-note #:getter ambitus-up-note
                    #:init-form (make <ambitus-note>))
   (ambitus-down-note #:getter ambitus-down-note
                      #:init-form (make <ambitus-note>))
   (start-c0 #:accessor ambitus-start-c0
             #:init-value #f)
   (start-key-sig #:accessor ambitus-start-key-sig
                  #:init-value '()))

%%% Accessor for the lower and upper note data of an ambitus
#(define-method (ambitus-note (ambitus <ambitus>) direction)
   "If @var{direction} is @code{UP}, then return the upper ambitus note
of @var{ambitus}, otherwise return the lower ambitus note."
   (if (= direction UP)
       (ambitus-up-note ambitus)
       (ambitus-down-note ambitus)))

%%% The <ambitus-note> class holds the grobs that are specific to ambitus
%%% (lower and upper) notes:
%%% - head: an AmbitusNoteHead grob;
%%% - accidental: an AmbitusAccidental grob, to be possibly printed next
%%% to the ambitus note head.
%%% Moreover:
%%% - pitch is the absolute pitch of the note
```

```
%%% - cause is the note event that causes this ambitus note, i.e. the lower
%%% or upper note of the considered music sequence.

#(define-class <ambitus-note> ()
   (head #:accessor ambitus-note-head
         #:init-value #f)
   (accidental #:accessor ambitus-note-accidental
               #:init-value #f)
   (cause #:accessor ambitus-note-cause
          #:init-value #f)
   (pitch #:accessor ambitus-note-pitch
          #:init-value #f))

%%%
%%% Ambitus engraving logics
%%%
%%% Rewrite of the code from @file{lily/ambitus-engraver.cc}.

#(define (make-ambitus translator)
   "Build an ambitus object: initialize all the grobs and their relations.

The Ambitus grob contain all other grobs:
 Ambitus
  |- AmbitusLine
  |- AmbitusNoteHead   for upper note
  |- AmbitusAccidental for upper note
  |- AmbitusNoteHead   for lower note
  |- AmbitusAccidental for lower note

The parent of an accidental is the corresponding note head,
and the accidental is set as the 'accidental-grob of the note head
so that is printed by the function that prints notes."
   ;; make the ambitus object
   (let ((ambitus (make <ambitus>)))
     ;; build the Ambitus grob, which will contain all other grobs
     (set! (ambitus-group ambitus) (ly:engraver-make-grob translator 'Ambitus '()))
     ;; build the AmbitusLine grob (line between lower and upper note)
     (set! (ambitus-line ambitus) (ly:engraver-make-grob translator 'AmbitusLine '()))
     ;; build the upper and lower AmbitusNoteHead and AmbitusAccidental
     (for-each (lambda (direction)
                 (let ((head (ly:engraver-make-grob translator 'AmbitusNoteHead '()))
                       (accidental (ly:engraver-make-grob translator 'AmbitusAccidental '()
                       (group (ambitus-group ambitus)))
                   ;; The parent of the AmbitusAccidental grob is the
                   ;; AmbitusNoteHead grob
                   (set! (ly:grob-parent accidental Y) head)
                   ;; The AmbitusAccidental grob is set as the accidental-grob
                   ;; object of the AmbitusNoteHead.  This is later used by the
                   ;; function that prints notes.
                   (set! (ly:grob-object head 'accidental-grob) accidental)
                   ;; both the note head and the accidental grobs are added
                   ;; to the main ambitus grob.
```

```
                    (ly:axis-group-interface::add-element group head)
                    (ly:axis-group-interface::add-element group accidental)
                    ;; the note head and the accidental grobs are added to the
                    ;; ambitus object
                    (set! (ambitus-note-head (ambitus-note ambitus direction))
                          head)
                    (set! (ambitus-note-accidental (ambitus-note ambitus direction))
                          accidental)))
                (list DOWN UP))
      ;; The parent of the ambitus line is the lower ambitus note head
      (set! (ly:grob-parent (ambitus-line ambitus) X)
            (ambitus-note-head (ambitus-note ambitus DOWN)))
      ;; the ambitus line is added to the ambitus main grob
      (ly:axis-group-interface::add-element (ambitus-group ambitus) (ambitus-line ambitus))
      ambitus))

#(define-method (initialize-ambitus-state (ambitus <ambitus>) translator)
   "Initialize the state of @var{ambitus}, by getting the starting
position of middle C and key signature from @var{translator}'s context."
   (if (not (ambitus-start-c0 ambitus))
       (begin
         (set! (ambitus-start-c0 ambitus)
               (ly:context-property (ly:translator-context translator)
                                    'middleCPosition
                                    0))
         (set! (ambitus-start-key-sig ambitus)
               (ly:context-property (ly:translator-context translator)
                                    'keyAlterations)))))

#(define-method (update-ambitus-notes (ambitus <ambitus>) note-grob)
   "Update the upper and lower ambitus pithes of @var{ambitus}, using
@var{note-grob}."
   ;; Get the event that caused the note-grob creation
   ;; and check that it is a note-event.
   (let ((note-event (ly:grob-property note-grob 'cause)))
     (if (ly:in-event-class? note-event 'note-event)
         ;; get the pitch from the note event
         (let ((pitch (ly:event-property note-event 'pitch)))
           ;; if this pitch is lower than the current ambitus lower
           ;; note pitch (or it has not been initialized yet),
           ;; then this pitch is the new ambitus lower pitch,
           ;; and conversely for upper pitch.
           (for-each (lambda (direction pitch-compare)
                       (if (or (not (ambitus-note-pitch (ambitus-note ambitus direction)))
                               (pitch-compare pitch
                                              (ambitus-note-pitch (ambitus-note ambitus dire
                           (begin
                             (set! (ambitus-note-pitch (ambitus-note ambitus direction))
                                   pitch)
                             (set! (ambitus-note-cause (ambitus-note ambitus direction))
                                   note-event))))
                     (list DOWN UP)
```

```
                    (list ly:pitch<? (lambda (p1 p2)
                                       (ly:pitch<? p2 p1))))))))

#(define-method (typeset-ambitus (ambitus <ambitus>) translator)
   "Typeset the ambitus:
- place the lower and upper ambitus notes according to their pitch and
  the position of the middle C;
- typeset or delete the note accidentals, according to the key signature.
  An accidental, if it is to be printed, is added to an AccidentalPlacement
  grob (a grob dedicated to the placement of accidentals near a chord);
- both note heads are added to the ambitus line grob, so that a line should
  be printed between them."
   ;; check if there are lower and upper pitches
   (if (and (ambitus-note-pitch (ambitus-note ambitus UP))
            (ambitus-note-pitch (ambitus-note ambitus DOWN)))
       ;; make an AccidentalPlacement grob, for placement of note accidentals
       (let ((accidental-placement (ly:engraver-make-grob
                                    translator
                                    'AccidentalPlacement
                                    (ambitus-note-accidental (ambitus-note ambitus DOWN)))))
         ;; For lower and upper ambitus notes:
         (for-each (lambda (direction)
                     (let ((pitch (ambitus-note-pitch (ambitus-note ambitus direction))))
                       ;; set the cause and the staff position of the ambitus note
                       ;; according to the associated pitch
                       (set! (ly:grob-property (ambitus-note-head (ambitus-note ambitus dire
                                               'cause)
                             (ambitus-note-cause (ambitus-note ambitus direction)))
                       (set! (ly:grob-property (ambitus-note-head (ambitus-note ambitus dire
                                               'staff-position)
                             (+ (ambitus-start-c0 ambitus)
                                (ly:pitch-steps pitch)))
                       ;; determine if an accidental shall be printed for this note,
                       ;; according to the key signature
                       (let* ((handle (or (assoc (cons (ly:pitch-octave pitch)
                                                       (ly:pitch-notename pitch))
                                                 (ambitus-start-key-sig ambitus))
                                          (assoc (ly:pitch-notename pitch)
                                                 (ambitus-start-key-sig ambitus))))
                              (sig-alter (if handle (cdr handle) 0)))
                         (cond ((= (ly:pitch-alteration pitch) sig-alter)
                                ;; the note alteration is in the key signature
                                ;; => it does not have to be printed
                                (ly:grob-suicide!
                                 (ambitus-note-accidental (ambitus-note ambitus direction)))
                                (set! (ly:grob-object (ambitus-note-head (ambitus-note ambit
                                                      'accidental-grob)
                                      '()))
                               (else
                                ;; otherwise, the accidental shall be printed
                                (set! (ly:grob-property (ambitus-note-accidental
                                                         (ambitus-note ambitus direction))
```

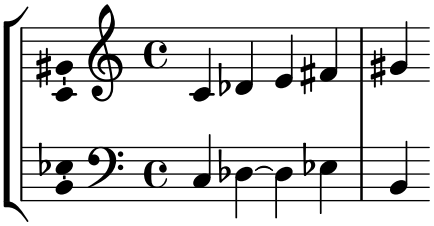
```
                                                    'alteration)
                                    (ly:pitch-alteration pitch)))))
                        ;; add the AccidentalPlacement grob to the
                        ;; conditional items of the AmbitusNoteHead
                        (ly:separation-item::add-conditional-item
                         (ambitus-note-head (ambitus-note ambitus direction))
                         accidental-placement)
                        ;; add the AmbitusAccidental to the list of the
                        ;; AccidentalPlacement grob accidentals
                        (ly:accidental-placement::add-accidental
                         accidental-placement
                         (ambitus-note-accidental (ambitus-note ambitus direction)))
                        ;; add the AmbitusNoteHead grob to the AmbitusLine grob
                        (ly:pointer-group-interface::add-grob
                         (ambitus-line ambitus)
                         'note-heads
                         (ambitus-note-head (ambitus-note ambitus direction)))))
                    (list DOWN UP))
          ;; add the AccidentalPlacement grob to the main Ambitus grob
          (ly:axis-group-interface::add-element (ambitus-group ambitus) accidental-placement)
        ;; no notes ==> suicide the grobs
        (begin
          (for-each (lambda (direction)
                      (ly:grob-suicide! (ambitus-note-accidental (ambitus-note ambitus direct
                      (ly:grob-suicide! (ambitus-note-head (ambitus-note ambitus direction))
                    (list DOWN UP))
          (ly:grob-suicide! ambitus-line))))

%%%
%%% Ambitus engraver definition
%%%
#(define ambitus-engraver
   (lambda (context)
     (let ((ambitus #f))
       ;; when music is processed: make the ambitus object, if not already built
       (make-engraver
    ((process-music translator)
     (if (not ambitus)
         (set! ambitus (make-ambitus translator))))
    ;; set the ambitus clef and key signature state
    ((stop-translation-timestep translator)
     (if ambitus
         (initialize-ambitus-state ambitus translator)))
    ;; when a note-head grob is built, update the ambitus notes
    (acknowledgers
          ((note-head-interface engraver grob source-engraver)
       (if ambitus
           (update-ambitus-notes ambitus grob))))
    ;; finally, typeset the ambitus according to its upper and lower notes
    ;; (if any).
    ((finalize translator)
     (if ambitus
```

```
        (typeset-ambitus ambitus translator)))))))

%%%
%%% Example
%%%

\score {
  \new StaffGroup <<
    \new Staff { c'4 des' e' fis' gis' }
    \new Staff { \clef "bass" c4 des ~ des ees b, }
  >>
  \layout { \context { \Staff \consists #ambitus-engraver } }
}
```



Affichage d'un système complet lorsqu'une seule portée est active

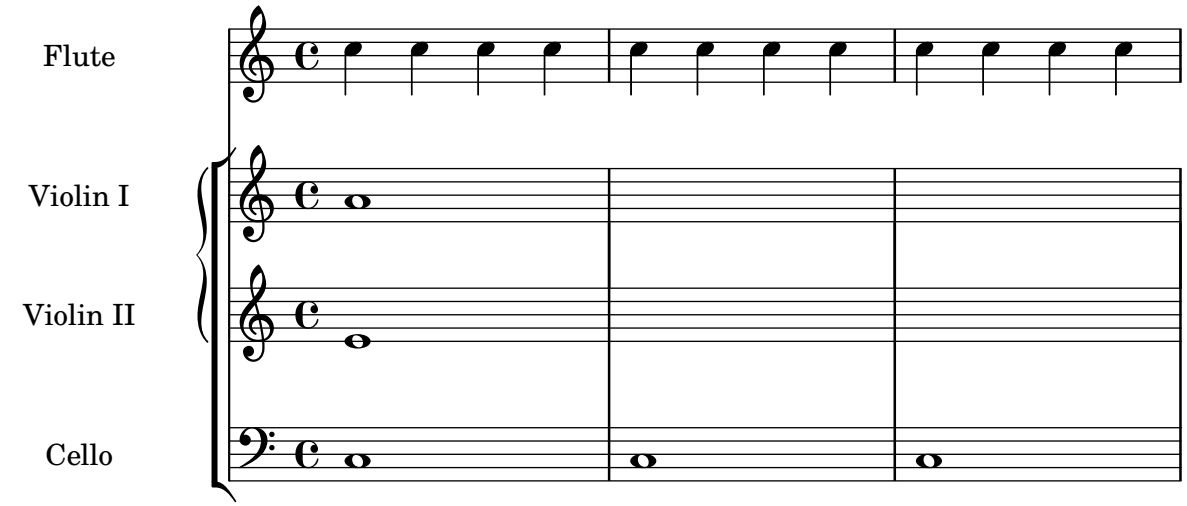
Il arrive, dans les partitions d'orchestre, qu'un instrument ou un groupe d'instruments ne joue pas pendant un moment et que leurs portées disparaisse pendant ce temps – à l'aide de `\removeEmptStaves`.

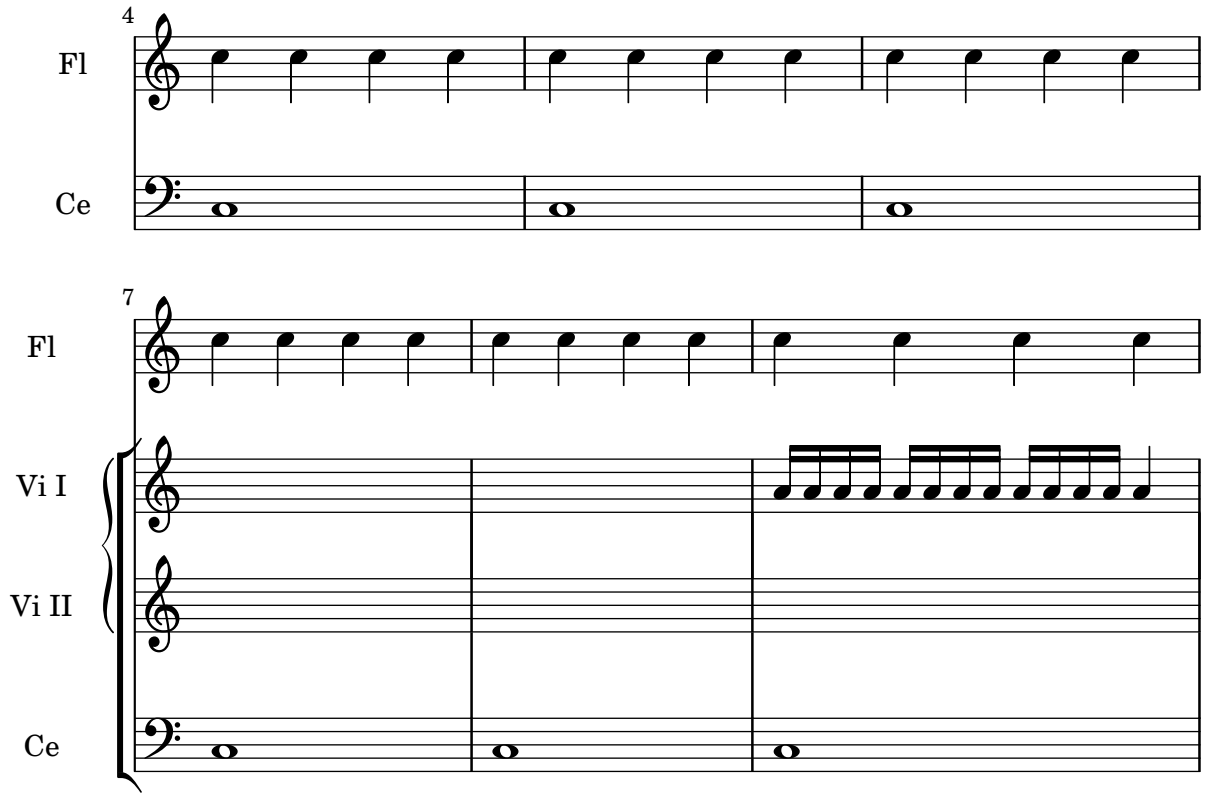
Il est préférable, lorsque l'un d'entre eux reprend, d'imprimer **toutes les portées du groupe en question** grâce au `Keep_alive_together_engraver` placé au niveau du regroupement, qu'il s'agisse d'un `GrandStaff` ou d'un `StaffGroup`.

Dans l'exemple suivant, les violons ne jouent pas lors des deuxième et troisième systèmes. Bien que seul le premier violon redémarre à la dernière mesure, la portée de second violon apparaît elle aussi.

```
\score {
  <<
    \new StaffGroup = "StaffGroup_woodwinds"
    <<
      \new Staff = "Staff_flute" \with {
        instrumentName = "Flute"
        shortInstrumentName = "Fl"
      }
      \relative c' {
        \repeat unfold 3 { c'4 c c c | c c c c | c c c c | \break }
      }
    >>
    \new StaffGroup = "StaffGroup_Strings"
    <<
      \new GrandStaff = "GrandStaff_violins"
      <<
        \new Staff = "StaffViolinI" \with {
          instrumentName = "Violin I"
```

```
          shortInstrumentName = "Vi I"
        }
        \relative c'' {
          a1 \repeat unfold 7 { s1 } \repeat unfold 12 a16  a4
        }
        \new Staff = "StaffViolinII" \with {
          instrumentName = "Violin II"
          shortInstrumentName = "Vi II"
        }
        \relative c' { e1 \repeat unfold 8 { s1 } }
      >>
      \new Staff = "Staff_cello" \with {
        instrumentName = "Cello"
        shortInstrumentName = "Ce"
      }
      \relative c { \clef bass \repeat unfold 9 { c1 }}
    >>
  >>
}
\layout {
  indent = 3.0\cm
  short-indent = 1.5\cm
  \context {
    \GrandStaff
    \consists Keep_alive_together_engraver
  }
  \context {
    \Staff
    \RemoveEmptyStaves
  }
}
```





Les graveurs un par un

Pour appréhender la notation, nous l'avons disséquée en éléments simples : chaque type de symbole est géré par un module différent. Chaque module est appelé « graveur ». Dans cet exemple, chaque graveur est appelé à son tour, dans l'ordre suivant :

– têtes de note,

– symboles de la portée,

– clef,

– hampes,

– ligatures, liaisons, accents,

– altérations, barres de mesure, métrique et armure.

Les graveurs se regroupent. Par exemple, têtes de note, liaisons, ligature, etc. forment un contexte de voix (`Voice`). Les graveurs chargés de la métrique, des altérations, des barres de mesure, etc. forment un contexte de portée (`Staff`).

```
%% sample music
topVoice = \relative c' {
  \key d \major
  es8([ g] a[ fis])
  b4
  b16[-. b-. b-. cis-.]
  d4->
}

botVoice = \relative c' {
  \key d \major
  c8[( f] b[ a)]
  es4
  es16[-. es-. es-. fis-.]
  b4->
```

```
}

hoom = \relative c {
  \key d \major
  \clef bass
  g8-. r
  r4
  fis8-.
  r8
  r4
  b'4->
}

pah = \relative c' {
  r8 b-.
  r4
  r8 g8-.
  r16 g-. r8
  \clef treble
  fis'4->
}

%
% setup for Request->Element conversion. Guru-only
%

MyStaff = \context {
  \type "Engraver_group"
  \name Staff

  \description "Handles clefs, bar lines, keys, accidentals.  It can contain
@code{Voice} contexts."

  \consists "Output_property_engraver"

  \consists "Font_size_engraver"

  \consists "Volta_engraver"
  \consists "Separating_line_group_engraver"
  \consists "Dot_column_engraver"

  \consists "Ottava_spanner_engraver"
  \consists "Rest_collision_engraver"
  \consists "Piano_pedal_engraver"
  \consists "Piano_pedal_align_engraver"
  \consists "Instrument_name_engraver"
  \consists "Grob_pq_engraver"
  \consists "Forbid_line_break_engraver"
  \consists "Axis_group_engraver"

  \consists "Pitch_squash_engraver"
```

```
  localAlterations = #'()

  % explicitly set instrumentName, so we don't get
  % weird effects when doing instrument names for
  % piano staves

  instrumentName = #'()
  shortInstrumentName = #'()

  \accepts "Voice"
  \defaultchild "Voice"
}


MyVoice = \context {
  \type "Engraver_group"
  \name Voice

  \description "
    Corresponds to a voice on a staff.  This context handles the
    conversion of dynamic signs, stems, beams, super- and subscripts,
    slurs, ties, and rests.

    You have to instantiate this explicitly if you want to have
    multiple voices on the same staff."

  localAlterations = #'()
  \consists "Font_size_engraver"

  % must come before all
  \consists "Output_property_engraver"
  \consists "Arpeggio_engraver"
  \consists "Multi_measure_rest_engraver"
  \consists "Text_spanner_engraver"
  \consists "Grob_pq_engraver"
  \consists "Note_head_line_engraver"
  \consists "Glissando_engraver"
  \consists "Ligature_bracket_engraver"
  \consists "Breathing_sign_engraver"
  % \consists "Rest_engraver"
  \consists "Grace_beam_engraver"
  \consists "New_fingering_engraver"
  \consists "Chord_tremolo_engraver"
  \consists "Percent_repeat_engraver"
  \consists "Slash_repeat_engraver"

  %{
    Must come before text_engraver, but after note_column engraver.
  %}
  \consists "Text_engraver"
  \consists "Dynamic_engraver"
  \consists "Dynamic_align_engraver"
```

```
  \consists "Fingering_engraver"

  \consists "Script_column_engraver"
  \consists "Rhythmic_column_engraver"
  \consists "Cluster_spanner_engraver"
  \consists "Tie_engraver"
  \consists "Tie_engraver"
  \consists "Tuplet_engraver"
  \consists "Note_heads_engraver"
  \consists "Rest_engraver"
}


\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}


MyStaff = \context {
  \MyStaff
  \consists "Staff_symbol_engraver"
}

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

MyStaff = \context {
  \MyStaff
  \consists "Clef_engraver"
  \remove "Pitch_squash_engraver"
}

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

MyVoice = \context {
  \MyVoice
  \consists "Stem_engraver"
```

```
}

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

MyVoice = \context {
  \MyVoice
  \consists "Beam_engraver"
}

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

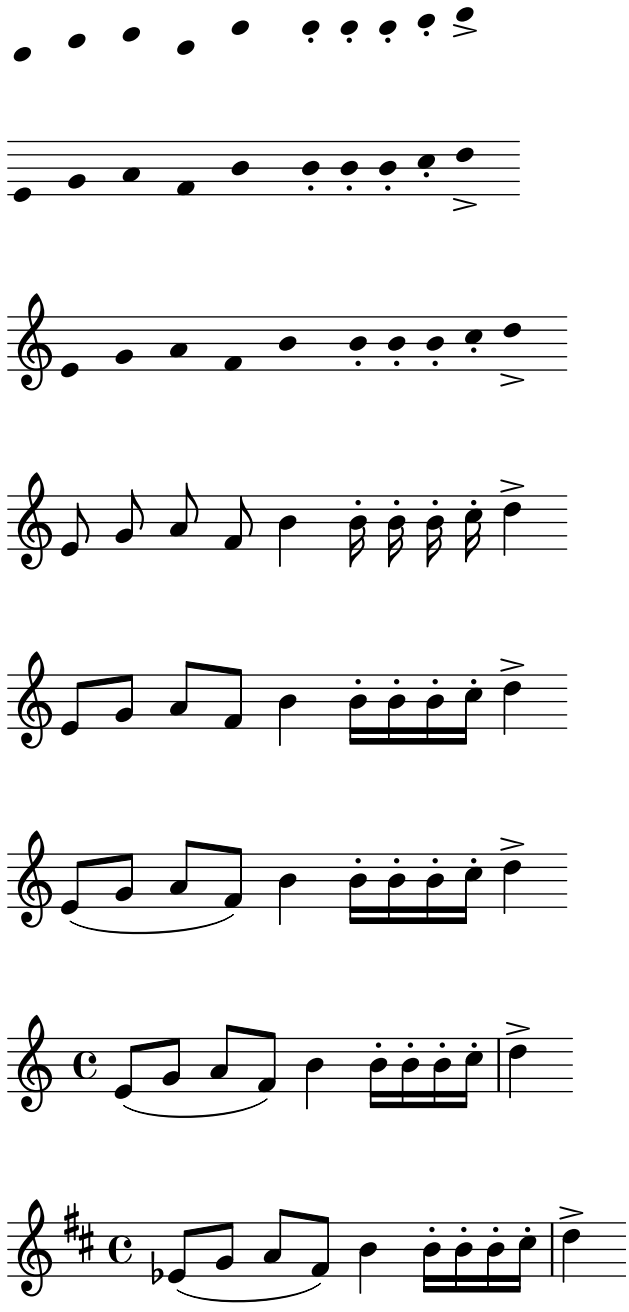
MyVoice = \context {
  \MyVoice
  \consists "Phrasing_slur_engraver"
  \consists "Slur_engraver"
  \consists "Script_engraver"
}


\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}

MyStaff = \context {
  \MyStaff
  \consists "Bar_engraver"
  \consists "Time_signature_engraver"
}

\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}
```

```
MyStaff = \context {
  \MyStaff
  \consists "Accidental_engraver"
  \consists "Key_engraver"
}
\score {
  \topVoice
  \layout {
    \context { \MyStaff }
    \context { \MyVoice }
  }
}
```

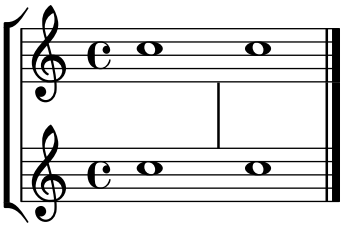


Présentation à l'ancienne (barres de mesure entre les portées)

En musique mensurale, les barres de mesure ne traversent pas les portées. Pour obtenir ce résultat avec un `StaffGroup` plutôt qu'en utilisant un `ChoirStaff`, il faudra masquer les portions de barre qui recouvrent les portées à l'aide d'un `\hide`.

```
global = {
  \hide Staff.BarLine
  s1 s
  % the final bar line is not interrupted
  \undo \hide Staff.BarLine
  \bar "|."
}

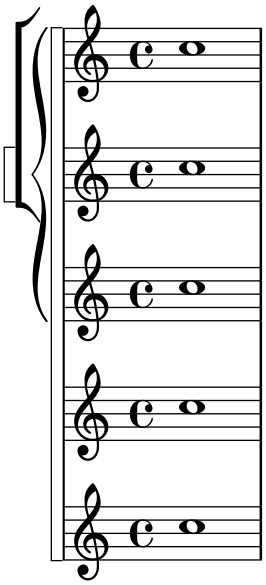
\new StaffGroup \relative c'' {
  <<
    \new Staff { << \global { c1 c } >> }
    \new Staff { << \global { c c } >> }
  >>
}
```



Imbrications de regroupements de portées

La propriété `systemStartDelimiterHierarchy` permet de créer des regroupements imbriqués complexes. La commande `\set StaffGroup.systemStartDelimiterHierarchy` prend en argument la liste alphabétique des sous-groupes à hiérarchiser. Chaque sous-groupe peut être affublé d'un délimiteur particulier. Chacun des regroupements intermédiaires doit être borné par des parenthèses. Bien que des éléments de la liste puissent être omis, le premier délimiteur embrassera toujours l'intégralité des portées. Vous disposez des quatre délimiteurs `SystemStartBar`, `SystemStartBracket`, `SystemStartBrace` et `SystemStartSquare`.

```
\new StaffGroup
\relative c'' <<
  \override StaffGroup.SystemStartSquare.collapse-height = #4
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare (SystemStartBrace (SystemStartBracket a
                             (SystemStartSquare b)  ) c ) d)
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
>>
```



Numérotation de groupes de mesures

Cet extrait illustre l'utilisation du `Measure_counter_engraver` aux fins de numéroter un groupe de mesures successives. La section à numéroter peut éventuellement comporter des reprises.

Le graveur doit être ajouté au contexte approprié. Il est ici attaché au contexte `Staff`, mais pourrait l'être à un contexte `Dynamics`.

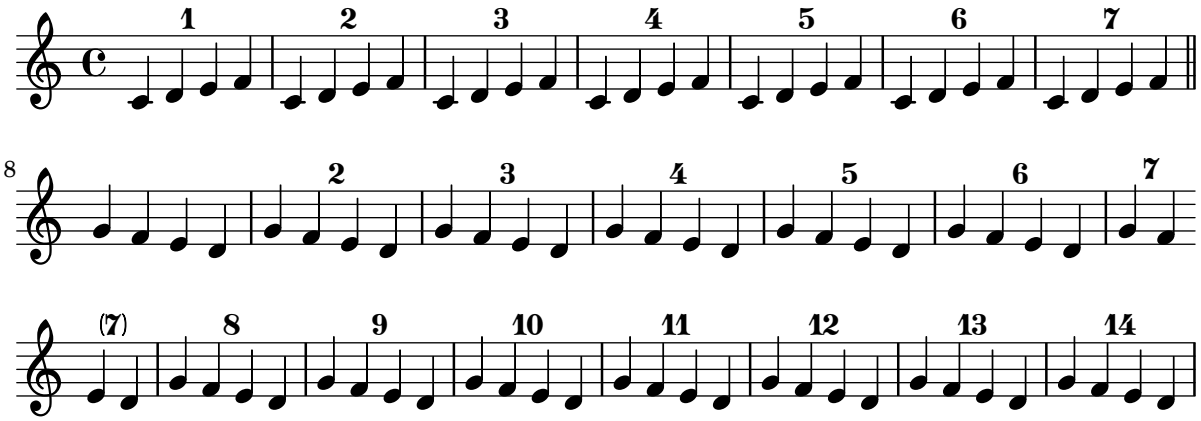
Le compteur est lancé par `\startMeasureCount` et arrêté par `\stopMeasureCount`. La numérotation commence à `1` par défaut, comportement qui peut se modifier à l'aide de la propriété `count-from`.

Lorsqu'une mesure s'étend après un saut de ligne, le numéro y est alors répété, entre parenthèses.

```
\layout {
  \context {
    \Staff
    \consists #Measure_counter_engraver
  }
}

\new Staff {
  \startMeasureCount
  \repeat unfold 7 {
    c'4 d' e' f'
  }
  \stopMeasureCount
  \bar "||"
  g'4 f' e' d'
  \override Staff.MeasureCounter.count-from = #2
  \startMeasureCount
  \repeat unfold 5 {
    g'4 f' e' d'
  }
  g'4 f'
  \bar ""
  \break
  e'4 d'
  \repeat unfold 7 {
```

```
    g'4 f' e' d'
  }
  \stopMeasureCount
}
```

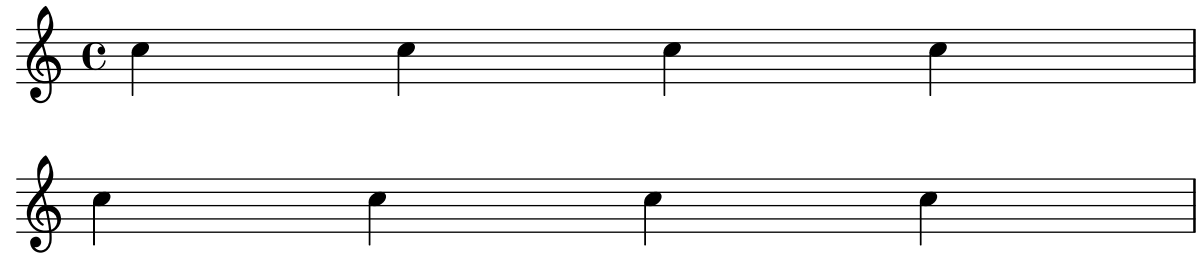


Suppression des numéros de mesure d'une partition

Désactiver le graveur concerné – `Bar_number_engraver` – donnera une partition – contexte `Score` – sans numéros de mesure.

```
\layout {
  \context {
    \Score
    \omit BarNumber
    % or:
    %\remove "Bar_number_engraver"
  }
}

\relative c'' {
  c4 c c c \break
  c4 c c c
}
```

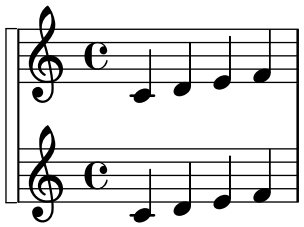


Indication de regroupement de portées par un rectangle

Un regroupement de portées sera indiqué par un simple rectangle – `SystemStartSquare` – en début de ligne dès lors que vous le mentionnerez explicitement au sein d'un contexte `StaffGroup` ou `ChoirStaff`.

```
\score {
  \new StaffGroup { <<
  \set StaffGroup.systemStartDelimiter = #'SystemStartSquare
```

```
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  >> }
}
```



Utilisation de lignes de repères dans une partition « à la française »

Le recours à des contextes `MarkLine` (tel qu'indiqué dans le LSR1010 (`http://lsr.di.unimi.it/LSR/Item?id=1010`)) dans une partition « à la française » peut s'avérer problématique lorsque toutes les portées entre deux contextes `Markline` sont absentes d'un système. La présence du `Keep_alive_together_engraver` au sein de chacun des `StaffGroup` permet de maintenir actif le `MarkLine` uniquement lorsque les autres portées du regroupement sont elles-mêmes actives.
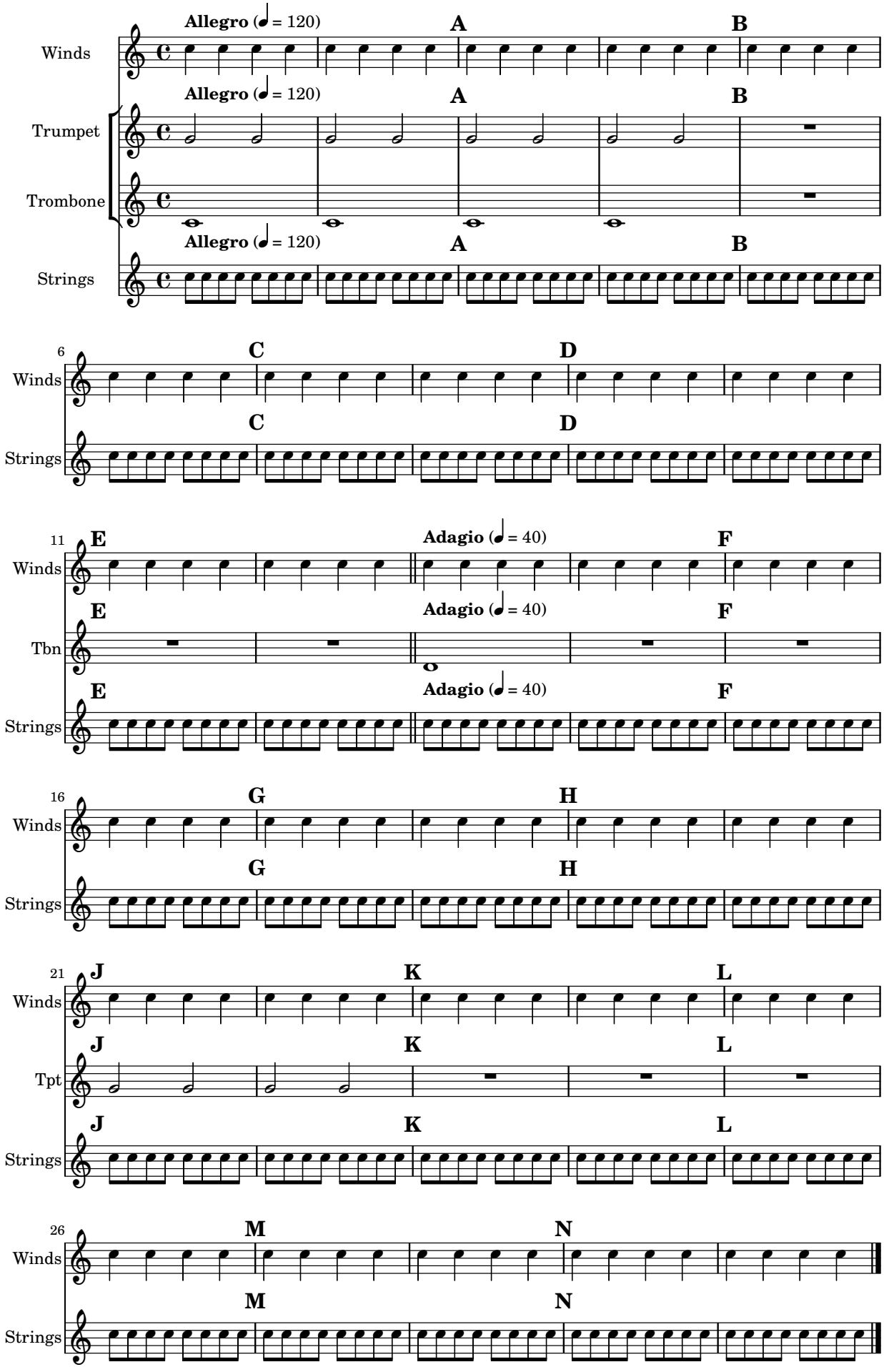
```
bars = {
  \tempo "Allegro" 4=120
  s1*2
  \repeat unfold 5 { \mark \default s1*2 }
  \bar "||"
  \tempo "Adagio" 4=40
  s1*2
  \repeat unfold 8 { \mark \default s1*2 }
  \bar "|."
}
winds = \repeat unfold 120 { c''4 }
trumpet = { \repeat unfold 8 g'2 R1*16 \repeat unfold 4 g'2 R1*8 }
trombone = { \repeat unfold 4 c'1 R1*8 d'1 R1*17 }
strings = \repeat unfold 240 { c''8 }

#(set-global-staff-size 16)
\paper {
  systems-per-page = 5
  ragged-last-bottom = ##f
}

\layout {
  indent = 15\mm
  short-indent = 5\mm
  \context {
    \name MarkLine
    \type Engraver_group
    \consists Output_property_engraver
    \consists Axis_group_engraver
    \consists Mark_engraver
    \consists Metronome_mark_engraver
```

```
    \override VerticalAxisGroup.remove-empty = ##t
    \override VerticalAxisGroup.remove-layer = #'any
    \override VerticalAxisGroup.staff-affinity = #DOWN
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing.basic-distance = 1
    keepAliveInterfaces = #'()
  }
  \context {
    \Staff
    \override VerticalAxisGroup.remove-empty = ##t
    \override VerticalAxisGroup.remove-layer = ##f
  }
  \context {
    \StaffGroup
    \accepts MarkLine
    \consists Keep_alive_together_engraver
  }
  \context {
    \Score
    \remove Mark_engraver
    \remove Metronome_mark_engraver
  }
}

\score {
  <<
    \new StaffGroup = "winds" \with {
      instrumentName = "Winds"
      shortInstrumentName = "Winds"
    } <<
      \new MarkLine \bars
      \new Staff \winds
    >>
    \new StaffGroup = "brass" <<
      \new MarkLine \bars
      \new Staff = "trumpet" \with {
        instrumentName = "Trumpet"
        shortInstrumentName = "Tpt"
      } \trumpet
      \new Staff = "trombone" \with {
        instrumentName = "Trombone"
        shortInstrumentName = "Tbn"
      } \trombone
    >>
    \new StaffGroup = "strings" \with {
      instrumentName = "Strings"
      shortInstrumentName = "Strings"
    } <<
      \new MarkLine \bars
      \new Staff = "strings" { \strings }
    >>
  >>
}
```


Allegro (♩ = 120)
A
B
Winds
Trumpet
Trombone
Strings
6
C
D
11
E
Adagio (♩ = 40)
F
Tbn
16
G
H
21
J
K
L
Tpt
26
M
N

Ensemble vocal avec alignement des paroles selon le contexte

Ce canevas ressemble beaucoup à celui pour chœur à quatre voix mixtes. La différence réside dans le fait que les paroles sont positionnées en ayant recours à `alignAboveContext` et `alignBelowContext`.

```
global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative c'' {
  c4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}
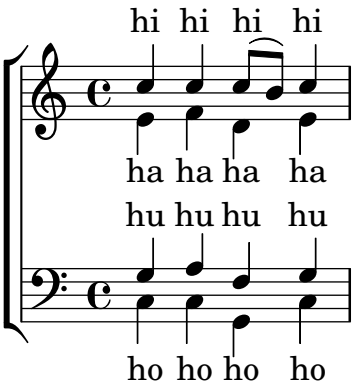
altoMusic = \relative c' {
  e4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

tenorMusic = \relative c' {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative c {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  \new ChoirStaff <<
    \new Staff = "women" <<
      \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
      \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = #"women" }
      \lyricsto "sopranos" \sopWords
    \new Lyrics \with { alignBelowContext = #"women" }
      \lyricsto "altos" \altoWords
    % we could remove the line about this with the line below, since
    % we want the alto lyrics to be below the alto Voice anyway.
    % \new Lyrics \lyricsto "altos" \altoWords
```

```
    \new Staff = "men" <<
      \clef bass
      \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
      \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = #"men" }
      \lyricsto "tenors" \tenorWords
    \new Lyrics \with { alignBelowContext = #"men" }
      \lyricsto "basses" \bassWords
    % again, we could replace the line above this with the line below.
    % \new Lyrics \lyricsto "basses" \bassWords
  >>
}
```



Ensemble vocal avec couplet et refrain

Ce canevas illustre la manière d'agencer une œuvre vocale où le couplet est chanté en solo et le refrain à deux voix. Vous noterez le recours aux silences invisibles dans la variable `\global` : ils permettent de positionner les changements de métrique et autres éléments communs à toutes les parties, ce pour l'intégralité du morceau.

