LilyPond

Il compositore tipografico per la musica

Learning Manual

Il team di sviluppo di LilyPond

Questo file fornisce un'introduzione alla versione di LilyPond 2.14.1.

Per maggiori informazioni su come questo manuale si integra col resto della documentazione, o per leggere questo manuale in altri formati, si veda Sezione "Manuals" in *Informazioni generali*.

Se ti manca qualche manuale, puoi trovare la completa documentazione all'indirizzo http://www.lilypond.org/.

Copyright © 1999–2011 by the authors.

La traduzione della seguente nota di copyright è gentilmente offerta per le persone che non parlano inglese, ma solo la nota in inglese ha valore legale.

The translation of the following copyright notice is provided for courtesy to non-English speakers, but only the notice in English legally counts.

> E' garantito il permesso di copiare, distribuire e/o modificare questo documento seguendo i termini della Licenza per Documentazione Libera GNU, Versione 1.1 o ogni versione successiva pubblicata dalla Free Software Foundation; senza alcuna sezione non modificabile. Una copia della licenza è acclusa nella sezione intitolata "Licenza per Documentazione Libera GNU".
>
> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections. A copy of the license is included in the section entitled "GNU Free Documentation License".

Per la versione di LilyPond 2.14.1

Sommario

1 Tutorial ... 1

1.1 Compilare un file ... 1
1.1.1 Inserire l'input ... 1
1.1.2 MacOS X ... 2
1.1.3 Windows ... 6
1.1.4 Linea di comando ... 13
1.2 Come scrivere i file di input ... 13
1.2.1 Notazione semplice ... 13
1.2.2 Lavorare sui file di input ... 18
1.3 Gestire gli errori ... 19
1.3.1 Consigli generali per la risoluzione dei problemi ... 19
1.3.2 Alcuni errori comuni ... 19
1.4 Come leggere i manuali ... 19
1.4.1 Materiale omesso ... 19
1.4.2 Esempi cliccabili ... 20
1.4.3 Panoramica dei manuali ... 20

2 Notazione comunemente utilizzata ... 21

2.1 Notazione su un solo pentagramma ... 21
2.1.1 Controlli di battuta ... 21
2.1.2 Alterazioni e armature di chiave ... 21
2.1.3 Legature di valore e di portamento ... 23
2.1.4 Articolazione e dinamica ... 24
2.1.5 Aggiungere il testo ... 25
2.1.6 Code automatiche e manuali ... 26
2.1.7 Comandi di tempo avanzati ... 26
2.2 Note simultanee ... 27
2.2.1 Espressioni musicali ... 27
2.2.2 Righi multipli ... 29
2.2.3 Gruppi di pentagrammi ... 30
2.2.4 Combinare le note negli accordi ... 31
2.2.5 Polifonia su un singolo rigo ... 31
2.3 Canzoni ... 32
2.3.1 Impostare canzoni semplici ... 32
2.3.2 Allineare il testo alla melodia ... 32
2.3.3 Testo su più righi ... 36
2.4 Ritocchi finali ... 37
2.4.1 Organizzare i brani con le variabili ... 37
2.4.2 Aggiungere i titoli ... 38
2.4.3 Nomi assoluti delle note ... 39
2.4.4 Dopo il tutorial ... 40

3 Concetti fondamentali ... 41

3.1 Come funzionano i file di input di LilyPond ... 41
3.1.1 Introduzione alla struttura di un file di LilyPond ... 41
3.1.2 La partitura è una (singola) espressione musicale composta ... 43
3.1.3 Annidare le espressioni musicali ... 46

3.1.4 Sul non annidamento di parentesi e legature di valore ... 47
3.2 Le voci contengono la musica ... 48
3.2.1 Sento le Voci ... 48
3.2.2 Definire esplicitamente le voci ... 53
3.2.3 Voci e musica vocale ... 56
3.3 Contesti e incisori ... 59
3.3.1 I contesti ... 59
3.3.2 Creare i contesti ... 60
3.3.3 Gli incisori ... 62
3.3.4 Modificare le proprietà di contesto ... 63
3.3.5 Aggiungere e togliere gli incisori ... 67
3.4 Estendere i modelli ... 70
3.4.1 Soprano e violoncello ... 70
3.4.2 Partitura vocale a quattro parti SATB ... 73
3.4.3 Scrivere una partitura da zero ... 78
3.4.4 Ridurre l'input grazie a variabili e funzioni ... 84
3.4.5 Partiture e parti ... 86

4 Tweaking output ... 88

Appendice A Templates ... 89

Appendice B GNU Free Documentation License ... 90

Appendice C Indice di LilyPond ... 97

1 Tutorial

Questo capitolo fornisce un'introduzione di base all'uso di LilyPond.

1.1 Compilare un file
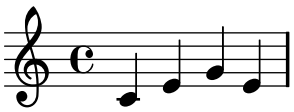
1.1.1 Inserire l'input

"Compilazione" è il termine usato per indicare l'elaborazione di un file di input in formato LilyPond per produrre uno o più file di output. I file di output generalmente sono PDF (per la stampa e la visualizzazione), MIDI (per la riproduzione audio) e PNG (per l'utilizzo online). I file di input di LilyPond sono semplici file di testo.

Questo esempio mostra un semplice file di input:

```
\version "2.14.1"
{
  c' e' g' e'
}
```

L'output grafico è:



> **Nota:** Le note e i testi nel file LilyPond devono essere sempre scritti tra `{` **parentesi graffe** `}`. Per evitare ambiguità, le parentesi dovrebbero essere delimitate da degli spazi, a meno che non si trovino all'inizio o alla fine di una linea. Può darsi che in alcuni esempi di questo manuale le parentesi verranno omesse, ma non dimenticarle nella tua musica! Per maggiori informazioni sull'aspetto degli esempi nel manuale, si veda Sezione 1.4 [Come leggere i manuali], pagina 19.

Inoltre, l'input di LilyPond è **sensibile alle maiuscole**. '`{ c d e }`' è un input valido; '`{ C D E }`' invece produrrà un messaggio di errore.

Generare l'output

Come generare l'output di LilyPond dipende dal tuo sistema operativo e dai programmi che usi.

- Sezione 1.1.2 [MacOS X], pagina 2 Sezione 1.1.2 [MacOS X], pagina 2 (graphical)
- Sezione 1.1.3 [Windows], pagina 6 Sezione 1.1.3 [Windows], pagina 6 (graphical)
- ⟨undefined⟩ [Command-line], pagina ⟨undefined⟩ ⟨undefined⟩ [Command-line], pagina ⟨undefined⟩ (command-line)

Si noti che sono disponibili molti altri editor di testo con un miglior supporto per LilyPond. Per maggiori informazioni, si veda Sezione "Editing facilitato" in *Informazioni generali*.

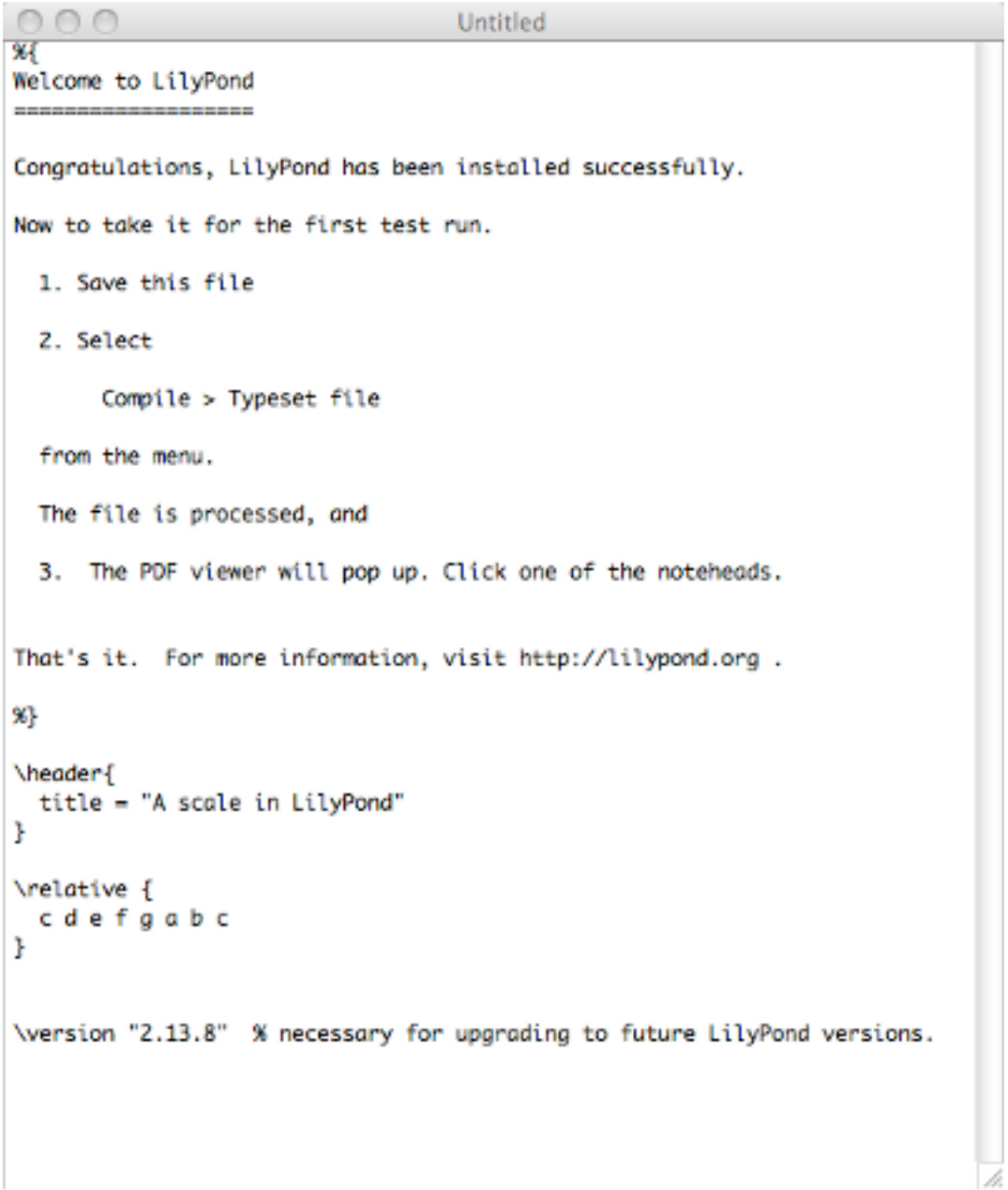
> **Nota:** La prima volta che si esegue LilyPond la compilazione potrà richiedere uno o due minuti perché tutti i font di sistema devono essere prima analizzati. In seguito, LilyPond sarà molto più veloce!