```
global = {
  \key g \major

  % verse
  \time 3/4
  s2.*2
  \break

  % refrain
  \time 2/4
  s2*2
  \bar "|."
}

SoloNotes = \relative g' {
  \clef "treble"

  % verse
  g4 g g |
  b4 b b |
```

```
  % refrain
  R2*2 |
}

SoloLyrics = \lyricmode {
  One two three |
  four five six |
}

SopranoNotes = \relative c'' {
  \clef "treble"

  % verse
  R2.*2 |

  % refrain
  c4 c |
  g4 g |
}

SopranoLyrics = \lyricmode {
  la la |
  la la |
}

BassNotes = \relative c {
  \clef "bass"

  % verse
  R2.*2 |

  % refrain
  c4 e |
  d4 d |
}

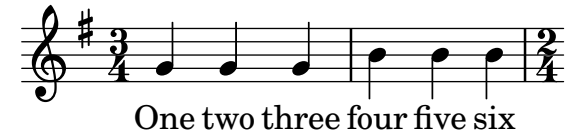
BassLyrics = \lyricmode {
  dum dum |
  dum dum |
}

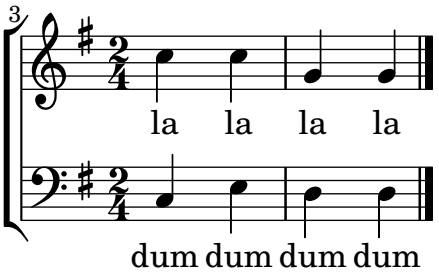
\score {
  <<
    \new Voice = "SoloVoice" << \global \SoloNotes >>
    \new Lyrics \lyricsto "SoloVoice" \SoloLyrics

    \new ChoirStaff <<
      \new Voice = "SopranoVoice" << \global \SopranoNotes >>
      \new Lyrics \lyricsto "SopranoVoice" \SopranoLyrics

      \new Voice = "BassVoice" << \global \BassNotes >>
```

```
      \new Lyrics \lyricsto "BassVoice" \BassLyrics
    >>
  >>
  \layout {
    ragged-right = ##t
    \context { \Staff
      % these lines prevent empty staves from being printed
      \RemoveEmptyStaves
      \override VerticalAxisGroup.remove-first = ##t
    }
  }
}
```





Tweaks and overrides

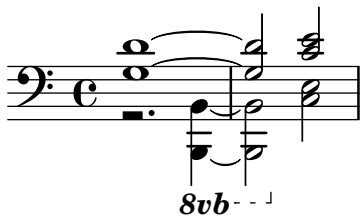
Section “Changing defaults” dans *Manuel de notation*

Section “Tweaking output” dans *Manuel d’initiation*

Ajout d’une indication d’octave pour une seule voix

Lorsque plusieurs voix cohabitent sur une même portée, déterminer l’octaviation d’une voix affectera la position des notes de toutes les voix, jusqu’à la fin du crochet d’octaviation. Si l’octaviation ne doit s’appliquer qu’à une seule voix, les positionnements du do central (propriété `middleCPosition`) et du crochet d’octaviation peuvent s’indiquer explicitement. Dans l’exemple suivant, le `middleCPosition` qui a normalement une valeur de 6 en clef de fa – soit six crans au-dessus de la ligne médiane – est déterminé à 7 crans au-dessus (une octave) pour la durée de l’*ottava*.

```
{
  \clef bass
  << { <g d'>1~ q2 <c' e'> }
  \\
    {
      r2.
      \set Staff.ottavation = #"8vb"
      \once \override Staff.OttavaBracket.direction = #DOWN
      \set Voice.middleCPosition = #(+ 6 7)
      <b,,, b,,>4 ~ |
      q2
      \unset Staff.ottavation
      \unset Voice.middleCPosition
      <c e>2
    }
  >>
}
```



Ajout de liens à des objets

La fonction `add-link`, telle que définie ci-dessous, permet d’ajouter un lien au stencil d’un objet graphique. Elle s’emploie au sein d’un `\override` ou d’un `\tweak`.

À noter que le fonctionnemment du `point-and-click` est perturbé sur les objets ainsi liés.

Cette fonction n’est opérationnelle que pour une sortie PDF.

La coloration des objets liés s’obtient par une commande séparée.

```
#(define (add-link url-strg)
  (lambda (grob)
    (let* ((stil (ly:grob-property grob 'stencil)))
      (if (ly:stencil? stil)
          (let* ((x-ext (ly:stencil-extent stil X))
```

```
                (y-ext (ly:stencil-extent stil Y))
                (url-expr `(url-link ,url-strg ,x-ext ,y-ext))
                (new-stil
                  (ly:stencil-add
                    (ly:make-stencil url-expr x-ext y-ext)
                    stil)))
          (ly:grob-set-property! grob 'stencil new-stil))))))

%%%% test

%% For easier maintenance of this snippet the URL is formatted to use the
%% actually used LilyPond version.
%% Of course a literal URL would work as well.

#(define major.minor-version
  (string-join (take (string-split (lilypond-version) #\.) 2) "."))

urlI =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/writing-pitches"
  major.minor-version)

urlII =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/rhythms"
  major.minor-version)

urlIII =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/note-heads"
  major.minor-version)

urlIV =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/beams"
  major.minor-version)

urlV =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/note-head-styles"
  major.minor-version)

urlVI =
#(format #f
  "http://lilypond.org/doc/v~a/Documentation/notation/writing-pitches"
  major.minor-version)

\relative c' {
   \key cis \minor
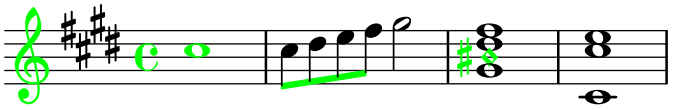
   \once \override Staff.Clef.color = #green
   \once \override Staff.Clef.after-line-breaking =
```

```
    #(add-link urlI)

  \once \override Staff.TimeSignature.color = #green
  \once \override Staff.TimeSignature.after-line-breaking =
    #(add-link urlII)

  \once \override NoteHead.color = #green
  \once \override NoteHead.after-line-breaking =
    #(add-link urlIII)

  cis'1
  \once \override Beam.color = #green
  \once \override Beam.after-line-breaking =
    #(add-link urlIV)
  cis8 dis e fis gis2
  <gis,
   \tweak Accidental.color #green
   \tweak Accidental.after-line-breaking #(add-link urlVI)
   \tweak color #green
   \tweak after-line-breaking #(add-link urlV)
   \tweak style #'harmonic
   bis
   dis
   fis
  >1
  <cis, cis' e>
}
```



Ajout de *markups* à une tablature

Par défaut, les *markups* n'apparaissent pas dans une tablature.

Il suffit, pour les voir apparaître, d'une simple commande `\revert TabStaff.TextScript.stencil`.

```
%% http://lsr.di.unimi.it/LSR/Item?id=919
% by P.P.Schneider on June 2014

high  = { r4 r8 <g c'> q r8 r4 }

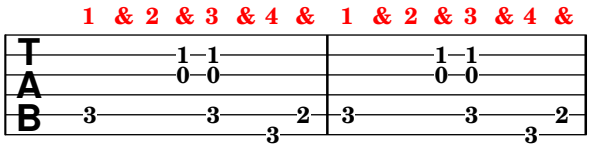
low = { c4 r4 c8 r8 g,8 b, }

pulse = { s8^"1" s^"&" s^"2" s^"&" s^"3" s^"&" s^"4" s^"&" }

\score {
 \new TabStaff {
   \repeat unfold 2 << \high \\ \low \\ \pulse >>
  }
  \layout {
    \context {
```

```
      \TabStaff
      \clef moderntab
      \revert TextScript.stencil
      \override TextScript.font-series = #'bold
      \override TextScript.font-size = #-2
      \override TextScript.color = #red

    }
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1/8)
    }
  }
}
```



Ajout de marques temporelles à un long glissando

Lorsqu'un glissando s'étend dans la durée, on trouve parfois des indications temporelles, matérialisées par des hampes sans tête de note. De telles hampes permettent aussi d'indiquer des éléments intermédiaires.

L'alignement des hampes avec la ligne de glissando peut requérir quelques aménagements.

```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}
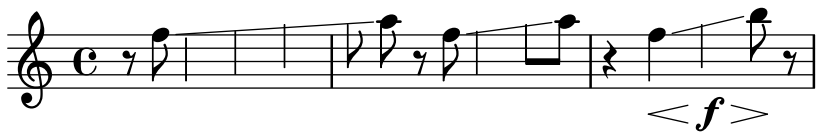
glissandoSkipOff = {
  \revert NoteColumn.glissando-skip
  \undo \hide NoteHead
  \revert NoteHead.no-ledgers
}

\relative c'' {
  r8 f8\glissando
  \glissandoSkipOn
  f4 g a a8\noBeam
  \glissandoSkipOff
  a8

  r8 f8\glissando
  \glissandoSkipOn
  g4 a8
  \glissandoSkipOff
  a8 |

  r4 f\glissando \<
```

```
  \glissandoSkipOn
  a4\f \>
  \glissandoSkipOff
  b8\! r |
}
```
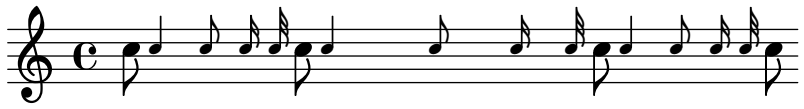


Ajustement de l'espacement des notes d'ornement

La propriété `spacing-increment` de `Score.GraceSpacing` permet d'ajuster l'espacement d'un groupe de notes d'ornement.

```
graceNotes = {
  \grace { c4 c8 c16 c32 }
  c8
}

\relative c'' {
  c8
  \graceNotes
  \override Score.GraceSpacing.spacing-increment = #2.0
  \graceNotes
  \revert Score.GraceSpacing.spacing-increment
  \graceNotes
}
```
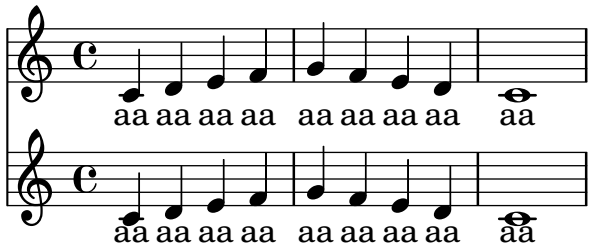


Ajustement de l'espacement vertical des paroles

Cet extrait illustre la manière de réduire l'espace entre la ligne de paroles et la portée.

```
% Default layout:
<<
  \new Staff \new Voice = melody \relative c' {
    c4 d e f
    g4 f e d
    c1
  }
  \new Lyrics \lyricsto melody { aa aa aa aa aa aa aa aa aa }

  \new Staff {
    \new Voice = melody \relative c' {
      c4 d e f
      g4 f e d
      c1
    }
  }
```

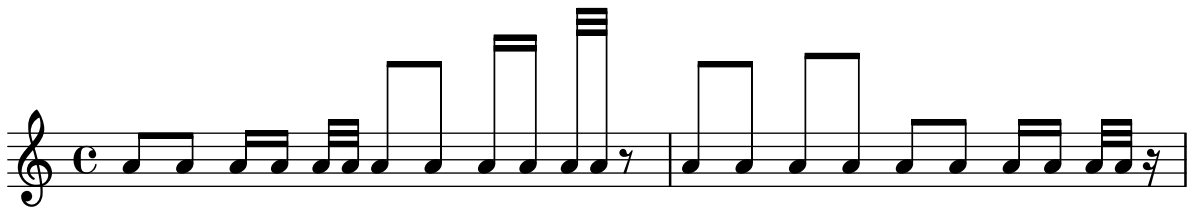
```
  % Reducing the minimum space below the staff and above the lyrics:
  \new Lyrics \with {
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing =
      #'((basic-distance . 1))
  }
  \lyricsto melody { aa aa aa aa aa aa aa aa aa }
>>
```



Modification de la hauteur de hampes ligaturées

La hauteur de hampe des notes ligaturées est gérée par la sous-propriété `beamed-lengths` des `details` de l'objet `Stem`. Lorsqu'elle ne comporte qu'une seule valeur, cette hauteur s'appliquera à toutes les hampes. En présence de plusieurs arguments, le premier affectera les croches, le second les doubles croches, et ainsi de suite. Le dernier argument s'appliquera aussi aux notes de plus courte durée. Les arguments peuvent être des valeurs décimales.

```
\relative c'' {
  \override Stem.details.beamed-lengths = #'(2)
  a8[ a] a16[ a] a32[ a]
  \override Stem.details.beamed-lengths = #'(8 10 12)
  a8[ a] a16[ a] a32[ a] r8
  \override Stem.details.beamed-lengths = #'(8)
  a8[ a]
  \override Stem.details.beamed-lengths = #'(8.5)
  a8[ a]
  \revert Stem.details.beamed-lengths
  a8[ a] a16[ a] a32[ a] r16
}
```
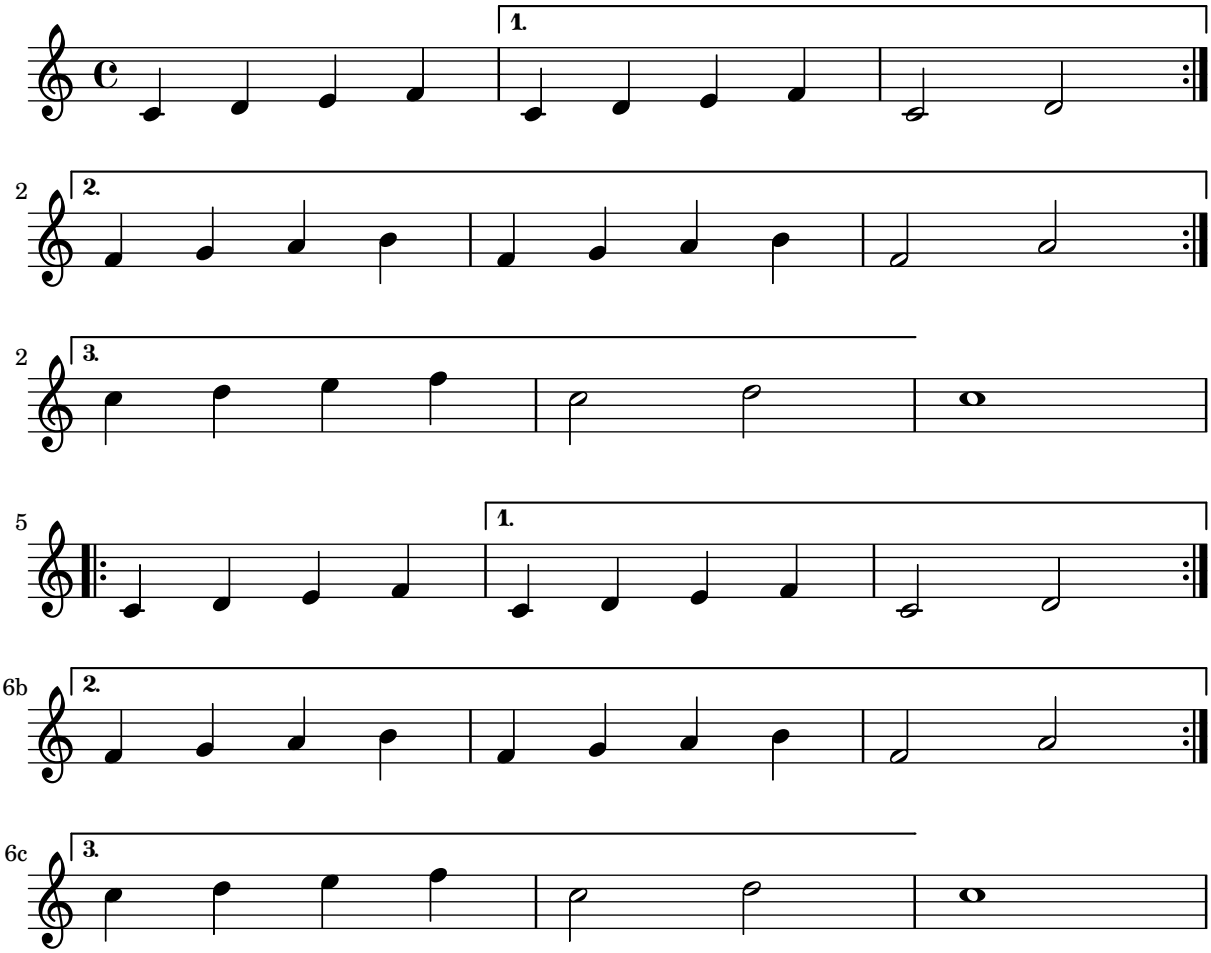


Numérotation des mesures et alternatives

Deux méthodes alternatives vous permettent de gérer la numérotation des mesures en cas de reprises.

```
\relative c'{
  \set Score.alternativeNumberingStyle = #'numbers
  \repeat volta 3 { c4 d e f | }
    \alternative {
      { c4 d e f | c2 d \break }
```

```
      { f4 g a b | f4 g a b | f2 a | \break }
      { c4 d e f | c2 d }
    }
  c1 \break
  \set Score.alternativeNumberingStyle = #'numbers-with-letters
  \repeat volta 3 { c,4 d e f | }
    \alternative {
      { c4 d e f | c2 d \break }
      { f4 g a b | f4 g a b | f2 a | \break }
      { c4 d e f | c2 d }
    }
  c1
}
```



Crochets d'analyse au-dessus de la portée

Les crochets d'analyse viennent par défaut se positionner au-dessous de la portée. L'exemple suivant vous indique comment les faire apparaître en surplomb de la portée.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}
```

```
\relative c'' {
  \once \override HorizontalBracket.direction = #UP
  c2\startGroup
  d2\stopGroup
}
```

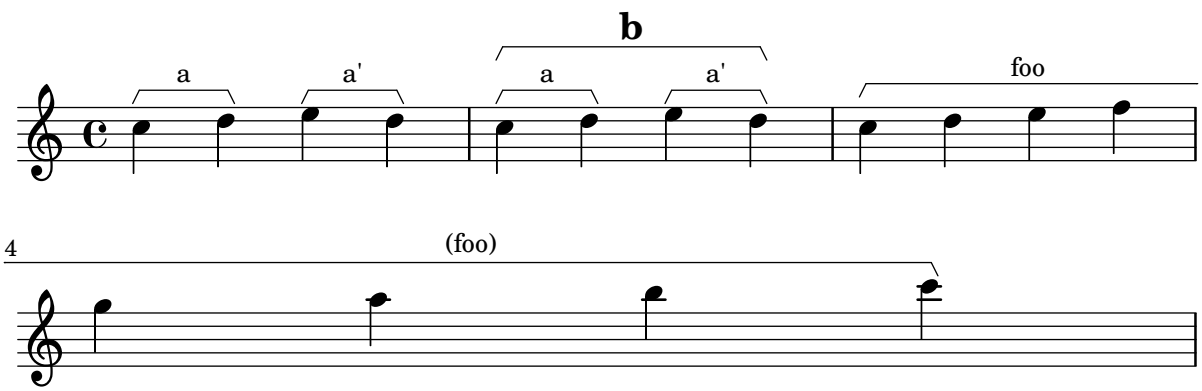


Crochet d'analyse avec texte

Du texte peut venir s'ajouter aux crochets d'analyse grâce à la propriété `texte` de l'objet graphique `HorizontalBracketText`. Plusieurs crochets présents en un même moment requièrent d'utiliser la commande `\tweak`. Le texte ajouté sera répété, entre parenthèse, après un saut de ligne.

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
    \override HorizontalBracket.direction = #UP
  }
}

{
  \once\override HorizontalBracketText.text = "a"
    c''\startGroup d''\stopGroup
    \once\override HorizontalBracketText.text = "a'"
    e''\startGroup d''\stopGroup |
  c''-\tweak HorizontalBracketText.text
        \markup \bold \huge "b" \startGroup
     -\tweak HorizontalBracketText.text "a" \startGroup
    d''\stopGroup
    e''-\tweak HorizontalBracketText.text "a'" \startGroup
    d''\stopGroup\stopGroup |
  c''-\tweak HorizontalBracketText.text foo \startGroup
    d'' e'' f'' | \break
  g'' a'' b'' c'''\stopGroup
}
```

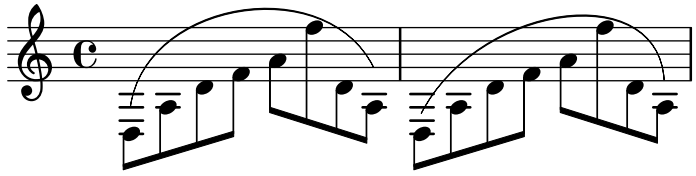


Liaison asymétrique

Une liaison peut adopter une courbe asymétrique afin de s'adapter au mieux à la ligne mélodique.

```
slurNotes = { d,8( a' d f a f' d, a) }

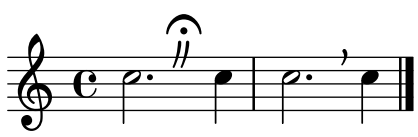
\relative c' {
  \stemDown
  \slurUp
  \slurNotes
  \once \override Slur.eccentricity = #3.0
  \slurNotes
}
```



Césure en forme de « voie de chemin de fer » avec point d'orgue

Une césure peut parfois s'indiquer par une double respiration surmontée d'un point d'orgue. Le code ci-dessous permet d'obtenir la combinaison répondant à cette esthétique.

```
\relative c'' {
  c2.
  % construct the symbol
  \override BreathingSign.text = \markup {
    \override #'(direction . 1)
    \override #'(baseline-skip . 1.8)
    \dir-column {
      \translate #'(0.155 . 0)
        \center-align \musicglyph "scripts.caesura.curved"
      \center-align \musicglyph "scripts.ufermata"
    }
  }
  \breathe c4
  % set the breathe mark back to normal
  \revert BreathingSign.text
  c2. \breathe c4
  \bar "|."
}
```

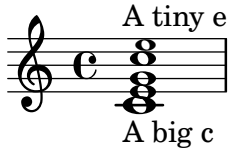


Modification de la taille d'une note particulière d'un accord

La commande `\tweak` permet de modifier la propriété `font-size` d'une note particulière d'un accord.

Il s'agit de placer, à l'intérieur même de l'accord (dans la construction `< >`) et avant la note considérée, l'instruction `\tweak` suivie de `font-size` et de définir la taille voulue – comme `#-2` pour une petite tête.
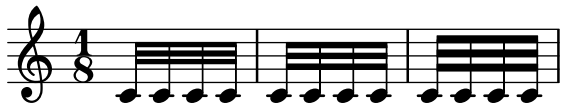
```
\relative c' {
  <\tweak font-size #+2 c e g c
   \tweak font-size #-2 e>1
   ^\markup { A tiny e }_\markup { A big c }
}
```



Modification de l'épaisseur et de l'écartement des ligatures

L'épaisseur des ligatures se gère au niveau de la propriété `Beam.beam-thickness`. L'ajustement de l'espace entre les ligatures est géré par la propriété `Beam.length-fraction`.

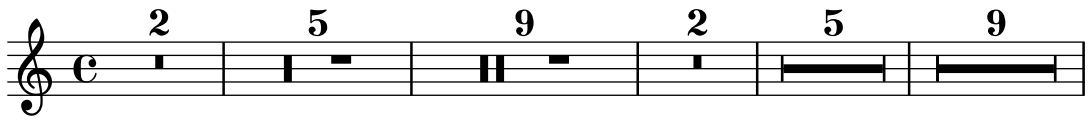
```
\relative f' {
  \time 1/8
  \override Beam.beam-thickness = #0.4
  \override Beam.length-fraction = #0.8
  c32 c c c
  \revert Beam.beam-thickness  % 0.48 is default thickness
  \revert Beam.length-fraction  % 1.0 is default spacing
  c32 c c c
  \override Beam.beam-thickness = #0.6
  \override Beam.length-fraction = #1.3
  c32 c c c
}
```



Modifier l'apparence d'un silence multimesure

Dans le cas où ce silence dure moins de dix mesures, LilyPond imprime sur la portée des « silences d'église » – *Kirchenpause* en allemand – et qui sont une simple suite de rectangles. La propriété `expand-limit` permet d'obtenir un silence unique :

```
\relative c'' {
  \compressMMRests {
    R1*2 | R1*5 | R1*9
    \override MultiMeasureRest.expand-limit = #3
    R1*2 | R1*5 | R1*9
  }
}
```
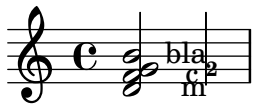


Modification des propriétés d'objets particuliers

La commande `\applyOutput` permet de personnaliser n'importe quel objet de rendu. Elle requiert une fonction Scheme à trois arguments.

```
#(define (mc-squared grob grob-origin context)
   (let ((sp (ly:grob-property grob 'staff-position)))
     (ly:grob-set-property!
      grob 'stencil
      (grob-interpret-markup grob
                             #{ \markup \lower #0.5
                                #(case sp
                                   ((-5) "m")
                                   ((-3) "c ")
                                   ((-2) #{ \markup \teeny \bold 2 #})
                                   (else "bla")) #}))))
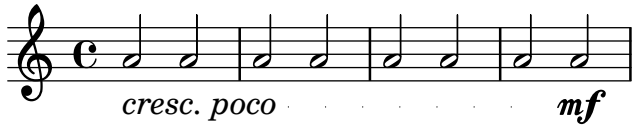
\relative c' {
  <d f g b>2
  \applyOutput Voice.NoteHead #mc-squared
  <d f g b>2
}
```



Modification du texte et de l'extension de nuances textuelles

Le texte par défaut des crescendos et decrescendos se change en modifiant les propriétés de contexte `crescendoText` et `decrescendoText`. L'aspect de la ligne d'extension est fonction de la propriété `style` du `DynamicTextSpanner`. Sa valeur par défaut est `'hairpin`, mais d'autres valeurs sont disponibles, comme `'line`, `'dashed-line` et `'dotted-line`.

```
\relative c'' {
  \set crescendoText = \markup { \italic { cresc. poco } }
  \set crescendoSpanner = #'text
  \override DynamicTextSpanner.style = #'dotted-line
  a2\< a
  a2 a
  a2 a
  a2 a\mf
}
```



Changement des fontes textuelles par défaut

Les familles de fontes par défaut pour le texte peuvent être changées à l'aide de la fonction `make-pango-font-tree`.

```
%{
```

```
You may have to install additional fonts.

Red Hat Fedora

  dejavu-fonts-all

Debian GNU/Linux, Ubuntu

  fonts-dejavu-core
  fonts-dejavu-extra
%}

\paper {
  % change for other default global staff size.
  myStaffSize = #20
  %{
     run
         lilypond -dshow-available-fonts
     to show all fonts available in the process log.
  %}

  #(define fonts
    (make-pango-font-tree "DejaVu Serif"
                          "DejaVu Sans"
                          "DejaVu Sans Mono"
     (/ myStaffSize 20)))
}

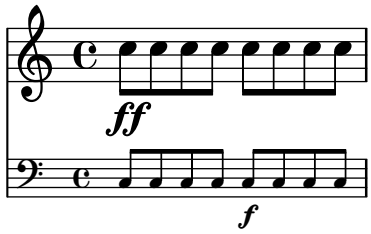
{
  g'''4^\markup {
    DejaVu Serif: \bold bold
                  \italic italic
                  \italic \bold { bold italic }
  }
  g4_\markup {
    \override #'(font-family . sans) {
      DejaVu Sans: \bold bold
                   \italic italic
                   \italic \bold { bold italic }
    }
  }
  g''2^\markup {
    \override #'(font-family . typewriter) {
      DejaVu Sans Mono: \bold bold
                        \italic italic
                        \italic \bold { bold italic }
    }
  }
}
```



Modification de la taille d'une portée

Bien que le meilleur moyen de définir la taille des portées consiste à utiliser `#(set-global-staff-size xx)`, une portée en particulier peut se redimensionner en affectant d'un coefficient ses propriétés `staff-space` et `fontSize`.
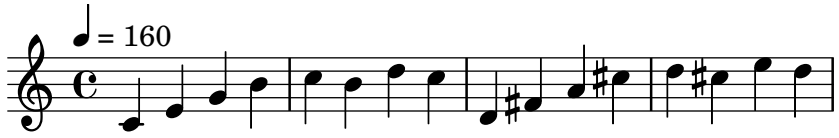
```
<<
  \new Staff {
    \relative c'' {
      \dynamicDown
      c8\ff c c c c c c c
    }
  }
  \new Staff \with {
    fontSize = #-3
    \override StaffSymbol.staff-space = #(magstep -3)
  } {
    \clef bass
    c8 c c c c\f c c c
  }
>>
```



Changement de tempo sans indication sur la partition

Vous pouvez indiquer un changement de tempo pour le fichier MIDI sans pour autant l'imprimer. Il suffit alors de le rendre invisible aux musiciens.

```
\score {
  \new Staff \relative c' {
    \tempo 4 = 160
    c4 e g b
    c4 b d c
    \set Score.tempoHideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
  \layout { }
  \midi { }
}
```
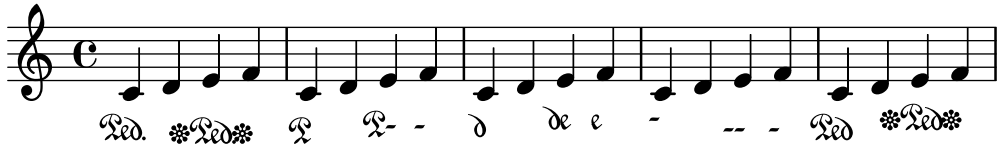


Modification du texte des indications de pédale

`Staff.pedalSustainStrings` permet de définir le texte affiché pour les instructions de pédale. Les seuls caractères autorisés sont les glyphes particuliers de pédale, comme vous pouvez le constater ici.

```
sustainNotes = { c4\sustainOn d e\sustainOff\sustainOn f\sustainOff }

\relative c' {
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("P" "P-" "-")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("d" "de" "e")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("M" "M-" "-")
  \sustainNotes
  \set Staff.pedalSustainStrings = #'("Ped" "*Ped" "*")
  \sustainNotes
}
```



Contrôle de la visibilité des extensions d'objet après saut de ligne

La visibilité des extensions qui se terminent sur la première note après un saut de ligne est contrôlée par un appel de `after-line-breaking` à la fonction `ly:spanner::kill-zero-spanned-time`.

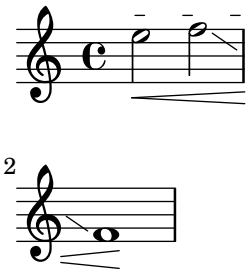
Pour des objets tels qu'un glissando ou un soufflet, le comportement par défaut est de ne pas être reportés après un saut de ligne. L'extension sera donc reprise dès lors que l'appel aura été désactivé.

Il en va inversement pour les extensions qui, par défaut pour les textes notamment, sont reportées après un saut de ligne ; il faudra alors activer l'appel pour empêcher leur report.

```
\paper { ragged-right = ##t }

\relative c'' {
  \override Hairpin.to-barline = ##f
  \override Glissando.breakable = ##t
  % show hairpin
  \override Hairpin.after-line-breaking = ##t
  % hide text span
  \override TextSpanner.after-line-breaking =
    #ly:spanner::kill-zero-spanned-time
  e2\<\startTextSpan
  % show glissando
  \override Glissando.after-line-breaking = ##t
  f2\glissando
```

```
  \break
  f,1\!\stopTextSpan
}
```
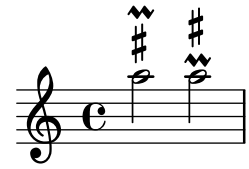


Contrôle de l'ordre vertical des articulations et ornements

Les symboles s'ordonnent verticalement suivant la propriété `script-priority`. Plus sa valeur numérique est faible, plus le symbole sera proche de la note. Dans l'exemple suivant, l'objet `TextScript` – le dièse – a d'abord la propriété la plus basse et se voit donc placé au plus près de la note ; ensuite, c'est l'objet `Script` – le mordant – qui a la propriété la plus basse, et se place alors sous le dièse. Lorsque deux objets ont la même priorité, c'est l'ordre dans lequel ils sont indiqués qui détermine lequel sera placé en premier.

```
\relative c''' {
  \once \override TextScript.script-priority = #-100
  a2^\prall^\markup { \sharp }

  \once \override Script.script-priority = #-100
  a2^\prall^\markup { \sharp }
}
```



Contrôle de l'impression des crochets de n-olet

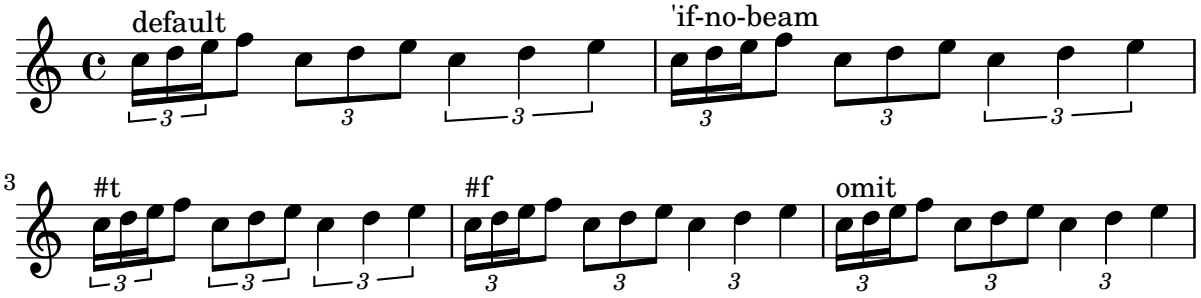
Selon la tradition, les crochets indicateurs de n-olet sont toujours imprimés, sauf dans le cas où ils seraient de la même longueur qu'une ligature.

LilyPond permet, au travers de la propriété `bracket-visibility`, de contôler précisément leur affichage : déterminée à `#t`, ils seront toujours imprimés ; `#f` permet de ne jamais les imprimer – donc omettre l'objet `TupletBracket` –, et `#'if-no-beam` les imprimera en l'absence de ligature (comportement par défaut).

```
music = \relative c'' {
  \tuplet 3/2 { c16[ d e } f8]
  \tuplet 3/2 { c8 d e }
  \tuplet 3/2 { c4 d e }
}

\new Voice {
  \relative c' {
    << \music s4^"default" >>
```
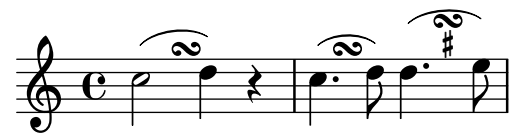
```
    \override TupletBracket.bracket-visibility = #'if-no-beam
    << \music s4^"'if-no-beam" >>
    \override TupletBracket.bracket-visibility = ##t
    << \music s4^"#t" >>
    \override TupletBracket.bracket-visibility = ##f
    << \music s4^"#f" >>
    \omit TupletBracket
    << \music s4^"omit" >>
  }
}
```



Création d'un groupetto retardé

Obtenir un groupetto retardé et dans lequel la note la plus basse est altérée requiert quelques surcharges. La propriété `outside-staff-priority` doit être désactivée (`#f`) pour éviter qu'elle prenne le pas sur la propriété `avoid-slur`. L'ajustement du positionnement horizontal s'effectue en jouant sur les fractions `2/3` et `1/3`.

```
\relative c'' {
  c2*2/3 ( s2*1/3\turn d4) r
  <<
    { c4.( d8) }
    { s4 s\turn }
  >>
  \transpose c d \relative c'' <<
    { c4.( d8) }
    {
      s4
      \once \set suggestAccidentals = ##t
      \once \override AccidentalSuggestion.outside-staff-priority = ##f
      \once \override AccidentalSuggestion.avoid-slur = #'inside
      \once \override AccidentalSuggestion.font-size = -3
      \once \override AccidentalSuggestion.script-priority = -1
      \single \hideNotes
      b8-\turn \noBeam
      s8
    }
  >>
}
```



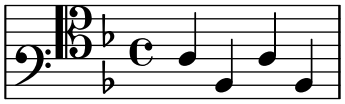
Création d'armures personnalisées

LilyPond accepte les armures personnalisées. Dans cet exemple, il s'agit d'une tonalité de ré mineur à laquelle on rajoute des bémols.

```
\new Staff \with {
  \override StaffSymbol.line-count = #8
  \override KeySignature.flat-positions = #'((-7 . 6))
  \override KeyCancellation.flat-positions = #'((-7 . 6))
  % presumably sharps are also printed in both octaves
  \override KeySignature.sharp-positions = #'((-6 . 7))
  \override KeyCancellation.sharp-positions = #'((-6 . 7))

  \override Clef.stencil = #
  (lambda (grob)(grob-interpret-markup grob
  #{ \markup\combine
    \musicglyph "clefs.C"
    \translate #'(-3 . -2)
    \musicglyph "clefs.F"
   #}))
    clefPosition = #3
    middleCPosition = #3
    middleCClefPosition = #3
}

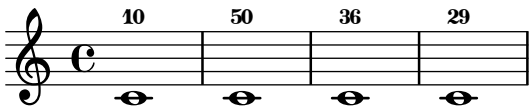
{
  \key d\minor
  f bes, f bes,
}
```



Création de doigtés sur deux caractères

Il est tout à fait possible de noter un doigté supérieur à 5.

```
\relative c' {
  c1-10
  c1-50
  c1-36
  c1-29
}
```



Création de repères simultanés
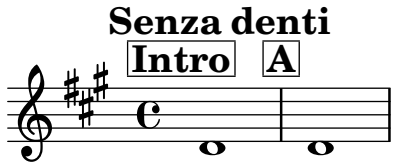
Les indications de repère, à la différence des scripts textuels, ne peuvent s'empiler en un même endroit de la partition : il n'est possible de créer qu'un seul objet `RehearsalMark` à la fois. Le recours à une mesure invisible et à une barre de mesure permet cependant d'ajouter une autre indication de repère, donnant ainsi l'impression d'un double repère sur la même colonne.

Cette méthode permet aussi de positionner un repère en fin de ligne et un autre au début de la ligne suivante.

```
{
  \key a \major
  \set Score.markFormatter = #format-mark-box-letters
  \once \override Score.RehearsalMark.outside-staff-priority = #5000
  \once \override Score.RehearsalMark.self-alignment-X = #LEFT
  \once \override Score.RehearsalMark.break-align-symbols = #'(key-signature)
  \mark \markup { \bold { Senza denti } }

  % the hidden measure and bar line
  % \cadenzaOn turns off automatic calculation of bar numbers
  \cadenzaOn
  \once \override Score.TimeSignature.stencil = ##f
  \time 1/16
  s16 \bar ""
  \cadenzaOff

  \time 4/4
  \once \override Score.RehearsalMark.self-alignment-X = #LEFT
  \mark \markup { \box \bold Intro }
  d'1
  \mark \default
  d'1
}
```



Création d'extensions de texte

Les commandes `\startTextSpan` et `\stopTextSpan` permettent d'ajouter une ligne de prolongation aux indications textuelles, à l'instar des indications de pédale ou d'octaviation. Jouer sur les propriétés de l'objet `TextSpanner` permet d'en modifier le rendu.

```
\paper { ragged-right = ##f }

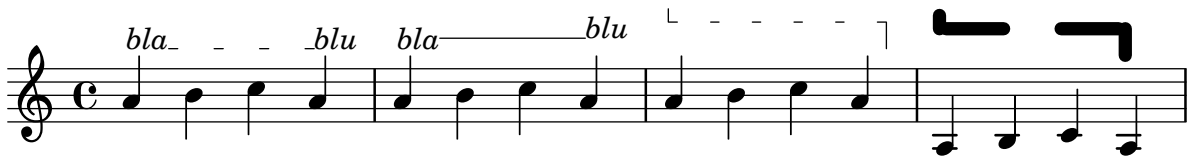
\relative c'' {
  \override TextSpanner.bound-details.left.text = #"bla"
  \override TextSpanner.bound-details.right.text = #"blu"
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \override TextSpanner.style = #'line
  \once \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan
```

```
  \override TextSpanner.style = #'dashed-line
  \override TextSpanner.bound-details.left.text =
    \markup { \draw-line #'(0 . 1) }
  \override TextSpanner.bound-details.right.text =
    \markup { \draw-line #'(0 . -2) }
  \once \override TextSpanner.bound-details.right.padding = #-2

  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan

  \set Staff.middleCPosition = #-13
  \override TextSpanner.dash-period = #10
  \override TextSpanner.dash-fraction = #0.5
  \override TextSpanner.thickness = #10
  a4 \startTextSpan
  b4 c
  a4 \stopTextSpan
}
```

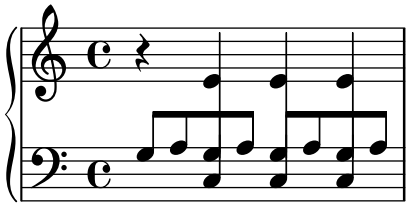


Accord distribué et problème de hampe - solution

Il est parfois préférable d'utiliser les hampes de la portée supérieure pour créer des accords distribués, afin d'éviter tout risque de collision au niveau des ligatures automatiques. Dans l'exemple suivant, le fait de partir des hampes de la portée inférieure aurait rendu nécessaire l'adaptation du détecteur de collision des ligatures, par une clause `\override Staff.Beam.collision-voice-only = ##t`, afin qu'il ne tienne pas compte des collisions entre portées.

```
\new PianoStaff <<
  \new Staff = up
    \relative c' {
      <<
        { r4
          \override Stem.cross-staff = ##t
          \override Stem.length = #19 % this is in half-spaces,
              % so it makes stems 9.5 staffspaces long
          \override Stem.Y-offset = #-6 % stems are normally lengthened
              % upwards, so here we must lower the stem by the amount
              % equal to the lengthening - in this case (19 - 7) / 2
              % (7 is default stem length)
          e e e }
        { s4
          \change Staff = "bottom"
          \override NoteColumn.ignore-collision = ##t
          c, c c
        }
```

```
      >>
    }
  \new Staff = bottom
    \relative c' {
      \clef bass
      \voiceOne
      g8 a g a g a g a
    }
>>
```



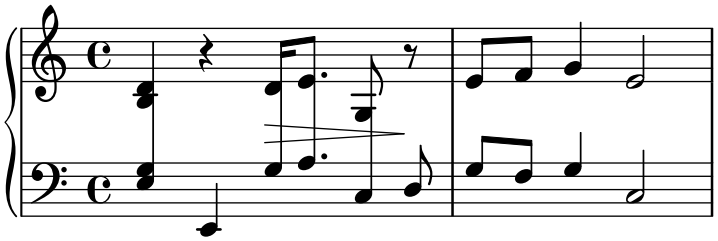
Hampes interportées

L'exemple ci-dessous illustre l'utilisation du `Span_stem_engraver` et de la commande `\crossStaff` afin de connecter des hampes entre les portées.

Nul n'est besoin de spécifier la taille des hampes ; le graveur calcule automatiquement la distance relative des têtes de note avec les portées.

```
\layout {
  \context {
    \PianoStaff
    \consists #Span_stem_engraver
  }
}

{
  \new PianoStaff <<
    \new Staff {
      <b d'>4 r d'16\> e'8. g8 r\!
      e'8 f' g'4 e'2
    }
    \new Staff {
      \clef bass
      \voiceOne
      \autoBeamOff
      \crossStaff { <e g>4 e, g16 a8. c8} d
      \autoBeamOn
      g8 f g4 c2
    }
  >>
}
```



Guidons

Les guidons peuvent adopter différents styles.

```
\layout { ragged-right = ##t }

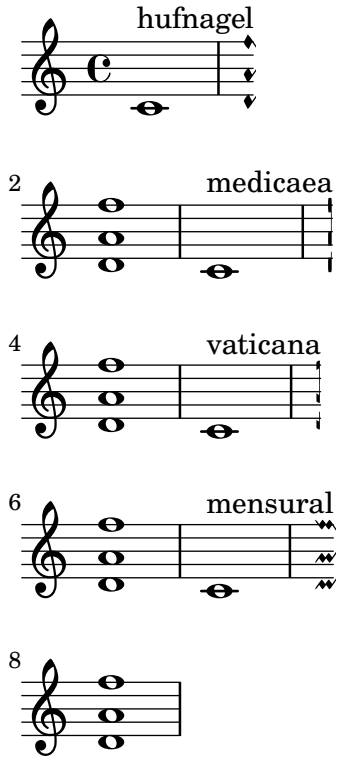
\new Staff \with { \consists "Custos_engraver" } \relative c' {
  \override Staff.Custos.neutral-position = #4

  \override Staff.Custos.style = #'hufnagel
  c1^"hufnagel" \break
  <d a' f'>1

  \override Staff.Custos.style = #'medicaea
  c1^"medicaea" \break
  <d a' f'>1

  \override Staff.Custos.style = #'vaticana
  c1^"vaticana" \break
  <d a' f'>1

  \override Staff.Custos.style = #'mensural
  c1^"mensural" \break
  <d a' f'>1
}
```



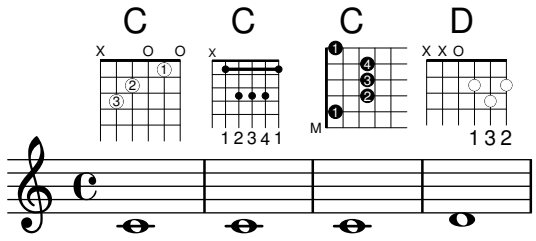
Personnalisation de diagrammes de fret

Les propriétés d'un diagramme de fret sont définies par les `fret-diagram-details`. En matière de diagramme de fret, les adaptations s'appliquent à l'objet `FretBoards.FretBoard`. Un `FretBoards` est comparable à un `Voice` : il s'agit d'un contexte du plus bas niveau, et il n'est donc pas primordial de l'instancier de manière explicite pour adapter ses propriétés.

```
\include "predefined-guitar-fretboards.ly"
\storePredefinedDiagram #default-fret-table \chordmode { c' }
                        #guitar-tuning
                        #"x;1-1-(;3-2;3-3;3-4;1-1-);"

% shorthand
oo = #(define-music-function
       (grob-path value)
       (list? scheme?)
       #{ \once \override $grob-path = #value #})

<<
  \new ChordNames {
    \chordmode { c1 | c | c | d }
  }
  \new FretBoards {
    % Set global properties of fret diagram
    \override FretBoards.FretBoard.size = #'1.2
    \override FretBoard.fret-diagram-details.finger-code = #'in-dot
    \override FretBoard.fret-diagram-details.dot-color = #'white
    \chordmode {
      c
      \oo FretBoard.size #'1.0
      \oo FretBoard.fret-diagram-details.barre-type #'straight
      \oo FretBoard.fret-diagram-details.dot-color #'black
      \oo FretBoard.fret-diagram-details.finger-code #'below-string
      c'
      \oo FretBoard.fret-diagram-details.barre-type #'none
      \oo FretBoard.fret-diagram-details.number-type #'arabic
      \oo FretBoard.fret-diagram-details.orientation #'landscape
      \oo FretBoard.fret-diagram-details.mute-string #"M"
      \oo FretBoard.fret-diagram-details.label-dir #LEFT
      \oo FretBoard.fret-diagram-details.dot-color #'black
      c'
      \oo FretBoard.fret-diagram-details.finger-code #'below-string
      \oo FretBoard.fret-diagram-details.dot-radius #0.35
      \oo FretBoard.fret-diagram-details.dot-position #0.5
      \oo FretBoard.fret-diagram-details.fret-count #3
      d
    }
  }
  \new Voice {
    c'1 | c' | c' | d'
  }
>>
```



Personnalisation des diagrammes de fret

Les propriétés d'un diagramme de fret sont modifiables grâce au `fret-diagram-details`. Lorsqu'ils sont générés sous forme de `\markup`, rien n'empêche de modifier les diagrammes en jouant sur les réglages de l'objet `Voice.TextScript` ou bien directement sur le *markup*.

```
<<
  \chords { c1 | c | c | d }

  \new Voice = "mel" {
    \textLengthOn
    % Set global properties of fret diagram
    \override TextScript.size = #'1.2
    \override TextScript.fret-diagram-details.finger-code = #'in-dot
    \override TextScript.fret-diagram-details.dot-color = #'white

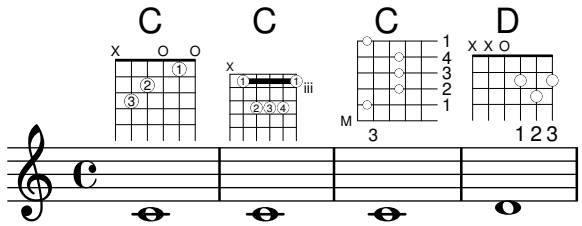
    %% C major for guitar, no barre, using defaults
       % terse style
    c'1^\markup { \fret-diagram-terse "x;3-3;2-2;o;1-1;o;" }

    %% C major for guitar, barred on third fret
       % verbose style
       % size 1.0
       % roman fret label, finger labels below string, straight barre
    c'1^\markup {
      % standard size
      \override #'(size . 1.0) {
        \override #'(fret-diagram-details . (
                     (number-type . roman-lower)
                     (finger-code . in-dot)
                     (barre-type . straight))) {
          \fret-diagram-verbose #'((mute 6)
                                   (place-fret 5 3 1)
                                   (place-fret 4 5 2)
                                   (place-fret 3 5 3)
                                   (place-fret 2 5 4)
                                   (place-fret 1 3 1)
                                   (barre 5 1 3))
        }
      }
    }

    %% C major for guitar, barred on third fret
       % verbose style
       % landscape orientation, arabic numbers, M for mute string
       % no barre, fret label down or left, small mute label font
    c'1^\markup {
      \override #'(fret-diagram-details . (
```

```
                    (finger-code . below-string)
                    (number-type . arabic)
                    (label-dir . -1)
                    (mute-string . "M")
                    (orientation . landscape)
                    (barre-type . none)
                    (xo-font-magnification . 0.4)
                    (xo-padding . 0.3))) {
        \fret-diagram-verbose #'((mute 6)
                                 (place-fret 5 3 1)
                                 (place-fret 4 5 2)
                                 (place-fret 3 5 3)
                                 (place-fret 2 5 4)
                                 (place-fret 1 3 1)
                                 (barre 5 1 3))
      }
    }

    %% simple D chord
       % terse style
       % larger dots, centered dots, fewer frets
       % label below string
    d'1^\markup {
      \override #'(fret-diagram-details . (
                    (finger-code . below-string)
                    (dot-radius . 0.35)
                    (dot-position . 0.5)
                    (fret-count . 3))) {
        \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
      }
    }
  }
>>
```
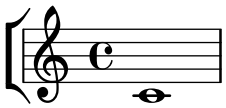


Indicateur de regroupement et portée unique
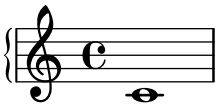
Lorsque, dans des regroupements de type `ChoirStaff` ou `StaffGroup`, une seule portée est active, aucune indication n'est donnée en début de ligne. Affecter à la propriété `collapse-height` un nombre de lignes inférieur à celui de la portée permet de modifier ce comportement par défaut.

Notez bien que dans le cas des `PianoStaff` et `GrandStaff`, pour lesquels le délimiteur de système est une accolade et non un crochet, il ne s'agit pas de la même propriété – voir le deuxième système de l'exemple.

```
\score {
```

```
  \new StaffGroup <<
    % Must be lower than the actual number of staff lines
    \override StaffGroup.SystemStartBracket.collapse-height = #4
    \override Score.SystemStartBar.collapse-height = #4
    \new Staff {
      c'1
    }
  >>
}
\score {
  \new PianoStaff <<
    \override PianoStaff.SystemStartBrace.collapse-height = #4
    \override Score.SystemStartBar.collapse-height = #4
    \new Staff {
      c'1
    }
  >>
}
```





Affichage de la généalogie d'un objet

Lorsque l'on manipule des rappels d'objet (*grob callbacks*), il peut être intéressant d'en maîtriser les ascendants. La plupart des objets graphiques ont des parents, lesquels auront une influence sur le positionnement de l'objet en question. Ainsi, les parents des côtés X et Y influenceront respectivement les positions horizontale et verticale de l'objet. De plus, chacun des parents peut avoir ses propres parents.

Certains aspects de la lignée d'un objet peuvent toutefois porter à confusion :

– Les types de parents d'un *grob* peuvent dépendre du contexte.

– Dans le cas de certains *grobs*, les parents X et Y peuvent être le même.

– Un « ascendant » particulier peut dépendre d'un *grob* de différentes manières.

– Le concept de « générations » est trompeur.

Par exemple, l'objet `System` peut, vis à vis d'un objet `VerticalAlignment`, être à la fois parent (par son côté Y) et grand parent (par deux fois du côté X).

La macro ci-dessous affiche à l'écran une représentation textuelle de l'ascendance d'un *grob*.

Elle se lance ainsi :

```
{ \once \override NoteHead.before-line-breaking = #display-ancestry c }
```

et génère la sortie suivante :

```
NoteHead
X,Y: NoteColumn
     X: PaperColumn
        X,Y: System
```

```
     Y: VerticalAxisGroup
        X: NonMusicalPaperColumn
           X,Y: System
        Y: VerticalAlignment
           X: NonMusicalPaperColumn
              X,Y: System
           Y: System

%% http://lsr.di.unimi.it/LSR/Item?id=622
%% see also http://www.lilypond.org/doc/v2.18/Documentation/snippets/tweaks-and-overrides#tw
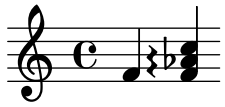
%% Remark:
%% grob::name is in the source since 2.19.x could be deleted during next LSR-upgrade
#(define (grob::name grob)
  (assq-ref (ly:grob-property grob 'meta) 'name))

#(define (get-ancestry grob)
  (if (not (null? (ly:grob-parent grob X)))
      (list (grob::name grob)
            (get-ancestry (ly:grob-parent grob X))
            (get-ancestry (ly:grob-parent grob Y)))
      (grob::name grob)))

#(define (format-ancestry lst padding)
   (string-append
    (symbol->string (car lst))
    "\n"
    (let ((X-ancestry
           (if (list? (cadr lst))
               (format-ancestry (cadr lst) (+ padding 3))
               (symbol->string (cadr lst))))
          (Y-ancestry
           (if (list? (caddr lst))
               (format-ancestry (caddr lst) (+ padding 3))
               (symbol->string (caddr lst)))))
      (if (equal? X-ancestry Y-ancestry)
          (string-append
           (format #f "~&")
           (make-string padding #\space)
           "X,Y: "
           (if (list? (cadr lst))
               (format-ancestry (cadr lst) (+ padding 5))
               (symbol->string (cadr lst))))
          (string-append
           (format #f "~&")
           (make-string padding #\space)
           "X: " X-ancestry
           "\n"
           (make-string padding #\space)
           "Y: " Y-ancestry
           (format #f "~&"))))
    (format #f "~&")))
```

```
#(define (display-ancestry grob)
   (format (current-error-port)
      "~3&~a~2%~a~&"
      (make-string 36 #\-)
      (if (ly:grob? grob)
          (format-ancestry (get-ancestry grob) 0)
          (format #f "~a is not a grob" grob))))
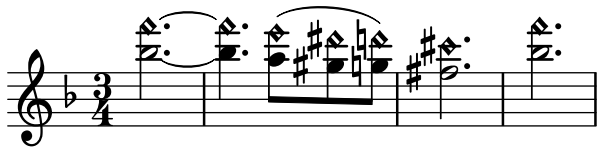
\relative c' {
  \once \override NoteHead.before-line-breaking = #display-ancestry
  f4
  \once \override Accidental.before-line-breaking = #display-ancestry
  \once \override Arpeggio.before-line-breaking = #display-ancestry
  <f as c>4\arpeggio
}
```



Harmoniques pointées

Les notes harmoniques artificielles, obtenues grâce à `\harmonic`, ne sont pas pointées. Ce comportement peut être modifié en activant la propriété de contexte `harmonicDots`.

```
\relative c''' {
  \time 3/4
  \key f \major
  \set harmonicDots = ##t
  <bes f'\harmonic>2. ~
  <bes f'\harmonic>4. <a e'\harmonic>8( <gis dis'\harmonic> <g d'\harmonic>)
  <fis cis'\harmonic>2.
  <bes f'\harmonic>2.
}
```



Encadrement d'objets

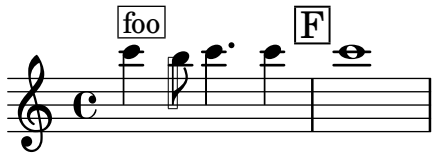
La fonction `print` peut se modifier pour obtenir l'encadrement de n'importe quel objet.

```
\relative c'' {
  \override TextScript.stencil =
    #(make-stencil-boxer 0.1 0.3 ly:text-interface::print)
  c'4^"foo"

  \override Stem.stencil =
    #(make-stencil-boxer 0.05 0.25 ly:stem::print)
  \override Score.RehearsalMark.stencil =
    #(make-stencil-boxer 0.15 0.3 ly:text-interface::print)
```

```
  b8

  \revert Stem.stencil
  \revert Flag.stencil
  c4. c4
  \mark "F"
  c1
}
```

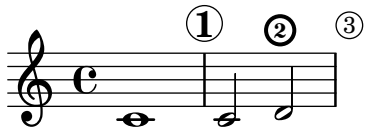


Encercler divers objets

La commande de *markup* `\circle` permet de dessiner un cercle autour de différents objets comme des indications de doigté. D'autres objets nécessitent de faire appel à des techniques spécifiques. Cet exemple illustre deux manières de procéder : pour les repères, et pour les numéros de mesure.

```
\relative c' {
  c1
  \set Score.markFormatter =
    #(lambda (mark context)
              (make-circle-markup (format-mark-numbers mark context)))
  \mark \default

  c2 d^\markup {
    \override #'(thickness . 3) {
      \circle \finger 2
    }
  }
  \override Score.BarNumber.break-visibility = #all-visible
  \override Score.BarNumber.stencil =
    #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
}
```



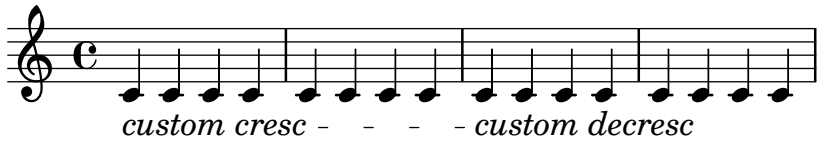
Personnalisation des extenseurs de nuance postfix

Il s'agit de fonctions postfix pour personnaliser l'extension des crescendos textuels. L'extension devrait débuter sur la première note de la mesure. Il faut utiliser `-\mycresc` – comme une articulation – sous peine que le départ de l'extension n'apparaisse qu'à la note suivante.

```
% Two functions for (de)crescendo spanners where you can explicitly
% give the spanner text.
mycresc =
#(define-music-function (mymarkup) (markup?)
   (make-music 'CrescendoEvent
```

```
               'span-direction START
               'span-type 'text
               'span-text mymarkup))
mydecresc =
#(define-music-function (mymarkup) (markup?)
   (make-music 'DecrescendoEvent
               'span-direction START
               'span-type 'text
               'span-text mymarkup))

\relative c' {
  c4-\mycresc "custom cresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4-\mydecresc "custom decresc" c4 c4 c4 |
  c4 c4\! c4 c4
}
```
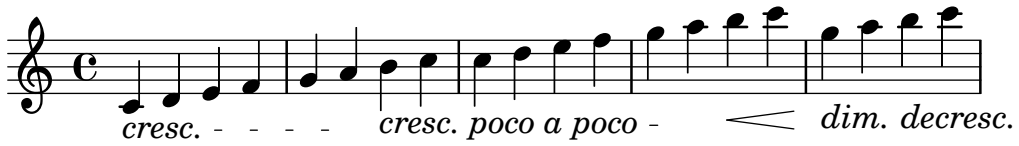


Extensions de nuance postfix

Les lignes d'extension des commandes `\cresc`, `\dim` et `\decresc` peuvent désormais être personnalisées facilement sous forme d'opérateurs postfix. Soufflets et (de)crescendos peuvent cohabiter. `\<` et `\>` produiront par défaut des soufflets, alors que `\cresc`, etc. produiront une indication textuelle avec extension.

```
% Some sample text dynamic spanners, to be used as postfix operators
crpoco =
#(make-music 'CrescendoEvent
             'span-direction START
             'span-type 'text
             'span-text "cresc. poco a poco")

\relative c' {
  c4\cresc d4 e4 f4 |
  g4 a4\! b4\crpoco c4 |
  c4 d4 e4 f4 |
  g4 a4\! b4\< c4 |
  g4\dim a4 b4\decresc c4\!
}
```

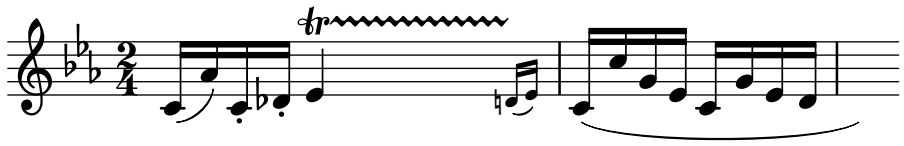


Allongement d'une marque de trille (`TrillSpanner`)

La propriété `minimum-length` d'une extension de trille – objet `TrillSpanner` – n'est effective qu'après un appel explicite à la procédure `set-spacing-rods`.

Pour ce faire, la propriété `set-spacing-rods` doit être définie à `ly:spanner::set-spacing-rods`.

```
\relative c' {
\key c\minor
  \time 2/4
  c16( as') c,-. des-.
  \once\override TrillSpanner.minimum-length = #15
  \once\override TrillSpanner.springs-and-rods = #ly:spanner::set-spacing-rods
  \afterGrace es4
  \startTrillSpan { d16[( \stopTrillSpan es)] }
  c( c' g es c g' es d
  \hideNotes
  c8)
}
```



Rappel du glissando à l'occasion d'une alternative

Un glissando qui se prolonge sur plusieurs sections `\alternative` peut se rappeler à l'aide d'une note d'ornement supplémentaire et masquée, à laquelle sera attaché le départ du glissando, ce dans chaque bloc `\alternative`. Cette note d'ornement devrait avoir la même hauteur que la note où commençait le glissando originel. Ceci est géré par une fonction musicale qui prendra en argument la hauteur de la note d'ornement.

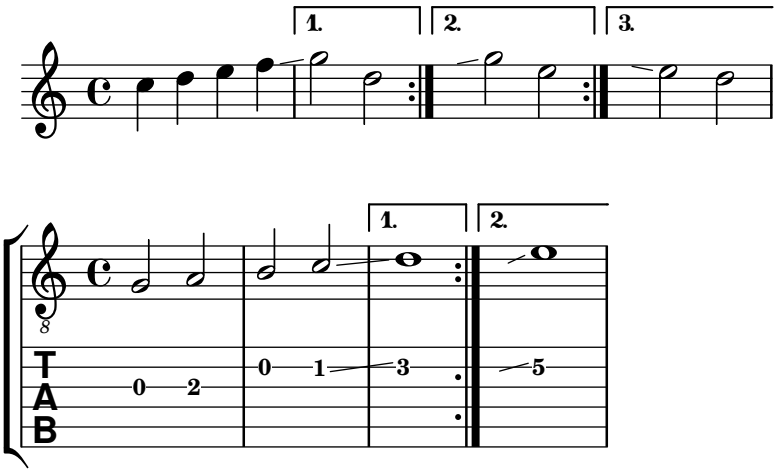
Dans le cadre d'un musique polyphonique, il ne faudra pas oublier d'ajouter une note d'ornement dans toutes les autres voix afin de préserver la synchronisation.

```
repeatGliss = #(define-music-function (grace)
  (ly:pitch?)
  #{
    % the next two lines ensure the glissando is long enough
    % to be visible
    \once \override Glissando.springs-and-rods
      = #ly:spanner::set-spacing-rods
    \once \override Glissando.minimum-length = #3.5
    \once \hideNotes
    \grace $grace \glissando
  #})

\score {
  \relative c'' {
    \repeat volta 3 { c4 d e f\glissando }
    \alternative {
      { g2 d }
      { \repeatGliss f g2 e }
      { \repeatGliss f e2 d }
    }
  }
}
```

```
music =  \relative c' {
  \voiceOne
  \repeat volta 2 {
    g a b c\glissando
  }
  \alternative {
    { d1 }
    { \repeatGliss c \once \omit StringNumber e1\2 }
  }
}
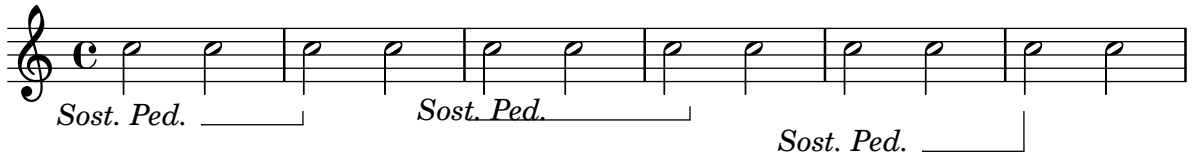
\score {
  \new StaffGroup <<
    \new Staff <<
      \new Voice { \clef "G_8" \music }
    >>
    \new TabStaff  <<
      \new TabVoice { \clef "moderntab" \music }
    >>
  >>
}
```



Affinage des indications de pédale

Les crochets d'indication de pédale peuvent se modifier de différentes manières.

```
\paper { ragged-right = ##f }
\relative c'' {
  c2\sostenutoOn c
  c2\sostenutoOff c
  \once \override Staff.PianoPedalBracket.shorten-pair = #'(-7 . -2)
  c2\sostenutoOn c
  c2\sostenutoOff c
  \once \override Staff.PianoPedalBracket.edge-height = #'(0 . 3)
  c2\sostenutoOn c
  c2\sostenutoOff c
}
```



Liaison de prolongation aplatie

La fonction ici présentée prend en argument le `Tie.stencil` par défaut et en recalcule le résultat d'après les étendues de cette valeur par défaut.

Des ajustements plus poussés sont possibles, en apportant des dérogations à `Tie.details.height-limit` ou à l'aide de la fonction `\shape`. La définition personnalisée peut aussi se modifier à la volée.

```
%% http://lsr.di.unimi.it/LSR/Item?id=1031

#(define ((flared-tie coords) grob)

  (define (pair-to-list pair)
     (list (car pair) (cdr pair)))

  (define (normalize-coords goods x y dir)
    (map
      (lambda (coord)
        ;(coord-scale coord (cons x (* y dir)))
        (cons (* x (car coord)) (* y dir (cdr coord))))
      goods))

  (define (my-c-p-s points thick)
    (make-connected-path-stencil
      points
      thick
      1.0
      1.0
      #f
      #f))

  ;; outer let to trigger suicide
  (let ((sten (ly:tie::print grob)))
    (if (grob::is-live? grob)
        (let* ((layout (ly:grob-layout grob))
               (line-thickness (ly:output-def-lookup layout 'line-thickness))
               (thickness (ly:grob-property grob 'thickness 0.1))
               (used-thick (* line-thickness thickness))
               (dir (ly:grob-property grob 'direction))
               (xex (ly:stencil-extent sten X))
               (yex (ly:stencil-extent sten Y))
               (lenx (interval-length xex))
               (leny (interval-length yex))
               (xtrans (car xex))
               (ytrans (if (> dir 0)(car yex) (cdr yex)))
               (uplist
                 (map pair-to-list
                      (normalize-coords coords lenx (* leny 2) dir))))

   (ly:stencil-translate
```

```
      (my-c-p-s uplist used-thick)
    (cons xtrans ytrans)))
  '())))

#(define flare-tie
  (flared-tie '((0 . 0)(0.1 . 0.2) (0.9 . 0.2) (1.0 . 0.0))))

\layout {
  \context {
    \Voice
    \override Tie.stencil = #flare-tie
  }
}

\paper { ragged-right = ##f }

\relative c' {
  a4~a
  \override Tie.height-limit = 4
  a'4~a
  a'4~a
  <a,, c e a c e a c e>~ q

  \break

  a'4~a
  \once \override Tie.details.height-limit = 14
  a4~a

  \break

  a4~a
  \once \override Tie.details.height-limit = 0.5
  a4~a

  \break
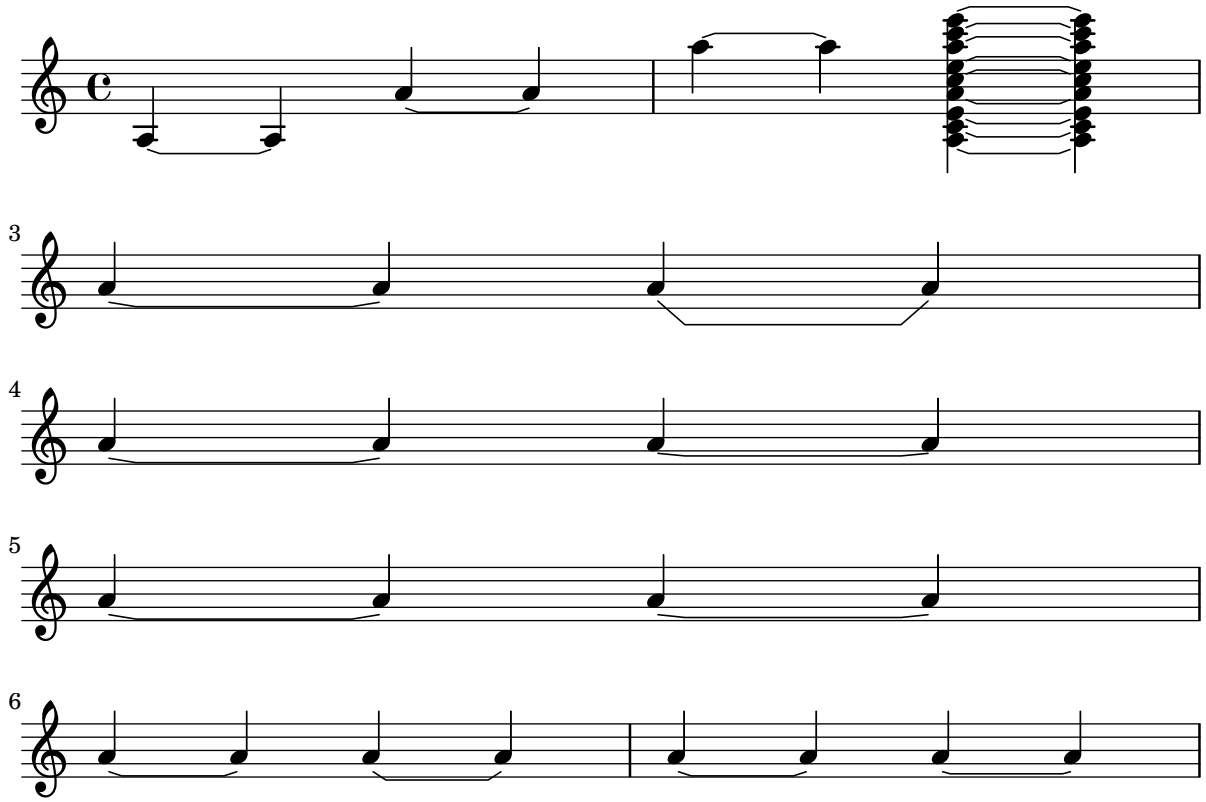
  a4~a
  \shape #'((0 . 0) (0 . 0.4) (0 . 0.4) (0 . 0)) Tie
  a4~a

  \break

  a4~a
  \once \override Tie.stencil =
    #(flared-tie '((0 . 0)(0.1 . 0.4) (0.9 . 0.4) (1.0 . 0.0)))
  a4~a

  a4~a
  \once \override Tie.stencil =
    #(flared-tie '((0 . 0)(0.06 . 0.1) (0.94 . 0.1) (1.0 . 0.0)))
  a4~a
```
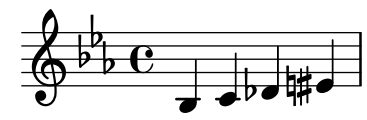
```
}
```



Impression forcée d'un bécarre avant une accidentelle

L'exemple suivant indique comment forcer l'impression d'un bécarre avant un changement d'altération.

```
\relative c' {
  \key es \major
  bes c des
  \tweak Accidental.restore-first ##t
  eis
}
```

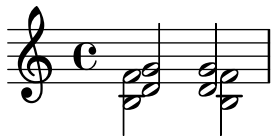


Décalage horizontal forcé

Quand LilyPond est dépassé, la propriété `force-hshift` de l'objet `NoteColumn` et des silences à hauteur déterminée peuvent s'avérer utiles pour dicter au programme les choix de placement. On travaille ici en espace de portée.

```
\relative c' <<
  {
    <d g>2 <d g>
  }
  \\
  {
```

```
    <b f'>2
    \once \override NoteColumn.force-hshift = #1.7
    <b f'>2
  }
>>
```



Construction et développement de diagrammes de fret

Voici différentes manières d'obtenir et de personnaliser des diagrammes de fret :

```
<<
  \chords {
    a2 a
    \repeat unfold 3 {
      c c c d d
    }
  }

  \new Voice = "mel" {
    \textLengthOn
    % Set global properties of fret diagram
    \override TextScript.size = #1.2
    \override TextScript.fret-diagram-details.finger-code = #'below-string
    \override TextScript.fret-diagram-details.dot-color = #'black

    %% A chord for ukulele
    a'2^\markup {
      \override #'(fret-diagram-details . (
                   (string-count . 4)
                   (dot-color . white)
                   (finger-code . in-dot))) {
        \fret-diagram "4-2-2;3-1-1;2-o;1-o;"
      }
    }

    %% A chord for ukulele, with formatting defined in definition string
    %  1.2 * size, 4 strings, 4 frets, fingerings below string
    %  dot radius .35 of fret spacing, dot position 0.55 of fret spacing
    a'2^\markup {
      \override #'(fret-diagram-details . (
                   (dot-color . white)
                   (open-string . "o"))) {
        \fret-diagram "s:1.2;w:4;h:3;f:2;d:0.35;p:0.55;4-2-2;3-1-1;2-o;1-o;"
      }
    }

      %% These chords will be in normal orientation
```

```
%% C major for guitar, barred on third fret
%  verbose style
%  roman fret label, finger labels below string, straight barre
c'2^\markup {
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                 (number-type . roman-lower)
                 (finger-code . below-string)
                 (barre-type . straight))) {
      \fret-diagram-verbose #'((mute 6)
                               (place-fret 5 3 1)
                               (place-fret 4 5 2)
                               (place-fret 3 5 3)
                               (place-fret 2 5 4)
                               (place-fret 1 3 1)
                               (barre 5 1 3))
    }
  }
}

%% C major for guitar, barred on third fret
%% Double barre used to test barre function
%  verbose style
c'2^\markup {
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                 (number-type . arabic)
                 (dot-label-font-mag . 0.9)
                 (finger-code . in-dot)
                 (fret-label-font-mag . 0.6)
                 (fret-label-vertical-offset . 0)
                 (label-dir . -1)
                 (mute-string . "M")
                 (xo-font-magnification . 0.4)
                 (xo-padding . 0.3))) {
      \fret-diagram-verbose #'((mute 6)
                               (place-fret 5 3 1)
                               (place-fret 4 5 2)
                               (place-fret 3 5 3)
                               (place-fret 2 5 4)
                               (place-fret 1 3 1)
                               (barre 4 2 5)
                               (barre 5 1 3))
    }
  }
}

%% C major for guitar, with capo on third fret
%  verbose style
c'2^\markup {
```

```
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                 (number-type . roman-upper)
                 (dot-label-font-mag . 0.9)
                 (finger-code . none)
                 (fret-label-vertical-offset . 0.5)
                 (xo-font-magnification . 0.4)
                 (xo-padding . 0.3))) {
      \fret-diagram-verbose #'((mute 6)
                               (capo 3)
                               (open 5)
                               (place-fret 4 5 1)
                               (place-fret 3 5 2)
                               (place-fret 2 5 3)
                               (open 1))
    }
  }
}

%% simple D chord
d'2^\markup {
  \override #'(fret-diagram-details . (
               (finger-code . below-string)
               (dot-radius . 0.35)
               (string-thickness-factor . 0.3)
               (dot-position . 0.5)
               (fret-count . 3))) {
    \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
  }
}

%% simple D chord, large top fret thickness
d'2^\markup  {
  \override #'(fret-diagram-details . (
               (finger-code . below-string)
               (dot-radius . 0.35)
               (dot-position . 0.5)
               (top-fret-thickness . 7)
               (fret-count . 3))) {
    \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
  }
}

  % These chords will be in landscape orientation
\override TextScript.fret-diagram-details.orientation = #'landscape

%% C major for guitar, barred on third fret
%  verbose style
%  roman fret label, finger labels below string, straight barre
c'2^\markup {
  % 110% of default size
```

```
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                 (number-type . roman-lower)
                 (finger-code . below-string)
                 (barre-type . straight))) {
      \fret-diagram-verbose #'((mute 6)
                               (place-fret 5 3 1)
                               (place-fret 4 5 2)
                               (place-fret 3 5 3)
                               (place-fret 2 5 4)
                               (place-fret 1 3 1)
                               (barre 5 1 3))
    }
  }
}

%% C major for guitar, barred on third fret
%% Double barre used to test barre function
%  verbose style
c'2^\markup {
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                 (number-type . arabic)
                 (dot-label-font-mag . 0.9)
                 (finger-code . in-dot)
                 (fret-label-font-mag . 0.6)
                 (fret-label-vertical-offset . 0)
                 (label-dir . -1)
                 (mute-string . "M")
                 (xo-font-magnification . 0.4)
                 (xo-padding . 0.3))) {
      \fret-diagram-verbose #'((mute 6)
                               (place-fret 5 3 1)
                               (place-fret 4 5 2)
                               (place-fret 3 5 3)
                               (place-fret 2 5 4)
                               (place-fret 1 3 1)
                               (barre 4 2 5)
                               (barre 5 1 3))
    }
  }
}

%% C major for guitar, with capo on third fret
%  verbose style
c'2^\markup {
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                 (number-type . roman-upper)
                 (dot-label-font-mag . 0.9)
```

```
                (finger-code . none)
                (fret-label-vertical-offset . 0.5)
                (xo-font-magnification . 0.4)
                (xo-padding . 0.3))) {
      \fret-diagram-verbose #'((mute 6)
                               (capo 3)
                               (open 5)
                               (place-fret 4 5 1)
                               (place-fret 3 5 2)
                               (place-fret 2 5 3)
                               (open 1))
    }
  }
}

%% simple D chord
d'2^\markup {
  \override #'(fret-diagram-details . (
                (finger-code . below-string)
                (dot-radius . 0.35)
                (dot-position . 0.5)
                (fret-count . 3))) {
    \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
  }
}

%% simple D chord, large top fret thickness
d'2^\markup {
  \override #'(fret-diagram-details . (
                (finger-code . below-string)
                (dot-radius . 0.35)
                (dot-position . 0.5)
                (top-fret-thickness . 7)
                (fret-count . 3))) {
    \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
  }
}

  % These chords will be in opposing-landscape orientation
\override TextScript.fret-diagram-details.orientation = #'opposing-landscape

%% C major for guitar, barred on third fret
%  verbose style
%  roman fret label, finger labels below string, straight barre
c'2^\markup {
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                (number-type . roman-lower)
                (finger-code . below-string)
                (barre-type . straight))) {
      \fret-diagram-verbose #'((mute 6)
```

```
                                (place-fret 5 3 1)
                                (place-fret 4 5 2)
                                (place-fret 3 5 3)
                                (place-fret 2 5 4)
                                (place-fret 1 3 1)
                                (barre 5 1 3))
    }
  }
}

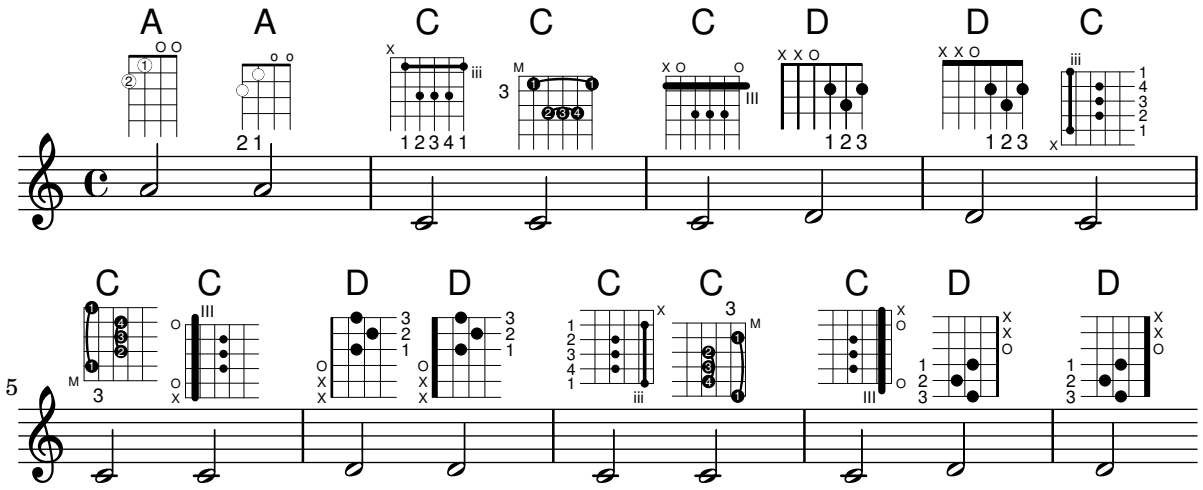
%% C major for guitar, barred on third fret
%% Double barre used to test barre function
%  verbose style
c'2^\markup {
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                (number-type . arabic)
                (dot-label-font-mag . 0.9)
                (finger-code . in-dot)
                (fret-label-font-mag . 0.6)
                (fret-label-vertical-offset . 0)
                (label-dir . -1)
                (mute-string . "M")
                (xo-font-magnification . 0.4)
                (xo-padding . 0.3))) {
      \fret-diagram-verbose #'((mute 6)
                               (place-fret 5 3 1)
                               (place-fret 4 5 2)
                               (place-fret 3 5 3)
                               (place-fret 2 5 4)
                               (place-fret 1 3 1)
                               (barre 4 2 5)
                               (barre 5 1 3))
    }
  }
}

%% C major for guitar, with capo on third fret
%  verbose style
c'2^\markup {
  % 110% of default size
  \override #'(size . 1.1) {
    \override #'(fret-diagram-details . (
                (number-type . roman-upper)
                (dot-label-font-mag . 0.9)
                (finger-code . none)
                (fret-label-vertical-offset . 0.5)
                (xo-font-magnification . 0.4)
                (xo-padding . 0.3))) {
      \fret-diagram-verbose #'((mute 6)
                               (capo 3)
```

```
                                    (open 5)
                                    (place-fret 4 5 1)
                                    (place-fret 3 5 2)
                                    (place-fret 2 5 3)
                                    (open 1))
        }
      }
    }

    %% simple D chord
    d'2^\markup {
      \override #'(fret-diagram-details . (
                   (finger-code . below-string)
                   (dot-radius . 0.35)
                   (dot-position . 0.5)
                   (fret-count . 3))) {
        \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
      }
    }

    %% simple D chord, large top fret thickness
    d'2^\markup {
      \override #'(fret-diagram-details . (
                   (finger-code . below-string)
                   (dot-radius . 0.35)
                   (dot-position . 0.5)
                   (top-fret-thickness . 7)
                   (fret-count . 3))) {
        \fret-diagram-terse "x;x;o;2-1;3-2;2-3;"
      }
    }
  }
>>
```



Génération de crochets personnalisés

Une fonction Scheme personnalisée permet de redéfinir la propriété `stencil` de l'objet `Flag`, de sorte à modifier le glyphe utilisé pour les crochets de croche.

```
#(define-public (weight-flag grob)
   (let* ((stem-grob (ly:grob-parent grob X))
          (log (- (ly:grob-property stem-grob 'duration-log) 2))
          (is-up? (eqv? (ly:grob-property stem-grob 'direction) UP))
          (yext (if is-up? (cons (* log -0.8) 0) (cons 0 (* log 0.8))))
          (flag-stencil (make-filled-box-stencil '(-0.4 . 0.4) yext))
          (stroke-style (ly:grob-property grob 'stroke-style))
          (stroke-stencil (if (equal? stroke-style "grace")
                              (make-line-stencil 0.2 -0.9 -0.4 0.9 -0.4)
                              empty-stencil)))
     (ly:stencil-add flag-stencil stroke-stencil)))


% Create a flag stencil by looking up the glyph from the font
#(define (inverted-flag grob)
   (let* ((stem-grob (ly:grob-parent grob X))
          (dir (if (eqv? (ly:grob-property stem-grob 'direction) UP) "d" "u"))
          (flag (retrieve-glyph-flag "" dir "" grob))
          (line-thickness (ly:staff-symbol-line-thickness grob))
          (stem-thickness (ly:grob-property stem-grob 'thickness))
          (stem-width (* line-thickness stem-thickness))
          (stroke-style (ly:grob-property grob 'stroke-style))
          (stencil (if (null? stroke-style)
                       flag
                       (add-stroke-glyph flag stem-grob dir stroke-style "")))
          (rotated-flag (ly:stencil-rotate-absolute stencil 180 0 0)))
     (ly:stencil-translate rotated-flag (cons (- (/ stem-width 2)) 0))))
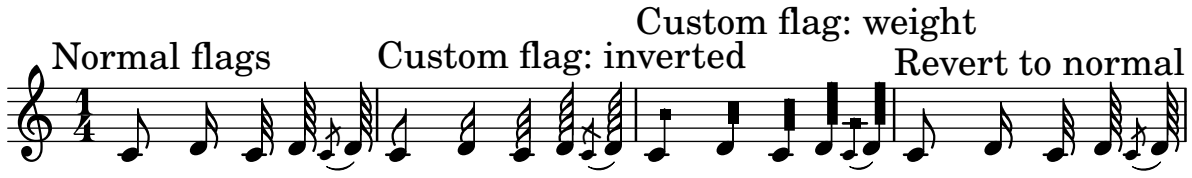
snippetexamplenotes =
{
  \autoBeamOff c'8 d'16 c'32 d'64 \acciaccatura {c'8} d'64
}

{
  \override Score.RehearsalMark.self-alignment-X = #LEFT
  \time 1/4
  \mark "Normal flags"
  \snippetexamplenotes

  \mark "Custom flag: inverted"
  \override Flag.stencil = #inverted-flag
  \snippetexamplenotes

  \mark "Custom flag: weight"
  \override Flag.stencil = #weight-flag
  \snippetexamplenotes
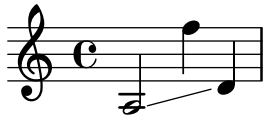
  \mark "Revert to normal"
  \revert Flag.stencil
  \snippetexamplenotes
}
```



Glissando par dessus un objet graphique

Un glissando peut sauter un objet `NoteColumn`.

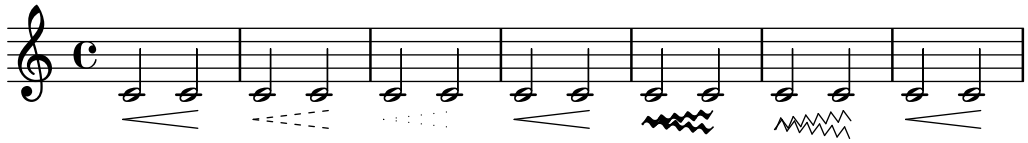
```
\relative c' {
  a2 \glissando
  \once \override NoteColumn.glissando-skip = ##t
  f''4 d,
}
```



Stylisation des lignes de soufflet

Les lignes d'un soufflet peuvent adopter tous les styles permis par la `line-interface` : discontinu, pointillé, continu, ondulé ou en zigzag.