1.1.2 MacOS X

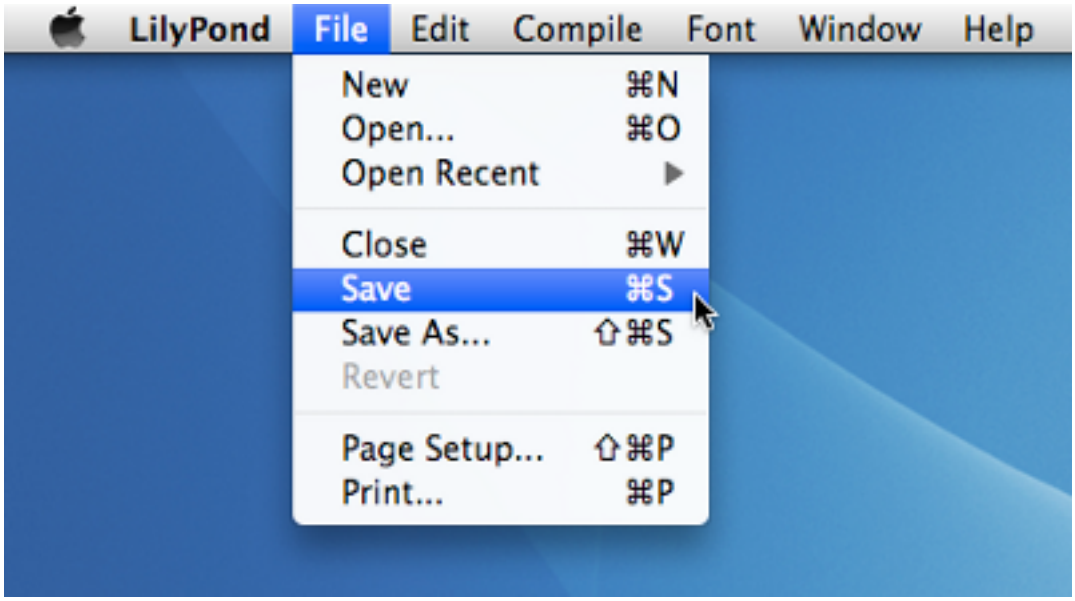
Nota: These instructions assume that you are using the LilyPond application. If you are using any of the programs described in Sezione "Easier editing" in *Informazioni generali*, please consult the documentation for those programs if you have any problems.

Step 1. Create your '`.ly`' file

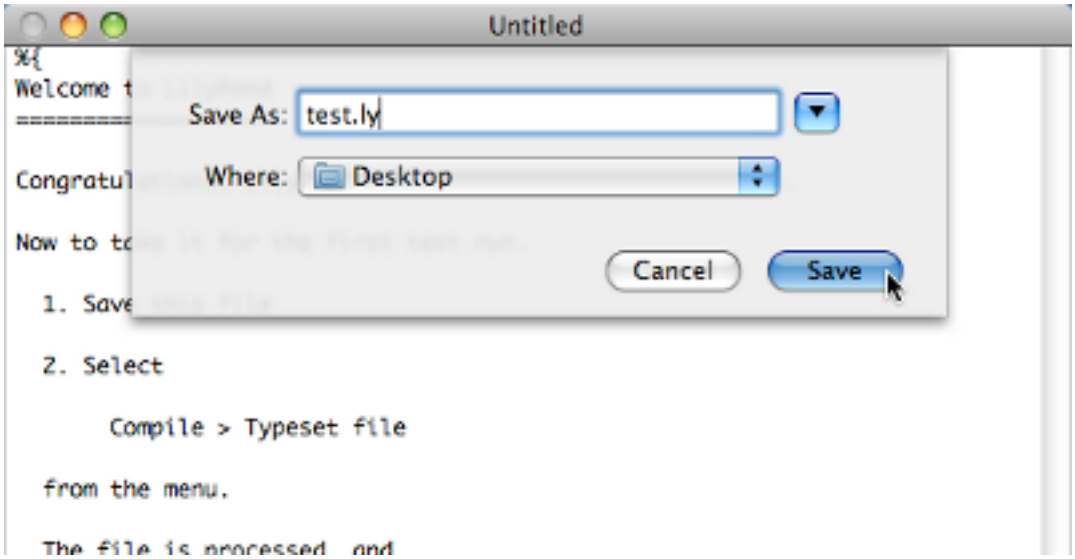
Double click the `LilyPond.app`, an example file will open.



From the menus along the top left of your screen, select `File > Save`.

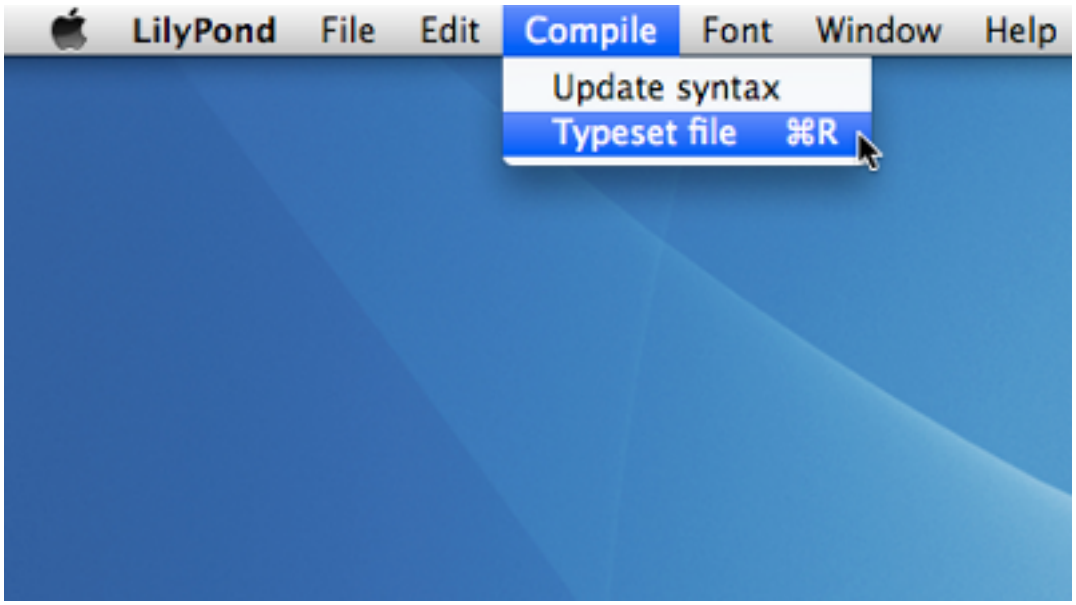


Choose a name for your file, for example '`test.ly`'.

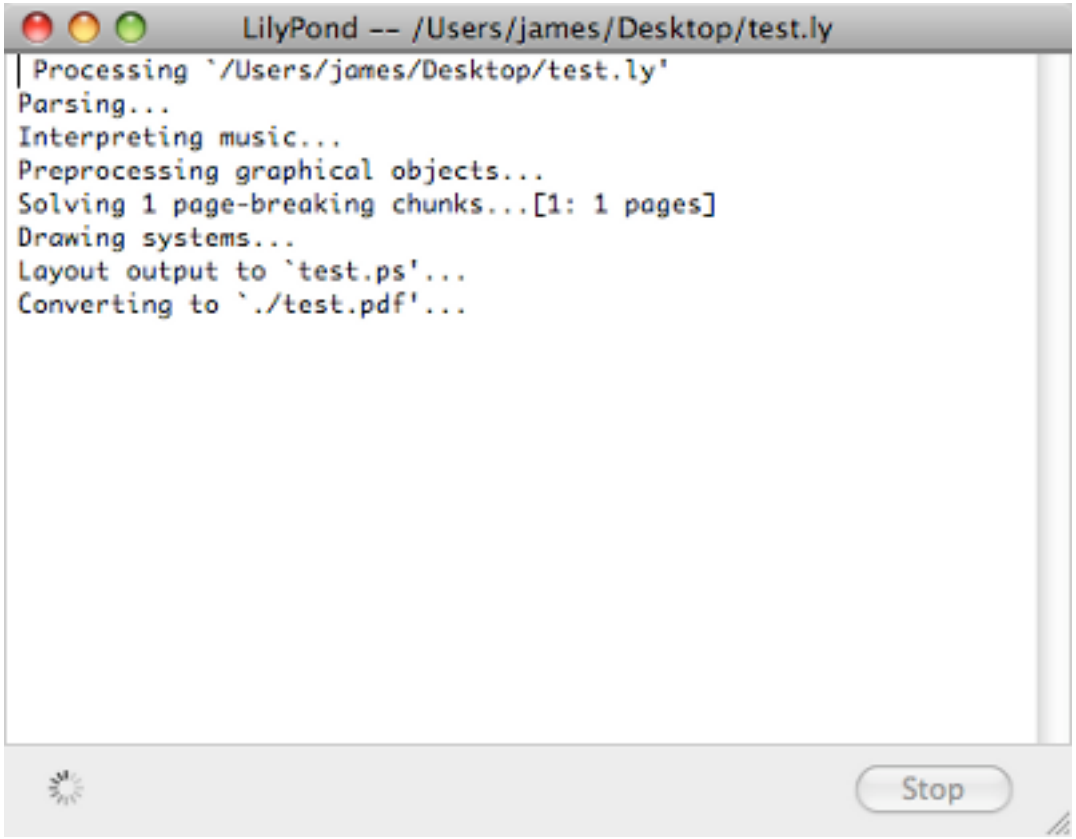


Step 2. Compile (with LilyPad)

From the same menus, select `Compile` > `Typeset`.