```
\relative c' {
  c2\< c\!
  \override Hairpin.style = #'dashed-line
  c2\< c\!
  \override Hairpin.style = #'dotted-line
  c2\< c\!
  \override Hairpin.style = #'line
  c2\< c\!
  \override Hairpin.style = #'trill
  c2\< c\!
  \override Hairpin.style = #'zigzag
  c2\< c\!
  \revert Hairpin.style
  c2\< c\!
}
```



Alignement de nuances personnalisées comme « sempre pp », « più f » ou « subito p »

Certaines indications de nuance requièrent un complément textuel, comme « sempre pp ». Dans la mesure où les nuances sont habituellement centrées sous la note, le *pp* se trouvera repoussé loin après la note à laquelle il s'applique.

Différentes approches permettent de correctement aligner horizontalement ce « sempre pp » :

– Un simple `\once \override DynamicText.X-offset = #-9.2` avant la note considérée de telle sorte que la nuance soit décalée manuellement à la bonne place. Inconvénient : il faut le faire manuellement à chaque fois qu'intervient ce *markup*.

– L'intégration d'un rembourrage (`#:hspace 7.1`) à la définition de cette nuance personnalisée afin qu'une fois centrée par LilyPond elle soit correctement alignée. Inconvénient : le rembourrage occupera exactement cet espace et ne permettra à aucun autre *markup* ou nuance d'apparaître à cet endroit.

– L'application d'un décalage au script de nuance `\once \override ... X-offset = ...`. Inconvénient : là aussi il faut le faire à chaque fois.

– L'attribution arbitraire d'une dimension à 0 du texte additionnel à l'aide d'un `#:with-dimensions '(0 . 0) '(0 . 0)`. Inconvénient : LilyPond considère que « sempre » n'occupe pas d'espace, et donc pourra mettre à cet endroit d'autres éléments, ce qui pourrait générer des collisions qui ne seront pas détectées par les routines *ad hoc*. D'autre part, il semble persister un espacement, ce qui laisse l'impression d'un alignement différent en l'absence de texte additionnel.

– L'ajout, explicite, du décalage directement dans la fonction Scheme du script de nuance.

– La définition d'un alignement explicite au sein du script de nuance. Ceci ne sera suivi d'effet, par défaut, qu'en jouant sur `X-offset`. Inconvénient : il faut définir `DynamicText.X-offset`, ce qui s'appliquera à toutes les nuances textuelles. Par ailleurs, l'alignement sera réalisé sur le bord droit du texte additionnel, non sur le milieu de *pp*.

```
\paper {
  ragged-right = ##f
  indent = 2.5\cm
}

% Solution 1: Using a simple markup with a particular halign value
% Drawback: It's a markup, not a dynamic command, so \dynamicDown
%           etc. will have no effect
semppMarkup = \markup { \halign #1.4 \italic "sempre" \dynamic "pp" }

% Solution 2: Using a dynamic script & shifting with
%             \once \override ...X-offset = ..
% Drawback: \once \override needed for every invocation
semppK =
#(make-dynamic-script
  (markup #:line
          (#:normal-text
           #:italic "sempre"
           #:dynamic "pp")))

% Solution 3: Padding the dynamic script so the center-alignment
%             puts it at the correct position
% Drawback: the padding really reserves the space, nothing else can be there
semppT =
#(make-dynamic-script
  (markup #:line
          (#:normal-text
           #:italic "sempre"
           #:dynamic "pp"
           #:hspace 7.1)))

% Solution 4: Dynamic, setting the dimensions of the additional text to 0
% Drawback: To lilypond "sempre" has no extent, so it might put
%           other stuff there => collisions
```

```
% Drawback: Also, there seems to be some spacing, so it's not exactly the
%           same alignment as without the additional text
semppM =
#(make-dynamic-script
  (markup #:line
          (#:with-dimensions '(0 . 0) '(0 . 0)
                             #:right-align
                             #:normal-text
                             #:italic "sempre"
                             #:dynamic "pp")))
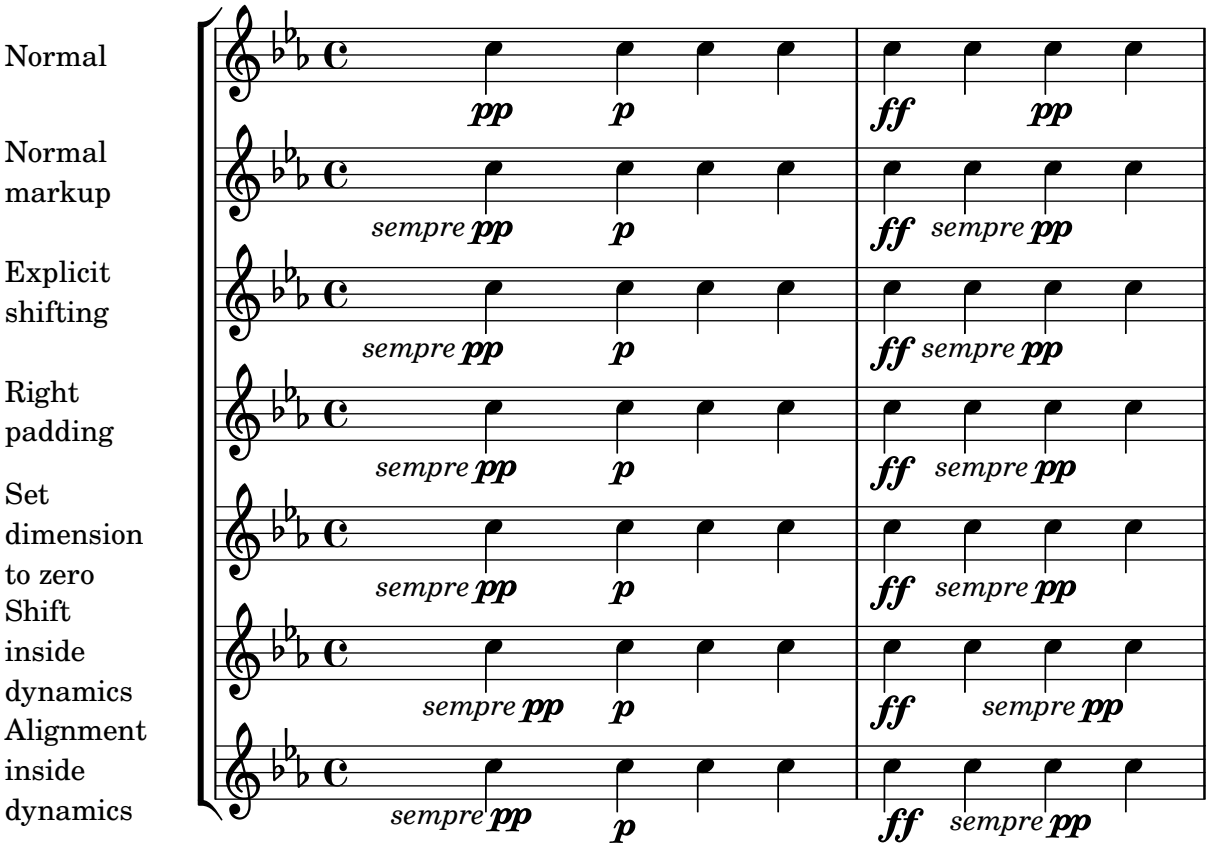
% Solution 5: Dynamic with explicit shifting inside the scheme function
semppG =
#(make-dynamic-script
  (markup #:hspace 0
          #:translate '(-18.85 . 0)
          #:line (#:normal-text
                  #:italic "sempre"
                  #:dynamic "pp")))

% Solution 6: Dynamic with explicit alignment. This has only effect
%             if one sets X-offset!
% Drawback: One needs to set DynamicText.X-offset!
% Drawback: Aligned at the right edge of the additional text,
%           not at the center of pp
semppMII =
#(make-dynamic-script
  (markup #:line (#:right-align
                  #:normal-text
                  #:italic "sempre"
                  #:dynamic "pp")))

\new StaffGroup <<
  \new Staff = "s" \with { instrumentName = \markup \column { Normal } }
  <<
    \relative c'' {
      \key es \major
      c4\pp c\p c c | c\ff c c\pp c
    }
  >>
  \new Staff = "sMarkup" \with {
    instrumentName = \markup \column { Normal markup }
  }
  <<
    \relative c'' {
      \key es \major
      c4-\semppMarkup c\p c c | c\ff c c-\semppMarkup c
    }
  >>
  \new Staff = "sK" \with {
    instrumentName = \markup \column { Explicit shifting }
  }
```

```
  <<
    \relative c'' {
      \key es \major
      \once \override DynamicText.X-offset = #-9.2
      c4\semppK c\p c c
      c4\ff c
      \once \override DynamicText.X-offset = #-9.2
      c4\semppK c
    }
  >>
  \new Staff = "sT" \with {
    instrumentName = \markup \column { Right padding }
  }
  <<
    \relative c'' {
      \key es \major
      c4\semppT c\p c c | c\ff c c\semppT c
    }
  >>
  \new Staff = "sM" \with {
    instrumentName = \markup \column { Set dimension "to zero" }
  }
  <<
    \relative c'' {
      \key es \major
      c4\semppM c\p c c | c\ff c c\semppM c
    }
  >>
  \new Staff = "sG" \with {
    instrumentName = \markup \column { Shift inside dynamics}
  }
  <<
    \relative c'' {
      \key es \major
      c4\semppG c\p c c | c\ff c c\semppG c
    }
  >>
  \new Staff = "sMII" \with {
    instrumentName = \markup \column { Alignment inside dynamics }
  }
  <<
    \relative c'' {
      \key es \major
      % Setting to ##f (false) gives the same result
      \override DynamicText.X-offset = #0
      c4\semppMII c\p c c | c\ff c c\semppMII c
    }
  >>
>>

\layout { \override Staff.InstrumentName.self-alignment-X = #LEFT }
```



Repositionnement d'un diagramme de fret

Différents moyens permettent de repositionner un diagramme de fret pour éviter des collisions ou le placer entre deux notes :

– La modification des valeurs de `padding` ou de `extra-offset` comme pour le second diagramme ;

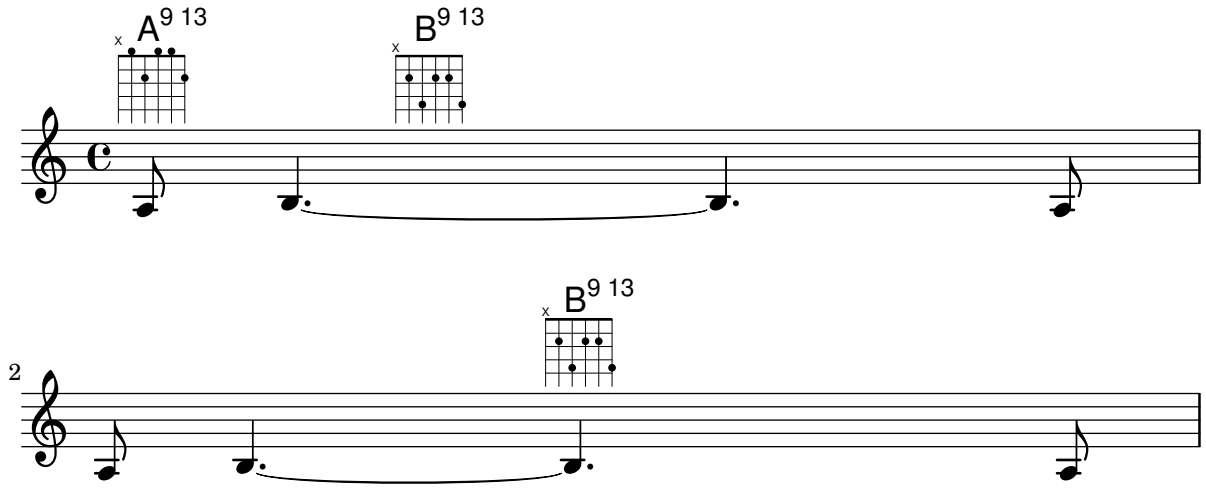
– L'adjonction d'une voix invisible dans laquelle les diagrammes sont attachés à des notes invisibles comme pour le troisième diagramme.

Lorsque le diagramme doit correspondre à une position rythmique dans la mesure, comme au troisième temps de la deuxième mesure, la seconde méthode est plus appropriée puisque le diagramme sera aligné sur le temps.

```
harmonies = \chordmode
{
  a8:13
% THE FOLLOWING IS THE COMMAND TO MOVE THE CHORD NAME
  \once \override ChordNames.ChordName.extra-offset = #'(10 . 0)
  b8:13 s2.
% THIS LINE IS THE SECOND METHOD
    s4 s4  b4:13
}

\score
{
  <<
    \new ChordNames \harmonies
    \new Staff
    {a8^\markup { \fret-diagram  "6-x;5-0;4-2;3-0;2-0;1-2;"  }
% THE FOLLOWING IS THE COMMAND TO MOVE THE FRET DIAGRAM
     \once \override TextScript.extra-offset = #'(10 . 0)
```

```
     b4.~^\markup { \fret-diagram  "6-x;5-2;4-4;3-2;2-2;1-4;"  } b4. a8\break
% HERE IS THE SECOND METHOD
     <<
       { a8 b4.~ b4. a8}
       { s4 s4 s4^\markup { \fret-diagram  "6-x;5-2;4-4;3-2;2-2;1-4;"  }
       }
     >>
   }
  >>
}
```



Impression d'un double repère sur une même barre (méthode 1)

Cette méthode permet d'imprimer deux marques-repères l'une au-dessus de l'autre. Le repère inférieur est décalé sous la portée, puis s'insère au-dessus de lui un espace pour positionner le repère supérieur en surplomb de la portée.

L'ajustement des valeurs de `extra-offset` et `baseline-skip` permet d'accroître ou réduire globalement l'espace entre le repère et la portée.

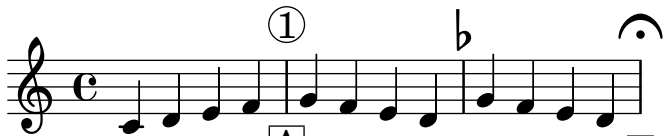
La capacité de pratiquement tout type de glyphe ou chaîne de caractères à faire l'objet d'un repère les rend capables de venir se centrer au-dessus ou en dessous d'une barre de mesure.

L'adjonction de la propriété `break-visibility`, comme dans cet autre cas (`\%22http://lsr.di.unimi.it/LSR/Item?id=1\%22`), permet de placer un double repère en fin de ligne.

Cette première méthode, bien que moins compliquée que la seconde, ne permet pas d'ajuster le positionnement d'un repère indépendamment de l'autre. Elle peut aussi générer quelques problèmes quant à l'espacement vertical, puisque le recours à `extra-offset` ne modifie en rien la valeur originelle de la boîte englobant le repère.

```
\relative c'{
  c d e f |
  \once \override Score.RehearsalMark.extra-offset = #'(0 . -8.5)
  \once \override Score.RehearsalMark.baseline-skip = #9
  \mark \markup \center-column { \circle 1 \box A }
  g f e d |
  \once \override Score.RehearsalMark.extra-offset = #'(0 . -8.5)
  \once \override Score.RehearsalMark.baseline-skip = #9
  \mark \markup \center-column { \flat { \bold \small \italic Fine. } }
```

```
  g f e d |
  \once \override Score.RehearsalMark.extra-offset = #'(0 . -8.5)
  \once \override Score.RehearsalMark.baseline-skip = #9
  \override Score.RehearsalMark.break-visibility = #begin-of-line-invisible
  \mark \markup \center-column { \fermata \box z }
}
```



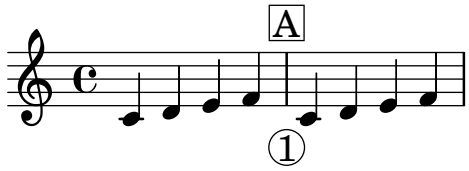
Impression d'un double repère sur une même barre (méthode 2)

Cette méthode permet d'imprimer deux marques-repères – l'une au-dessus de la portée et l'autre en dessous – à l'aide de deux voix séparées. Le graveur de repères est ensuite déplacé dans chacune des voix, sans quoi aucun repère ne sera imprimé. Enfin, chaque voix se voit attribuer un positionnement pour les repères.

Cette méthode, bien que plus complexe que la première, apporte plus de flexibilité dans l'ajustement individuel des deux composants du repère.

```
\score {
  \relative c'
  <<
    \new Staff {
      <<
        \new Voice \with {
          \consists Mark_engraver
          \consists "Staff_collecting_engraver"
        }
        { c4 d e f
          \mark \markup { \box A }
          c4 d e f
        }
        \new Voice \with {
          \consists Mark_engraver
          \consists "Staff_collecting_engraver"
          \override RehearsalMark.direction = #DOWN
        }
        { s4 s s s
          \mark \markup { \circle 1 }
          s4 s s s
        }
      >>
    }
  >>
  \layout {
    \context {
      \Score
```

```
      \remove "Mark_engraver"
      \remove "Staff_collecting_engraver"
    }
  }
}
```
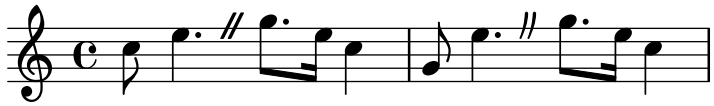


Insertion d'une césure

Une surcharge de la propriété `text` de l'objet `BreathingSign` permet de créer une marque de césure.

LilyPond dispose également d'une variante courbée.

```
\relative c'' {
  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.straight"
  }
  c8 e4. \breathe g8. e16 c4

  \override BreathingSign.text = \markup {
    \musicglyph "scripts.caesura.curved"
  }
  g8 e'4. \breathe g8. e16 c4
}
```
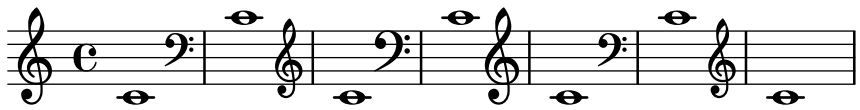


Conservation de la taille lors d'un changement de clef

Lorsqu'un changement de clef intervient, le nouveau signe s'imprime dans une taille inférieure. On peut y déroger avec `full-size-change`.

```
\relative c' {
  \clef "treble"
  c1
  \clef "bass"
  c1
  \clef "treble"
  c1
  \override Staff.Clef.full-size-change = ##t
  \clef "bass"
  c1
  \clef "treble"
  c1
  \revert Staff.Clef.full-size-change
  \clef "bass"
  c1
```

```
  \clef "treble"
  c1
}
```



Terminaison de ligne en flèche

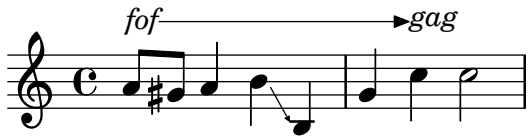
Les extensions de texte, tout comme les indications sous forme de ligne tel un glissando, peuvent se voir pourvues d'une extrémité en flèche.

```
\relative c'' {
  \override TextSpanner.bound-padding = #1.0
  \override TextSpanner.style = #'line
  \override TextSpanner.bound-details.right.arrow = ##t
  \override TextSpanner.bound-details.left.text = #"fof"
  \override TextSpanner.bound-details.right.text = #"gag"
  \override TextSpanner.bound-details.right.padding = #0.6

  \override TextSpanner.bound-details.right.stencil-align-dir-y = #CENTER
  \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER

  \override Glissando.bound-details.right.arrow = ##t
  \override Glissando.arrow-length = #0.5
  \override Glissando.arrow-width = #0.25

  a8\startTextSpan gis a4 b\glissando b,
  g'4 c\stopTextSpan c2
}
```
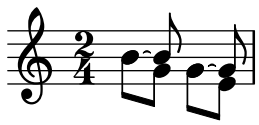


Recours à la propriété `'transparent` pour rendre des objets invisibles

Une modification de la propriété `transparent` permet d'imprimer un objet « à l'encre sympathique » : l'objet n'est pas affiché bien que tous ses attributs soient préservés. L'objet en question occupe donc sa place, est pris en compte lors de collisions, et peut se voir attaché liaisons ou ligatures.

Cet exemple illustre la manière de connecter deux voix par une liaison de prolongation. Les liaisons de prolongation ne peuvent normalement intervenir que dans la même voix. Dès lors que la liaison est entamée dans une autre voix et que la première hampe ascendante est rendue transparente dans cette même voix, la liaison semble passer d'une voix à l'autre.

```
\relative {
  \time 2/4
  <<
    {
      \once \hide Stem
```

```
      \once \override Stem.length = #8
      b'8 ~ 8\noBeam
      \once \hide Stem
      \once \override Stem.length = #8
      g8 ~ 8\noBeam
    }
    \\
    {
      b8 g g e
    }
  >>
}
```

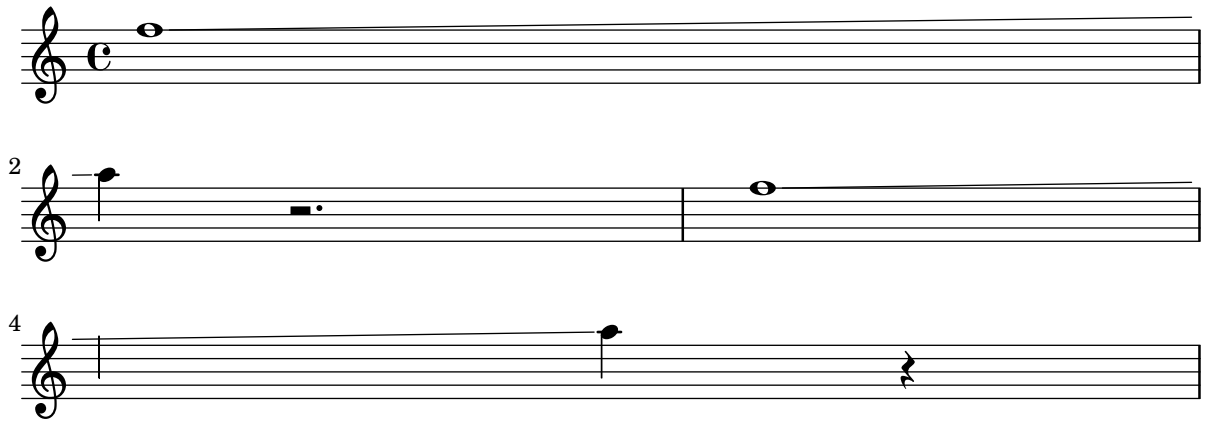


Saut de ligne et glissando

L'affectation de la valeur `#t` à la propriété `breakable`, combinée à `after-line-breaking`, permet la rupture d'une indication de glissando lors d'un saut de ligne.

```
glissandoSkipOn = {
  \override NoteColumn.glissando-skip = ##t
  \hide NoteHead
  \override NoteHead.no-ledgers = ##t
}
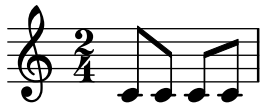
\relative c'' {
  \override Glissando.breakable = ##t
  \override Glissando.after-line-breaking = ##t
  f1\glissando |
  \break
  a4 r2. |
  f1\glissando
  \once \glissandoSkipOn
  \break
  a2 a4 r4 |
}
```



Contrôle manuel du positionnement des ligatures

Le positionnement et la pente des ligatures peuvent être contrôlés manuellement à l'aide d'une adaptation de la propriété `positions` de l'objet `Beam`.

```
\relative c' {
  \time 2/4
  % from upper staff-line (position 2) to center (position 0)
  \override Beam.positions = #'(2 . 0)
  c8 c
  % from center to one above center (position 1)
  \override Beam.positions = #'(0 . 1)
  c8 c
}
```

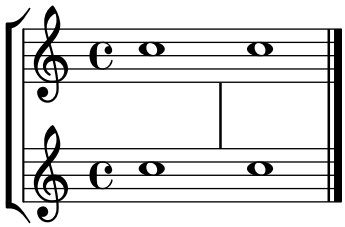


Présentation à l'ancienne (barres de mesure entre les portées)

En musique mensurale, les barres de mesure ne traversent pas les portées. Pour obtenir ce résultat avec un `StaffGroup` plutôt qu'en utilisant un `ChoirStaff`, il faudra masquer les portions de barre qui recouvrent les portées à l'aide d'un `\hide`.

```
global = {
  \hide Staff.BarLine
  s1 s
  % the final bar line is not interrupted
  \undo \hide Staff.BarLine
  \bar "|."
}

\new StaffGroup \relative c'' {
  <<
    \new Staff { << \global { c1 c } >> }
    \new Staff { << \global { c c } >> }
  >>
}
```

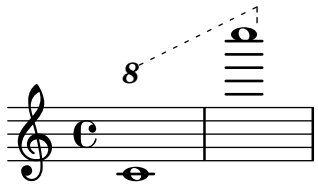


Modification de la pente de l'extension d'octaviation

Il est possible d'adapter la pente d'une indication d'octaviation.

```
\relative c'' {
  \override Staff.OttavaBracket.stencil = #ly:line-spanner::print
  \override Staff.OttavaBracket.bound-details =
    #`((left . ((Y . 0) ; Change the integer here
```
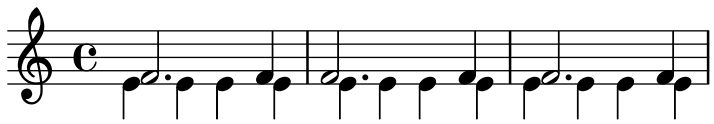
```
                (attach-dir . ,LEFT)
                (padding . 0)
                (stencil-align-dir-y . ,CENTER)))
       (right . ((Y . 5) ; Change the integer here
                 (padding . 0)
                 (attach-dir . ,RIGHT)
                 (text . ,(make-draw-dashed-line-markup
                           (cons 0 -1.2))))))
  \override Staff.OttavaBracket.left-bound-info =
     #ly:line-spanner::calc-left-bound-info-and-text
  \override Staff.OttavaBracket.right-bound-info =
     #ly:line-spanner::calc-right-bound-info
  \ottava #1
  c1
  c'''1
}
```



Déplacement des notes pointées dans une polyphonie

Une note pointée appartenant à la voix supérieure d'une portée polyphonique sera par défaut décalée vers la droite afin d'éviter les collisions avec les autres voix. Ce comportement peut être outrepassé à l'aide de la propriété `prefer-dotted-right` de `NoteCollision`.

```
\new Staff \relative c' <<
  {
    f2. f4
    \override Staff.NoteCollision.prefer-dotted-right = ##f
    f2. f4
    \override Staff.NoteCollision.prefer-dotted-right = ##t
    f2. f4
  }
  \\
  { e4 e e e e e e e e e e }
>>
```
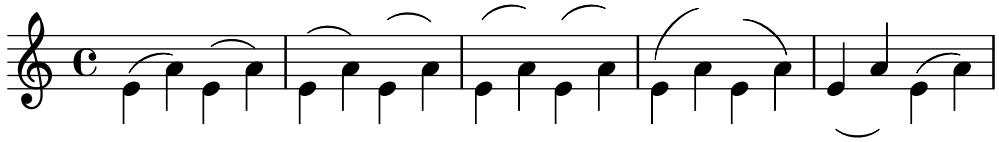


Déplacement vertical des liaisons d'articulation

Le positionnement vertical d'une liaison se contrôle par la propriété `positions` de l'objet `Slur`. Cette propriété dispose de deux paramètres : le premier affecte l'extrémité gauche de la liaison, le second son extrémité droite. La valeur des paramètres n'aura aucune influence sur le galbe de la liaison. LilyPond ne s'en servira que pour adapter le positionnement de la courbe. Des valeurs

positives décalent la liaison vers le haut et s'utilisent pour des hampes descendantes. Des valeurs négatives entraînent un décalage vers le bas.

```
\relative c' {
  \stemDown
  e4( a)
  \override Slur.positions = #'(1 . 1)
  e4( a)
  \override Slur.positions = #'(2 . 2)
  e4( a)
  \override Slur.positions = #'(3 . 3)
  e4( a)
  \override Slur.positions = #'(4 . 4)
  e4( a)
  \override Slur.positions = #'(5 . 5)
  e4( a)
  \override Slur.positions = #'(0 . 5)
  e4( a)
  \override Slur.positions = #'(5 . 0)
  e4( a)
  \stemUp
  \override Slur.positions = #'(-5 . -5)
  e4( a)
  \stemDown
  \revert Slur.positions
  e4( a)
}
```

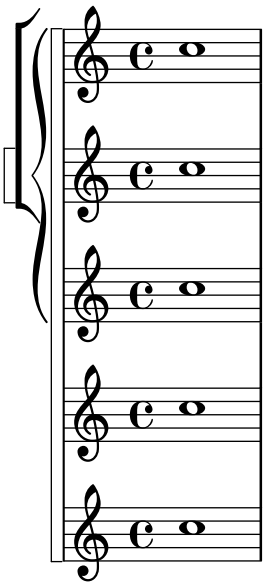


Imbrications de regroupements de portées

La propriété `systemStartDelimiterHierarchy` permet de créer des regroupements imbriqués complexes. La commande `\set StaffGroup.systemStartDelimiterHierarchy` prend en argument la liste alphabétique des sous-groupes à hiérarchiser. Chaque sous-groupe peut être affublé d'un délimiteur particulier. Chacun des regroupements intermédiaires doit être borné par des parenthèses. Bien que des éléments de la liste puissent être omis, le premier délimiteur embrassera toujours l'intégralité des portées. Vous disposez des quatre délimiteurs `SystemStartBar`, `SystemStartBracket`, `SystemStartBrace` et `SystemStartSquare`.

```
\new StaffGroup
\relative c'' <<
  \override StaffGroup.SystemStartSquare.collapse-height = #4
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare (SystemStartBrace (SystemStartBracket a
                            (SystemStartSquare b)  ) c ) d)
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
```

>>



Personnalisation de certains types d'articulation

On peut parfois vouloir modifier un seul type d'articulation. Bien que ce soit tout à fait faisable avec un `\tweak`, cela devient vite fastidieux d'affiner chaque occurrence d'un même signe dans toute une partition. Le code ci-dessous illustre la manière de modifier des articulations grâce à une liste de réglages personnalisés. Ceci peut servir à créer des feuilles de style.

La fonction `\customScripts` ici proposée peut se placer, avec la version 2.16.2, dans un bloc `\layout`.

```
% Code by David Nalesnik and Thomas Morley

#(define (custom-script-tweaks ls)
  (lambda (grob)
    (let* ((type (ly:prob-property
                   (ly:grob-property grob 'cause)
                   'articulation-type))
           (tweaks (assoc-ref ls type)))
      (if tweaks
          (for-each
            (lambda (x) (ly:grob-set-property! grob (car x) (cdr x)))
            tweaks)))))

customScripts =
#(define-music-function (settings)(list?)
#{
  \override Script.before-line-breaking =
    #(custom-script-tweaks settings)
#})

revertCustomScripts = { \revert Script.before-line-breaking }

%%%%%%%%%%%%
% Example:
```

```
%%%%%%%%%%%

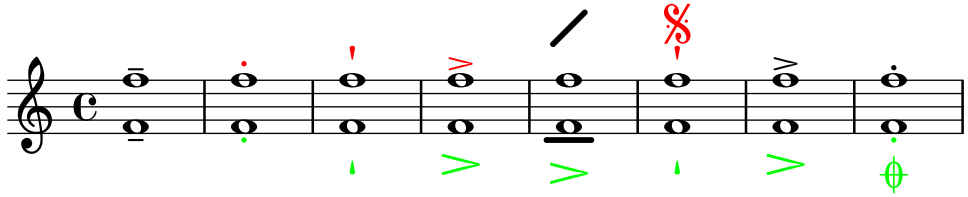
% Predefine a list of desired tweaks.
#(define my-settings-1
  '(
    ("staccato" . ((color . (1 0 0))(padding . 0.5)))
    ("accent" . ((font-size . 0)(color . (1 0 0))))
    ("tenuto" . ((rotation . (45 0 0)) (padding . 2)(font-size . 10)))
    ("staccatissimo" . ((padding . 1) (color . (1 0 0))))
    ("segno" . ((font-size . 0)(color . (1 0 0))))
    ))

#(define my-settings-2
  '(
    ("staccato" . ((color . (0 1 0))))
    ("accent" . ((font-size . 4)(color . (0 1 0))(padding . 1.5)))
    ("tenuto" . ((font-size . 10)))
    ("staccatissimo" . ((padding . 2) (color . (0 1 0))))
    ("coda" . ((color . (0 1 0)) (padding . 1)))
    ))

one =
\relative c'' {
  f1--
  \customScripts #my-settings-1
  f-. f-! f-> f-- f-!\segno
  \revertCustomScripts
  f-> f-.
}

two =
\relative c' {
  f1--
  \customScripts #my-settings-2
  f-. f-! f-> f---> f-!
  f-> f-.\coda
}

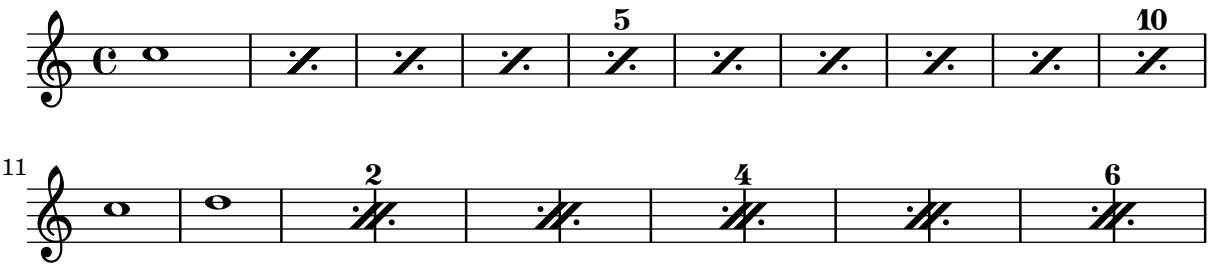
\new Staff <<
   \new Voice { \voiceOne \one }
   \new Voice { \voiceTwo \two }
   >>
```



Affichage du numéro de répétition en pourcent

Le numéro de mesure répétée sera imprimé à intervalle régulier si vous déterminez la propriété de contexte `repeatCountVisibility`.

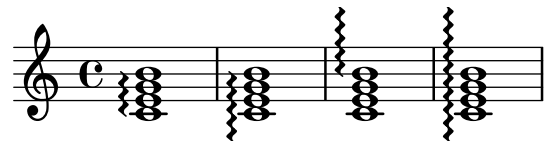
```
\relative c'' {
  \set countPercentRepeats = ##t
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
  \repeat percent 10 { c1 } \break
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
  \repeat percent 6 { c1 d1 }
}
```



Positionnement des arpeggios

L'ajustement de la taille d'une indication d'arpeggio s'effectue au travers du positionnement de ses extrémités haute ou basse.

```
\relative c' {
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(-5 . 0)
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(0 . 5)
  <c e g b>1\arpeggio
  \once \override Arpeggio.positions = #'(-5 . 5)
  <c e g b>1\arpeggio
}
```



Positionnement des silences multimesures

Si l'on peut positionner verticalement un silence simple en le rattachant à une note, il n'en va pas de même pour un silence multimesure. Néanmoins, et uniquement dans le cadre de musique polyphonique, les silences multimesures sont positionnés différemment selon qu'ils appartiennent à une voix au numéro pair ou impair. Le positionnement des silences multimesures peut se contrôler ainsi :

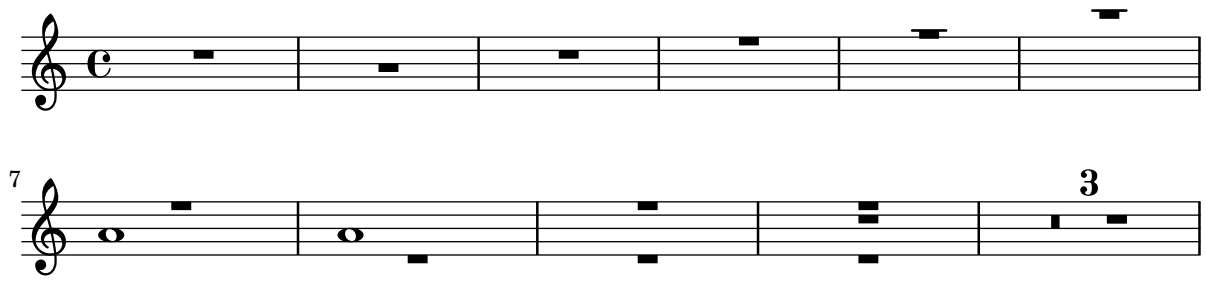
```
\relative c'' {
  % Multi-measure rests by default are set under the fourth line
  R1
  % They can be moved using an override
  \override MultiMeasureRest.staff-position = #-2
  R1
```

```
  \override MultiMeasureRest.staff-position = #0
  R1
  \override MultiMeasureRest.staff-position = #2
  R1
  \override MultiMeasureRest.staff-position = #3
  R1
  \override MultiMeasureRest.staff-position = #6
  R1
  \revert MultiMeasureRest.staff-position
  \break

  % In two Voices, odd-numbered voices are under the top line
  << { R1 } \\ { a1 } >>
  % Even-numbered voices are under the bottom line
  << { a1 } \\ { R1 } >>
  % Multi-measure rests in both voices remain separate
  << { R1 } \\ { R1 } >>

  % Separating multi-measure rests in more than two voices
  % requires an override
  << { R1 } \\ { R1 } \\
     \once \override MultiMeasureRest.staff-position = #0
     { R1 }
  >>

  % Using compressed bars in multiple voices requires another override
  % in all voices to avoid multiple instances being printed
  \compressMMRests
  <<
   \revert MultiMeasureRest.direction
    { R1*3 }
    \\
   \revert MultiMeasureRest.direction
    { R1*3 }
  >>
}
```
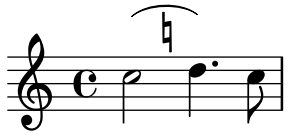


Positionnement d'une annotation à l'intérieur d'une liaison

Lorsqu'une annotation doit s'incrire à l'intérieur d'une liaison, la propriété `outside-staff-priority` doit être désactivée.

```
\relative c'' {
  \override TextScript.avoid-slur = #'inside
```

```
  \override TextScript.outside-staff-priority = ##f
  c2(^\markup { \halign #-10 \natural } d4.) c8
}
```



Inscrire le numéro de mesure dans un cadre ou un cercle

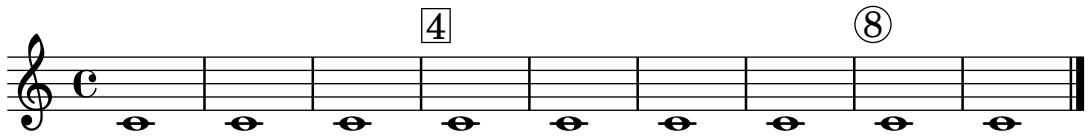
Les numéros de mesure peuvent être encadrés ou entourés d'un cercle.

```
\relative c' {
  % Prevent bar numbers at the end of a line and permit them elsewhere
  \override Score.BarNumber.break-visibility = #end-of-line-invisible
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 4)

  % Increase the size of the bar number by 2
  \override Score.BarNumber.font-size = #2

  % Draw a box round the following bar number(s)
  \override Score.BarNumber.stencil
    = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
  \repeat unfold 5 { c1 }

  % Draw a circle round the following bar number(s)
  \override Score.BarNumber.stencil
    = #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
  \repeat unfold 4 { c1 } \bar "|."
}
```



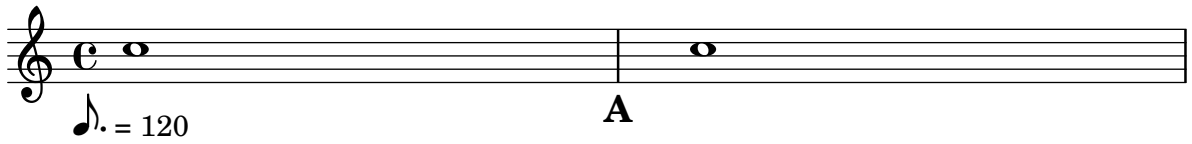
Impression du métronome et des repères sous la portée

Les indications de tempo et les marques de repère s'impriment par défaut au-dessus de la portée. Le fait de régler en conséquence la propriété `direction` des objets `MetronomeMark` et `RehearsalMark` les placera au-dessous de la portée.

```
\layout {
  indent = 0
  ragged-right = ##f
}

{
  % Metronome marks below the staff
  \override Score.MetronomeMark.direction = #DOWN
  \tempo 8. = 120
  c''1
```

```
  % Rehearsal marks below the staff
  \override Score.RehearsalMark.direction = #DOWN
  \mark \default
  c''1
}
```
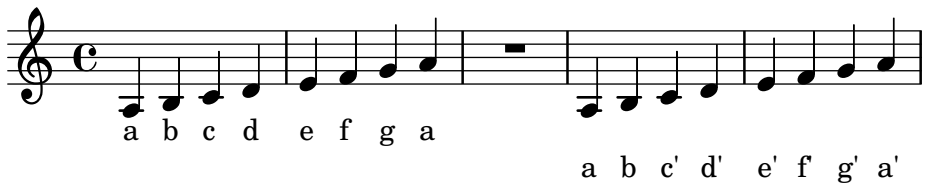


Impression du nom des notes avec ou sans indication d'octave

Le contexte `NoteNames` permet d'imprimer le nom des notes. La propriété `printOctaveNames`, une fois activée, leur adjoindra une indication d'octave.

```
scale = \relative c' {
  a4 b c d
  e4 f g a
}

\new Staff {
  <<
    \scale
    \context NoteNames {
      \set printOctaveNames = ##f
      \scale
    }
  >>
  R1
  <<
    \scale
    \context NoteNames {
      \set printOctaveNames = ##t
      \scale
    }
  >>
}
```

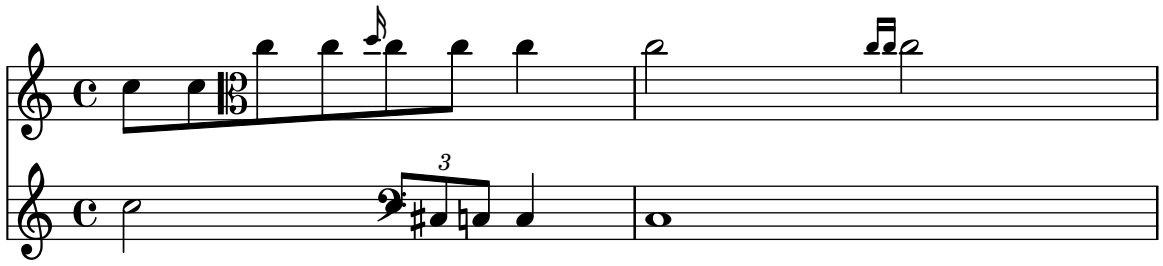


Espacement strictement proportionnel des notes

Lorsque la propriété `strict-note-spacing` est activée, l'espacement des notes dans un système ne tient compte ni des barres de mesure ni des clefs, qui se retrouvent placées juste avant la note qui tombe au même moment. Ceci peut entraîner certaines collisions.

```
\relative c'' <<
  \override Score.SpacingSpanner.strict-note-spacing = ##t
  \set Score.proportionalNotationDuration = #(ly:make-moment 1/16)
```

```
  \new Staff {
    c8[ c \clef alto c c \grace { d16 } c8 c] c4
    c2 \grace { c16[ c16] } c2
  }
  \new Staff {
    c2 \tuplet 3/2 { c8 \clef bass cis,, c } c4
    c1
  }
>>
```



Retrait de l'accolade à la première ligne d'une pièce pour piano

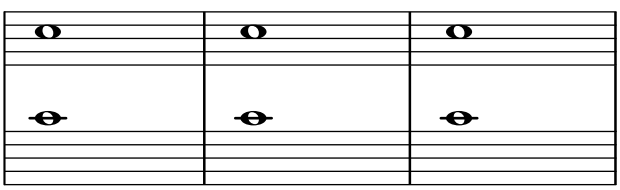
Dans cet extrait est supprimée la première accolade d'un `PianoStaff` ou d'un `GrandStaff`.

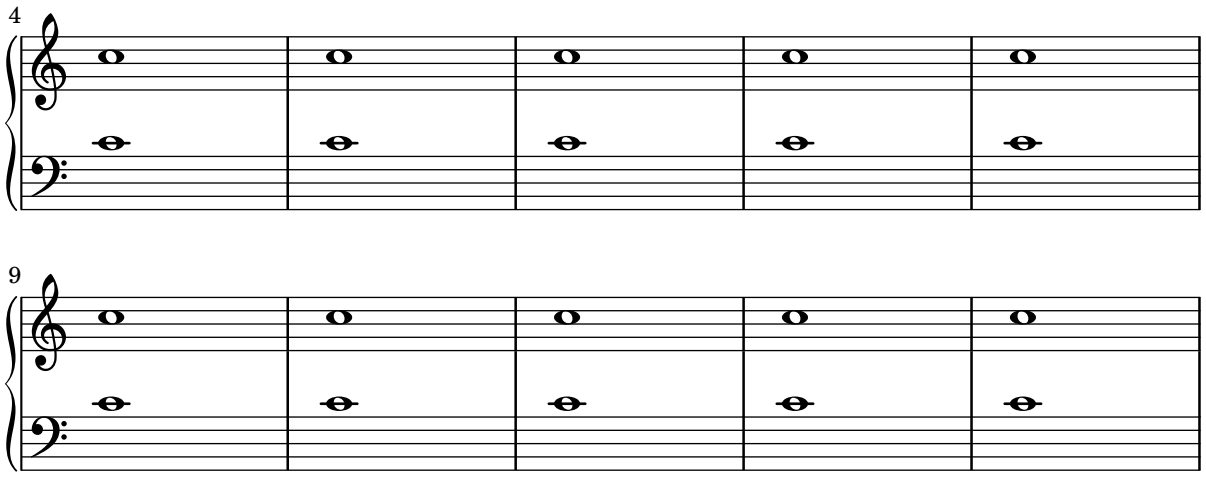
Ceci peut s'avérer utile pour couper et coller l'image générée dans de la musique préexistante.

Est utilisée la fonction `\alterBroken`.

```
someMusic =  {
  \once \override Staff.Clef.stencil = ##f
  \once \override Staff.TimeSignature.stencil = ##f
  \repeat unfold 3 c1 \break
  \repeat unfold 5 c1 \break
  \repeat unfold 5 c1
}

\score {
  \new PianoStaff
  <<
    \new Staff = "right" \relative c'' \someMusic
    \new Staff = "left" \relative c' { \clef F \someMusic }
  >>
  \layout {
    indent=75
    \context {
      \PianoStaff
      \alterBroken transparent #'(#t) SystemStartBrace
    }
  }
}
```

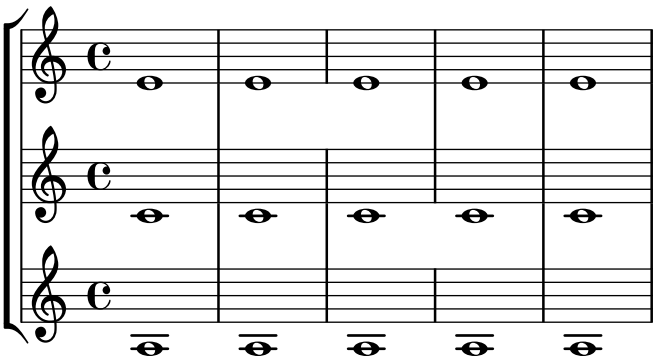




Suppression de la partie interportée des barres de mesure d'un regroupement autre que `ChoirStaff`

Les barres de mesure des regroupements `StaffGroup`, `PianoStaff` et `GrandStaff` sont par défaut d'un seul tenant. La portion entre les portées – l'objet `SpanBar` – peut néanmoins être supprimée, portée par portée.

```
\relative c' {
  \new StaffGroup <<
    \new Staff {
      e1 | e
      \once \override Staff.BarLine.allow-span-bar = ##f
      e1 | e | e
    }
    \new Staff {
      c1 | c | c
      \once \override Staff.BarLine.allow-span-bar = ##f
      c1 | c
    }
    \new Staff {
      a1 | a | a | a | a
    }
  >>
}
```



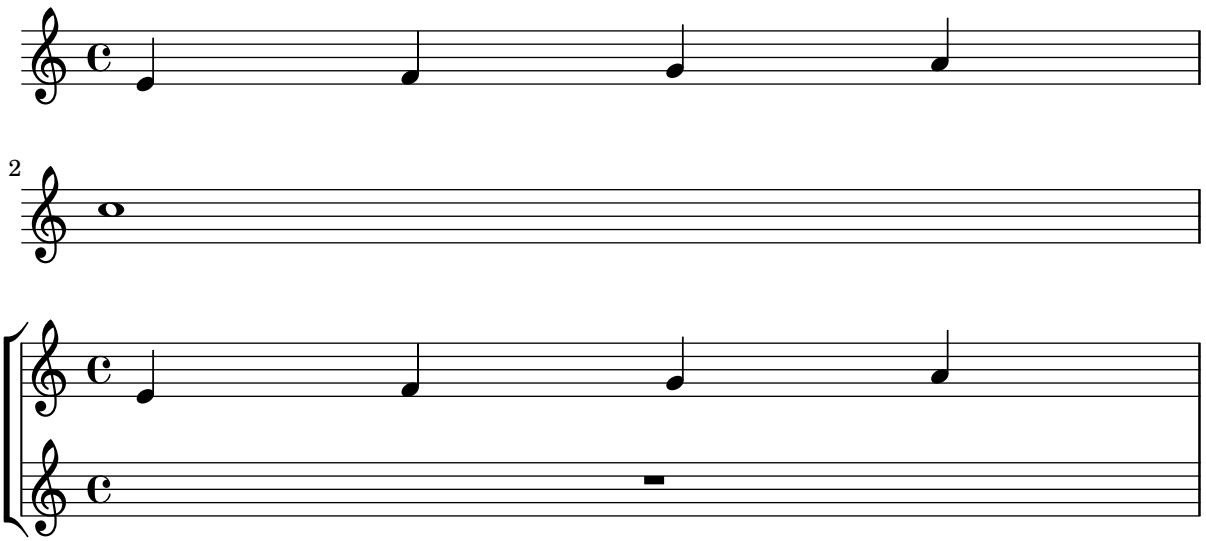
Masquage de la première ligne si elle est vide

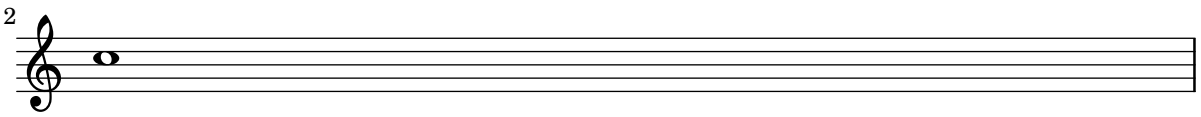
Par défaut, le premier système comportera absolument toutes les portées. Si vous préférez masquer les portées vides y compris pour le premier système, vous devrez activer la propriété `remove-first` du `VerticalAxisGroup`. Mentionnée dans un bloc `\layout`, cette commande

agira de manière globale. Pour qu'elle ne soit effective que pour une portée particulière, vous devrez également spécifier le contexte (`Staff` pour qu'il ne concerne que la portée en cours) en préfixe de la propriété.

La première ligne inférieure du deuxième `StaffGroup` est bien présente, pour la simple raison que le réglage en question ne s'applique qu'à la portée dans laquelle il a été inscrit.

```
\layout {
  \context {
    \Staff \RemoveEmptyStaves
    % To use the setting globally, uncomment the following line:
    % \override VerticalAxisGroup.remove-first = ##t
  }
}
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    % To use the setting globally, comment this line,
    % uncomment the line in the \layout block above
    \override Staff.VerticalAxisGroup.remove-first = ##t
    R1 \break
    R
  }
>>
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    R1 \break
    R
  }
>>
```





Styles de silences

Les silences peuvent être gravés selon différents styles.

```
\new Staff \relative c {
  \omit Score.TimeSignature
  \cadenzaOn

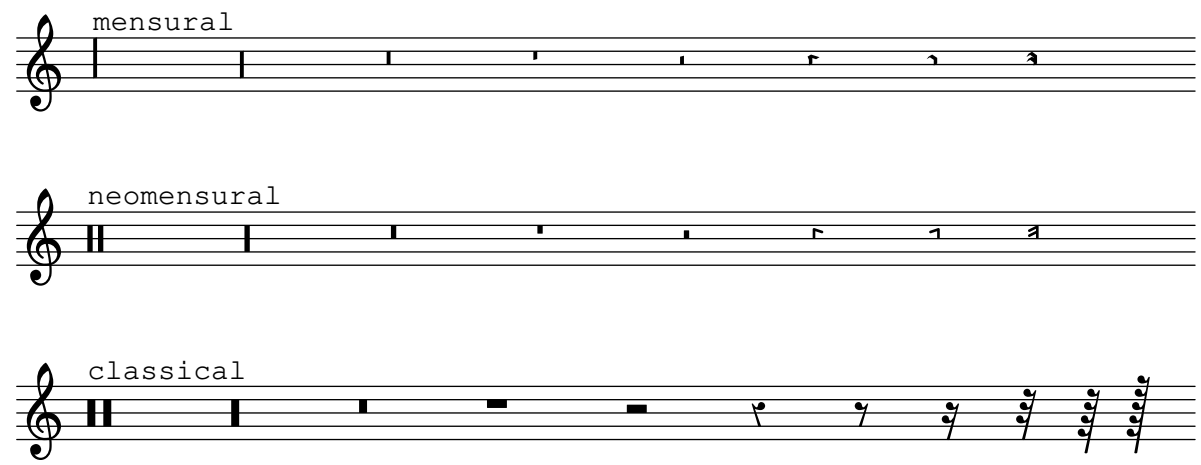
  \override Staff.Rest.style = #'mensural
  r\maxima^\markup \typewriter { mensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""
  \break

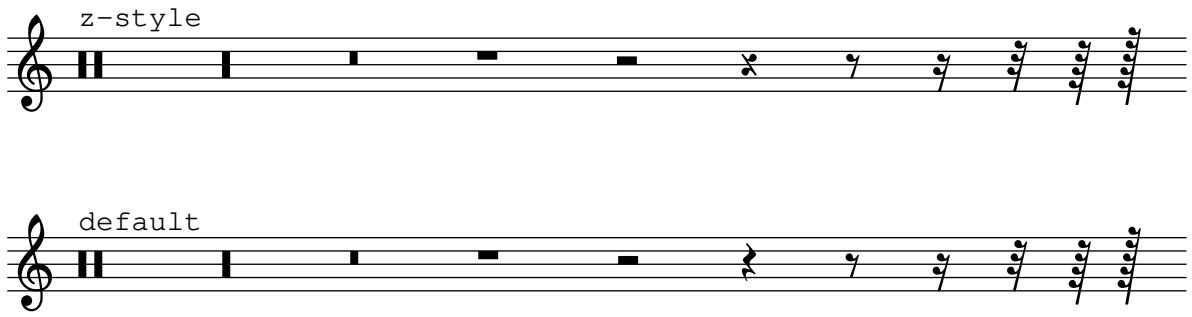
  \override Staff.Rest.style = #'neomensural
  r\maxima^\markup \typewriter { neomensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'classical
  r\maxima^\markup \typewriter { classical }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'z
  r\maxima^\markup \typewriter { z-style }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
  \bar ""
  \break

  \override Staff.Rest.style = #'default
  r\maxima^\markup \typewriter { default }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
}
```





Barres rythmiques

Il arrive, dans une feuille de chant « simple », que les notes soient remplacées par une « pulsation » et que la structure de la chanson soit indiquée par les accords au-dessus des mesures. Ceci peut être utile lorsque l'on crée ou retranscrit la structure d'une chanson, ainsi que pour donner au guitariste et musiciens de jazz une pseudo partition.

Le fonctionnement standard d'un `\repeat percent` n'est pas ici applicable puisque le premier temps doit être une note ou un silence.

Le code ci-dessous propose deux alternatives à ce problème en redéfinissant l'aspect d'un silence. Si la durée d'un temps ne correspond pas à la noire, le `r4` inclus dans la définition devra être remplacé par un silence de durée appropriée.

```
% Macro to print single slash
rs = {
  \once \override Rest.stencil = #ly:percent-repeat-item-interface::beat-slash
  \once \override Rest.thickness = #0.48
  \once \override Rest.slope = #1.7
  r4
}

% Function to print a specified number of slashes
comp = #(define-music-function (count) (integer?)
  #{
    \override Rest.stencil = #ly:percent-repeat-item-interface::beat-slash
    \override Rest.thickness = #0.48
    \override Rest.slope = #1.7
    \repeat unfold $count { r4 }
    \revert Rest.stencil
  #}
)

\score {
  \relative c' {
    c4 d e f |
    \rs \rs \rs \rs |
    \comp #4 |
  }
}
```



Séparation entre altérations annulées et nouvelle armure

Les altérations supprimées lors d'un changement de tonalité sont par défaut accolées à la nouvelle armure. Ce comportement peut s'adapter grâce à la propriété `break-align-orders` de l'objet `BreakAlignment`.
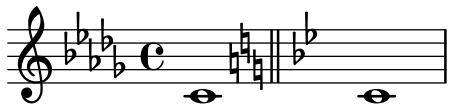
La valeur de `break-align-orders` est constituée d'un vecteur à trois composantes listant l'ordre des éléments à l'occasion d'une rupture. Seule la deuxième liste est ici modifiée, `key-cancellation` étant déplacé avant `staff-bar`. En ne modifiant que la deuxième liste, cette modification dans l'ordre d'apparition des éléments sera effective seulement en cours de système, et non en début ou fin de ligne.

```
\new Staff {
  \override Score.BreakAlignment.break-align-orders =
    ##((left-edge ambitus breathing-sign clef staff-bar
                  key-cancellation key-signature time-signature custos)

       (left-edge ambitus breathing-sign clef key-cancellation
                  staff-bar key-signature time-signature custos)

       (left-edge ambitus breathing-sign clef key-cancellation
                  key-signature staff-bar time-signature custos))

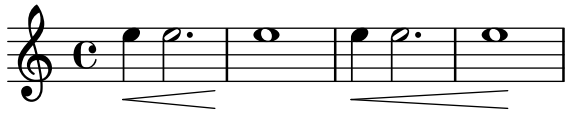
  \key des \major
  c'1
  \bar "||"
  \key bes \major
  c'1
}
```



Soufflets et barres de mesure

En principe, un soufflet – (de)crescendo imprimé sous forme graphique – commence au bord gauche de la note de départ, et se termine au bord droit de la note d'arrivée. Cependant, si la note d'arrivée est sur un premier temps, le soufflet s'arrêtera au niveau de la barre de mesure qui la précède. Ce comportement peut être annulé en assignant *faux* (`#f`) à la propriété `to-barline`.

```
\relative c'' {
  e4\< e2.
  e1\!
  \override Hairpin.to-barline = ##f
  e4\< e2.
  e1\!
}
```



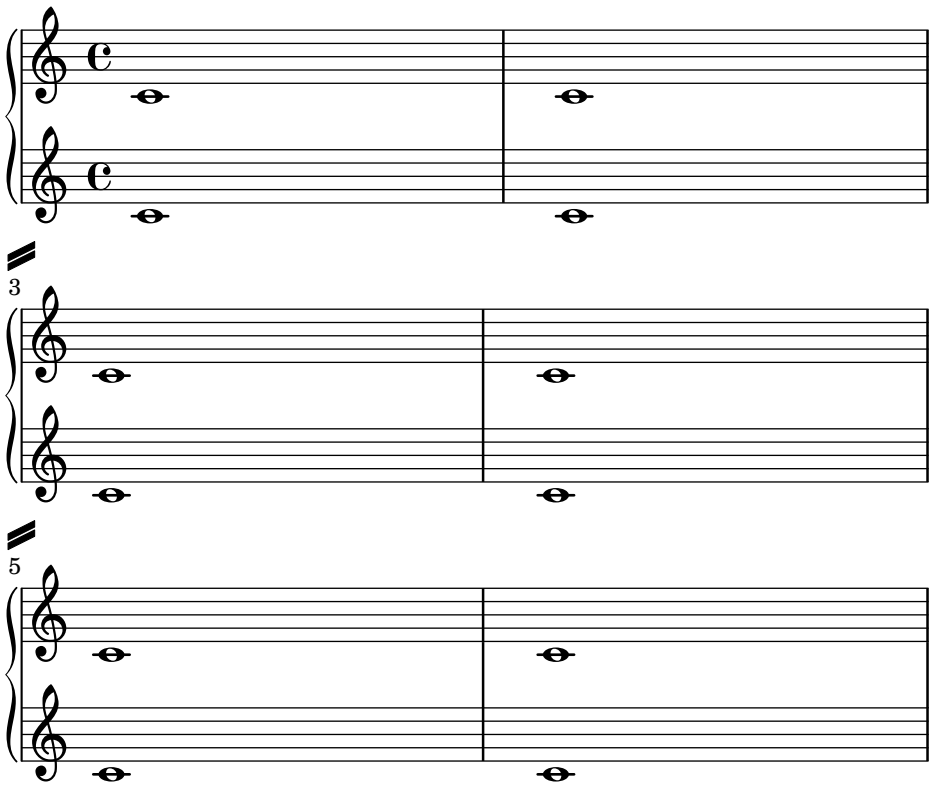
Séparation visuelle entre les systèmes

La séparation entre deux systèmes consécutifs peut être mise en évidence par n'importe quel *markup*. LilyPond dispose à cet effet d'une double oblique inversée : `\slashSeparator`.

```
\paper {
  system-separator-markup = \slashSeparator
  line-width = 120
}

notes = \relative c' {
  c1 | c \break
  c1 | c \break
  c1 | c
}

\book {
  \score {
    \new GrandStaff <<
      \new Staff \notes
      \new Staff \notes
    >>
  }
}
```


3
5

Impression d'une même articulation des deux côtés d'une note ou d'un accord

LilyPond ne permet pas, par défaut, qu'une même articulation (accent, flageolet, point d'orgue, etc.) se retrouve à la fois au-dessus et au-dessous d'une note. Par exemple, `c4_\fermata^\fermata` ne donnera qu'un seul point d'orgue en dessous du do ; celui du dessus sera tout bonnement ignoré.

On peut néanmoins accoler des scripts, tels des doigtés, à l'intérieur d'un accord ; il peut donc y avoir autant d'articulations que de besoin, ce qui, par voie de conséquence, permet de s'affranchir de la présence des hampes et de positionner l'articulation relativement à la tête de note comme dans le cas du flageolet ci-dessous. L'imitation du traitement d'un script externe à un accord requérant un `'add-stem-support` demande de libeller la note comme étant un accord et d'ajouter les articulations au sein de la construction `<...>`.

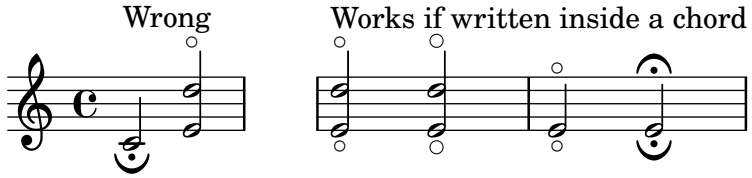
Un simple amendement permettra de rectifier le positionnement habituel en surplomb : `<c-\tweak direction #DOWN-\fermata^\fermata>`

```
% The same as \flageolet, just a little smaller
smallFlageolet = \tweak font-size #-2 \flageolet

\relative c' {
  <>^"Wrong"
  c2_\fermata^\fermata % The second fermata is ignored!
  <e d'>2^\smallFlageolet_\smallFlageolet

  \stopStaff s1 \startStaff

  <>^"Works if written inside a chord"
  <e_\smallFlageolet d'^\smallFlageolet>2
  <e_\flageolet d'^\flageolet>2
  <e_\smallFlageolet^\smallFlageolet>2
  <e_\fermata^\fermata>2
}
```

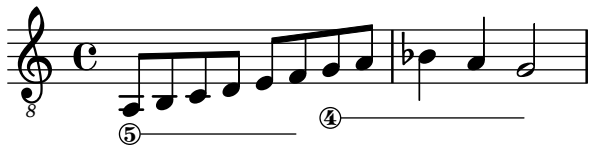


Ligne de prolongation pour numéro de corde

Voici comment ajouter une ligne de prolongation à une indication de numéro de corde, afin de stipuler que les notes qui suivent doivent être jouées sur la corde en question.

```
stringNumberSpanner =
  #(define-music-function (StringNumber) (string?)
    #{
      \override TextSpanner.style = #'solid
      \override TextSpanner.font-size = #-5
      \override TextSpanner.bound-details.left.stencil-align-dir-y = #CENTER
      \override TextSpanner.bound-details.left.text =
        \markup { \circle \number $StringNumber }
    #})
```

```
\relative c {
  \clef "treble_8"
  \stringNumberSpanner "5"
  \textSpannerDown
  a8\startTextSpan
  b c d e f\stopTextSpan
  \stringNumberSpanner "4"
  g\startTextSpan a
  bes4 a g2\stopTextSpan
}
```
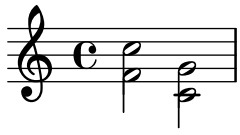


Suppression des avertissements de chevauchement

If notes from two voices with stems in the same direction are placed at the same position, and both voices have no shift or the same shift specified, the error message '`warning: ignoring too many clashing note columns`' will appear when compiling the LilyPond file. This message can be suppressed by setting the `'ignore-collision` property of the `NoteColumn` object to `#t`. Please note that this does not just suppress warnings but stops LilyPond trying to resolve collisions at all and so may have unintended results unless used with care.

```
ignore = \override NoteColumn.ignore-collision = ##t
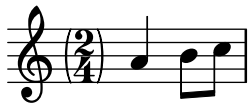
\relative c' {
  \new Staff <<
    \new Voice { \ignore \stemDown f2 g }
    \new Voice { c2 \stemDown c, }
  >>
}
```



Métrique entre parenthèses – méthode 3

Autre manière de mettre la métrique entre parenthèses.
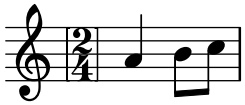
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (parenthesize-stencil (ly:time-signature::print grob) 0.1 0.4 0.4 0.1 ))
  \time 2/4
  a4 b8 c
}
```



Métrique entre parenthèses

Une métrique peut être mise entre parenthèses.
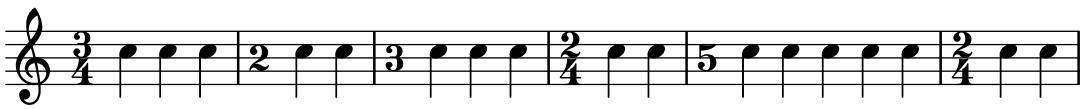
```
\relative c'' {
  \override Staff.TimeSignature.stencil = #(lambda (grob)
    (bracketify-stencil (ly:time-signature::print grob) Y 0.1 0.2 0.1))
  \time 2/4
  a4 b8 c
}
```



Affichage seulement du numérateur d'une métrique (au lieu d'une fraction)

La métrique est parfois indiquée non pas par une fraction (par ex. 7/4) mais simplement par son numérateur (7 dans ce cas). L'instruction `\override Staff.TimeSignature.style = #'single-digit` permet de déroger au style par défaut de manière permanente – un `\revert Staff.TimeSignature.style` annulera ces modifications. Lorsque cette métrique sous la forme d'un seul chiffre ne se présente qu'une seule fois, il suffit de faire précéder l'instruction `\override` d'un simple `\once`.

```
\relative c'' {
  \time 3/4
  c4 c c
  % Change the style permanently
  \override Staff.TimeSignature.style = #'single-digit
  \time 2/4
  c4 c
  \time 3/4
  c4 c c
  % Revert to default style:
  \revert Staff.TimeSignature.style
  \time 2/4
  c4 c
  % single-digit style only for the next time signature
  \once \override Staff.TimeSignature.style = #'single-digit
  \time 5/4
  c4 c c c c
  \time 2/4
  c4 c
}
```



Crochet de n-olet et changement de portée

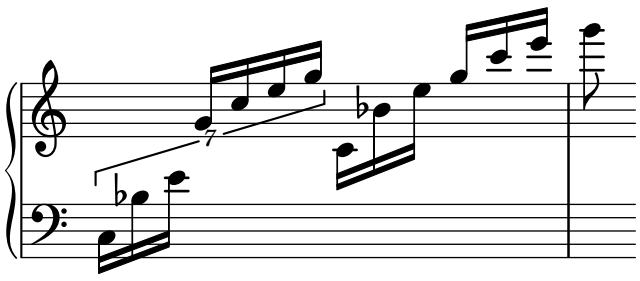
Voici comment obtenir un crochet de n-olet qui débute sur une portée et se termine dans l'autre.

```
aigues = \relative c' {
```

```
  \time 6/8
  s4.
  \stemDown
  c16[ bes' e]
  \stemUp
  g c e
  \stemDown
  g8
}

basses = \relative c {
  \time 3/4
  \clef F
  \tweak positions #'(4.5 . 9.5)
  \tweak edge-height #'(1 . -1)
  \tuplet 7/6 {
    c16[ bes' e]
    \change Staff = md
    \stemUp
    g[ c e g]
  }
  s4.s8
}

\new PianoStaff
\with { \omit TimeSignature }
<<
  \new Staff = md \aigues
  \new Staff = mg \basses
>>
```



Affinage des propriétés d'une clef

Modifier le glyphe, la position de la clef ou son octaviation ne changeront pas la position des notes ; il faut pour y parvenir modifier aussi la position du do médium. La redéfinition préalable de `middleCClefPosition` permet de placer l'armure sur les bonnes lignes. Le positionnement est relatif à la ligne médiane, un nombre positif faisant monter, un nombre négatif abaissant.

Par exemple, la commande `\clef "treble_8"` équivaut à définir `clefGlyph`, `clefPosition` – qui contrôle la position verticale de la clef – `middleCPosition` et `clefOctavation`. Une nouvelle clef apparaîtra dès lors que l'une de ces propriétés, à l'exception de `middleCPosition`, aura été modifiée.

Les exemples qui suivent illustrent les différentes possibilités de définir ces propriétés manuellement. Sur la première ligne, la position relative des notes par rapport aux clefs est préservée, ce qui n'est pas le cas pour la deuxième ligne.

```
{
  % The default treble clef
  \key f \major
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  \set Staff.middleCPosition = #6
  \set Staff.middleCClefPosition = #6
  \key g \major
  c'1
  % The baritone clef
  \set Staff.clefGlyph = #"clefs.C"
  \set Staff.clefPosition = #4
  \set Staff.middleCPosition = #4
  \set Staff.middleCClefPosition = #4
  \key f \major
  c'1
  % The standard choral tenor clef
  \set Staff.clefGlyph = #"clefs.G"
  \set Staff.clefPosition = #-2
  \set Staff.clefTransposition = #-7
  \set Staff.middleCPosition = #1
  \set Staff.middleCClefPosition = #1
  \key f \major
  c'1
  % A non-standard clef
  \set Staff.clefPosition = #0
  \set Staff.clefTransposition = #0
  \set Staff.middleCPosition = #-4
  \set Staff.middleCClefPosition = #-4
  \key g \major
  c'1 \break

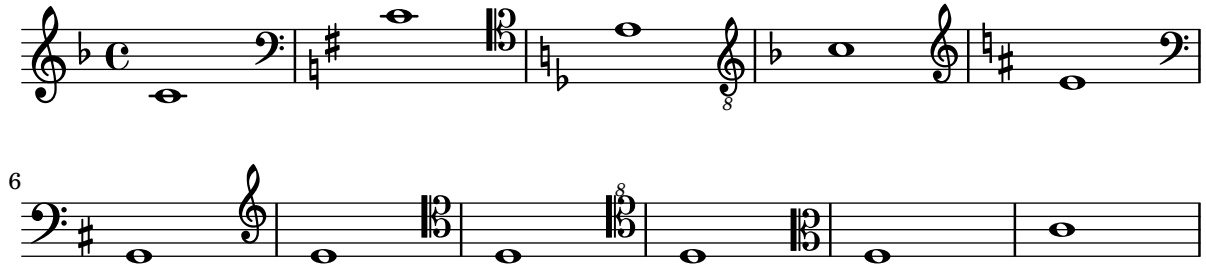
  % The following clef changes do not preserve
  % the normal relationship between notes, key signatures
  % and clefs:

  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  c'1
  \set Staff.clefGlyph = #"clefs.G"
  c'1
  \set Staff.clefGlyph = #"clefs.C"
  c'1
  \set Staff.clefTransposition = #7
  c'1
  \set Staff.clefTransposition = #0
```

```
  \set Staff.clefPosition = #0
  c'1

  % Return to the normal clef:

  \set Staff.middleCPosition = #0
  c'1
}
```
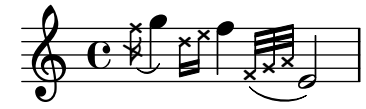


Mise en forme des notes d'ornement

Il est possible de changer globalement la mise en forme des notes d'ornement dans un morceau, au moyen des fonctions `add-grace-property` et `remove-grace-property`.

Ici, par exemple, on ôte la définition de l'orientation des objets `Stem` pour toutes les petites notes, afin que les hampes ne soient pas toujours orientées vers le haut, et on leur préfère des têtes en forme de croix.

```
\relative c'' {
  \new Staff {
    $(remove-grace-property 'Voice 'Stem 'direction)
    $(add-grace-property 'Voice 'NoteHead 'style 'cross)
    \new Voice {
       \acciaccatura { f16 } g4
       \grace { d16 e } f4
       \appoggiatura { f,32 g a } e2
    }
  }
}
```



Crochets de style alternatif

Une dérogation à la propriété `stencil` de l'objet `Flag` permet aux croches et notes de durée inférieure d'adopter une autre forme de crochet. Sont disponibles les variantes `modern-straight-flag`, `old-straight-flag` et `flat-flag`.

```
testnotes = {
  \autoBeamOff
  c8 d16 c32 d64 \acciaccatura { c8 } d64 r4
}

\score {
```
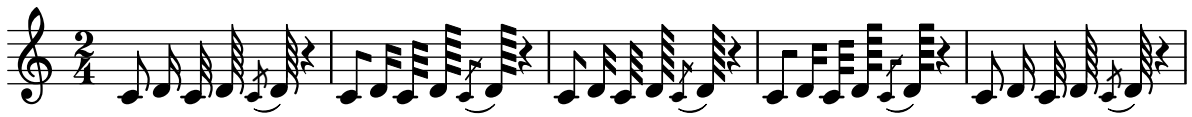
```
  \relative c' {
    \time 2/4
    \testnotes

    \override Flag.stencil = #modern-straight-flag
    \testnotes

    \override Flag.stencil = #old-straight-flag
    \testnotes

    \override Flag.stencil = #flat-flag
    \testnotes

    \revert Flag.stencil
    \testnotes
  }
  \layout {
    indent = 0
    \context {
      \Score
      \override NonMusicalPaperColumn.line-break-permission = ##f
    }
  }
}
```



Utilisation de `ly:grob-object` pour accéder aux *grobs* avec `\tweak`

Certains objets graphiques ne sont accessibles que par le biais d'un *callback* à partir d'un autre `grob`. Ils sont normalement listés dans les « *layout objects* » au sein de la section « Propriétés internes » d'une *grob-interface*. La fonction `ly:grob-object` permet d'accéder à ces objets.

Voici plusieurs moyens d'accéder aux objets par un *callback* sur `NoteHead`. D'autres biais sont naturellement possibles ; `NoteHead` a cependant l'avantage incontestable d'être utilisé implicitement par la commande `\tweak`.

La fonction `display-grobs` définie ci-dessous n'est probablement pas très utile. Elle indique toutefois qu'il est tout à fait possible d'accéder aux objets.

Voici par exemple ce qui sera émis dans la console :

```
  --------------------
  #<Grob Accidental >
  #<Grob Arpeggio >
  #<Grob Stem >
```

```
#(define (notehead-get-accidental notehead)
   ;; notehead is grob
   (ly:grob-object notehead 'accidental-grob))

#(define (notehead-get-arpeggio notehead)
   ;; notehead is grob
```
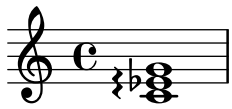
```
   (let ((notecolumn (notehead-get-notecolumn notehead)))
     (ly:grob-object notecolumn 'arpeggio)))

#(define (notehead-get-notecolumn notehead)
   ;; notehead is grob
   (ly:grob-parent notehead X))

#(define (notehead-get-stem notehead)
   ;; notehead is grob
   (let ((notecolumn (notehead-get-notecolumn notehead)))
     (ly:grob-object notecolumn 'stem)))

#(define (display-grobs notehead)
   ;; notehead is grob
   (let ((accidental (notehead-get-accidental notehead))
         (arpeggio (notehead-get-arpeggio notehead))
         (stem (notehead-get-stem notehead)))
     (format (current-error-port) "~2&~a\n" (make-string 20 #\-))
     (for-each
      (lambda (x) (format (current-error-port) "~a\n" x))
      (list accidental arpeggio stem))))

\relative c' {
  %% display grobs for each note head:
  %\override NoteHead.before-line-breaking = #display-grobs
  <c
  %% or just for one:
  \tweak before-line-breaking #display-grobs
  es
  g>1\arpeggio
}
```



Utilisation de PostScript pour générer des têtes de note à l'allure particulière
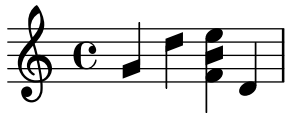
Lorsqu'il est impossible d'obtenir facilement une allure particulière pour les têtes de note en recourant à la technique du `\markup`, un code PostScript peut vous tirer d'embarras. Voici comment générer des têtes ressemblant à des parallélogrammes.

```
parallelogram =
  #(ly:make-stencil (list 'embedded-ps
    "gsave
      currentpoint translate
      newpath
      0 0.25 moveto
      1.3125 0.75 lineto
      1.3125 -0.25 lineto
      0 -0.75 lineto
      closepath
```

```
      fill
      grestore" )
    (cons 0 1.3125)
    (cons -.75 .75))

myNoteHeads = \override NoteHead.stencil = \parallelogram
normalNoteHeads = \revert NoteHead.stencil

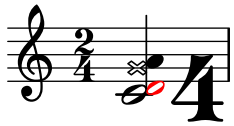
\relative c'' {
  \myNoteHeads
  g4 d'
  \normalNoteHeads
  <f, \tweak stencil \parallelogram b e>4 d
}
```



Utilisation de \tweak pour retoucher des objets particuliers

La commande `\tweak` permet de retoucher directement n'importe quel objet graphique. En voici quelques exemples :

```
\relative c' {
  \time 2/4
  \set fingeringOrientations = #'(right)
  <
    \tweak font-size #3 c
    \tweak color #red d-\tweak font-size #8 -4
    \tweak style #'cross g
    \tweak duration-log #2 a
  >2
}
```



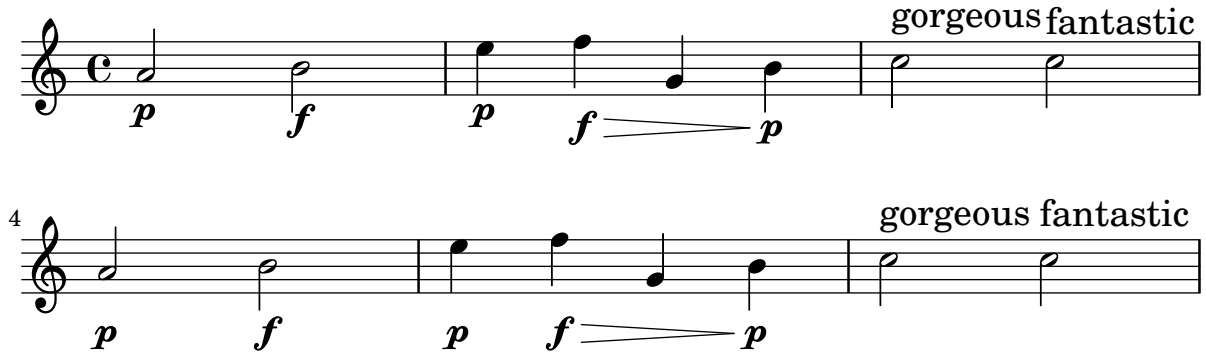
Alignement vertical des nuances indications textuelles

Tous les objets `DynamicLineSpanner` (soufflets ou nuances textuelles) viennent s'aligner sur une ligne de référence placée, par rapport à la portée, à au moins la valeur de `'staff-padding` sauf lorsque d'autres éléments de notation les en éloignent plus. Les nuances seront centrés sur une même ligne dès lors que `'staff-padding` aura été défini à une valeur suffisante.