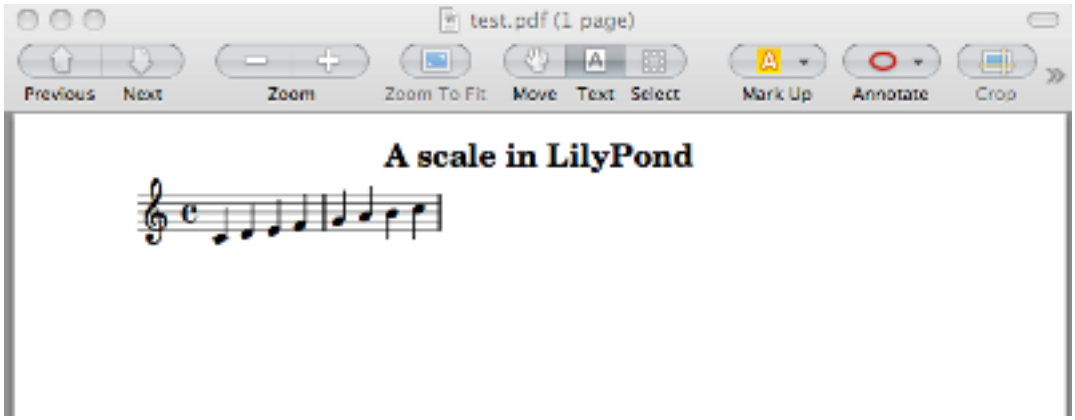


A new window will open showing a progress log of the compilation of the file you have just saved.



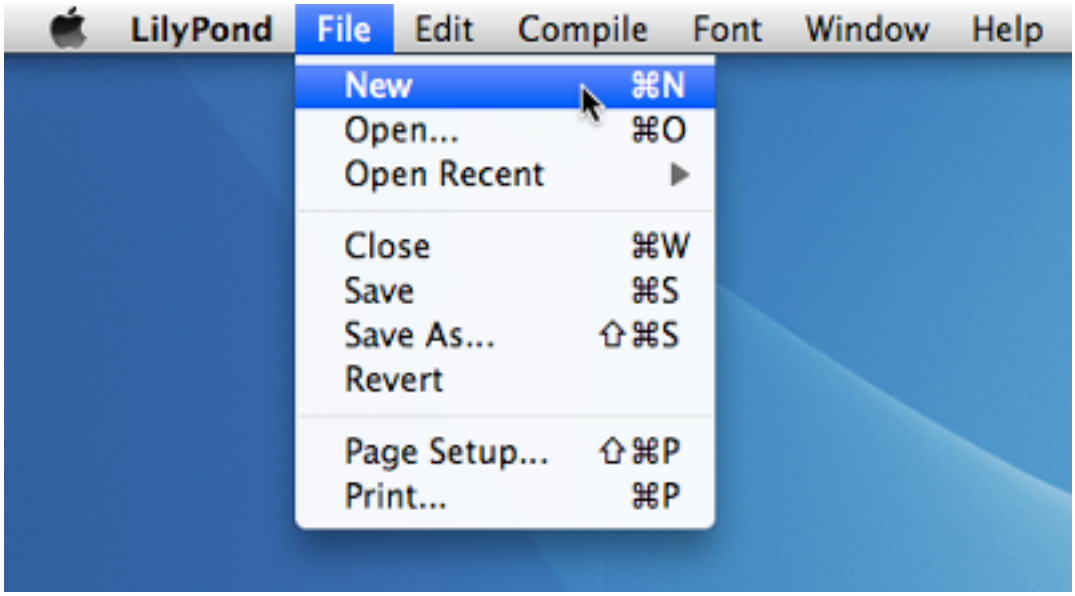
Step 3. View output

Once the compilation has finished, a PDF file will be created with the same name as the original file and will be automatically opened in the default PDF viewer and displayed on your screen.

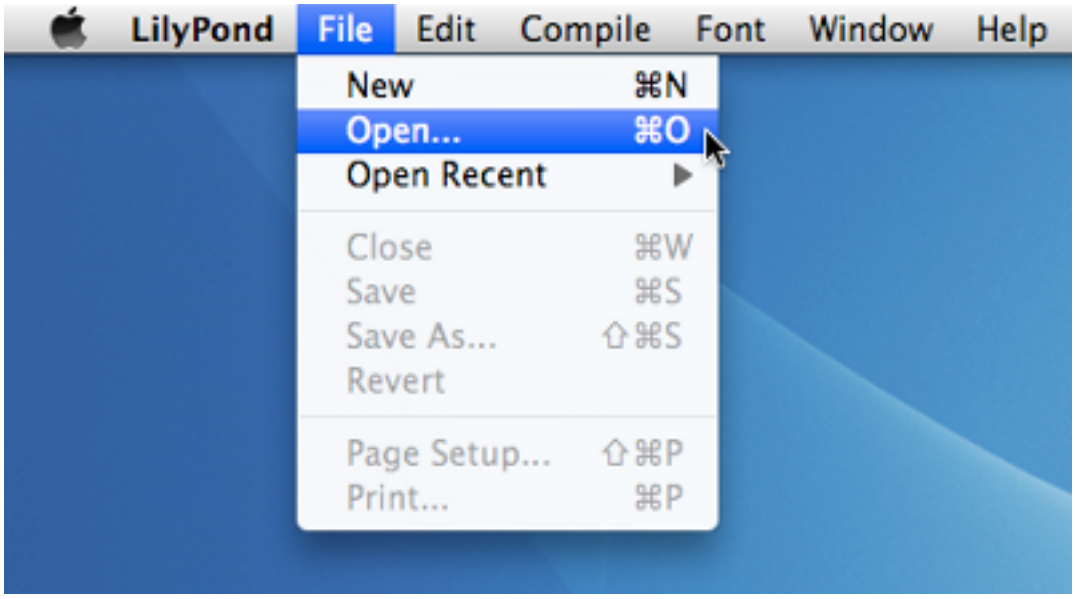


Other commands

To create new files for LilyPond, begin by selecting `File > New`



or `File > Open` to open and edit existing files you have saved previously.



You must save any new edits you make to your file before you `Compile > Typeset` and if the PDF file is not displayed check the window with the progress log for any errors.

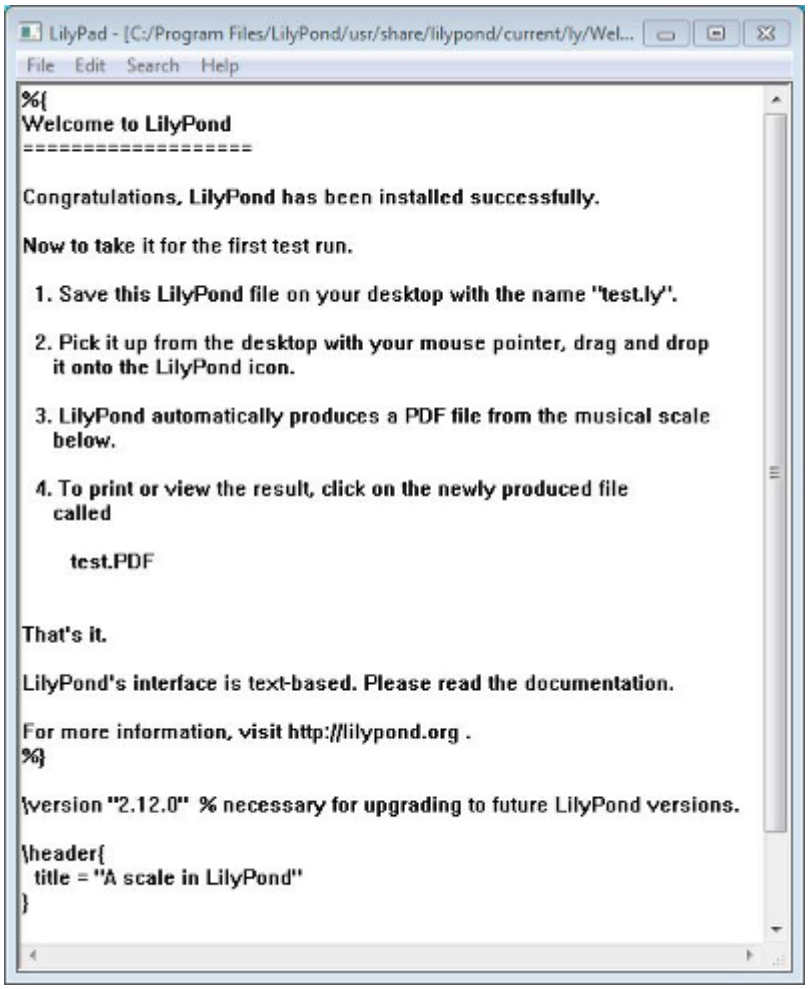
If you are not using the default Preview PDF viewer that comes with the Mac Operating system and you have the PDF file generated from a previous compilation open, then any further compilations may fail to generate an update PDF until you close the original.