C'est le même principe – en combinaison avec `\textLengthOn` – qui sert à aligner les indications textuelles sur une ligne de référence.

```
music = \relative c' {
  a'2\p b\f
  e4\p f\f\> g, b\p
  c2^\markup { \huge gorgeous } c^\markup { \huge fantastic }
}
```

```
{
  \music
  \break
  \override DynamicLineSpanner.staff-padding = #3
  \textLengthOn
  \override TextScript.staff-padding = #1
  \music
}
```



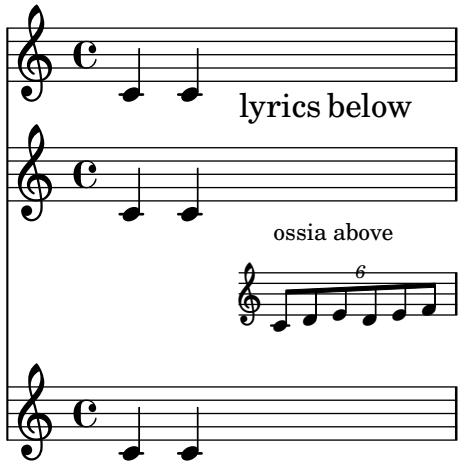
Positionnement d'une ossia et des paroles

Cet exemple illustre la manière de positionner une portée d'ossia et des paroles à l'aide des propriétés de contexte `alignBelowContext` et `alignAboveContext`.

```
\paper {
  ragged-right = ##t
}

\relative c' <<
  \new Staff = "1" { c4 c s2 }
  \new Staff = "2" { c4 c s2 }
  \new Staff = "3" { c4 c s2 }
  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = #"1"
        lyrics4 below
      }
      \new Staff \with {
        alignAboveContext = #"3"
        fontSize = #-2
        \override StaffSymbol.staff-space = #(magstep -2)
        \remove "Time_signature_engraver"
      } {
        \tuplet 6/4 {
          \override TextScript.padding = #3
          c8[^"ossia above" d e d e f]
        }
      }
    >>
  }
```

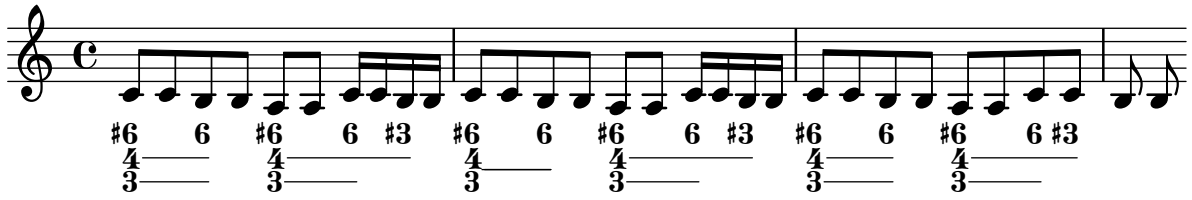
```
>>
```



Prolongateur commun de basse figurée

L'activation de la propriété `useBassFigureExtenders` permet d'afficher des lignes de prolongation pour les chiffres qui se répètent. Deux chiffres prolongés sur la même durée se verront affublés d'un unique prolongateur, verticalement centré entre eux, dès lors que la propriété `figuredBassCenterContinuations` aura elle aussi été activée.

```
<<
  \relative c' {
    c8 c b b a a c16 c b b
    c8 c b b a a c16 c b b
    c8 c b b a a c c b b
  }
  \figures {
    \set useBassFigureExtenders = ##t
    <6+ 4 3>4 <6 4 3>8 r
    <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
    \set figuredBassCenterContinuations = ##t
    <6+ 4 3>4 <6 4 3>8 r
    <6+ 4 3>4 <6 4 3>8 <4 3+>16 r
    \set figuredBassCenterContinuations = ##f
    <6+ 4 3>4 <6 4 3>8 r
    <6+ 4 3>4 <6 4 3>8 <4 3+>8
  }
>>
```



Paper and layout

Section “Spacing issues” dans *Manuel de notation*

Alignement des noms d’instrument

L’alignement horizontal des noms d’instrument se gère à l’aide de la propriété `Staff.InstrumentName.self-alignment-X`. Les variables `indent` et `short-indent`, attachées au bloc `\layout`, déterminent l’espace alloué à l’alignement des noms d’instrument, respectivement dans leurs formes développée et abrégée.

```
\paper { left-margin = 3\cm }

\score {
  \new StaffGroup <<

    \new Staff \with {
      \override InstrumentName.self-alignment-X = #LEFT
      instrumentName = \markup \left-column {
        "Left aligned"
        "instrument name"
        }
        shortInstrumentName = "Left"
      }

      {  c''1 \break c''1 }

    \new Staff \with {
      \override InstrumentName.self-alignment-X = #CENTER
      instrumentName = \markup \center-column {
        Centered
        "instrument name"
        }
      shortInstrumentName = "Centered"
    }

    { g'1 g'1}

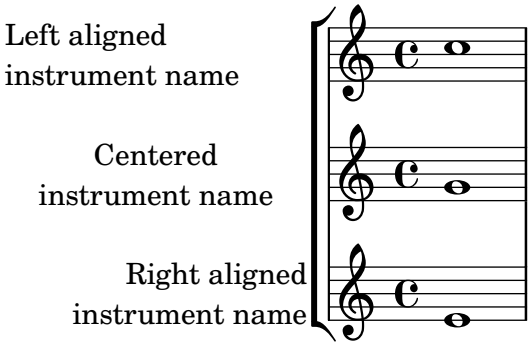
    \new Staff \with {
      \override InstrumentName.self-alignment-X = #RIGHT
      instrumentName = \markup \right-column {
        "Right aligned"
        "instrument name"
      }
      shortInstrumentName = "Right"
    }

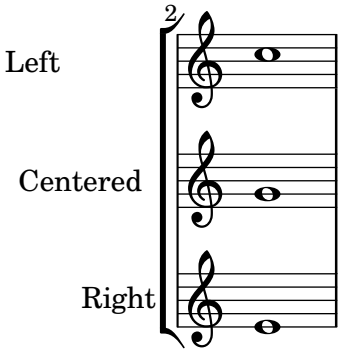
    { e'1 e'1 }

  >>

  \layout {
    ragged-right = ##t
```

```
    indent = 4\cm
    short-indent = 2\cm
  }
}
```





Agencement de paroles séparées sur une seule ligne

On peut vouloir positionner les paroles de différents intervenants sur une même ligne, notamment pour un dialogue par bribes. Cet extrait montre comment y parvenir avec l'instruction `\override VerticalAxisGroup.nonstaff-nonstaff-spacing.minimum-distance = ##f`.

```
\layout {
  \context {
    \Lyrics
    \override VerticalAxisGroup.nonstaff-nonstaff-spacing.minimum-distance = ##f
  }
}

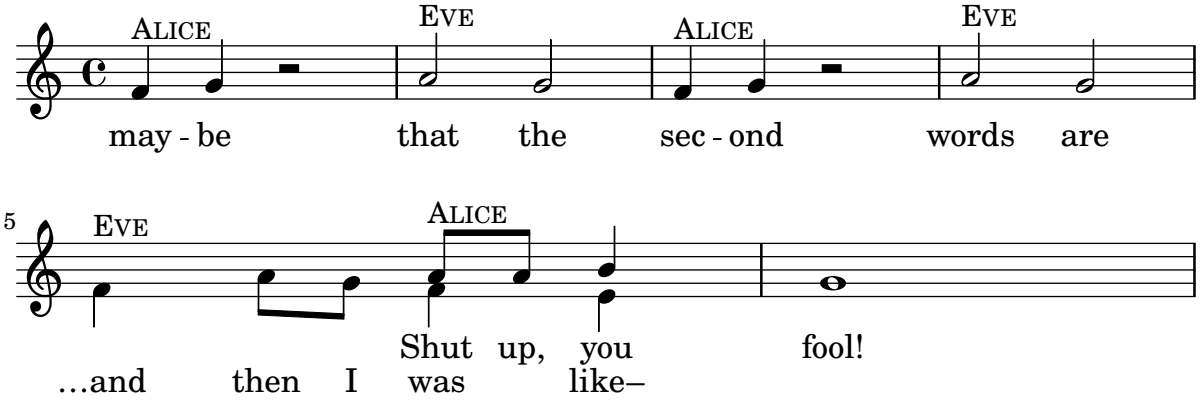
aliceSings = \markup { \smallCaps "Alice" }
eveSings = \markup { \smallCaps "Eve" }

<<
  \new Staff <<
    \new Voice = "alice" {
      f'4^\aliceSings g' r2 |
      s1 |
      f'4^\aliceSings g' r2 |
      s1 | \break
      % ...

      \voiceOne
```

```
      s2 a'8^\aliceSings a' b'4 |
      \oneVoice
      g'1
    }
    \new Voice = "eve" {
      s1 |
      a'2^\eveSings g' |
      s1 |
      a'2^\eveSings g'
      % ...

      \voiceTwo
      f'4^\eveSings a'8 g' f'4 e' |
      \oneVoice
      s1
    }
  >>
  \new Lyrics \lyricsto "alice" {
    may -- be
    sec -- ond
    % ...
    Shut up, you fool!
  }
  \new Lyrics \lyricsto "eve" {
    that the
    words are
    % ...
    ...and then I was like--
  }
>>
```



Partitionnement d'un ouvrage

`\bookpart` permet de découper un ouvrage en plusieurs parties. La dernière page de chaque partie peut être affectée d'un `ragged-last-bottom`. Les routines de formatage font la distinction entre les *markups* d'entête et de pied de page selon qu'ils se rattachent à une partie ou bien à l'ouvrage entier.

```
#(set-default-paper-size "a6")

\book {
```

```
  %% book paper, which is inherited by all children bookparts
  \paper {
    ragged-last-bottom = ##t
    %% Page footer: add a different part-tagline at part last page
    oddFooterMarkup = \markup {
      \column {
        \fill-line {
          %% Copyright header field only on book first page.
          \on-the-fly #first-page \fromproperty #'header:copyright
        }
        \fill-line {
          %% Part tagline header field only on each part last page.
          \on-the-fly #part-last-page \fromproperty #'header:parttagline
        }
        \fill-line {
          %% Tagline header field only on book last page.
          \on-the-fly #last-page \fromproperty #'header:tagline
        }
      }
    }
  }

  %% book header, which is inherited by the first bookpart
  \header {
    title = "Book title"
    copyright = "Copyright line on book first page"
    parttagline = "Part tagline"
    tagline = "Book tagline"
  }

  \bookpart {
    %% a different page breaking function may be used on each part
    \paper { page-breaking = #ly:minimal-breaking }
    \header { subtitle = "First part" }
    \markup { The first book part }
    \markup { a page break }
    \pageBreak
    \markup { first part last page }
    \markup \wordwrap { with ragged-last-bottom (see the space below this text) }
  }

  \bookpart {
    \header { subtitle = "Second part" }
    { c'4 }
  }

}
```

Book title

First part

The first book part

a page break

Copyright line on book first page
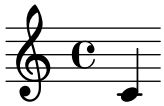
2
first part last page
with ragged-last-bottom (see the space below this text)

Part tagline

3

Book title

Second part



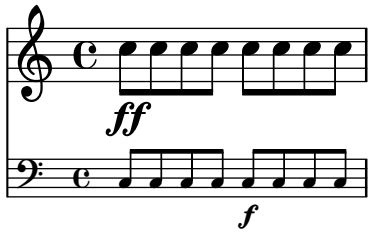
Part tagline

Book tagline

Modification de la taille d'une portée

Bien que le meilleur moyen de définir la taille des portées consiste à utiliser `#(set-global-staff-size xx)`, une portée en particulier peut se redimensionner en affectant d'un coefficient ses propriétés `staff-space` et `fontSize`.

```
<<
  \new Staff {
    \relative c'' {
      \dynamicDown
      c8\ff c c c c c c c
    }
  }
  \new Staff \with {
    fontSize = #-3
    \override StaffSymbol.staff-space = #(magstep -3)
  } {
    \clef bass
    c8 c c c c\f c c c
  }
>>
```



Découpe de systèmes

Le code ci-dessous permet de réaliser des découpes à l'emporte pièce, autrement dit d'extraire des fragments, au sein d'une partition.

Dans la mesure où le rendu inclus dans ce manuel pourrait ne pas être des plus évidents, nous vous conseillons de traiter ce fichier individuellement, avec l'option `-dclip-systems`.

Les fichiers résultants sont de la forme '`base-from-début-to-fin[-compteur].eps`'.

Lorsque sont inclus le début ou la fin d'un système, l'extraction comportera les objets rattachés au système, comme le nom d'instrument.

Les notes d'ornement qui se trouveraient en fin de fragment ne sont pas incluses.

La découpe peut couvrir plusieurs systèmes, auquel cas seront générés plusieurs fichiers EPS.

```
#(ly:set-option 'clip-systems)
#(define output-suffix "1")

origScore = \score {
  \relative c' {
    \new Staff \with { instrumentName = "Instrument" }
    c1
    d1
    \grace c16 e1
    \key d \major
    f1 \break
    \clef bass
    g,1
    fis1
  }
}

\book {
  \score {
    \origScore
    \layout {
      % Each clip-region is a (START . END) pair
      % where both are rhythmic-locations.

      % (make-rhythmic-locations BAR-NUMBER NUM DEN)
      % means NUM/DEN whole-notes into bar numbered BAR-NUMBER

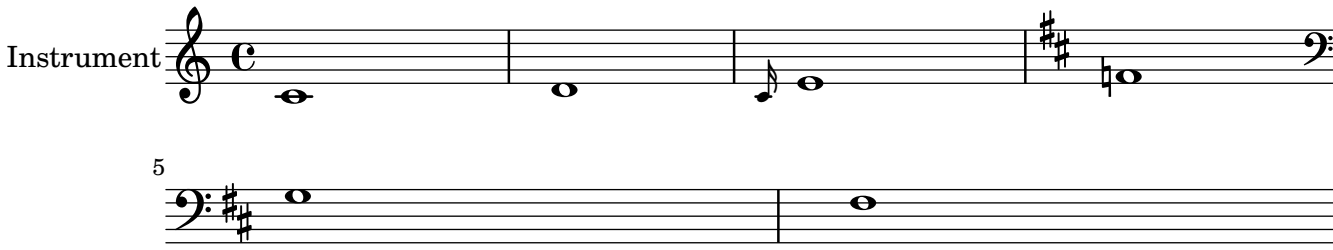
      clip-regions = #(list
      (cons
       (make-rhythmic-location 2 0 1)
       (make-rhythmic-location 4 0 1))

      (cons
       (make-rhythmic-location 0 0 1)
       (make-rhythmic-location 4 0 1))
```
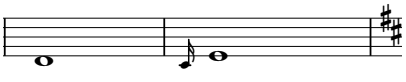
```
      (cons
       (make-rhythmic-location 0 0 1)
       (make-rhythmic-location 6 0 1))
      )
    }
  }
}

#(ly:set-option 'clip-systems #f)
#(define output-suffix #f)

\book {
  \score { \origScore }
  \markup { \bold \fontsize #6 clips }
  \score {
    \lyrics {
      \markup { from-2.0.1-to-4.0.1-clip.eps }
      \markup {
        \epsfile #X #30.0 #(format #f "~a-1-from-2.0.1-to-4.0.1-clip.eps"
                           (ly:parser-output-name)) }
    }
  }
}
```



clips

from-2.0.1-to-4.0.1-clip.eps 

Impression de papier à musique

Rien n'est plus simple, pour obtenir une feuille blanche avec des portées, que de générer des mesures vides, de supprimer le `Bar_number_engraver` du contexte `Score`, et les `Time_signature_engraver`, `Clef_engraver` et `Bar_engraver` du contexte `Staff`.

```
#(set-global-staff-size 20)

\score {
  {
    \repeat unfold 12 { s1 \break }
  }
  \layout {
    indent = 0\in
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Clef_engraver"
      \remove "Bar_engraver"
    }
    \context {
      \Score
      \remove "Bar_number_engraver"
    }
  }
}

% uncomment these lines for "letter" size
%{
\paper {
  #(set-paper-size "letter")
  ragged-last-bottom = ##f
  line-width = 7.5\in
  left-margin = 0.5\in
  bottom-margin = 0.25\in
  top-margin = 0.25\in
}
%}

% uncomment these lines for "A4" size
%{
\paper {
  #(set-paper-size "a4")
  ragged-last-bottom = ##f
  line-width = 180
  left-margin = 15
  bottom-margin = 10
  top-margin = 10
}
%}
```



Champs d'entête

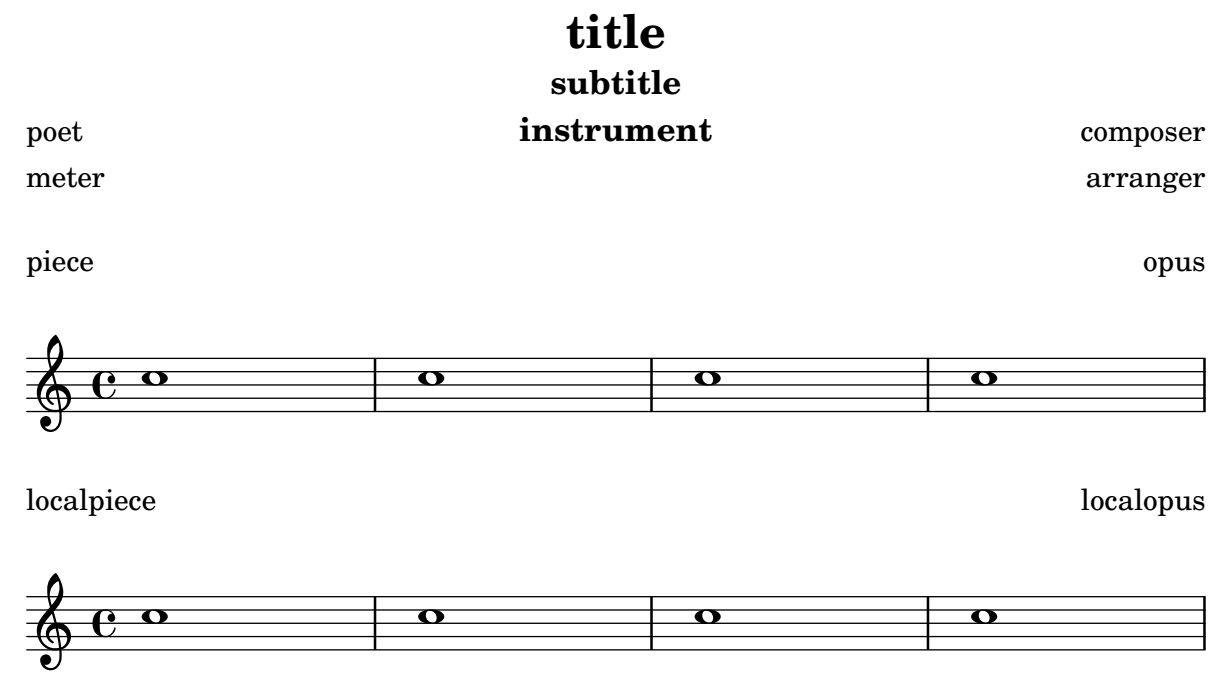
Voici la liste de tous les champs d'entête :

```
\header {
  copyright = "copyright"
  title = "title"
  subtitle = "subtitle"
  composer = "composer"
  arranger = "arranger"
  instrument = "instrument"
  meter = "meter"
  opus = "opus"
  piece = "piece"
  poet = "poet"
  texidoc = "All header fields with special meanings."
  copyright = "public domain"
  enteredby = "jcn"
  source = "urtext"
}
```

```
\layout {
  ragged-right = ##f
}

\score {
  \relative c'' { c1 | c | c | c }
}

\score {
   \relative c'' { c1 | c | c | c }
   \header {
     title = "localtitle"
     subtitle = "localsubtitle"
     composer = "localcomposer"
     arranger = "localarranger"
     instrument = "localinstrument"
     metre = "localmetre"
     opus = "localopus"
     piece = "localpiece"
     poet = "localpoet"
     copyright = "localcopyright"
   }
}
```



Séparation visuelle entre les systèmes

La séparation entre deux systèmes consécutifs peut être mise en évidence par n'importe quel *markup*. LilyPond dispose à cet effet d'une double oblique inversée : `\slashSeparator`.

```
\paper {
  system-separator-markup = \slashSeparator
  line-width = 120
}
```

```
notes = \relative c' {
  c1 | c \break
  c1 | c \break
  c1 | c
}

\book {
  \score {
    \new GrandStaff <<
      \new Staff \notes
      \new Staff \notes
    >>
  }
}
```

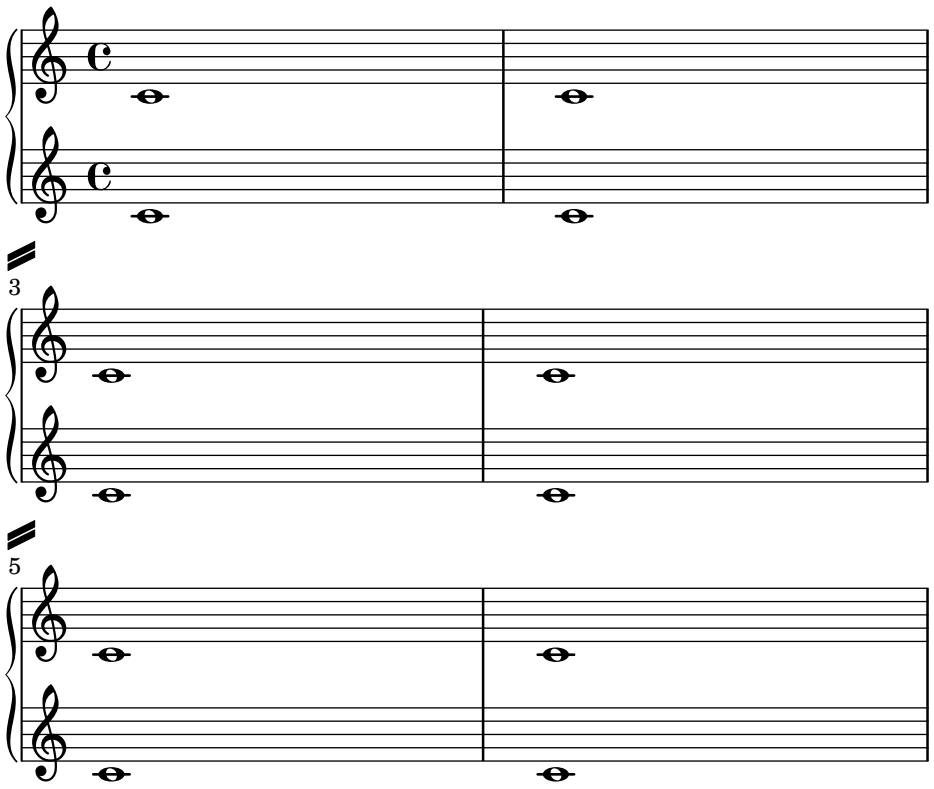

3
5

Table des matières

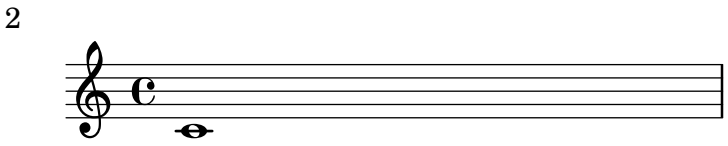
L'instruction `\markuplist \table-of-contents` permet d'inclure une table des matières. Ses éléments sont ajoutés à l'aide de la commande `\tocItem`.

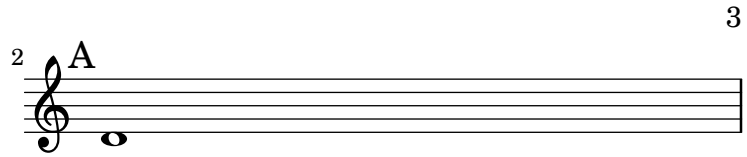
```
#(set-default-paper-size "a6")

\book {
  \markuplist \table-of-contents
  \pageBreak
  \tocItem \markup { The first score }
  \score {
    {
      c'1 \pageBreak
      \mark "A" \tocItem \markup { Mark A }
      d'1
    }
  }
  \pageBreak
  \tocItem \markup { The second score }
  \score {
    { e'1 }
    \header { piece = "Second score" }
  }
}
```

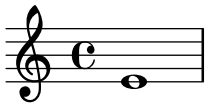
Table of Contents

The first score **2**
Mark A **3**
The second score **4**





4
Second score



Music engraving by LilyPond 2.23.2—www.lilypond.org

Alignement vertical de `StaffGroups` sans `SystemStartBar`

Cet extrait indique comment obtenir des regroupements `StaffGroup` alignés verticalement, disposant chacun de leur indication de groupe `SystemStartBar`, sans qu'il ne soient connectés entre eux.

```
% by Thomas Morley

#(set-global-staff-size 18)

\paper {
  indent = 0
  ragged-right = ##f
  print-all-headers = ##t
}

\layout {
  \context {
    \Staff
    \consists "Mark_engraver"
    \override RehearsalMark.self-alignment-X = #LEFT
  }
  \context {
```

```
    \StaffGroup
    systemStartDelimiterHierarchy =
      #'(SystemStartBrace (SystemStartBracket a b))
  }
  \context {
    \Score
    \override SystemStartBrace.style = #'bar-line
    \omit SystemStartBar
    \override SystemStartBrace.padding = #-0.1
    \override SystemStartBrace.thickness = #1.6
    \remove "Mark_engraver"
    \override StaffGrouper.staffgroup-staff-spacing.basic-distance = #15
  }
}

%%%% EXAMPLE

txt =
\lyricmode {
  Wer4 nur den lie -- ben Gott läßt wal2 -- ten4
  und4 hof -- fet auf ihn al -- le Zeit2.
}

% First StaffGroup "exercise"

eI =
\relative c' {
        \mark \markup {
                \bold Teacher:
                This is a simple setting of the choral. Please improve it.
                }
        \key a \minor
        \time 4/4
        \voiceOne

        \partial 4
        e4
        a b c b
        a b gis2
        e4\fermata g! g f
        e a a gis
        a2.\fermata
        \bar ":|."
}

eII =
\relative c' {
        \key a \minor
        \time 4/4
        \voiceTwo
        \partial 4
        c4
```

```
        e e e gis
        a f e2
        b4 b d d
        c c d d
        c2.
        \bar ":|."
}

eIII =
\relative c' {
        \key a \minor
        \time 4/4
        \clef bass
        \voiceOne

        \partial 4
        a4
        c b a b
        c d b2
        gis4 g g b
        c a f e
        e2.
}

eIV =
\relative c' {
        \key a \minor
        \time 4/4
        \clef bass
        \voiceTwo

        \partial 4
        a,4
        a' gis a e
        a, d e2
        e,4\fermata e' b g
        c f d e
        a,2.\fermata
        \bar ":|."
}

exercise =
\new StaffGroup = "exercise"
<<

  \new Staff
    <<
      \new Voice \eI
      \new Voice \eII
    >>

  \new Lyrics \txt
```

```
  \new Staff
    <<
      \new Voice \eIII
      \new Voice \eIV
    >>
>>

% Second StaffGRoup "simple Bach"

sbI =
\relative c' {
        \mark \markup { \bold" Pupil:" Here's my version! }
        \key a \minor
        \time 4/4
        \voiceOne

        \partial 4
        e4
        a b c b
        a b gis2
        e4\fermata g! g f
        e a a gis
        a2.\fermata
        \bar ":|."
}

sbII =
\relative c' {
        \key a \minor
        \time 4/4
        \voiceTwo
        \partial 4
        c8 d
        e4 e e8 f g4
        f f e2
        b4 b8 c d4 d
        e8 d c4 b8 c d4
        c2.
        \bar ":|."
}

sbIII =
\relative c' {
        \key a \minor
        \time 4/4
        \clef bass
        \voiceOne

        \partial 4
        a8 b
        c4 b a b8 c
```

```
        d4 d8 c b2
        gis4 g g8 a b4
        b a8 g f4 e
        e2.
}

sbIV =
\relative c' {
        \key a \minor
        \time 4/4
        \clef bass
        \voiceTwo

        \partial 4
        a,4
        a' gis a e
        f8 e d4 e2
        e,4\fermata e' b a8 g
        c4 f8 e d4 e
        a,2.\fermata
        \bar ":|."
}

simpleBach =
\new StaffGroup = "simple Bach"
<<

  \new Staff
    <<
      \new Voice \sbI
      \new Voice \sbII
    >>

  \new Lyrics \txt

  \new Staff
    <<
      \new Voice \sbIII
      \new Voice \sbIV
    >>
>>

% Third StaffGroup "chromatic Bach"

cbI =
\relative c' {
        \mark \markup {
          \bold "Teacher:"
          \column {
            "Well, you simply copied and transposed a version of J.S.Bach."
            "Do you know this one?"
          }
```

```
        }
        \key a \minor
        \time 4/4
        \voiceOne

        \partial 4
        e4
        a b c b
        a b gis4. fis8
        e4\fermata g! g f
        e a a8 b gis4
        a2.\fermata
        \bar ":|."
}

cbII =
\relative c' {
        \key a \minor
        \time 4/4
        \voiceTwo
        \partial 4
        c8 d
        e4 e e8 fis gis4
        a8 g! f!4 e2
        b4 e e d
        d8[ cis] d dis e fis e4
        e2.
        \bar ":|."
}

cbIII =
\relative c' {
        \key a \minor
        \time 4/4
        \clef bass
        \voiceOne

        \partial 4
        a8 b
        c[ b] a gis8 a4 d,
        e8[ e'] d c b4. a8
        gis4 b c d8 c
        b[ a] a b c b b c16 d
        c2.
}

cbIV =
\relative c' {
        \key a \minor
        \time 4/4
        \clef bass
        \voiceTwo
```

```
        \partial 4
        a4
        c, e a, b
        c d e2
        e4\fermata e a b8 c
        gis[ g] fis f e dis e4
        a,2.\fermata
        \bar ":|."
}

chromaticBach =
\new StaffGroup = "chromatic Bach"
<<

  \new Staff
    <<
      \new Voice \cbI
      \new Voice \cbII
    >>

  \new Lyrics \txt

  \new Staff
    <<
      \new Voice \cbIII
      \new Voice \cbIV
    >>
>>

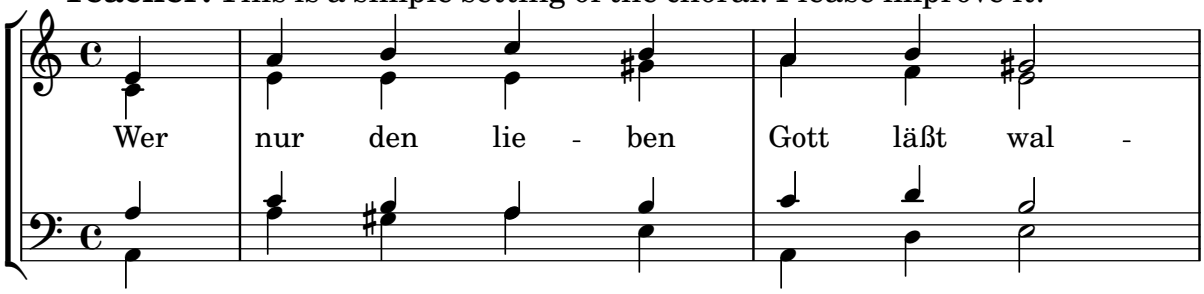

% Score

\score {
        <<
        \exercise
        \simpleBach
        \chromaticBach
        >>
        \header {
                title = \markup
                           \column {
                             \combine \null \vspace #1
                             "Exercise: Improve the given choral"
                             " "
                            }
        }
        \layout {
                \context {
                        \Lyrics
                        \override LyricText.X-offset = #-1
                }
```

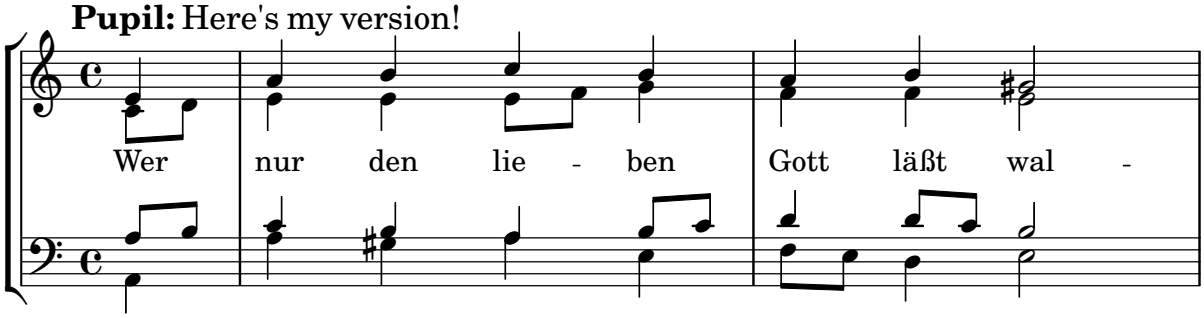
```
        }
}
```

Exercise: Improve the given choral

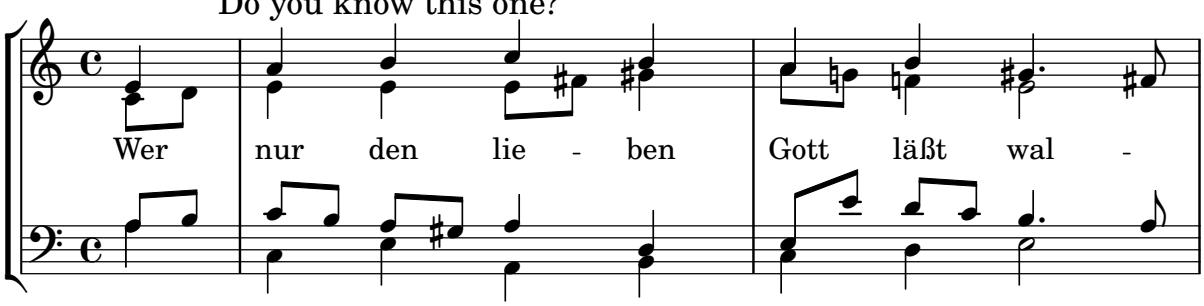
Teacher: This is a simple setting of the choral. Please improve it.
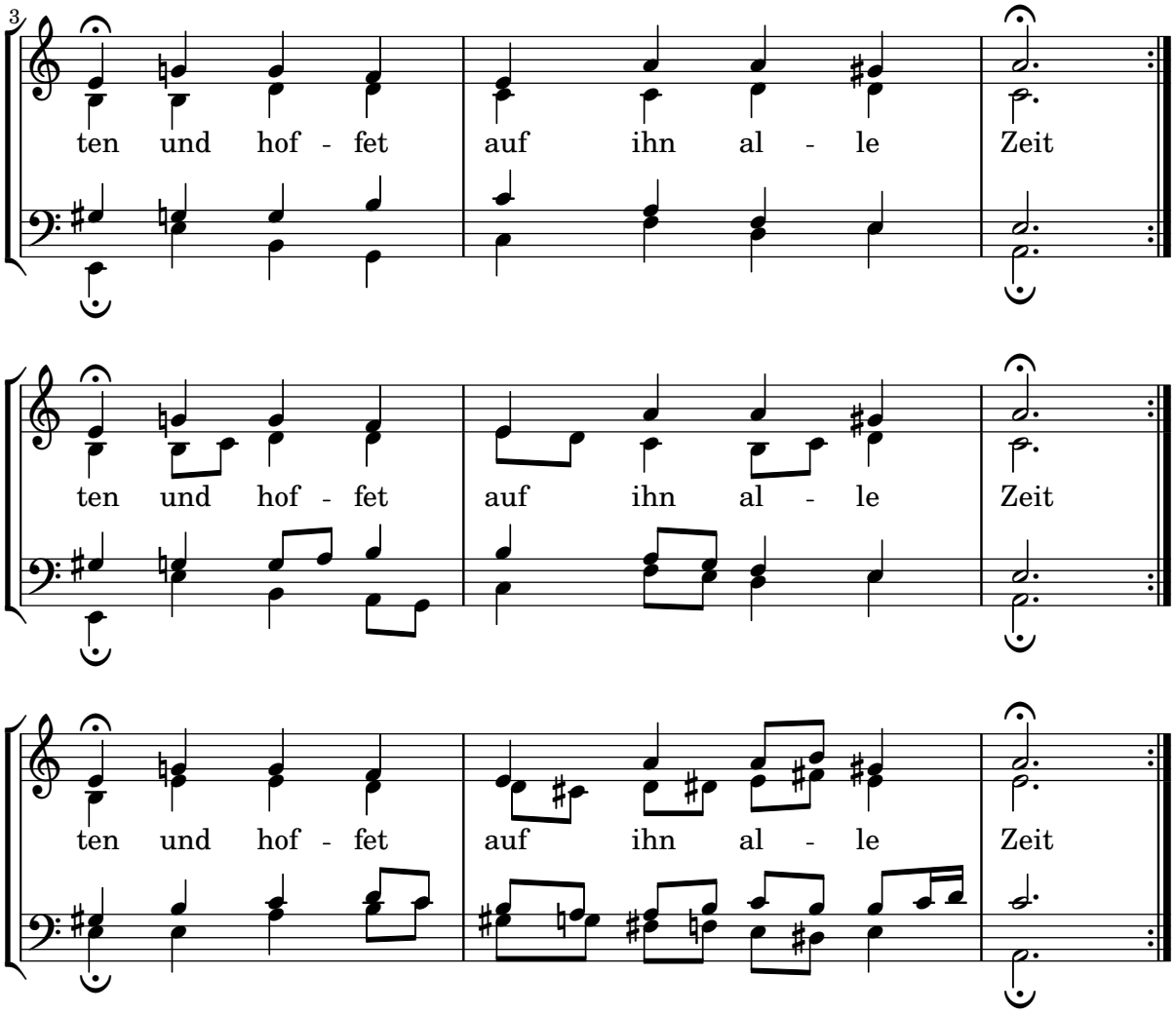


Pupil: Here's my version!



Teacher: Well, you simply copied and transposed a version of J.S.Bach. Do you know this one?




3
ten und hof - fet auf ihn al - le Zeit
ten und hof - fet auf ihn al - le Zeit
ten und hof - fet auf ihn al - le Zeit

Titles

Section “Titles and headers” dans *Manuel de notation*

Ajout de la date du jour à une partition

Avec un peu de code Scheme, voici comment ajouter facilement la date du jour à votre partition.
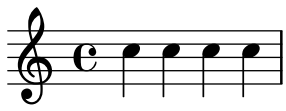
```
% first, define a variable to hold the formatted date:
date = #(strftime "%d-%m-%Y" (localtime (current-time)))

% use it in the title block:
\header {
  title = "Including the date!"
  subtitle = \date
}

\score {
  \relative c'' {
    c4 c c c
  }
}
% and use it in a \markup block:
\markup {
  \date
}
```

Including the date!

11-04-2021



11-04-2021

Alignement des noms d’instrument

L’alignement horizontal des noms d’instrument se gère à l’aide de la propriété `Staff.InstrumentName.self-alignment-X`. Les variables `indent` et `short-indent`, attachées au bloc `\layout`, déterminent l’espace alloué à l’alignement des noms d’instrument, respectivement dans leurs formes développée et abrégée.

```
\paper { left-margin = 3\cm }

\score {
  \new StaffGroup <<

    \new Staff \with {
      \override InstrumentName.self-alignment-X = #LEFT
      instrumentName = \markup \left-column {
        "Left aligned"
        "instrument name"
```

```
        }
        shortInstrumentName = "Left"
      }

      {  c''1 \break c''1 }

    \new Staff \with {
      \override InstrumentName.self-alignment-X = #CENTER
      instrumentName = \markup \center-column {
        Centered
        "instrument name"
        }
      shortInstrumentName = "Centered"
    }

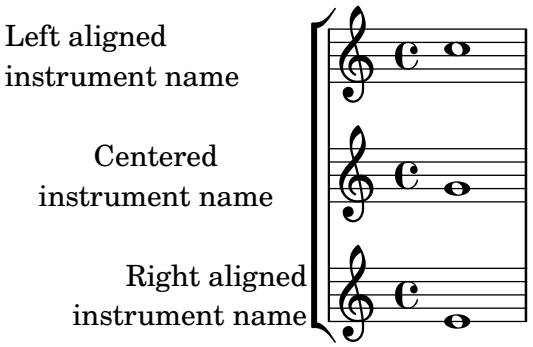
    { g'1 g'1}

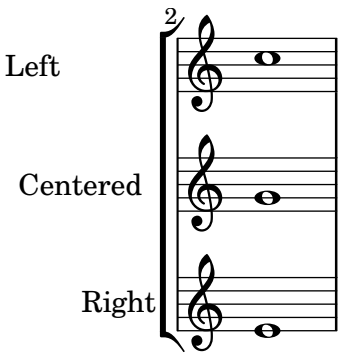
    \new Staff \with {
      \override InstrumentName.self-alignment-X = #RIGHT
      instrumentName = \markup \right-column {
        "Right aligned"
        "instrument name"
      }
      shortInstrumentName = "Right"
    }

    { e'1 e'1 }

  >>

  \layout {
    ragged-right = ##t
    indent = 4\cm
    short-indent = 2\cm
  }
}
```





Champs d'entête

Voici la liste de tous les champs d'entête :

```
\header {
  copyright = "copyright"
  title = "title"
  subtitle = "subtitle"
  composer = "composer"
  arranger = "arranger"
  instrument = "instrument"
  meter = "meter"
  opus = "opus"
  piece = "piece"
  poet = "poet"
  texidoc = "All header fields with special meanings."
  copyright = "public domain"
  enteredby = "jcn"
  source = "urtext"
}

\layout {
  ragged-right = ##f
}

\score {
  \relative c'' { c1 | c | c | c }
}

\score {
   \relative c'' { c1 | c | c | c }
   \header {
     title = "localtitle"
     subtitle = "localsubtitle"
     composer = "localcomposer"
     arranger = "localarranger"
     instrument = "localinstrument"
     metre = "localmetre"
     opus = "localopus"
     piece = "localpiece"
     poet = "localpoet"
     copyright = "localcopyright"
   }
}
```

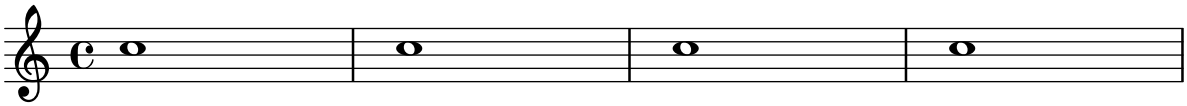
title

subtitle
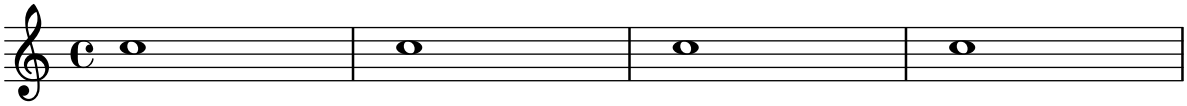
poet **instrument** composer

meter arranger

piece opus



localpiece localopus



Impression du numéro de version

L'insertion du résultat de `lilypond-version` dans des paroles permet d'ajouter à la partition ou au document généré par `lilypond-book` le numéro de la version de LilyPond utilisée. Une autre possibilité consiste à l'ajouter à une chaîne textuelle comme ci-dessous.

```
\score {
  \new Lyrics {
    \override Score.RehearsalMark.self-alignment-X = #LEFT
    \mark #(string-append "Processed with LilyPond version " (lilypond-version))
    s2
  }
}
```

Processed with LilyPond version 2.23.2

Spacing
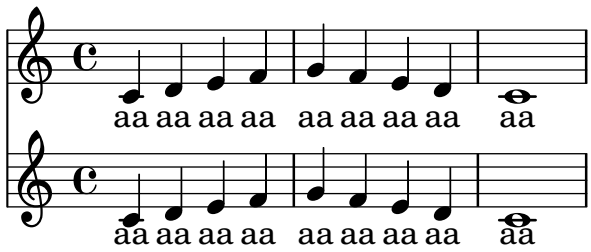
Section “Spacing issues” dans *Manuel de notation*

Ajustement de l’espacement vertical des paroles

Cet extrait illustre la manière de réduire l’espace entre la ligne de paroles et la portée.

```
% Default layout:
<<
  \new Staff \new Voice = melody \relative c' {
    c4 d e f
    g4 f e d
    c1
  }
  \new Lyrics \lyricsto melody { aa aa aa aa aa aa aa aa aa }

  \new Staff {
    \new Voice = melody \relative c' {
      c4 d e f
      g4 f e d
      c1
    }
  }
  % Reducing the minimum space below the staff and above the lyrics:
  \new Lyrics \with {
    \override VerticalAxisGroup.nonstaff-relatedstaff-spacing =
      #'((basic-distance . 1))
  }
  \lyricsto melody { aa aa aa aa aa aa aa aa aa }
>>
```
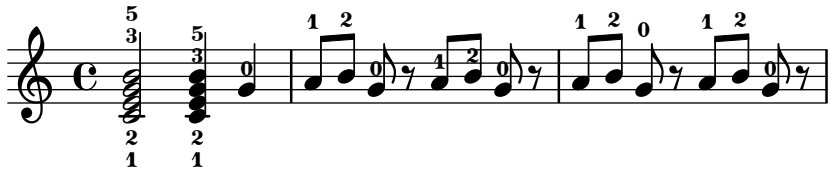


Impression des doigtés à l’intérieur de la portée

L’empilement des indications de doigté se fait par défaut à l’extérieur de la portée. Il est néanmoins possible d’annuler ce comportement. Une attention particulière doit toutefois être portée dans les cas où doigté et hampe vont dans la même direction : les indications de doigté n’évitent les hampe qu’en présence de ligature. Ce réglage peut s’adapter pour éviter toutes les hampes ou aucune d’elles. L’exemple suivant illustre ces deux options, ainsi que la manière de revenir au comportement par défaut.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering.staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 g'-0
  a8[-1 b]-2 g-0 r
```

```
  \override Fingering.add-stem-support = ##f
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = ##t
  a[-1 b]-2 g-0 r
  \override Fingering.add-stem-support = #only-if-beamed
  a[-1 b]-2 g-0 r
}
```



Référencement de page

Des références de page peuvent prendre place aussi bien dans la musique qu'en tête de partition, puis être reprises dans un *markup*.

```
#(set-default-paper-size "a6")

#(define-markup-command (toc-line layout props label text)
  (symbol? markup?)
  (interpret-markup layout props
   (markup #:fill-line (text #:page-ref label "8" "?"))))

\book {
  \markup \huge \fill-line { \null Title Page \null }

  \pageBreak

  \label #'toc
  \markup \column {
    \large \fill-line { \null Table of contents \null }
    \toc-line #'toc "Table of contents"
    \toc-line #'firstScore "First Score"
    \toc-line #'markA "Mark A"
    \toc-line #'markB "Mark B"
    \toc-line #'markC "Mark C"
    \toc-line #'unknown "Unknown label"
  }

  \pageBreak

  \label #'firstScore
  \score {
    \new Staff \relative c' {
      c2 c
      \mark \markup {
        A (page \concat { \page-ref #'markA "0" "?" ) }
      } \label #'markA
      c2 c
      \pageBreak
```

```
      \mark "B" \label #'markB
      d2 d
      d2 d
      \once \override Score.RehearsalMark.break-visibility =
        #begin-of-line-invisible
      \mark "C" \label #'markC
    }
    \header { piece = "First score" }
  }
}
```

Title Page

2

Table of contents

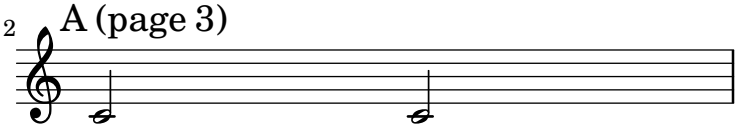
Table of contents 2
First Score 3
Mark A 3
Mark B 4
Mark C 4
Unknown label ?
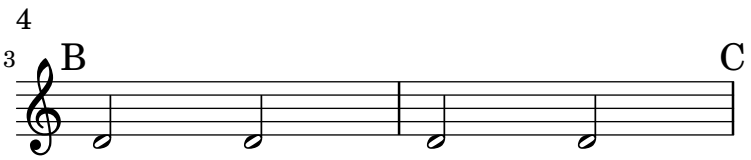
3

First score



 A (page 3)
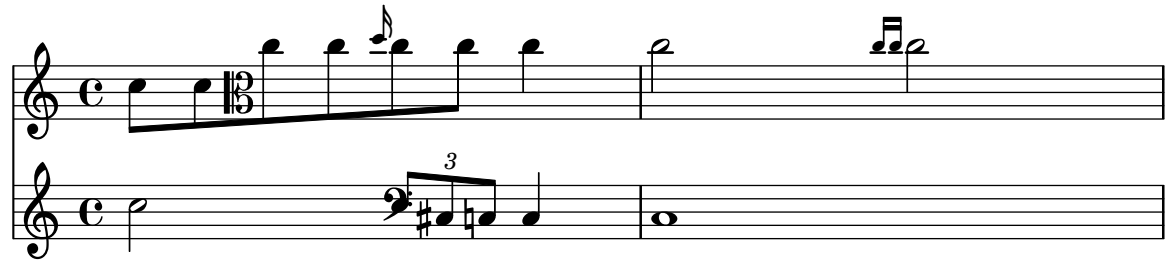




Music engraving by LilyPond 2.23.2—www.lilypond.org

Espacement strictement proportionnel des notes

Lorsque la propriété `strict-note-spacing` est activée, l'espacement des notes dans un système ne tient compte ni des barres de mesure ni des clefs, qui se retrouvent placées juste avant la note qui tombe au même moment. Ceci peut entraîner certaines collisions.

```
\relative c'' <<
  \override Score.SpacingSpanner.strict-note-spacing = ##t
  \set Score.proportionalNotationDuration = #(ly:make-moment 1/16)
  \new Staff {
    c8[ c \clef alto c c \grace { d16 } c8 c] c4
    c2 \grace { c16[ c16] } c2
  }
  \new Staff {
    c2 \tuplet 3/2 { c8 \clef bass cis,, c } c4
    c1
  }
>>
```



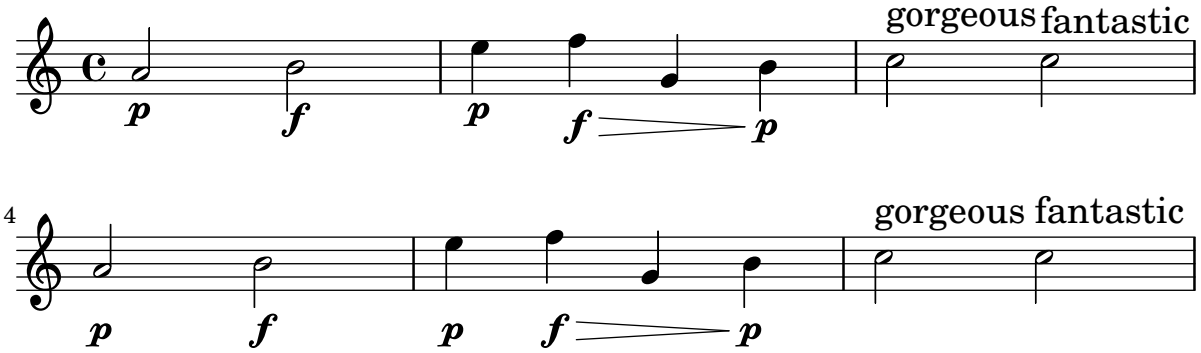
Alignement vertical des nuances indications textuelles

Tous les objets `DynamicLineSpanner` (soufflets ou nuances textuelles) viennent s'aligner sur une ligne de référence placée, par rapport à la portée, à au moins la valeur de `'staff-padding` sauf lorsque d'autres éléments de notation les en éloignent plus. Les nuances seront centrés sur une même ligne dès lors que `'staff-padding` aura été défini à une valeur suffisante.

C'est le même principe – en combinaison avec `\textLengthOn` – qui sert à aligner les indications textuelles sur une ligne de référence.

```
music = \relative c' {
  a'2\p b\f
  e4\p f\f\> g, b\p
  c2^\markup { \huge gorgeous } c^\markup { \huge fantastic }
}

{
  \music
  \break
  \override DynamicLineSpanner.staff-padding = #3
  \textLengthOn
  \override TextScript.staff-padding = #1
  \music
}
```



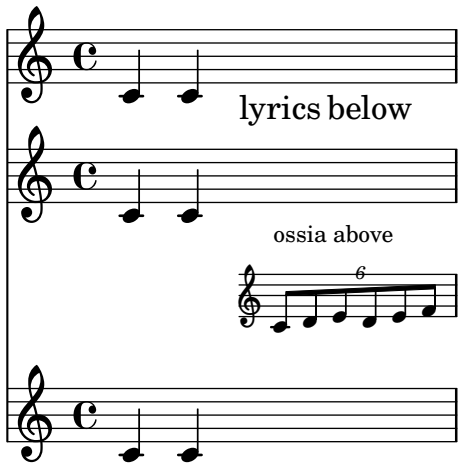
Positionnement d'une ossia et des paroles

Cet exemple illustre la manière de positionner une portée d'ossia et des paroles à l'aide des propriétés de contexte `alignBelowContext` et `alignAboveContext`.

```
\paper {
  ragged-right = ##t
}

\relative c' <<
  \new Staff = "1" { c4 c s2 }
  \new Staff = "2" { c4 c s2 }
  \new Staff = "3" { c4 c s2 }
```

```
  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = #"1"
        lyrics4 below
      }
      \new Staff \with {
        alignAboveContext = #"3"
        fontSize = #-2
        \override StaffSymbol.staff-space = #(magstep -2)
        \remove "Time_signature_engraver"
      } {
        \tuplet 6/4 {
          \override TextScript.padding = #3
          c8[^"ossia above" d e d e f]
        }
      }
    >>
  }
>>
```



MIDI
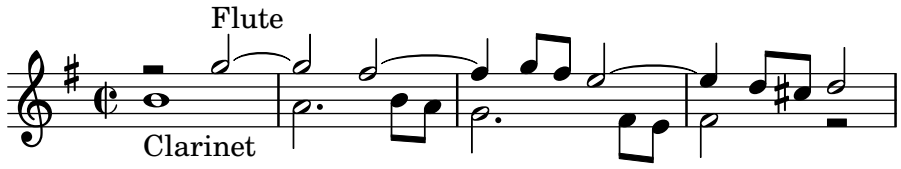
Section “Creating MIDI output” dans *Manuel de notation*

Affectation d’un canal MIDI par voix

Lorsque LilyPond génère un fichier MIDI, chaque portée sera par défaut affectée à un canal, quel que soit le nombre de voix qu’elle contient. Ceci permet d’éviter de se retrouver à court de canaux, sachant qu’il n’y en a que seize de disponibles par piste.

Le fait de déplacer le `Staff_performer` dans le contexte `Voice` permet d’affecter à chaque voix d’une même portée un canal MIDI spécifique. Dans l’exemple suivant, la même portée donnera lieu à deux canaux MIDI différents, chacun étant affecté de son propre `midiInstrument`.
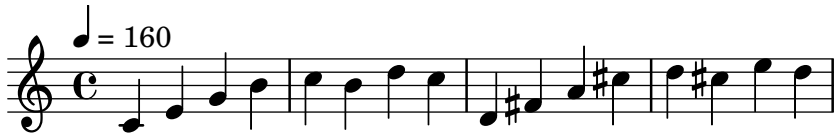
```
\score {
  \new Staff <<
    \new Voice \relative c''' {
      \set midiInstrument = #"flute"
      \voiceOne
      \key g \major
      \time 2/2
      r2 g-"Flute" ~
      g fis ~
      fis4 g8 fis e2 ~
      e4 d8 cis d2
    }
    \new Voice \relative c'' {
      \set midiInstrument = #"clarinet"
      \voiceTwo
      b1-"Clarinet"
      a2. b8 a
      g2. fis8 e
      fis2 r
    }
  >>
  \layout { }
  \midi {
    \context {
      \Staff
      \remove "Staff_performer"
    }
    \context {
      \Voice
      \consists "Staff_performer"
    }
    \tempo 2 = 72
  }
}
```



Changement de tempo sans indication sur la partition

Vous pouvez indiquer un changement de tempo pour le fichier MIDI sans pour autant l'imprimer. Il suffit alors de le rendre invisible aux musiciens.

```
\score {
  \new Staff \relative c' {
    \tempo 4 = 160
    c4 e g b
    c4 b d c
    \set Score.tempoHideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
  \layout { }
  \midi { }
}
```



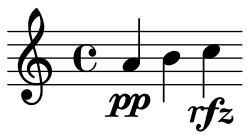
Création de nuance particulière pour la sortie MIDI

L'exemple suivant illustre la manière de créer une indication de nuance, absente de la liste par défaut, et de lui assigner une valeur spécifique utile à la sortie MIDI.

L'indication de nuance `\rfz` (*rinforzando*) se voit attribuer une valeur de `0.9`.

```
#(define (myDynamics dynamic)
    (if (equal? dynamic "rfz")
      0.9
      (default-dynamic-absolute-volume dynamic)))

\score {
  \new Staff {
    \set Staff.midiInstrument = #"cello"
    \set Score.dynamicAbsoluteVolumeFunction = #myDynamics
    \new Voice {
      \relative {
        a'4\pp b c-\rfz
      }
    }
  }
  \layout {}
  \midi {}
}
```



Personnalisation de batterie pour les sorties imprimable et MIDI

Des noms de « hauteurs » de batterie personnalisée peuvent s'utiliser à la fois pour la sortie imprimable et le fichier MIDI généré. Ces variables doivent être définies en tant que `drumPitchNames`, `drumStyleTable` et `midiDrumPitches`, comme indiqué ci-dessous. En résumé, cet extrait

– définit quelques noms de « hauteur »,
– définit comment ils seront rendus,
– indique à LilyPond de les utiliser pour la mise en forme,
– assigne les hauteurs aux noms,
– indique à LilyPond de les utiliser pour le rendu MIDI.

```
%% This snippet tries to amend
%% NR 2.5.1 Common notation for percussion - Custom percussion staves
%% http://lilypond.org/doc/v2.18/Documentation/notation/common-notation-for-percussion#custo

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% To use custom drum-pitch-names for your score and midi you need to follow
%% this route:
%%
%%%%%%%%%%%%
%% LAYOUT:
%%%%%%%%%%%%
%%
%% (1) Define a name and put it in `drumPitchNames'
%%     This can be done at toplevel with
%%         drumPitchNames.my-name  = #'my-name
%%     It's possible to add an alias as well.
%% (2) Define how it should be printed
%%     Therefore put them into a toplevel-list, where each entry should look:
%%         (my-name
%%           note-head-style-or-default
%%             articulation-string-or-#f
%%               staff-position)
%%     Example:
%%         #(define my-style
%%           '(
%%             (my-name default "tenuto" -1)
%%             ; ...
%%             ))
%% (3) Tell LilyPond to use this custom-definitions, with
%%         drumStyleTable = #(alist->hash-table my-style)
%%     in a \layout or \with
%%
%%     Now we're done for layout, here a short, but complete example:
%%         \new DrumStaff
%%           \with { drumStyleTable = #(alist->hash-table my-style) }
%%           \drummode { my-name }
%%
%%%%%%%%%%%%
%% MIDI:
%%%%%%%%%%%%
%%
```

```
%% (1) Again at toplvel, assign a pitch to your custom-note-name
%%         midiDrumPitches.my-name = ges
%%     Note that you have to use the name, which is in drumPitchNames, no alias
%% (2) Tell LilyPond to use this pitch(es), with
%%         drumPitchTable = #(alist->hash-table midiDrumPitches)
%%
%%     Example:
%%         \score {
%%            \new DrumStaff
%%             \with {
%%               drumStyleTable = #(alist->hash-table my-style)
%%               drumPitchTable = #(alist->hash-table midiDrumPitches)
%%             }
%%             \drummode { my-name4 }
%%           \layout {}
%%           \midi {}
%%         }
%%
%%%%%%%%%%%%
%% TESTING
%%%%%%%%%%%%
%%
%% To test whether all is fine, run the following sequence in terminal:
%%         lilypond my-file.ly
%%         midi2ly my-file.midi
%%         gedit my-file-midi.ly
%%
%% Which will do:
%% 1. create pdf and midi
%% 2. transform the midi back to a .ly-file
%%    (note: midi2ly is not always good in correctly identifying enharmonic pitches)
%% 3. open this file in gedit (or use another editor)
%% Now watch what you've got.
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% FULL EXAMPLE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
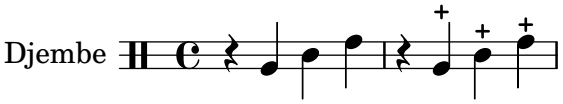
drumPitchNames.dbass      = #'dbass
drumPitchNames.dba        = #'dbass  % 'db is in use already
drumPitchNames.dbassmute  = #'dbassmute
drumPitchNames.dbm        = #'dbassmute
drumPitchNames.do         = #'dopen
drumPitchNames.dopenmute  = #'dopenmute
drumPitchNames.dom        = #'dopenmute
drumPitchNames.dslap      = #'dslap
drumPitchNames.ds         = #'dslap
drumPitchNames.dslapmute  = #'dslapmute
drumPitchNames.dsm        = #'dslapmute
```

```
#(define djembe
  '((dbass      default  #f         -2)
    (dbassmute  default  "stopped"  -2)
    (dopen      default  #f          0)
    (dopenmute  default  "stopped"   0)
    (dslap      default  #f          2)
    (dslapmute  default  "stopped"   2)))

midiDrumPitches.dbass = g
midiDrumPitches.dbassmute = fis
midiDrumPitches.dopen =  a
midiDrumPitches.dopenmute = gis
midiDrumPitches.dslap =  b
midiDrumPitches.dslapmute = ais

one = \drummode { r4 dba4 do ds r dbm dom dsm }

\score {
   \new DrumStaff
    \with {
      \override StaffSymbol.line-count =  #3
      instrumentName = #"Djembe "
      drumStyleTable = #(alist->hash-table djembe)
      drumPitchTable = #(alist->hash-table midiDrumPitches)
    }
    \one
  \layout {}
  \midi {}
}
```



Démonstration de `midiInstrument`

Partant du problème que l'on ignore quel `midiInstrument` serait le plus adapté à sa composition, voici un fichier LilyPond de démonstration.

```
\header {
  title = "Demo of all midi sounds"
  arranger = "Myself "
}

baseMelody = \relative c' {
  c4.\mf g c16 b' c d
  e16 d e f g4 g'4 r
  R1
}
melody = {
  \tempo 4 = 150
  \baseMelody
}
```

```
\score {
  \new Staff <<
    \new Voice \melody
  >>
  \layout { }
}

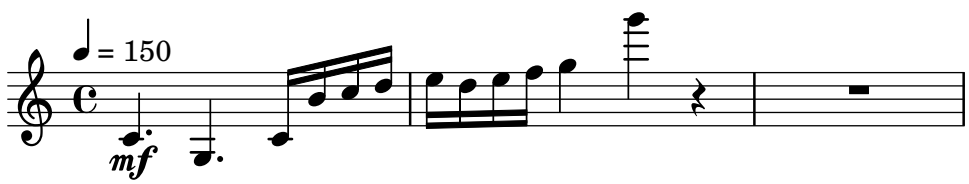
\score {
  \new Staff <<
    \new Voice {
      r\mf
      \set Staff.midiInstrument = #"acoustic grand" \melody
      \set Staff.midiInstrument = #"bright acoustic" \melody
      \set Staff.midiInstrument = #"electric grand" \melody
      \set Staff.midiInstrument = #"honky-tonk" \melody
      \set Staff.midiInstrument = #"electric piano 1" \melody
      \set Staff.midiInstrument = #"electric piano 2" \melody
      \set Staff.midiInstrument = #"harpsichord" \melody
      \set Staff.midiInstrument = #"clav" \melody
      \set Staff.midiInstrument = #"celesta" \melody
      \set Staff.midiInstrument = #"glockenspiel" \melody
      \set Staff.midiInstrument = #"music box" \melody
      \set Staff.midiInstrument = #"vibraphone" \melody
      \set Staff.midiInstrument = #"marimba" \melody
      \set Staff.midiInstrument = #"xylophone" \melody
      \set Staff.midiInstrument = #"tubular bells" \melody
      \set Staff.midiInstrument = #"dulcimer" \melody
      \set Staff.midiInstrument = #"drawbar organ" \melody
      \set Staff.midiInstrument = #"percussive organ" \melody
      \set Staff.midiInstrument = #"rock organ" \melody
      \set Staff.midiInstrument = #"church organ" \melody
      \set Staff.midiInstrument = #"reed organ" \melody
      \set Staff.midiInstrument = #"accordion" \melody
      \set Staff.midiInstrument = #"harmonica" \melody
      \set Staff.midiInstrument = #"concertina" \melody
      \set Staff.midiInstrument = #"acoustic guitar (nylon)" \melody
      \set Staff.midiInstrument = #"acoustic guitar (steel)" \melody
      \set Staff.midiInstrument = #"electric guitar (jazz)" \melody
      \set Staff.midiInstrument = #"electric guitar (clean)" \melody
      \set Staff.midiInstrument = #"electric guitar (muted)" \melody
      \set Staff.midiInstrument = #"overdriven guitar" \melody
      \set Staff.midiInstrument = #"distorted guitar" \melody
      \set Staff.midiInstrument = #"acoustic bass" \melody
      \set Staff.midiInstrument = #"electric bass (finger)" \melody
      \set Staff.midiInstrument = #"electric bass (pick)" \melody
      \set Staff.midiInstrument = #"fretless bass" \melody
      \set Staff.midiInstrument = #"slap bass 1" \melody
      \set Staff.midiInstrument = #"slap bass 2" \melody
      \set Staff.midiInstrument = #"synth bass 1" \melody
      \set Staff.midiInstrument = #"synth bass 2" \melody
      \set Staff.midiInstrument = #"violin" \melody
```

```
\set Staff.midiInstrument = #"viola" \melody
\set Staff.midiInstrument = #"cello" \melody
\set Staff.midiInstrument = #"contrabass" \melody
\set Staff.midiInstrument = #"tremolo strings" \melody
\set Staff.midiInstrument = #"pizzicato strings" \melody
\set Staff.midiInstrument = #"orchestral harp" \melody
\set Staff.midiInstrument = #"timpani" \melody
\set Staff.midiInstrument = #"string ensemble 1" \melody
\set Staff.midiInstrument = #"string ensemble 2" \melody
\set Staff.midiInstrument = #"synthstrings 1" \melody
\set Staff.midiInstrument = #"synthstrings 2" \melody
\set Staff.midiInstrument = #"choir aahs" \melody
\set Staff.midiInstrument = #"voice oohs" \melody
\set Staff.midiInstrument = #"synth voice" \melody
\set Staff.midiInstrument = #"orchestra hit" \melody
\set Staff.midiInstrument = #"trumpet" \melody
\set Staff.midiInstrument = #"trombone" \melody
\set Staff.midiInstrument = #"tuba" \melody
\set Staff.midiInstrument = #"muted trumpet" \melody
\set Staff.midiInstrument = #"french horn" \melody
\set Staff.midiInstrument = #"brass section" \melody
\set Staff.midiInstrument = #"synthbrass 1" \melody
\set Staff.midiInstrument = #"synthbrass 2" \melody
\set Staff.midiInstrument = #"soprano sax" \melody
\set Staff.midiInstrument = #"alto sax" \melody
\set Staff.midiInstrument = #"tenor sax" \melody
\set Staff.midiInstrument = #"baritone sax" \melody
\set Staff.midiInstrument = #"oboe" \melody
\set Staff.midiInstrument = #"english horn" \melody
\set Staff.midiInstrument = #"bassoon" \melody
\set Staff.midiInstrument = #"clarinet" \melody
\set Staff.midiInstrument = #"piccolo" \melody
\set Staff.midiInstrument = #"flute" \melody
\set Staff.midiInstrument = #"recorder" \melody
\set Staff.midiInstrument = #"pan flute" \melody
\set Staff.midiInstrument = #"blown bottle" \melody
\set Staff.midiInstrument = #"shakuhachi" \melody
\set Staff.midiInstrument = #"whistle" \melody
\set Staff.midiInstrument = #"ocarina" \melody
\set Staff.midiInstrument = #"lead 1 (square)" \melody
\set Staff.midiInstrument = #"lead 2 (sawtooth)" \melody
\set Staff.midiInstrument = #"lead 3 (calliope)" \melody
\set Staff.midiInstrument = #"lead 4 (chiff)" \melody
\set Staff.midiInstrument = #"lead 5 (charang)" \melody
\set Staff.midiInstrument = #"lead 6 (voice)" \melody
\set Staff.midiInstrument = #"lead 7 (fifths)" \melody
\set Staff.midiInstrument = #"lead 8 (bass+lead)" \melody
\set Staff.midiInstrument = #"pad 1 (new age)" \melody
\set Staff.midiInstrument = #"pad 2 (warm)" \melody
\set Staff.midiInstrument = #"pad 3 (polysynth)" \melody
\set Staff.midiInstrument = #"pad 4 (choir)" \melody
\set Staff.midiInstrument = #"pad 5 (bowed)" \melody
```

```
      \set Staff.midiInstrument = #"pad 6 (metallic)" \melody
      \set Staff.midiInstrument = #"pad 7 (halo)" \melody
      \set Staff.midiInstrument = #"pad 8 (sweep)" \melody
      \set Staff.midiInstrument = #"fx 1 (rain)" \melody
      \set Staff.midiInstrument = #"fx 2 (soundtrack)" \melody
      \set Staff.midiInstrument = #"fx 3 (crystal)" \melody
      \set Staff.midiInstrument = #"fx 4 (atmosphere)" \melody
      \set Staff.midiInstrument = #"fx 5 (brightness)" \melody
      \set Staff.midiInstrument = #"fx 6 (goblins)" \melody
      \set Staff.midiInstrument = #"fx 7 (echoes)" \melody
      \set Staff.midiInstrument = #"fx 8 (sci-fi)" \melody
      \set Staff.midiInstrument = #"sitar" \melody
      \set Staff.midiInstrument = #"banjo" \melody
      \set Staff.midiInstrument = #"shamisen" \melody
      \set Staff.midiInstrument = #"koto" \melody
      \set Staff.midiInstrument = #"kalimba" \melody
      \set Staff.midiInstrument = #"bagpipe" \melody
      \set Staff.midiInstrument = #"fiddle" \melody
      \set Staff.midiInstrument = #"shanai" \melody
      \set Staff.midiInstrument = #"tinkle bell" \melody
      \set Staff.midiInstrument = #"agogo" \melody
      \set Staff.midiInstrument = #"steel drums" \melody
      \set Staff.midiInstrument = #"woodblock" \melody
      \set Staff.midiInstrument = #"taiko drum" \melody
      \set Staff.midiInstrument = #"melodic tom" \melody
      \set Staff.midiInstrument = #"synth drum" \melody
      \set Staff.midiInstrument = #"reverse cymbal" \melody
      \set Staff.midiInstrument = #"guitar fret noise" \melody
      \set Staff.midiInstrument = #"breath noise" \melody
      \set Staff.midiInstrument = #"seashore" \melody
      \set Staff.midiInstrument = #"bird tweet" \melody
      \set Staff.midiInstrument = #"telephone ring" \melody
      \set Staff.midiInstrument = #"helicopter" \melody
      \set Staff.midiInstrument = #"applause" \melody
      \set Staff.midiInstrument = #"gunshot" \melody
    }
  >>
  \midi { }
}
```

Demo of all midi sounds

Myself



Réglage de l'égalisation par défaut des instruments MIDI

L'égaliseur basique peut être modifié par la définition d'une nouvelle procédure Scheme `instrumentEqualizer` au sein du contexte `Score`. Cette procédure prend en unique argument le nom d'un instrument MIDI et renverra une paire de fractions correspondant aux minimum et maximum de volume alloué à cet instrument.
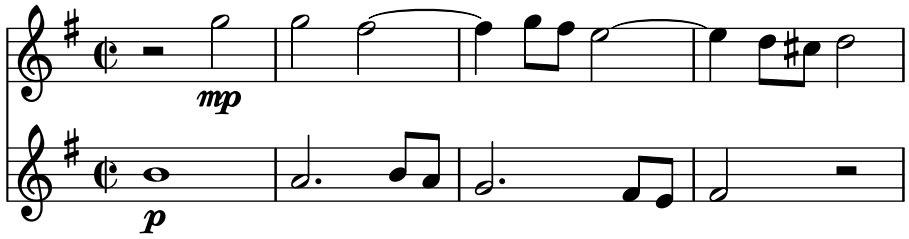
Dans l'exemple suivant sont réglés les volumes relatifs de la flûte et de la clarinette.

```
#(define my-instrument-equalizer-alist '())

#(set! my-instrument-equalizer-alist
  (append
    '(
      ("flute" . (0.7 . 0.9))
      ("clarinet" . (0.3 . 0.6)))
    my-instrument-equalizer-alist))

#(define (my-instrument-equalizer s)
  (let ((entry (assoc s my-instrument-equalizer-alist)))
    (if entry
      (cdr entry))))

\score {
  <<
    \new Staff {
      \key g \major
      \time 2/2
      \set Score.instrumentEqualizer = #my-instrument-equalizer
      \set Staff.midiInstrument = "flute"
      \new Voice \relative {
        r2 g''\mp g fis~
        4 g8 fis e2~
        4 d8 cis d2
      }
    }
    \new Staff {
      \key g \major
      \set Staff.midiInstrument = "clarinet"
      \new Voice \relative {
        b'1\p a2. b8 a
        g2. fis8 e
        fis2 r
      }
    }
  >>
  \layout { }
  \midi {  }
}
```


mp
p

Templates

Exemples de notation ancienne – transcription moderne de musique grégorienne
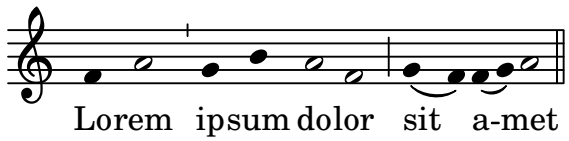
Voici comment vous pourriez transcrire du grégorien. Pour mémoire, il n'y a en grégorien ni découpage en mesure, ni hampe ; seules sont utilisées des têtes de note blanches ou noires, ainsi que des signes spécifiques permettant d'indiquer des silences de différentes durées.

```
\include "gregorian.ly"

chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g) a2 \finalis
}

verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met
}

\score {
  \new Staff <<
    \new Voice = "melody" \chant
    \new Lyrics = "one" \lyricsto melody \verba
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Bar_engraver"
      \hide Stem
    }
    \context {
      \Voice
      \override Stem.length = #0
    }
    \context {
      \Score
      barAlways = ##t
    }
  }
}
```



Psalmodie anglicane

Cet exemple illustre la manière de présenter un cantique tel qu'on le trouve dans l'église anglicane. Vous noterez comment sont ajoutés les couplets indépendamment de la musique. Dans le but de vous montrer plusieurs styles, comparez le code des deux couplets.

```
SopranoMusic = \relative g' {
  g1 | c2 b | a1 | \bar "||"
  a1 | d2 c | c b | c1 | \bar "||"
}

AltoMusic = \relative c' {
  e1 | g2 g | f1 |
  f1 | f2 e | d d | e1 |
}

TenorMusic = \relative a {
  c1 | c2 c | c1 |
  d1 | g,2 g | g g | g1 |
}

BassMusic =  \relative c {
  c1 | e2 e | f1 |
  d1 | b2 c | g' g | c,1 |
}

global = {
  \time 2/2
}

dot = \markup {
  \raise #0.7 \musicglyph "dots.dot"
}

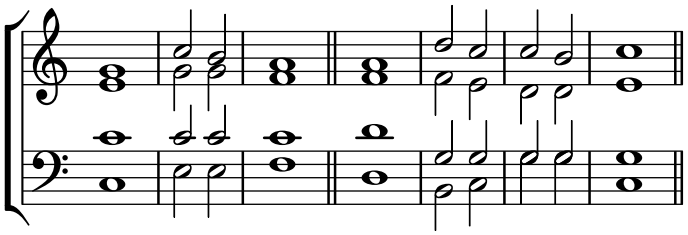
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph "scripts.rvarcomma"
}

% Use markup to center the chant on the page
\markup {
  \fill-line {
    \score {  % centered
      <<
        \new ChoirStaff <<
          \new Staff <<
            \global
            \clef "treble"
            \new Voice = "Soprano" <<
              \voiceOne
              \SopranoMusic
            >>
            \new Voice = "Alto" <<
              \voiceTwo
```

```
            \AltoMusic
          >>
        >>
        \new Staff <<
          \clef "bass"
          \global
          \new Voice = "Tenor" <<
            \voiceOne
            \TenorMusic
          >>
          \new Voice = "Bass" <<
            \voiceTwo
            \BassMusic
          >>
        >>
      >>
      \layout {
        \context {
          \Score
          \override SpacingSpanner.base-shortest-duration = #(ly:make-moment 1/2)
        }
        \context {
          \Staff
          \remove "Time_signature_engraver"
        }
      }
    }  % End score
  }
}  % End markup

\markup {
  \fill-line {
    \column {
      \left-align {
        \null \null \null
        \line {
          \fontsize #5 O
          \fontsize #3 come
          let us \bold sing | unto \dot the | Lord : let
        }
        \line {
          us heartily
          \concat { re \bold joice }
          in the | strength of | our
        }
        \line {
          sal | vation.
        }
        \null
        \line {
          \hspace #2.5 8. Today if ye will hear his voice *
```

```
        }
        \line {
          \concat { \bold hard en }
          \tick not your \tick hearts : as in the pro-
        }
        \line {
          vocation * and as in the \bold day of tempt- \tick
        }
        \line {
          -ation \tick in the \tick wilderness.
        }
      }
    }
  }
}
```



O come let us **sing** | unto • the | Lord : let us heartily re**joice** in the | strength of | our sal | vation.

8. Today if ye will hear his voice * **hard**en ' not your ' hearts : as in the provocation * and as in the **day** of tempt- ' -ation ' in the ' wilderness.

Modèle pour cantique

Le code ci-dessous illustre la manière d'agencer un cantique liturgique dans lequel chaque ligne débute et se termine par une mesure incomplète. Vous noterez par ailleurs l'affichage des paroles indépendamment de la musique.

```
Timeline = {
  \time 4/4
  \tempo 4=96
  \partial 2
  s2 | s1 | s2 \breathe s2 | s1 | s2 \bar "||" \break
  s2 | s1 | s2 \breathe s2 | s1 | s2 \bar "||"
}

SopranoMusic = \relative g' {
  g4 g | g g g g | g g g g | g g g g | g2
```

```
  g4 g | g g g g | g g g g | g g g g | g2
}

AltoMusic = \relative c' {
  d4 d | d d d d | d d d d | d d d d | d2
  d4 d | d d d d | d d d d | d d d d | d2
}

TenorMusic = \relative a {
  b4 b | b b b b | b b b b | b b b b | b2
  b4 b | b b b b | b b b b | b b b b | b2
}

BassMusic =  \relative g {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

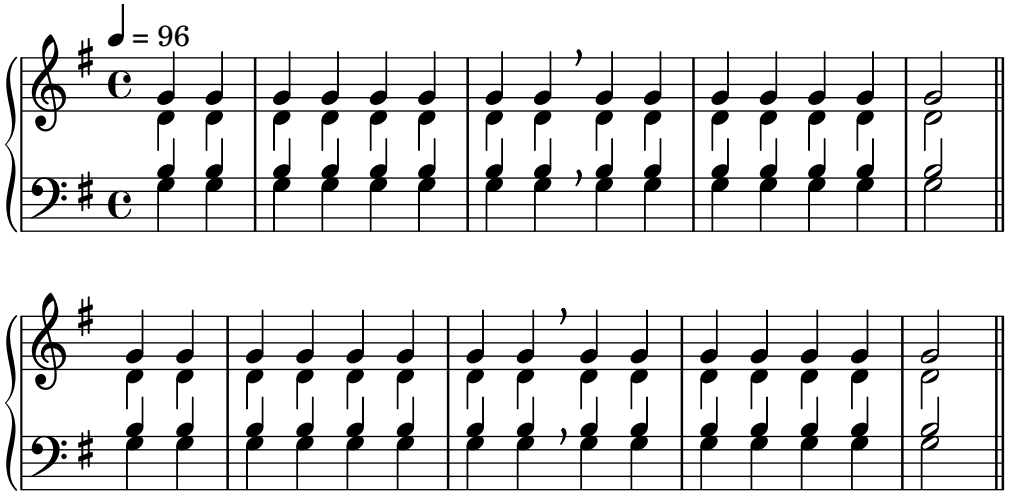
global = {
 \key g \major
}

\score {  % Start score
  <<
    \new PianoStaff <<  % Start pianostaff
      \new Staff <<  % Start Staff = RH
        \global
        \clef "treble"
        \new Voice = "Soprano" <<  % Start Voice = "Soprano"
          \Timeline
          \voiceOne
          \SopranoMusic
        >>  % End Voice = "Soprano"
        \new Voice = "Alto" <<  % Start Voice = "Alto"
          \Timeline
          \voiceTwo
          \AltoMusic
        >>  % End Voice = "Alto"
      >>  % End Staff = RH
      \new Staff <<  % Start Staff = LH
        \global
        \clef "bass"
        \new Voice = "Tenor" <<  % Start Voice = "Tenor"
          \Timeline
          \voiceOne
          \TenorMusic
        >>  % End Voice = "Tenor"
        \new Voice = "Bass" <<  % Start Voice = "Bass"
          \Timeline
          \voiceTwo
          \BassMusic
        >>  % End Voice = "Bass"
```

```
      >>  % End Staff = LH
    >>  % End pianostaff
  >>
}  % End score

\markup {
  \fill-line {
    ""
    {
      \column {
        \left-align {
          "This is line one of the first verse"
          "This is line two of the same"
          "And here's line three of the first verse"
          "And the last line of the same"
        }
      }
    }
    ""
  }
}

\paper {  % Start paper block
  indent = 0     % don't indent first system
  line-width = 130   % shorten line length to suit music
}  % End paper block
```



This is line one of the first verse
This is line two of the same
And here's line three of the first verse
And the last line of the same

Symboles de jazz

Bien que compliqué de prime abord, voici un canevas tout à fait indiqué pour les ensembles de jazz. Vous noterez que tous les instruments sont notés en ut (`\key c \major`), la tonalité de

concert. Les notes seront automatiquement transposées dès lors qu'elles seront inscrites dans une section `\transpose`.

```
\header {
  title = "Song"
  subtitle = "(tune)"
  composer = "Me"
  meter = "moderato"
  piece = "Swing"
  tagline = \markup {
    \column {
      "LilyPond example file by Amelie Zapf,"
      "Berlin 07/07/2003"
    }
  }
}
% To make the example display in the documentation
\paper {
  paper-width = 130
}
%#(set-global-staff-size 16)
\include "english.ly"

%%%%%%%%%%% Some macros %%%%%%%%%%%%%%%%%%

sl = {
  \override NoteHead.style = #'slash
  \hide Stem
}
nsl = {
  \revert NoteHead.style
  \undo \hide Stem
}
crOn = \override NoteHead.style = #'cross
crOff = \revert NoteHead.style

%% insert chord name style stuff here.

jazzChords = { }

%%%%%%%%%%% Keys'n'thangs %%%%%%%%%%%%%%%%

global = { \time 4/4 }

Key = { \key c \major }

% ############ Horns ############

% ------ Trumpet ------
trpt = \transpose c d \relative c'' {
  \Key
  c1 | c | c |
```

```
}
trpHarmony = \transpose c' d {
  \jazzChords
}
trumpet = {
  \global
  \clef treble
  <<
    \trpt
  >>
}

% ------ Alto Saxophone ------
alto = \transpose c a \relative c' {
  \Key
  c1 | c | c |
}
altoHarmony = \transpose c' a {
  \jazzChords
}
altoSax = {
  \global
  \clef treble
  <<
    \alto
  >>
}

% ------ Baritone Saxophone ------
bari = \transpose c a' \relative c {
  \Key
  c1
  c1
  \sl
  d4^"Solo" d d d
  \nsl
}
bariHarmony = \transpose c' a \chordmode {
  \jazzChords s1 s d2:maj e:m7
}
bariSax = {
  \global
  \clef treble
  <<
    \bari
  >>
}

% ------ Trombone ------
tbone = \relative c {
  \Key
  c1 | c | c
```

```
}
tboneHarmony = \chordmode {
  \jazzChords
}
trombone = {
  \global
  \clef bass
  <<
    \tbone
  >>
}

% ############ Rhythm Section #############

% ------ Guitar ------
gtr = \relative c'' {
  \Key
  c1
  \sl
  b4 b b b
  \nsl
  c1
}
gtrHarmony = \chordmode {
  \jazzChords
  s1 c2:min7+ d2:maj9
}
guitar = {
  \global
  \clef treble
  <<
    \gtr
  >>
}

%% ------ Piano ------
rhUpper = \relative c'' {
  \voiceOne
  \Key
  c1 | c | c
}
rhLower = \relative c' {
  \voiceTwo
  \Key
  e1 | e | e
}

lhUpper = \relative c' {
  \voiceOne
  \Key
  g1 | g | g
}
```

```
lhLower = \relative c {
  \voiceTwo
  \Key
  c1 | c | c
}

PianoRH = {
  \clef treble
  \global
  <<
    \new Voice = "one" \rhUpper
    \new Voice = "two" \rhLower
  >>
}
PianoLH = {
  \clef bass
  \global
  <<
    \new Voice = "one" \lhUpper
    \new Voice = "two" \lhLower
  >>
}

piano = {
  <<
    \new Staff = "upper" \PianoRH
    \new Staff = "lower" \PianoLH
  >>
}

% ------ Bass Guitar ------
Bass = \relative c {
  \Key
  c1 | c | c
}
bass = {
  \global
  \clef bass
  <<
    \Bass
  >>
}

% ------ Drums ------
up = \drummode {
  \voiceOne
  hh4 <hh sn> hh <hh sn>
  hh4 <hh sn> hh <hh sn>
  hh4 <hh sn> hh <hh sn>
}
down = \drummode {
  \voiceTwo
```

```
  bd4 s bd s
  bd4 s bd s
  bd4 s bd s
}

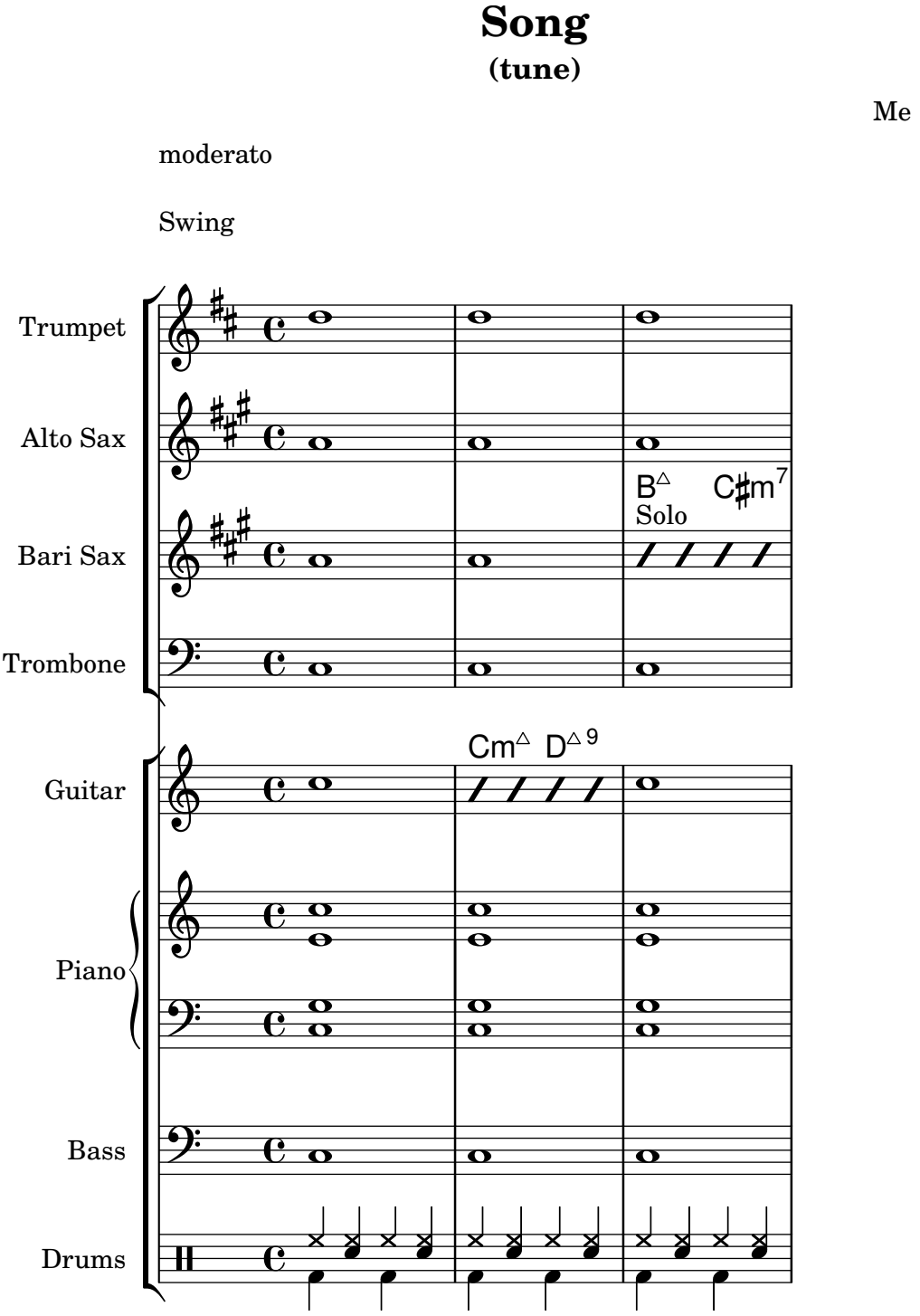
drumContents = {
  \global
  <<
    \new DrumVoice \up
    \new DrumVoice \down
  >>
}

%%%%%%%% It All Goes Together Here %%%%%%%%%%%%%%%%%%%%

\score {
  <<
    \new StaffGroup = "horns" <<
      \new Staff = "trumpet" \with { instrumentName = "Trumpet" }
      \trumpet
      \new Staff = "altosax" \with { instrumentName = "Alto Sax" }
      \altoSax
      \new ChordNames = "barichords" \with { instrumentName = "Trumpet" }
      \bariHarmony
      \new Staff = "barisax" \with { instrumentName = "Bari Sax" }
      \bariSax
      \new Staff = "trombone" \with { instrumentName = "Trombone" }
      \trombone
    >>

    \new StaffGroup = "rhythm" <<
      \new ChordNames = "chords" \gtrHarmony
      \new Staff = "guitar" \with { instrumentName = "Guitar" }
      \guitar
      \new PianoStaff = "piano" \with {
        instrumentName = "Piano"
        midiInstrument = "acoustic grand"
      }
      \piano
      \new Staff = "bass" \with { instrumentName = "Bass" }
      \bass
      \new DrumStaff \with { instrumentName = "Drums" }
      \drumContents
    >>
  >>
  \layout {
    \context { \Staff \RemoveEmptyStaves }
    \context {
      \Score
      \override BarNumber.padding = #3
      \override RehearsalMark.padding = #2
      skipBars = ##t
```

```
    }
  }
  \midi { }
}
```



Orchestre, chœur et piano

Ce canevas illustre l'utilisation de contextes `StaffGroup` pour regrouper les instruments selon leur famille, imbriqués dans un `GrandStaff`, ainsi que le recours à la fonction `\transpose` pour les instruments transpositeurs. Dans tous les identificateurs, la musique est stockée en ut. Les

notes peuvent tout aussi bien être saisies en ut ou dans la tonalité particulière de l'instrument avant d'être transposées puis affectées à une variable.