1.1.3 Windows

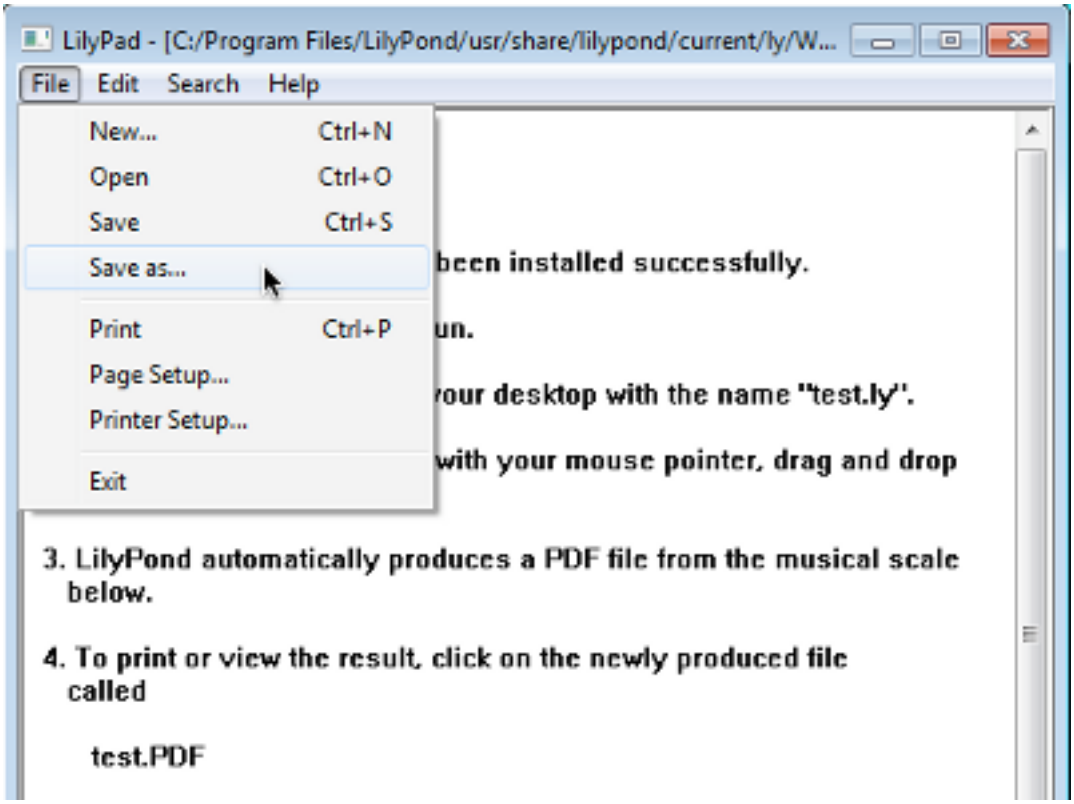
Nota: These instructions assume that you are using the built-in LilyPad editor. If you are using any of the programs described in Sezione "Easier editing" in *Informazioni generali*, please consult the documentation for those programs if you have any problems compiling a file.

Step 1. Create your '.ly' file

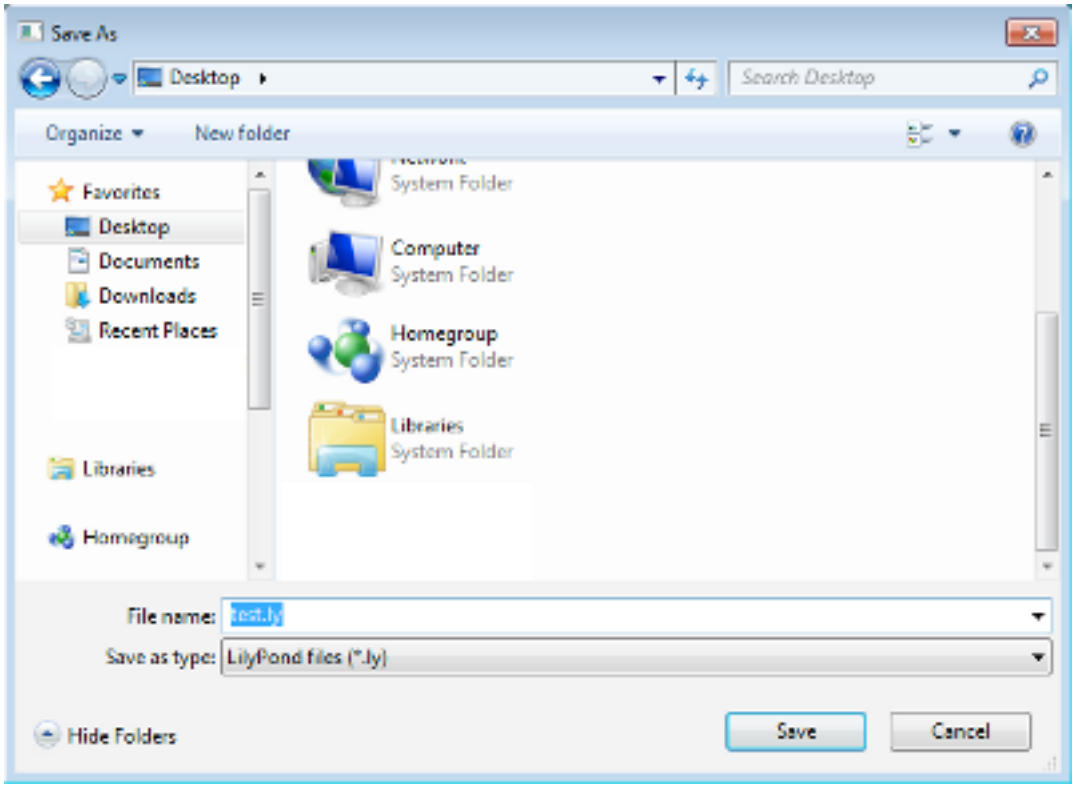
Double-click the LilyPond icon on your desktop, an example file will open.



From the menus that appear along the top of the example file, select `File > Save as`. Do not use the `File > Save` for the example file as this will not work until you have given it a valid LilyPond file name.



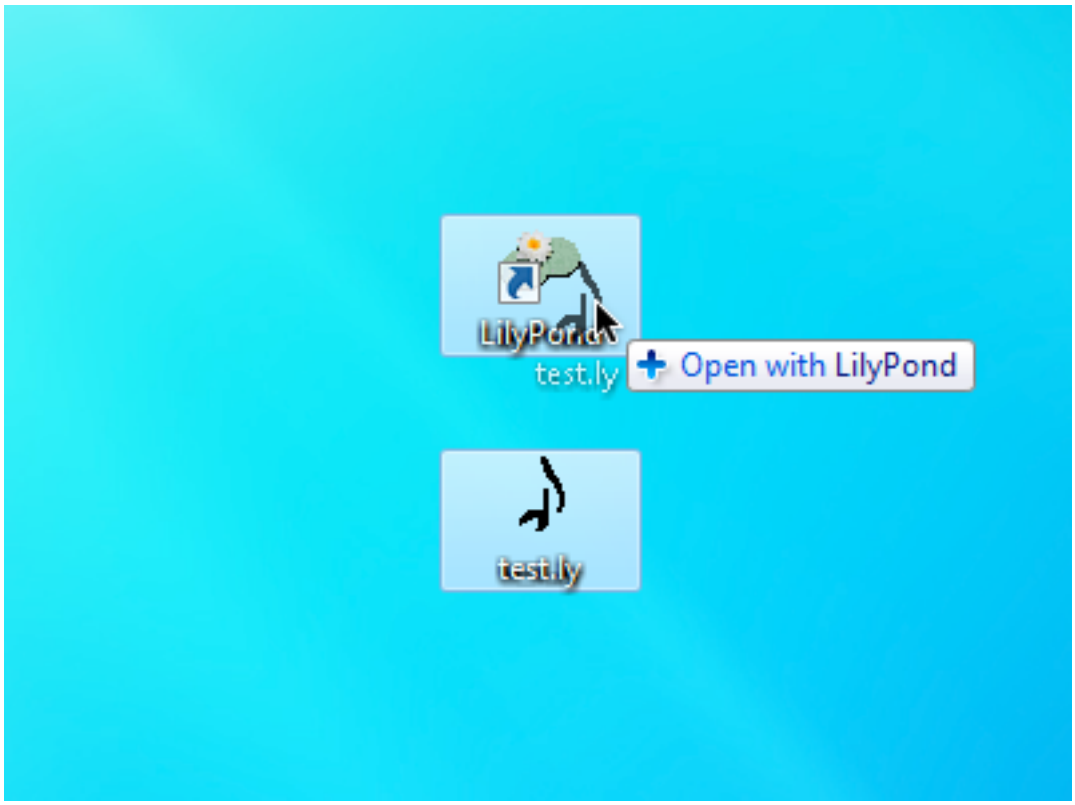
Choose a name for your file, for example '`test.ly`'.



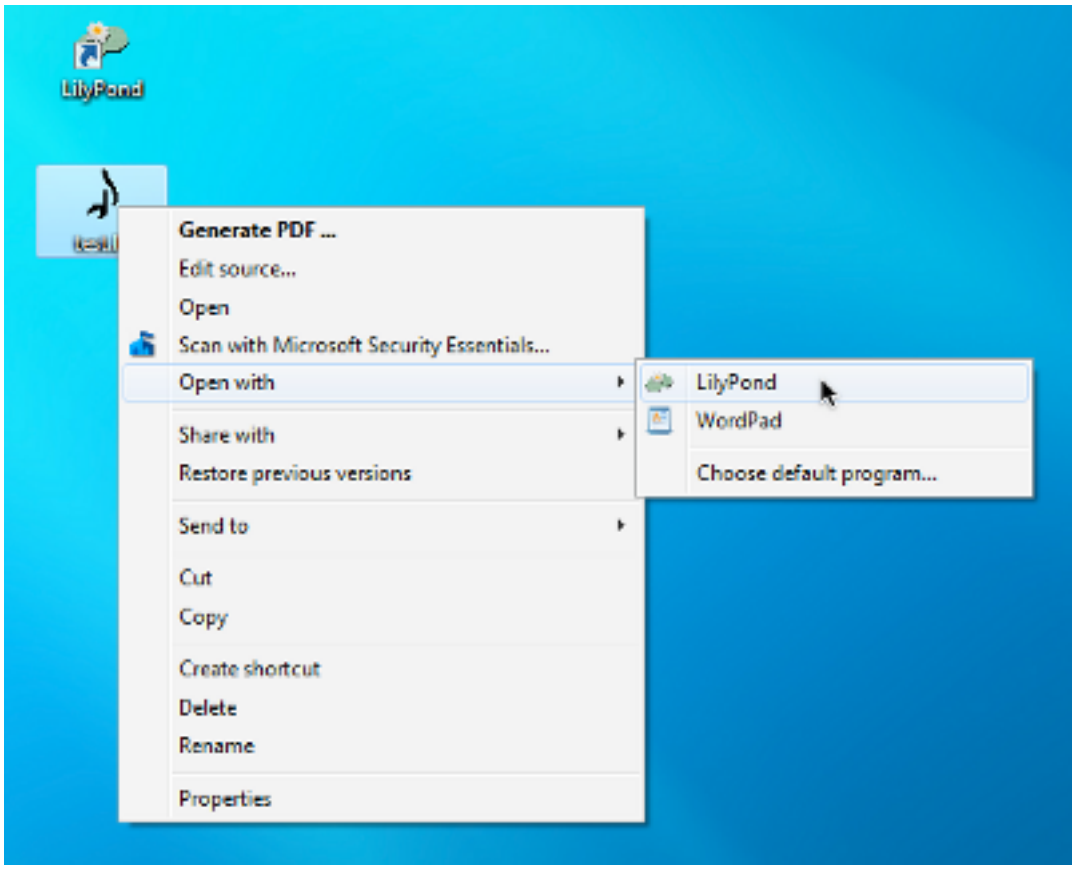
Step 2a. Compile (with drag-and-drop)

Depending on what you prefer, to compile your file either:

Drag-and-drop the file directly onto the LilyPond icon.



Right-click on the file and from the pop-up context menu choose `Open with > LilyPond`.

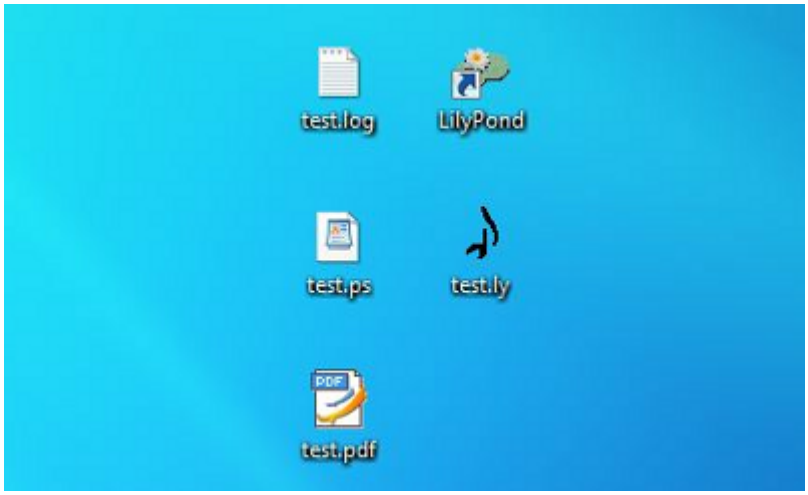


Step 2b. Compile (with double-clicking)

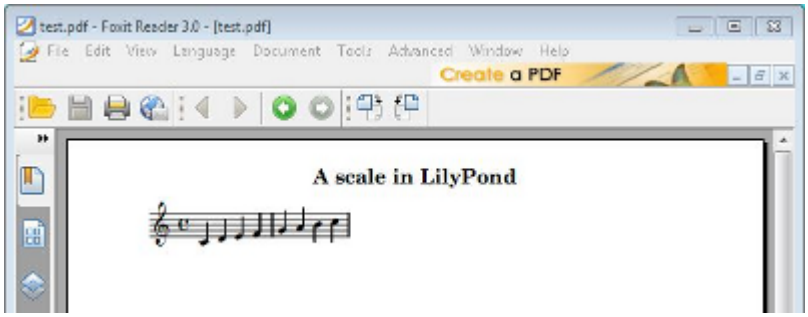
Or simply double-click the '`test.ly`'.

Step 3. View output

During the compilation of the '`test.ly`' file, a command window will, very briefly open and then close. Three additional files will have been created during this process.

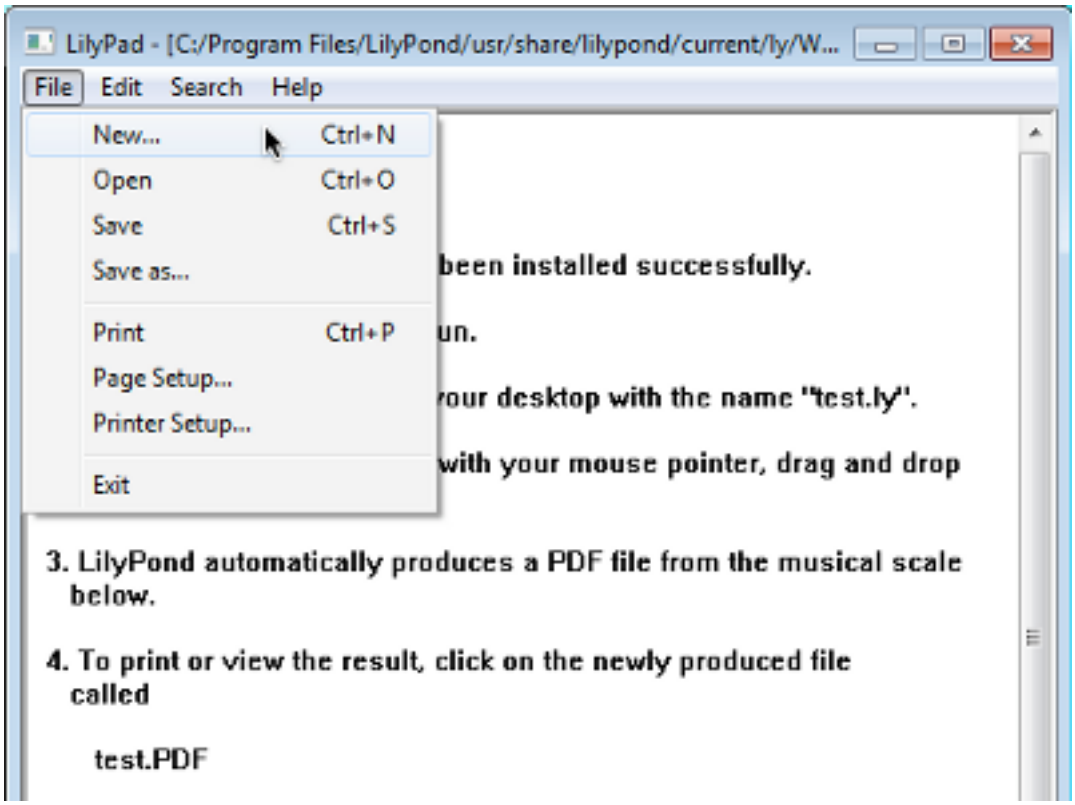


The PDF file contains the engraved '`test.ly`' file.

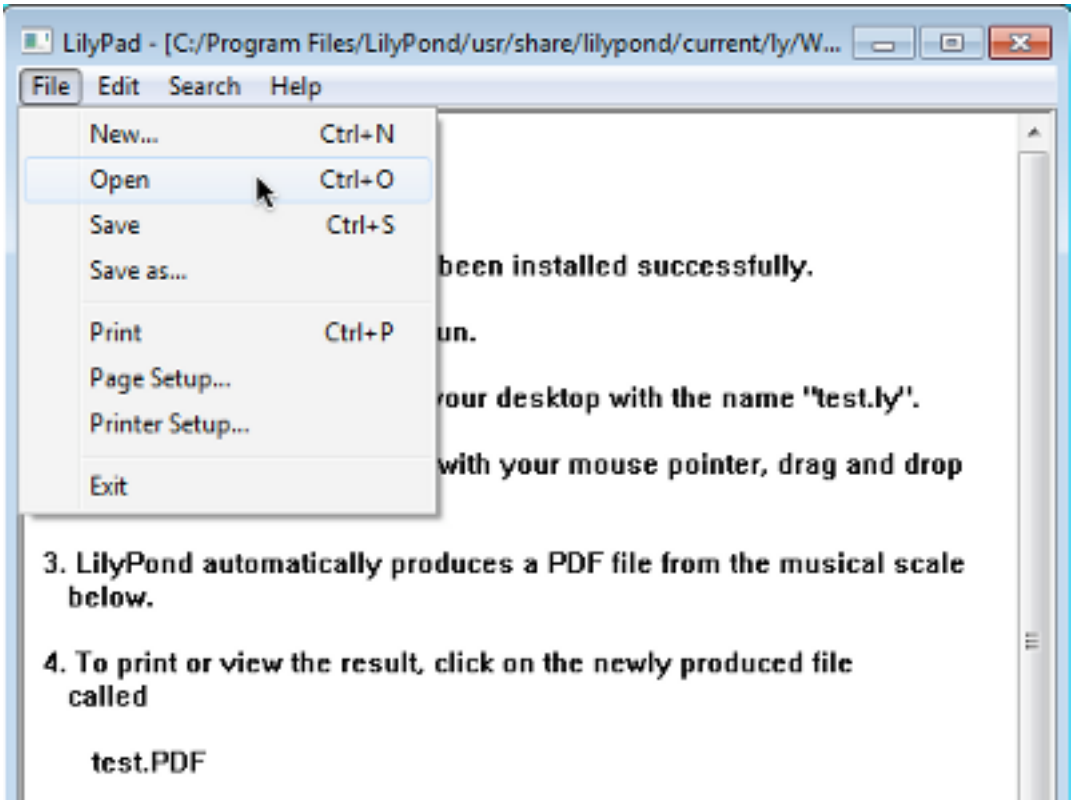


Other commands

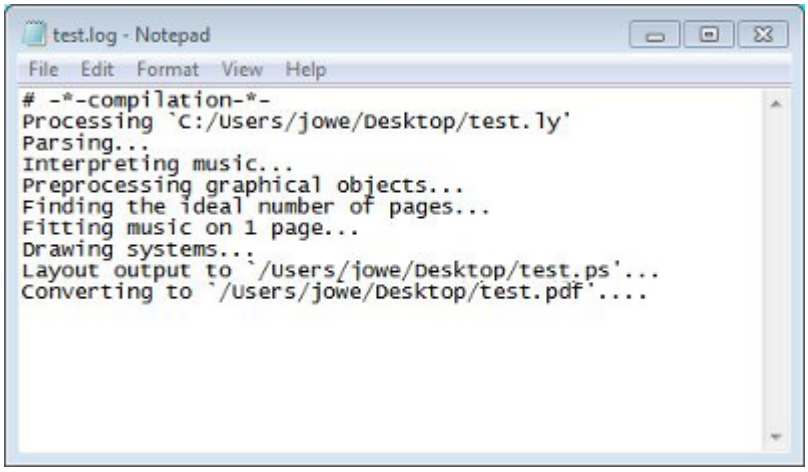
To create a new file, begin by selecting `File > New` from within any previously created file.



or `File > Open` to open and edit any files you have saved before.