```
#(set-global-staff-size 17)
\paper {
  indent = 3.0\cm  % add space for instrumentName
  short-indent = 1.5\cm  % add less space for shortInstrumentName
}

fluteMusic = \relative c' { \key g \major g'1 b }

% Pitches as written on a manuscript for Clarinet in A
% are transposed to concert pitch.

clarinetMusic = \transpose c' a
  \relative c'' { \key bes \major bes1 d }

trumpetMusic = \relative c { \key g \major g''1 b }

% Key signature is often omitted for horns

hornMusic = \transpose c' f
  \relative c { d'1 fis }

percussionMusic = \relative c { \key g \major g1 b }

sopranoMusic = \relative c'' { \key g \major g'1 b }

sopranoLyrics = \lyricmode { Lyr -- ics }

altoIMusic = \relative c' { \key g \major g'1 b }

altoIIMusic = \relative c' { \key g \major g'1 b }

altoILyrics =  \sopranoLyrics

altoIILyrics = \lyricmode { Ah -- ah }

tenorMusic = \relative c' { \clef "treble_8" \key g \major g1 b }

tenorLyrics = \sopranoLyrics

pianoRHMusic = \relative c { \key g \major g''1 b }

pianoLHMusic = \relative c { \clef bass \key g \major g1 b }

violinIMusic = \relative c' { \key g \major g'1 b }

violinIIMusic = \relative c' { \key g \major g'1 b }

violaMusic = \relative c { \clef alto \key g \major g'1 b }
```

```
celloMusic = \relative c { \clef bass \key g \major g1 b }

bassMusic = \relative c { \clef "bass_8" \key g \major g,1 b }

\score {
  <<
    \new StaffGroup = "StaffGroup_woodwinds" <<
      \new Staff = "Staff_flute" \with { instrumentName = "Flute" }
      \fluteMusic

      \new Staff = "Staff_clarinet" \with {
        instrumentName = \markup { \concat { "Clarinet in B" \flat } }
      }

      % Declare that written Middle C in the music
      % to follow sounds a concert B flat, for
      % output using sounded pitches such as MIDI.
      %\transposition bes

      % Print music for a B-flat clarinet
      \transpose bes c' \clarinetMusic
    >>

    \new StaffGroup = "StaffGroup_brass" <<
      \new Staff = "Staff_hornI" \with { instrumentName = "Horn in F" }
       % \transposition f
        \transpose f c' \hornMusic

      \new Staff = "Staff_trumpet" \with { instrumentName = "Trumpet in  C" }
      \trumpetMusic

    >>
    \new RhythmicStaff = "RhythmicStaff_percussion"
    \with { instrumentName = "Percussion" }
    <<
      \percussionMusic
    >>
    \new PianoStaff \with { instrumentName = "Piano" }
    <<
      \new Staff { \pianoRHMusic }
      \new Staff { \pianoLHMusic }
    >>
    \new ChoirStaff = "ChoirStaff_choir" <<
      \new Staff = "Staff_soprano" \with { instrumentName = "Soprano" }
      \new Voice = "soprano"
      \sopranoMusic

      \new Lyrics \lyricsto "soprano" { \sopranoLyrics }
      \new GrandStaff = "GrandStaff_altos"
      \with { \accepts Lyrics } <<
        \new Staff = "Staff_altoI"  \with { instrumentName = "Alto I" }
        \new Voice = "altoI"
```

```
        \altoIMusic

        \new Lyrics \lyricsto "altoI" { \altoILyrics }
        \new Staff = "Staff_altoII" \with { instrumentName = "Alto II" }
        \new Voice = "altoII"
        \altoIIMusic

        \new Lyrics \lyricsto "altoII" { \altoIILyrics }
      >>

      \new Staff = "Staff_tenor" \with { instrumentName = "Tenor" }
        \new Voice = "tenor"
        \tenorMusic

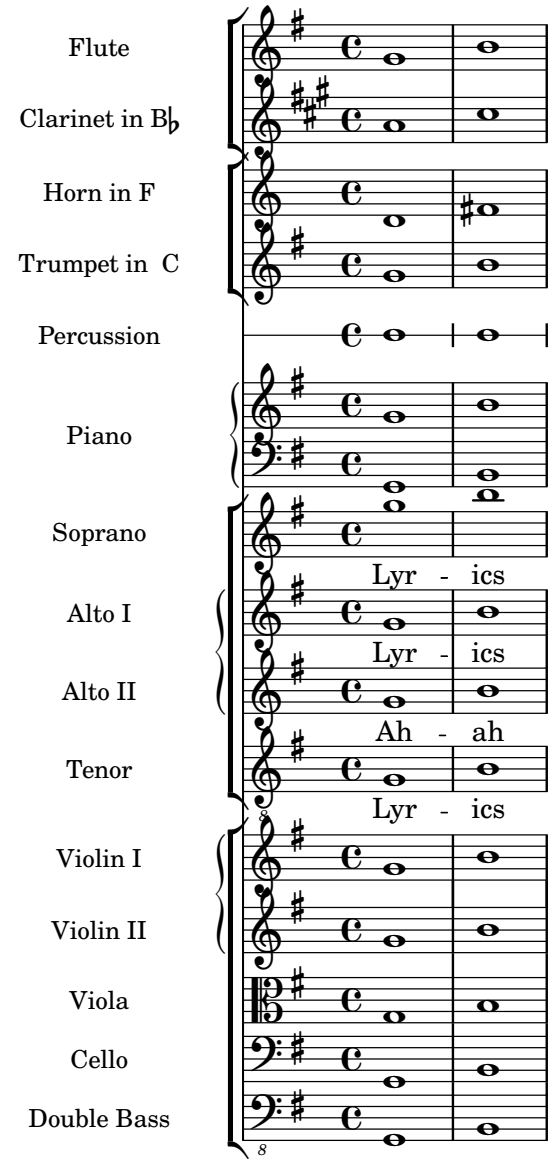
      \new Lyrics \lyricsto "tenor" { \tenorLyrics }
    >>
    \new StaffGroup = "StaffGroup_strings" <<
      \new GrandStaff = "GrandStaff_violins" <<
        \new Staff = "Staff_violinI" \with { instrumentName = "Violin I" }
        \violinIMusic

        \new Staff = "Staff_violinII" \with { instrumentName = "Violin II" }
        \violinIIMusic
      >>

      \new Staff = "Staff_viola" \with { instrumentName = "Viola" }
      \violaMusic

      \new Staff = "Staff_cello" \with { instrumentName = "Cello" }
      \celloMusic

      \new Staff = "Staff_bass" \with { instrumentName = "Double Bass" }
      \bassMusic
    >>
  >>
  \layout { }
}
```



Piano – cannevas simple

Voici une simple partition pour piano avec quelques notes.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4
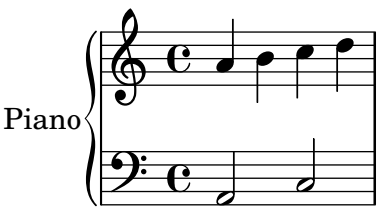
  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

\score {
```

```
  \new PianoStaff \with { instrumentName = "Piano" }
  <<
    \new Staff = "upper" \upper
    \new Staff = "lower" \lower
  >>
  \layout { }
  \midi { }
}
```



Piano et paroles entre les portées

Lorsque la mélodie est doublée au piano, cela ne nécessite pas forcément une portée spécifique. Les paroles peuvent s'insérer entre les deux portées de la partition pour piano.

```
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

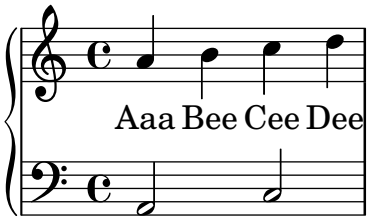
  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

\score {
  \new PianoStaff <<
    \new Staff = upper { \new Voice = "singer" \upper }
    \new Lyrics \lyricsto "singer" \text
    \new Staff = lower { \lower }
  >>
  \layout { }
  \midi { }
}
```



Piano, mélodie et paroles

Il s'agit du format classique pour le chant : une portée pour la mélodie et les paroles au-dessus de l'accompagnement au piano.

```
melody = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

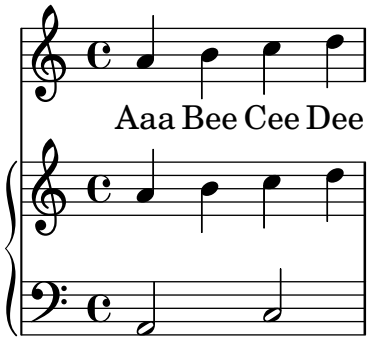
upper = \relative c'' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

lower = \relative c {
  \clef bass
  \key c \major
  \time 4/4

  a2 c
}

\score {
  <<
    \new Voice = "mel" { \autoBeamOff \melody }
    \new Lyrics \lyricsto mel \text
    \new PianoStaff <<
      \new Staff = "upper" \upper
      \new Staff = "lower" \lower
    >>
  >>
  \layout {
    \context { \Staff \RemoveEmptyStaves }
  }
  \midi { }
}
```
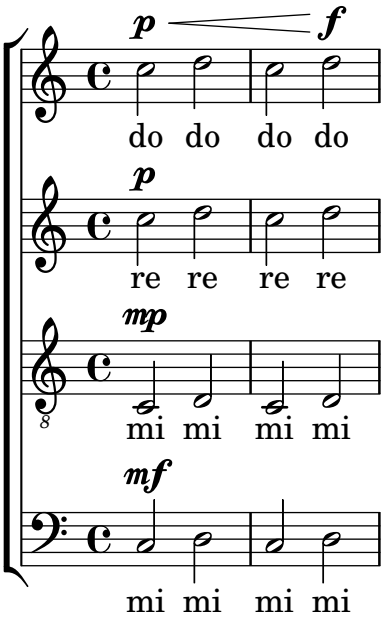


Modèle pour chœur SATB, sur quatre portées

Modèle pour chœur à quatre voix mixtes, chaque pupitre ayant sa propre portée.

```
global = {
  \key c \major
  \time 4/4
  \dynamicUp
}
sopranonotes = \relative c'' {
  c2 \p \< d c d \f
}
sopranowords = \lyricmode { do do do do }
altonotes = \relative c'' {
  c2\p d c d
}
altowords = \lyricmode { re re re re }
tenornotes = {
  \clef "G_8"
  c2\mp d c d
}
tenorwords = \lyricmode { mi mi mi mi }
bassnotes = {
  \clef bass
  c2\mf d c d
}
basswords = \lyricmode { mi mi mi mi }

\score {
  \new ChoirStaff <<
    \new Staff <<
      \new Voice = "soprano" <<
        \global
        \sopranonotes
      >>
      \new Lyrics \lyricsto "soprano" \sopranowords
    >>
    \new Staff <<
      \new Voice = "alto" <<
        \global
        \altonotes
      >>
      \new Lyrics \lyricsto "alto" \altowords
    >>
    \new Staff <<
```

```
      \new Voice = "tenor" <<
        \global
        \tenornotes
      >>
      \new Lyrics \lyricsto "tenor" \tenorwords
    >>
    \new Staff <<
      \new Voice = "bass" <<
        \global
        \bassnotes
      >>
      \new Lyrics \lyricsto "bass" \basswords
    >>
  >>
}
```



Score for diatonic accordion

A template to write a score for a diatonic accordion.

- There is a horizontal staff indicating if the accordion must be pushed (thick line) or pulled (thin line)

- There is a small rhythmic staff with lyrics that describes the bass buttons to press. The bar lines are made from gridlines

- The tabulator staff for diatonic accordions shows the geographic position of the buttons and not (as for every other instrument) the pitch of the notes; the keys on the melody-side of the accordion are placed in three columns and about 12 rows

In the tabulator staff notation the outermost column is described with notes between lines, the innermost column is described with notes between lines and a cross as accidental, and the middle column is described with notes on a line, whereby the row in the middle is represented on the middle line in the staff.

Some words to transpose piano notes to the diatonic accordion:

1. Every diatonic accordion is built for some keys only (for example, for the keys of C major and F major), so it is important to transpose a piano melody to match one of these keys.

Transpose the source code, not only the output because this code is required later on to translate it once more to the tabulator staff. This can be done with the command `displayLilyMusic`.

2. You have to alternate the push- and pull-direction of the accordion regularly. If the player has a too long part to pull the accordion gets broken. On the other hand, some harmonies are only available in one direction. Considering this, decide which parts of the melody are the push-parts and which the pull-parts.

3. For each pull- or push-part translate the piano notes to the according tabulature representation.

```
verse = \lyricmode { Wie gross bist du! Wie gross bist du! }

harmonies = \new ChordNames \chordmode {
  \germanChords
  \set chordChanges = ##t
  bes8 bes8 bes8
  es2 f
  bes1
}

NoStem = { \hide Stem }
NoNoteHead = \hide NoteHead
ZeroBeam = \override Beam.positions = #'(0 . 0)

staffTabLine = \new Staff \with {
  \remove "Time_signature_engraver"
  \remove "Clef_engraver"
} {
  \override Staff.StaffSymbol.line-positions = #'(0)
  % Shows one horizontal line. The vertical line
  %  (simulating a bar-line) is simulated with a gridline
  \set Staff.midiInstrument = #"choir aahs"
  \key c \major
  \relative c''
  {
    % disable the following line to see the noteheads while writing the song
    \NoNoteHead
    \override NoteHead.no-ledgers = ##t

    % The beam between 8th-notes is used to draw the push-line
    %How to fast write the push-lines:
    %  1. write repeatedly 'c c c c c c c c |' for the whole length of the song
    %  2. uncomment the line \NoNoteHead
    %  3. compile
    %  4. Mark the positions on which push/pull changes.
    %     In the score-picture click on the position
    %        the push- or pull-part starts
    %           (on the noteHead, the cursor will change to a hand-icon).
    %     The cursor in the source code will jump just at this position.
    %   a) If a push-part starts there, replace the 'c' by an 'e['
    %   b) If a pull-part starts there, replace the 'c' by an 's'
    %  5. Switch into 'overwrite-mode' by pressing the 'ins' key.
    %  6. For the pull-parts overwrite the 'c' with 's'
```

```
    %  7. For every push-part replace the last 'c' with 'e]'
    %        8. Switch into 'insert-mode' again
    %  9. At last it should look like e.g.
    %        (s s e[ c | c c c c c c c c | c c c c c c e] s s)
    % 10. re-enable the line \NoNoteHead
    \autoBeamOff
    \ZeroBeam
    s8 s s e[ c c c c c c e] | s s s s s
  }
}

% Accordion melody in tabulator score
% 1. Place a copy of the piano melody below
% 2. Separate piano melody into pull- and push-parts
%      according to the staffTabLine you've already made
% 3. For each line: Double the line. Remark the 1st one
%      (Keeps unchanged as reference) and then change the second
%      line using the transformation paper
%    or the macros 'conv2diaton push.bsh' and 'conv2diaton pull.bsh'
% Tips:
% - In jEdit Search & Replace mark the Option 'Keep Dialog'

AccordionTabTwoCBesDur = {
  % pull 1
  %<f' bes'>8 <f' a'>8 <d' bes'>8 |
  <g'' a''>8 <g'' b''>8 <e'' a''>8 |
  % push 2
  %<g' c''>4 <f' d''> <g' ees''> <f' a'> |
  <g'' a''>4 <d'' eisis''> <g'' bisis''> <d'' f''> |
  % pull 3
  % <f' bes'>2 r8 }
  <g'' a''>2 r8 }

AccordionTab= { \dynamicUp
  % 1. Place a copy of the piano melody above
  % 2. Separate piano melody into pull- and push-parts
  %      according to the staffTabLine you've already made
  % 3. For each line: Double the line. Remark the 1st one
  %      (Keeps unchanged as reference) and then
  %      change the second line using the transformation paper
  % Tips:
  % - In jEdit Search & Replace mark the Option 'Keep Dialog'
  % -
  \AccordionTabTwoCBesDur
}



\layout {
  \context {
    \Score
    % The vertical line (simulating a bar-line) in
```

```
    % the staffBassRhytm is a gridline
    \consists "Grid_line_span_engraver"
  }
  \context {
    \Staff
    \consists "Grid_point_engraver"
    gridInterval = #(ly:make-moment 4/4) % 4/4 - tact. How many beats per bar
    % The following line has to be adjusted O-F-T-E-N.
    \override GridPoint.Y-extent = #'(-2 . -21)
  }
  \context {
    \ChoirStaff
    \remove "System_start_delimiter_engraver"
  }
}

staffVoice = \new Staff = astaffvoice  {
  \time 4/4
  \set Staff.instrumentName = "Voice"
  \set Staff.midiInstrument = "voice oohs"
  \key bes \major
  \partial 8*3
  \clef treble
  {
    \context Voice = "melodyVoi"
    {
      <f' bes'>8 <f' a'>8 <d' bes'>8 |
      <g' c''>4 <f' d''> <g' es''> <f' a'> |
      <f' bes'>2 r8
    }
    \bar "|."
  }
}

staffAccordionMel =
\new Staff  \with { \remove "Clef_engraver" } {
  \accidentalStyle forget %Set the accidentals (Vorzeichen) for each note,
  %do not remember them for the rest of the measure.
  \time 4/4
  \set Staff.instrumentName="Accordion"
  \set Staff.midiInstrument="voice oohs"
  \key c \major
  \clef treble
  { \AccordionTab \bar "|." }
}

AltOn =
#(define-music-function (mag) (number?)
  #{ \override Stem.length = #(* 7.0 mag)
      \override NoteHead.font-size =
#(inexact->exact (* (/ 6.0 (log 2.0)) (log mag))) #})
```
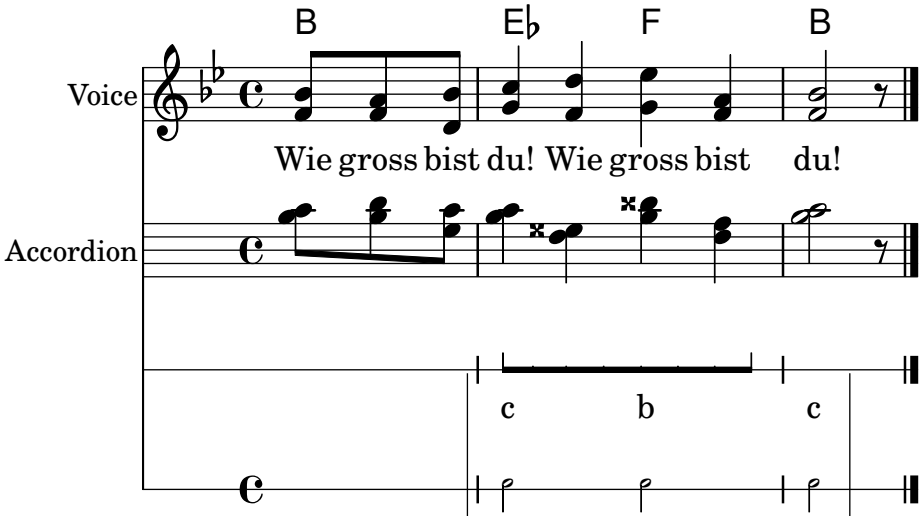
```
AltOff = {
  \revert Stem.length
  \revert NoteHead.font-size
}

BassRhytm = {s4 s8 | c2 c2 | c2 s8 }
LyricBassRhythmI=  \lyricmode { c b | c }

staffBassRhytm =
\new Staff = staffbass \with { \remove "Clef_engraver" } {
  % This is not a RhythmicStaff  because it must be possible to append lyrics.

  \override Score.GridLine.extra-offset = #'( 13.0 . 0.0 ) % x.y
  \override Staff.StaffSymbol.line-positions = #'( 0 )
  % Shows one horizontal line. The vertical line
  %  (simulating a bar-line) is simulated by a grid
  % Search for 'grid' in this page to find all related functions
  \time 4/4
  {
    \context Voice = "VoiceBassRhytm"
    \stemDown \AltOn #0.6
    \relative c''
    {
      \BassRhytm
    }
    \AltOff
    \bar "|."
  }
}

\score {
  \new ChoirStaff <<
    \harmonies
    \staffVoice
    \context Lyrics = "lmelodyVoi"
      \with { alignBelowContext = astaffvoice }
      \lyricsto melodyVoi \verse
    \staffAccordionMel
    \staffTabLine
    \staffBassRhytm
    \context Lyrics = "lBassRhytmAboveI"
      \with { alignAboveContext = staffbass }
      \lyricsto VoiceBassRhytm \LyricBassRhythmI
  >>
}
```



Paroles, musique et accords

Ce canevas comporte tous les éléments d'une chanson : la mélodie, les paroles, les accords.

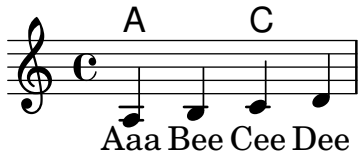
```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}

harmonies = \chordmode {
  a2 c
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Voice = "one" { \autoBeamOff \melody }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}
```



Paroles, musique, accords et diagrammes de fret

Ce canevas comporte, en plus de la mélodie, des paroles et des accords, les diagrammes de fret.

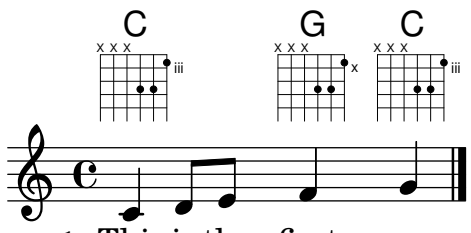
```
verseI = \lyricmode {
  \set stanza = #"1."
  This is the first verse
}

verseII = \lyricmode {
  \set stanza = #"2."
  This is the second verse.
}

theChords = \chordmode {
  % insert chords for chordnames and fretboards here
  c2 g4 c
}

staffMelody = \relative c' {
   \key c \major
   \clef treble
   % Type notes for melody here
   c4 d8 e f4 g
   \bar "|."
}

\score {
  <<
    \context ChordNames { \theChords }
    \context FretBoards { \theChords }
    \new Staff {
      \context Voice = "voiceMelody" { \staffMelody }
    }
    \new Lyrics = "lyricsI" {
      \lyricsto "voiceMelody" \verseI
    }
    \new Lyrics = "lyricsII" {
      \lyricsto "voiceMelody" \verseII
    }
  >>
  \layout { }
  \midi { }
}
```



1. This is the first verse
2. This is the second verse.

Mélodie simple et accords
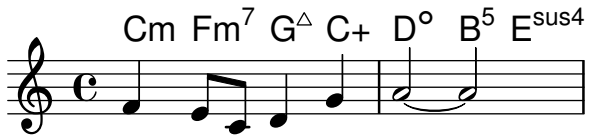
Vous avez besoin de la partition d'une mélodie avec les accords ? N'allez pas plus loin !

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  f4 e8[ c] d4 g
  a2 ~ a
}

harmonies = \chordmode {
  c4:m f:min7 g:maj c:aug
  d2:dim b4:5 e:sus
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \harmonies
    }
    \new Staff \melody
  >>
  \layout{ }
  \midi { }
}
```



Portée unique et paroles
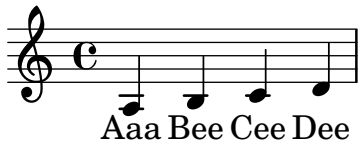
Ce canevas comporte une simple ligne mélodique agrémentée de paroles. Recopiez-le, ajoutez-y d'autres notes et paroles. Les ligatures automatiques sont ici désactivées comme il est d'usage en matière de musique vocale. Pour activer la fonction de ligature automatique, modifiez ou commentez la ligne en question.

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

text = \lyricmode {
  Aaa Bee Cee Dee
}
```

```
\score{
  <<
    \new Voice = "one" {
      \autoBeamOff
      \melody
    }
    \new Lyrics \lyricsto "one" \text
  >>
  \layout { }
  \midi { }
}
```
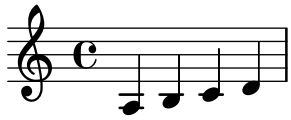


Portée unique avec quelques notes

Cet exemple simpliste se compose d'une portée agrémentée de quelques notes. Il convient tout à fait pour un instrument seul ou un fragment mélodique. Recopiez-le dans un nouveau fichier, ajoutez-y d'autres notes et c'est prêt !

```
melody = \relative c' {
  \clef treble
  \key c \major
  \time 4/4

  a4 b c d
}

\score {
  \new Staff \melody
  \layout { }
  \midi { }
}
```



Quatuor à cordes (conducteur)

Voici un canevas pour quatuor à cordes. Notez l'utilisation de la variable `\global` pour traiter la métrique et la tonalité.

```
global= {
  \time 4/4
  \key c \major
}

violinOne = \new Voice \relative c'' {
  c2 d
```

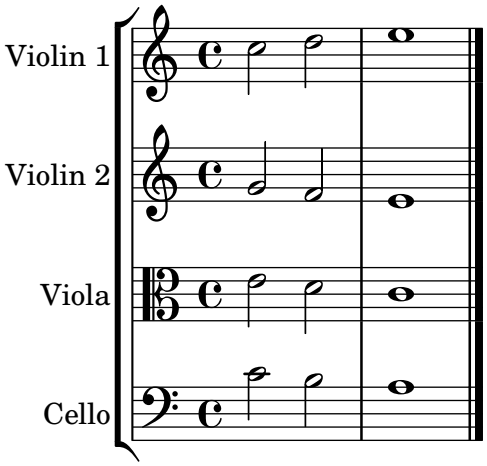
```
  e1
  \bar "|."
}

violinTwo = \new Voice \relative c'' {
  g2 f
  e1
  \bar "|."
}

viola = \new Voice \relative c' {
  \clef alto
  e2 d
  c1
  \bar "|."
}

cello = \new Voice \relative c' {
  \clef bass
  c2 b
  a1
  \bar "|."
}

\score {
  \new StaffGroup <<
    \new Staff \with { instrumentName = "Violin 1" }
    << \global \violinOne >>
    \new Staff \with { instrumentName = "Violin 2" }
    << \global \violinTwo >>
    \new Staff \with { instrumentName = "Viola" }
    << \global \viola >>
    \new Staff \with { instrumentName = "Cello" }
    << \global \cello >>
  >>
  \layout { }
  \midi { }
}
```



Quatuor à cordes, avec parties séparées

Grâce à ce canevas, vous pouvez obtenir une partition d'excellente facture pour quatuor à cordes mais aussi, si le besoin s'en faisait sentir, une partie séparée par instrument. Par ailleurs, cet exemple illustre l'utilisation de la fonction `\tag` dans le but d'extraire des parties séparées.

Il vous faudra découper ce canevas en plusieurs fichiers séparés ; leur nom respectif est indiqué en commentaire : `piece.ly` comporte tout ce qui a trait à la musique, les autres fichiers – `score.ly`, `vn1.ly`, `vn2.ly`, `vla.ly` et `vlc.ly` – vous permettront d'obtenir les parties selon le pupitre.

N'oubliez pas de supprimer les commentaires superflus des fichiers individualisés !

```
%%%%% piece.ly
%%%%% (This is the global definitions file)

global= {
  \time 4/4
  \key c \major
}


Violinone = \new Voice {
  \relative c'' {
    c2 d e1
    \bar "|."
  }
}


Violintwo = \new Voice {
  \relative c'' {
    g2 f e1
    \bar "|."
  }
}


Viola = \new Voice {
  \relative c' {
    \clef alto
    e2 d c1
    \bar "|."
```

```
  }
}


Cello = \new Voice {
  \relative c' {
    \clef bass
    c2 b a1
    \bar "|."
  }
}


music = {
  <<
    \tag #'score \tag #'vn1
    \new Staff \with { instrumentName = "Violin 1" }
    << \global \Violinone >>

    \tag #'score \tag #'vn2
    \new Staff \with { instrumentName = "Violin 2" }
    << \global \Violintwo>>

    \tag #'score \tag #'vla
    \new Staff \with { instrumentName = "Viola" }
    << \global \Viola>>

    \tag #'score \tag #'vlc
    \new Staff \with { instrumentName = "Cello" }
    << \global \Cello >>
  >>
}

% These are the other files you need to save on your computer

% score.ly
% (This is the main file)

% uncomment the line below when using a separate file
%\include "piece.ly"

#(set-global-staff-size 14)

\score {
  \new StaffGroup \keepWithTag #'score \music
  \layout { }
  \midi { }
}


%{ Uncomment this block when using separate files
```

```
% vn1.ly
% (This is the Violin 1 part file)

\include "piece.ly"
\score {
  \keepWithTag #'vn1 \music
  \layout { }
}


% vn2.ly
% (This is the Violin 2 part file)

\include "piece.ly"
\score {
  \keepWithTag #'vn2 \music
  \layout { }
}

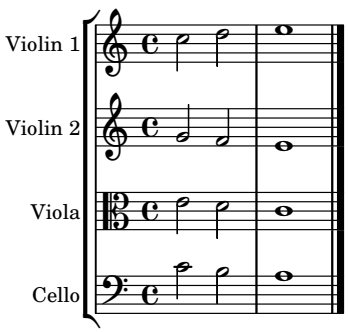

% vla.ly
% (This is the Viola part file)

\include "piece.ly"
\score {
  \keepWithTag #'vla \music
  \layout { }
}


% vlc.ly
% (This is the Cello part file)

\include "piece.ly"
\score {
  \keepWithTag #'vlc \music
  \layout { }
}

%}
```



Ensemble vocal avec réduction pour piano

Ce canevas ajoute une réduction pour piano à une partition standard pour chœur à quatre voix mixtes. Ceci illustre l'un des avantages de LilyPond : une expression musicale peut être réutilisée sans effort. Toute modification apportée à l'une des voix, mettons `tenorMusique`, sera automatiquement reportée dans la réduction pour piano.

```
\paper {
  top-system-spacing.basic-distance = #10
  score-system-spacing.basic-distance = #20
  system-system-spacing.basic-distance = #20
  last-bottom-spacing.basic-distance = #10
}

global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative {
  c''4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

altoMusic = \relative {
  e'4 f d e
}
altoWords =\lyricmode {
  ha ha ha ha
}

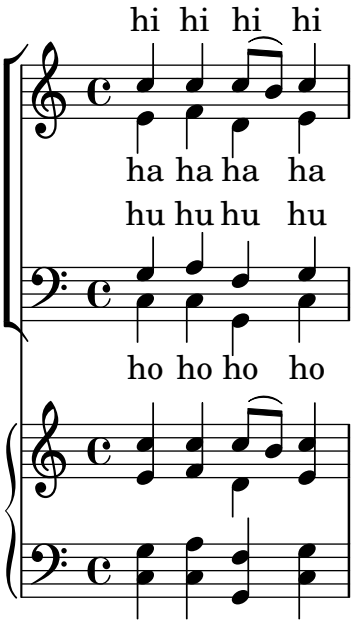
tenorMusic = \relative {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  <<
    \new ChoirStaff <<
      \new Lyrics = "sopranos" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
```

```
      }
      \new Staff = "women" <<
        \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
        \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
      >>
      \new Lyrics = "altos"
      \new Lyrics = "tenors" \with {
        % This is needed for lyrics above a staff
        \override VerticalAxisGroup.staff-affinity = #DOWN
      }

      \new Staff = "men" <<
        \clef bass
        \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
        \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
      >>
      \new Lyrics = "basses"
      \context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
      \context Lyrics = "altos" \lyricsto "altos" \altoWords
      \context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
      \context Lyrics = "basses" \lyricsto "basses" \bassWords
    >>
    \new PianoStaff <<
      \new Staff <<
        \set Staff.printPartCombineTexts = ##f
        \partCombine
        << \global \sopMusic >>
        << \global \altoMusic >>
      >>
      \new Staff <<
        \clef bass
        \set Staff.printPartCombineTexts = ##f
        \partCombine
        << \global \tenorMusic >>
        << \global \bassMusic >>
      >>
    >>
  >>
}
```



Ensemble vocal avec alignement des paroles selon le contexte

Ce canevas ressemble beaucoup à celui pour chœur à quatre voix mixtes. La différence réside dans le fait que les paroles sont positionnées en ayant recours à `alignAboveContext` et `alignBelowContext`.

```
global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative c'' {
  c4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

altoMusic = \relative c' {
  e4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}
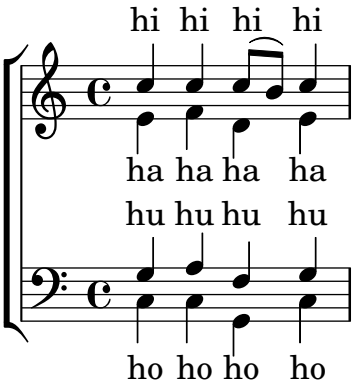
tenorMusic = \relative c' {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative c {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
```

```
}

\score {
  \new ChoirStaff <<
    \new Staff = "women" <<
      \new Voice = "sopranos" { \voiceOne << \global \sopMusic >> }
      \new Voice = "altos" { \voiceTwo << \global \altoMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = #"women" }
      \lyricsto "sopranos" \sopWords
    \new Lyrics \with { alignBelowContext = #"women" }
      \lyricsto "altos" \altoWords
    % we could remove the line about this with the line below, since
    % we want the alto lyrics to be below the alto Voice anyway.
    % \new Lyrics \lyricsto "altos" \altoWords

    \new Staff = "men" <<
      \clef bass
      \new Voice = "tenors" { \voiceOne << \global \tenorMusic >> }
      \new Voice = "basses" { \voiceTwo << \global \bassMusic >> }
    >>
    \new Lyrics \with { alignAboveContext = #"men" }
      \lyricsto "tenors" \tenorWords
    \new Lyrics \with { alignBelowContext = #"men" }
      \lyricsto "basses" \bassWords
    % again, we could replace the line above this with the line below.
    % \new Lyrics \lyricsto "basses" \bassWords
  >>
}
```



Ensemble vocal avec couplet et refrain

Ce canevas illustre la manière d'agencer une œuvre vocale où le couplet est chanté en solo et le refrain à deux voix. Vous noterez le recours aux silences invisibles dans la variable `\global` : ils permettent de positionner les changements de métrique et autres éléments communs à toutes les parties, ce pour l'intégralité du morceau.

```
global = {
  \key g \major

  % verse
```

```
  \time 3/4
  s2.*2
  \break

  % refrain
  \time 2/4
  s2*2
  \bar "|."
}

SoloNotes = \relative g' {
  \clef "treble"

  % verse
  g4 g g |
  b4 b b |

  % refrain
  R2*2 |
}

SoloLyrics = \lyricmode {
  One two three |
  four five six |
}

SopranoNotes = \relative c'' {
  \clef "treble"

  % verse
  R2.*2 |

  % refrain
  c4 c |
  g4 g |
}

SopranoLyrics = \lyricmode {
  la la |
  la la |
}

BassNotes = \relative c {
  \clef "bass"

  % verse
  R2.*2 |

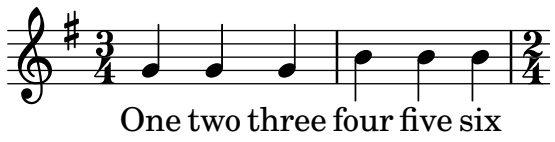
  % refrain
  c4 e |
  d4 d |
}
```

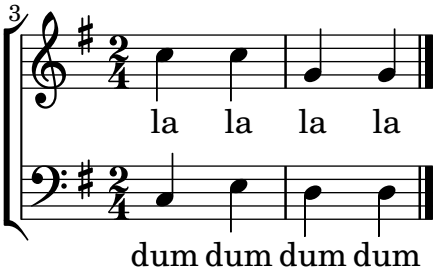
```
BassLyrics = \lyricmode {
  dum dum |
  dum dum |
}

\score {
  <<
    \new Voice = "SoloVoice" << \global \SoloNotes >>
    \new Lyrics \lyricsto "SoloVoice" \SoloLyrics

    \new ChoirStaff <<
      \new Voice = "SopranoVoice" << \global \SopranoNotes >>
      \new Lyrics \lyricsto "SopranoVoice" \SopranoLyrics

      \new Voice = "BassVoice" << \global \BassNotes >>
      \new Lyrics \lyricsto "BassVoice" \BassLyrics
    >>
  >>
  \layout {
    ragged-right = ##t
    \context { \Staff
      % these lines prevent empty staves from being printed
      \RemoveEmptyStaves
      \override VerticalAxisGroup.remove-first = ##t
    }
  }
}
```





Ensemble vocal (simple)

Ce fichier constitue un canevas standard de partition pour chœur à quatre voix mixtes. Lorsque les ensembles s'étoffent, il est judicieux de recourir à une section spécifique incluse dans chacune des parties, tout particulièrement pour gérer la métrique et la tonalité qui, la plupart du temps, sont communes à tous les pupitres. Comme il est d'usage pour les hymnes, les quatre voix sont réparties sur deux portées.

```
\paper {
  top-system-spacing.basic-distance = #10
```

```
  score-system-spacing.basic-distance = #20
  system-system-spacing.basic-distance = #20
  last-bottom-spacing.basic-distance = #10
}

global = {
  \key c \major
  \time 4/4
}

sopMusic = \relative {
  c''4 c c8[( b)] c4
}
sopWords = \lyricmode {
  hi hi hi hi
}

altoMusic = \relative {
  e'4 f d e
}
altoWords = \lyricmode {
  ha ha ha ha
}

tenorMusic = \relative {
  g4 a f g
}
tenorWords = \lyricmode {
  hu hu hu hu
}

bassMusic = \relative {
  c4 c g c
}
bassWords = \lyricmode {
  ho ho ho ho
}

\score {
  \new ChoirStaff <<
    \new Lyrics = "sopranos" \with {
      % this is needed for lyrics above a staff
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff = "women" <<
      \new Voice = "sopranos" {
        \voiceOne
        << \global \sopMusic >>
      }
      \new Voice = "altos" {
        \voiceTwo
        << \global \altoMusic >>
```
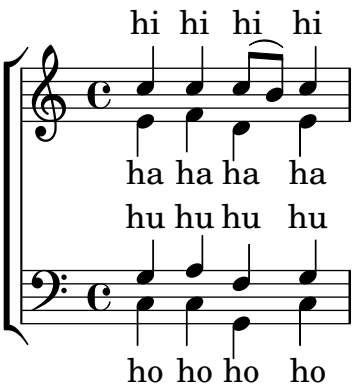
```
      }
    >>
    \new Lyrics = "altos"
    \new Lyrics = "tenors" \with {
      % this is needed for lyrics above a staff
      \override VerticalAxisGroup.staff-affinity = #DOWN
    }
    \new Staff = "men" <<
      \clef bass
      \new Voice = "tenors" {
        \voiceOne
        << \global \tenorMusic >>
      }
      \new Voice = "basses" {
        \voiceTwo << \global \bassMusic >>
      }
    >>
    \new Lyrics = "basses"
    \context Lyrics = "sopranos" \lyricsto "sopranos" \sopWords
    \context Lyrics = "altos" \lyricsto "altos" \altoWords
    \context Lyrics = "tenors" \lyricsto "tenors" \tenorWords
    \context Lyrics = "basses" \lyricsto "basses" \bassWords
  >>
}
```