You must save any new edits you make before you compile it and if the PDF file is not created, check the log file that will have been created during the compilation attempt, for any errors.



This log file is overwritten each time you compile your LilyPond file.

The PS file is used internally by LilyPond to create the PDF file and can be ignored. It also gets overwritten each time you compile your file.

If you are viewing your file in a PDF viewer, then you must close the PDF if you wish to make a new compilation as it may fail to create the new PDF while it is still being viewed.

1.1.4 Linea di comando

> **Nota:** These instructions assume that you are familiar with command-line programs. If you are using any of the programs described in Sezione "Easier editing" in *Informazioni generali*, please consult the documentation for those programs if you have any problems compiling a file.

Step 1. Create your '`.ly`' file

Create a text file called '`test.ly`' and enter:

```
\version "2.14.0"
{
c' e' g' e'
}
```

Step 2. Compile (with command-line)

To process '`test.ly`', type the following at the command prompt:

```
lilypond test.ly
```

You will see something resembling:

```
GNU LilyPond 2.14.0
Processing `test.ly'
Parsing...
Interpreting music...
Preprocessing graphical objects...
Solving 1 page-breaking chunks...[1: 1 pages]
Drawing systems...
Layout output to `test.ps'...
Converting to `./test.pdf'...
```

Step 3. View output

You may view or print the resulting '`test.pdf`'.
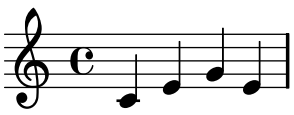
1.2 Come scrivere i file di input

Questa sezione introduce le basi della sintassi di LilyPond e ha l'obiettivo di aiutarti ad iniziare a scrivere i file di input.

1.2.1 Notazione semplice

LilyPond aggiungerà automaticamente alcuni elementi della notazione. Nell'esempio seguente, abbiamo specificato soltanto quattro note, ma LilyPond ha aggiunto una chiave, il tempo e le durate.

```
{
  c' e' g' e'
}
```
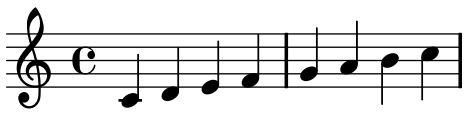


Questo comportamento può essere modificato, ma nella maggior parte dei casi questi valori sono utili.

Altezze

Glossario musicale: Sezione “pitch” in *Glossario Musicale*, Sezione “interval” in *Glossario Musicale*, Sezione “scale” in *Glossario Musicale*, Sezione “middle C” in *Glossario Musicale*, Sezione “octave” in *Glossario Musicale*, Sezione “accidental” in *Glossario Musicale*.
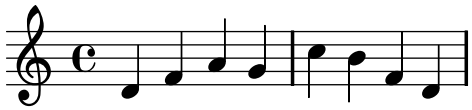
Il modo più semplice per inserire le note è usare il modo `\relative` (relativo). In questo modo, l’ottava viene scelta automaticamente in base al principio per cui la nota che segue deve essere posizionata vicino a quella precedente, ovvero deve essere posizionata nell’ottava che si trova entro tre spazi di pentagramma dalla nota precedente. Per iniziare, scriveremo il pezzo musicale più elementare, una *scala*, in cui ogni nota si trova entro una distanza di appena uno spazio di pentagramma dalla nota precedente.

```
% set the starting point to middle C
\relative c' {
  c d e f
  g a b c
}
```



La nota iniziale è un *Do centrale*. Ogni nota successiva viene posta il più vicino possibile alla nota precedente – in altre parole, la prima `c` è il Do più vicino al Do centrale. Questo è seguito dal Re più vicino alla nota precedente. Possiamo creare melodie che hanno intervalli più ampi, sempre usando soltanto il modo `\relative`:
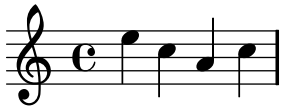
```
\relative c' {
  d f a g
  c b f d
}
```



Non è necessario che la prima nota della melodia inizi con la nota che specifica l’altezza iniziale. Nell’esempio precedente, la prima nota – il `Re` – è il Re più vicino al Do centrale.
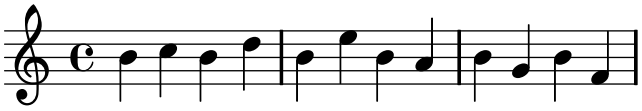
Se si aggiungono (o si rimuovono) apostrofi `'` o virgole `,` dal comando ‘`\relative c'`’, possiamo cambiare l’ottava di partenza:

```
% one octave above middle C
\relative c'' {
  e c a c
}
```



Il modo relativo all’inizio può apparire disorientante, ma è il sistema più semplice per inserire gran parte delle melodie. Vediamo come questo calcolo relativo funziona in pratica. Se si parte da un Si, che si trova sulla linea centrale in chiave di violino, si possono raggiungere un Do, un Re e un Mi entro 3 spazi di pentagramma andando in su, e un La, un Sol e un Fa entro 3 spazi di pentagramma andando in giù. Quindi se la nota che segue il Si è un Do, un Re o un Mi, si troverà sopra il Si, mentre il La, il Sol o il Fa si troveranno sotto.
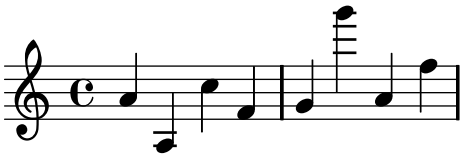
```
\relative c'' {
  b c  % c is 1 staff space up, so is the c above
  b d  % d is 2 up or 5 down, so is the d above
  b e  % e is 3 up or 4 down, so is the e above
  b a  % a is 6 up or 1 down, so is the a below
  b g  % g is 5 up or 2 down, so is the g below
  b f  % f is 4 up or 3 down, so is the f below
}
```



Accade esattamente lo stesso quando qualcuna di queste note ha un diesis o un bemolle. Le *Alterazioni* sono **completamente ignorate** nel calcolo della posizione relativa. Viene fatto lo stesso conteggio dello spazio del pentagramma da una nota in una qualsiasi altra posizione sul pentagramma.

Per inserire intervalli che sono più ampi di tre spazi di pentagramma, si può alzare l'*ottava* aggiungendo al nome della nota una virgoletta ad apice singolo `'` (o apostrofo). Si può abbassare l'ottava aggiungendo al nome della nota una virgola `,`.

```
\relative c'' {
  a a, c' f,
  g g'' a,, f'
}
```



Per cambiare una nota di due (o più!) ottave, si usano multipli di `''` o `,,` – ma stai attento a usare due virgolette ad apici semplici `''` e non quelle a doppi apici `"` !

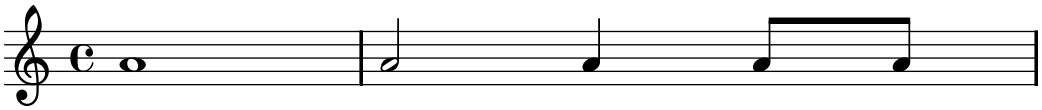
Durate (ritmi)
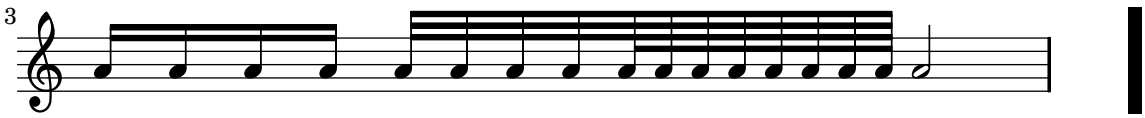
Glossario musicale: Sezione "beam" in *Glossario Musicale*, Sezione "duration" in *Glossario Musicale*, Sezione "whole note" in *Glossario Musicale*, Sezione "half note" in *Glossario Musicale*, Sezione "quarter note" in *Glossario Musicale*, Sezione "dotted note" in *Glossario Musicale*.

La *durata* di una nota viene indicata da un numero che segue il nome della nota: `1` per una *semibreve*, `2` per una *minima*, `4` per una *semiminima* e così via. Le *Code* vengono aggiunte in automatico.

Se non si specifica una durata, si usa la durata precedente anche per la nota successiva. La durata predefinita della prima nota è di un quarto.

```
\relative c'' {
  a1
  a2 a4 a8 a
  a16 a a a a32 a a a a64 a a a a a a a a2
}
```





Per creare *note puntate*, aggiungi un punto `.` al numero di durata. La durata di una nota puntata deve essere dichiarata esplicitamente (cioè con un numero).

```
\relative c'' {
  a a a4. a8
  a8. a16 a a8. a8 a4.
}
```
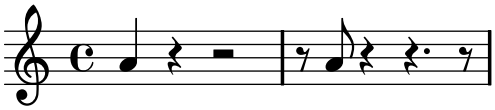


Pause

Glossario musicale: Sezione “rest” in *Glossario Musicale*.

Una *pausa* viene inserita proprio come una nota ma col nome `r` :

```
\relative c'' {
  a4 r r2
  r8 a r4 r4. r8
}
```
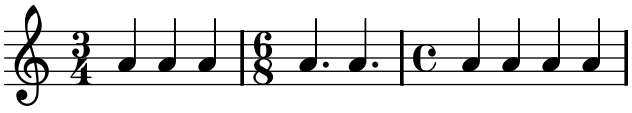


Indicazione di tempo

Glossario musicale: Sezione “time signature” in *Glossario Musicale*.

Il *tempo* si imposta con il comando `\time`:

```
\relative c'' {
  \time 3/4
  a4 a a
  \time 6/8
  a4. a
  \time 4/4
  a4 a a a
}
```



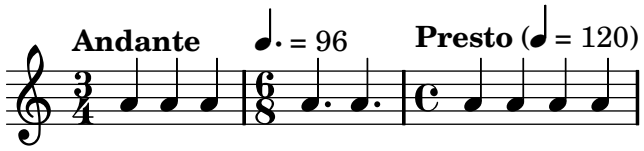
Tempo marks

Glossario musicale: Sezione “tempo indication” in *Glossario Musicale*, Sezione “metronome” in *Glossario Musicale*.

L’ *indicazione di tempo* e l’ *indicazione metronomica* si impostano col comando `\tempo`:

```
\relative c'' {
  \time 3/4
  \tempo "Andante"
  a4 a a
  \time 6/8
```

```
  \tempo 4. = 96
  a4. a
  \time 4/4
  \tempo  "Presto" 4 = 120
  a4 a a a
}
```
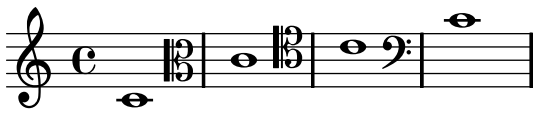


Chiave

Glossario musicale: Sezione “clef” in *Glossario Musicale*.

La *chiave* si imposta con il comando `\clef`:
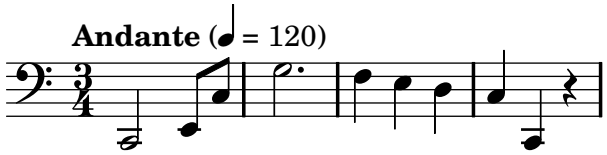
```
\relative c' {
  \clef treble
  c1
  \clef alto
  c1
  \clef tenor
  c1
  \clef bass
  c1
}
```



Tutto insieme

Ecco un piccolo esempio che mostra tutti questi elementi insieme:

```
\relative c, {
  \clef "bass"
  \time 3/4
  \tempo "Andante" 4 = 120
  c2 e8 c'
  g'2.
  f4 e d
  c4 c, r
}
```



Vedi anche

Guida alla notazione: Sezione “Writing pitches” in *Guida alla Notazione*, Sezione “Writing rhythms” in *Guida alla Notazione*, Sezione “Writing rests” in *Guida alla Notazione*, Sezione “Time signature” in *Guida alla Notazione*, Sezione “Clef” in *Guida alla Notazione*.

1.2.2 Lavorare sui file di input

I file di input di LilyPond sono simili ai file sorgenti di molti comuni linguaggi di programmazione. Contengono una dichiarazione di versione, sono sensibili alle maiuscole, e in generale gli spazi bianchi vengono ignorati. Le espressioni musicali si formano con parentesi graffe `{ }`, e i commenti sono indicati con `%` o `%{ ... %}` .

Se le frasi precedenti ti sembrano assurde, non preoccuparti! Spiegheremo cosa significano tutti questi termini:

- **Dichiarazione di versione**: Ogni file LilyPond dovrebbe contenere una dichiarazione di versione. Una dichiarazione di versione è una linea che indica la versione di LilyPond per la quale il file è stato scritto, come nel seguente esempio:

  ```
  \version "2.14.1"
  ```

  Per convenzione, la dichiarazione di versione viene posta all'inizio del file LilyPond.

  La dichiarazione di versione è importante per almeno due ragioni. Primo, permette l'aggiornamento automatico del file di input file via via che la sintassi di LilyPond cambia. Secondo, indica la versione di LilyPond richiesta per compilare il file.

  Se la dichiarazione di versione è omessa dal file di input, LilyPond stampa un avvertimento durante la compilazione del file.

- **Sensibile alle maiuscole**: distingue tra lettere in minuscolo (es: `a, b, s, t`) o in maiuscolo (es: `A, B, S, T`). Le note sono in minuscolo: `{ c d e }` è un input valido; `{ C D E }` causerà un messaggio di errore.

- **Insensibile agli spazi**: non importa quanti spazi (o tabulazioni o nuove linee) aggiungi. '`{ c4 d e }`' ha lo stesso significato di '`{ c4 d e }`' e:

  ```
  { c4                        d
                        e   }
  ```

  Certo, l'esempio precedente è scomodo da leggere. Una buona regola pratica è indentare i blocchi di codice con una tabulazione o due spazi:

  ```
  {
    c4 d e
  }
  ```

  Tuttavia, uno spazio bianco *è* necessario per separare molti elementi sintattici gli uni dagli altri. In altre parole, lo spazio bianco può essere *aggiunto*, ma non può essere *eliminato*. Dal momento che uno spazio bianco mancante può dare adito a strani errori, si consiglia di inserire sempre uno spazio bianco prima e dopo ogni elemento sintattico, ad esempio, prima e dopo ogni parentesi graffa.

- **Espressioni**: ogni parte dell'input di LilyPond deve avere `{` **parentesi graffe** `}` intorno. Queste parentesi dicono a LilyPond che l'input costituisce un'espressione musicale singola, proprio come le parentesi `()` in matematica. Per evitare ambiguità, le parentesi dovrebbero essere racchiuse tra spazi, a meno che non si trovino all'inizio o alla fine di una linea.

  Anche un comando di LilyPond seguito da un'espressione semplice in parentesi (come '`\relative c' { ... }`') conta come un'espressione musicale singola.

- **Commenti**: un commento è un appunto per il lettore umano dell'input musicale; viene ignorato quando l'input viene analizzato, dunque non ha alcun effetto sull'output finale. Ci sono due tipi di commenti. Il simbolo di percentuale `%` introduce un commento di linea; tutto quello che sta dopo `%` su quella linea verrà ignorato. Per convenzione, un commento di linea viene posto *sopra* il codice a cui si riferisce.

  ```
  a4 a a a
  % this comment refers to the Bs
  b2 b
  ```

Un commento di blocco segna come commento un'intera sezione di input musicale. Tutto ciò che è compreso tra `%{` e `%}` viene ignorato. Tuttavia, i commenti di blocco non si 'annidano'. Ovvero non si può inserire un commento di blocco dentro a un altro commento di blocco. Se ci provi, il primo `%}` interromperà *entrambi* i commenti di blocco. Il seguente frammento mostra gli usi possibili per i commenti:

```
% notes for twinkle twinkle follow
  c4 c g' g a a g2

%{
  This line, and the notes below are ignored,
  since they are in a block comment.

  f4 f e e d d c2
%}
```

1.3 Gestire gli errori

Talvolta LilyPond non genera l'output che vorresti. Questa sezione fornisce alcuni link che possono aiutarti a risolvere i problemi che potrai incontrare.

1.3.1 Consigli generali per la risoluzione dei problemi

La risoluzione dei problemi in LilyPond può essere un compito impegnativo per le persone che sono abituate a un'interfaccia grafica, perché è possibile creare dei file di input non validi. Quando questo accade, un approccio logico è il modo migliore per identificare e risolvere il problema. Alcune linee guida che ti aiutano a imparare questo sono indicate in Sezione "Troubleshooting" in *Uso del Programma*.

1.3.2 Alcuni errori comuni

Ci sono alcuni errori comuni che sono difficili da risolvere basandosi soltanto sui messaggi di errore che compaiono. Questi sono descritti in Sezione "Common errors" in *Uso del Programma*.

1.4 Come leggere i manuali

Questa sezione spiega come leggere la documentazione in modo efficiente, e introduce anche alcune utili funzionalità interattive che sono disponibili nella versione online della documentazione.

1.4.1 Materiale omesso

L'input di LilyPond deve essere compreso dai segni `{ }` o da '`\relative c'' { ... }`', come abbiamo visto in Sezione 1.2.2 [Lavorare sui file di input], pagina 18. Nella parte restante di questo manuale, gran parte degli esempi ometterà questi segni. Per replicare gli esempi, puoi copiare e incollare l'input mostrato, ma **devi** aggiungere '`\relative c'' { ... }`' in questo modo:

```
\relative c'' {
  ...example goes here...
}
```

Perché omettere le parentesi? Gran parte degli esempi in questo manuale possono essere inseriti nel mezzo di un pezzo più ampio. Per questi esempi, non ha senso includere '`\relative c'' { ... }`' – non si deve mettere un `\relative` dentro un altro `\relative`! Se includessimo '`\relative c'' { ... }`' in ogni esempio, non potresti copiare un piccolo esempio della documentazione e incollarlo dentro a un tuo brano più esteso. La maggior parte delle persone vuole aggiungere del materiale a un brano esistente, per questo abbiamo strutturato il manuale in questo modo.

Ricorda che i file LilyPond dovrebbero avere una dichiarazione di versione (`\version`). Nei manuali la dichiarazione è omessa perché gli esempi sono frammenti di codice e non file completi. Ma dovresti prendere l'abitudine di includerla nei tuoi file.

1.4.2 Esempi cliccabili

> **Nota:** Queste funzionalità sono disponibili soltanto nei manuali in HTML.

Molte persone imparano ad usare un programma provando e smanettando. Questo è possibile anche con LilyPond. Se clicchi su un'immagine nella versione HTML di questo manuale, vedrai l'esatto input LilyPond usato per generare quell'immagine. Provalo su questa immagine:



Click here.

Tagliando e copiando quel che si trova nella sezione "ly snippet", puoi ricavare un modello di partenza per sperimentare. Per vedere esattamente lo stesso output (larghezza della linea e tutto il resto), copia tutto da "Start cut-&-pastable section" alla fine del file.

1.4.3 Panoramica dei manuali

La documentazione su LilyPond è vasta. I nuovi utenti talvolta sono confusi su quali parti dovrebbero leggere, e ogni tanto saltano la lettura di parti fondamentali.

> **Nota:** Non saltare le parti importanti della documentazione. Altrimenti ti risulterà molto più difficile comprendere le sezioni successive.

- **Prima di fare** *qualsiasi* **tentativo**: leggi il Capitolo 1 [Tutorial], pagina 1 e la Capitolo 2 [Notazione comunemente utilizzata], pagina 21 del Manuale di apprendimento. Se ti imbatti in termini musicali che non conosci, cercali nel Sezione "Glossario" in *Glossario Musicale*.
- **Prima di tentare di scrivere un pezzo musicale completo**: leggi i Capitolo 3 [Concetti fondamentali], pagina 41 del Manuale di apprendimento. Dopo puoi dare un'occhiata alle sezioni rilevanti della Sezione "Notation reference" in *Guida alla Notazione*.
- **Prima di cercare di cambiare l'output predefinito**: leggi il Capitolo 4 [Tweaking output], pagina 88 del Manuale di apprendimento.
- **Prima di iniziare un grande progetto**: leggi il documento Sezione "Suggestions for writing files" in *Uso del Programma* del manuale di Uso del programma.

2 Notazione comunemente utilizzata

Questo capitolo spiega come creare delle bella musica a stampa che faccia uso della notazione musicale comunemente utilizzata, seguendo il materiale esposto in Capitolo 1 [Tutorial], pagina 1.
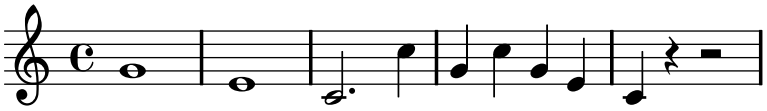
2.1 Notazione su un solo pentagramma

Questa sezione presenta la notazione comunemente usata per una singola voce su un solo pentagramma.

2.1.1 Controlli di battuta

I *controlli di battuta*, pur se non strettamente necessari, dovrebbero essere usati nel codice di input per mostrare dove si vuole che cadano le stanghette. Vengono inseriti col simbolo della barra verticale, `|`. Grazie ai controlli di battuta, il programma può verificare che tu abbia inserito delle durate che facciano sì che ogni misura raggiunga la giusta durata. I controlli di battuta rendono anche il codice di input più facile da leggere, perché aiutano a tenere tutto in ordine.

```
g1 | e1 | c2. c'4 | g4 c g e | c4 r r2 |
```



Vedi anche

Guida alla notazione: Sezione “Bar and bar number checks” in *Guida alla Notazione*.

2.1.2 Alterazioni e armature di chiave
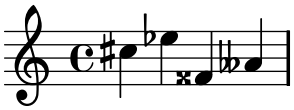
> **Nota:** I nuovi utenti sono spesso confusi riguardo a questi concetti – leggi il messaggio di avviso in fondo a questa pagina, soprattutto se non hai una buona conoscenza della teoria musicale!

Alterazioni

Glossario musicale: Sezione “sharp” in *Glossario Musicale*, Sezione “flat” in *Glossario Musicale*, Sezione “double sharp” in *Glossario Musicale*, Sezione “double flat” in *Glossario Musicale*, Sezione “accidental” in *Glossario Musicale*.

Un *diesis* si ottiene aggiungendo il suffisso `is` al nome della nota, e un *bemolle* aggiungendo `es`. Come puoi immaginare, un *doppio diesis* o un *doppio bemolle* si ottengono aggiungendo `isis` o `eses`. Questa sintassi deriva dalle convenzioni per i nomi delle note presenti nelle lingue nordiche e germaniche, come il tedesco e l’olandese. Per usare altri nomi per le *alterazioni*, si veda Sezione “Nomi delle note in altre lingue” in *Guida alla Notazione*.

```
cis4 ees fisis, aeses
```
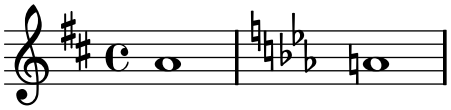


Armature di chiave

Glossario musicale: Sezione “key signature” in *Glossario Musicale*, Sezione “major” in *Glossario Musicale*, Sezione “minor” in *Glossario Musicale*.

L’*armatura di chiave* viene impostata col comando `\key` seguito da un’altezza e da `\major` o `\minor`.

```
\key d \major
a1 |
\key c \minor
a1 |
```



Attenzione: armature di chiave e altezze
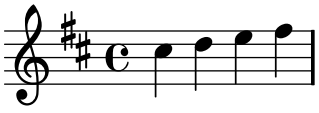
Glossario musicale: Sezione “accidental” in *Glossario Musicale*, Sezione “key signature” in *Glossario Musicale*, Sezione “pitch” in *Glossario Musicale*, Sezione “flat” in *Glossario Musicale*, Sezione “natural” in *Glossario Musicale*, Sezione “sharp” in *Glossario Musicale*, Sezione “transposition” in *Glossario Musicale*.

Per determinare se stampare o meno un’*alterazione*, LilyPond esamina le altezze e l’*armatura di chiave*. L’armatura di chiave influisce soltanto sulle alterazioni che vengono *stampate*, non sull’*altezza* della nota! Questa è una caratteristica che spesso causa confusione ai nuovi utenti, quindi la spiegheremo più dettagliatamente.

LilyPond fa una netta distinzione tra contenuto musicale e layout. L’alterazione di (*bemolle*, il segno di *bequadro* o il *diesis*) di una nota fanno parte dell’altezza, e sono quindi contenuto musicale. Se un’alterazione (un segno *stampato* di bemolle, bequadro o diesis) venga posta oppure no di fronte alla nota corrispondente è una questione di layout. Il layout segue delle regole, dunque le alterazioni sono stampate automaticamente in base a queste regole. Le altezze nella tua musica sono opere d’arte, quindi non verranno aggiunte in automatico: sei tu a dover inserire la nota che vuoi sentire.

In questo esempio:
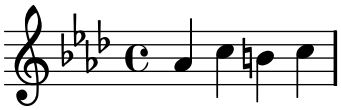
```
\key d \major
cis4 d e fis
```



Nessuna nota ha un’alterazione rispetto all’armatura di chiave, ma devi comunque aggiungere `is` e scrivere `cis` e `fis` nel file di input.

Il codice `b` non significa “stampa un punto nero esattamente nella linea centrale del pentagramma.” Piuttosto, significa “c’è una nota con altezza Si-bequadro.” Nella tonalità di La bemolle maggiore, *deve* avere un’alterazione:

```
\key aes \major
aes4 c b c
```



Se l'esempio precedente sembra poco chiaro, considera questo: se tu stessi suonando un pianoforte, quale tasto premeresti? Se premi un tasto nero, allora *devi* aggiungere `-is` o `-es` al nome della nota!

Aggiungere esplicitamente tutte le alterazioni richiederà un po' più di lavoro in fase di scrittura, ma il vantaggio è che la *trasposizione* è più semplice, e le alterazioni possono essere prodotte usando diverse convenzioni. Per alcuni esempi che mostrano come sia possibile produrre delle alterazioni in base a regole diverse, si veda Sezione "Alterazioni automatiche" in *Guida alla Notazione*.

Vedi anche

Guida alla notazione: Sezione "Note names in other languages" in *Guida alla Notazione*, Sezione "Accidentals" in *Guida alla Notazione*, Sezione "Automatic accidentals" in *Guida alla Notazione*, Sezione "Key signature" in *Guida alla Notazione*.
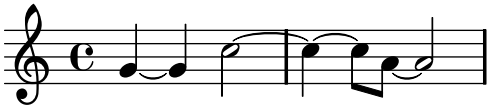
2.1.3 Legature di valore e di portamento

Legature di valore

Glossario musicale: Sezione "tie" in *Glossario Musicale*.

Una *legatura di valore* si ottiene apponendo una tilde `~` alla prima nota della legatura.

```
g4~ g c2~ | c4~ c8 a~ a2 |
```
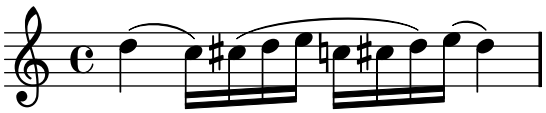


Legature di portamento

Glossario musicale: Sezione "slur" in *Glossario Musicale*.

Una *legatura di portamento* è una linea curva che collega più note. La nota iniziale e quella finale sono indicate rispettivamente con `(` e `)`.

```
d4( c16) cis( d e c cis d) e( d4)
```
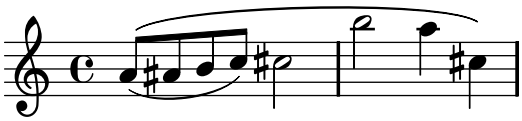


Legature di fraseggio

Glossario musicale: Sezione "slur" in *Glossario Musicale*, Sezione "phrasing" in *Glossario Musicale*.

Le legature di portamento che indicano una *frase* più lunga possono essere inserite con `\(` e `\)`. E' possibile avere allo stesso tempo sia le *legature di portamento* sia le legature di frase, ma non si possono avere simultaneamente diverse legature di portamento, o diverse legature di frase.

```
a8(\( ais b c) cis2 b'2 a4 cis,\)
```
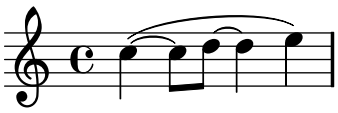


Attenzione: legature di portamento vs. legature di valore

Glossario musicale: Sezione “articulation” in *Glossario Musicale*, Sezione “slur” in *Glossario Musicale*, Sezione “tie” in *Glossario Musicale*.

Una *legatura di portamento* ha lo stesso aspetto di una *legatura di valore*, ma un significato diverso. Una legatura di valore rende semplicemente la nota più lunga, e può essere usata solo con coppie di note della stessa altezza. Le legature di portamento indicano l’*articolazione* delle note, e possono essere usate con ampi gruppi di note. Legature di valore e legature di portamento possono essere annidate le une dentro le altre.

```
c4~( c8 d~ d4 e)
```



Vedi anche

Guida alla notazione: Sezione “Ties” in *Guida alla Notazione*, Sezione “Slurs” in *Guida alla Notazione*, Sezione “Phrasing slurs” in *Guida alla Notazione*.
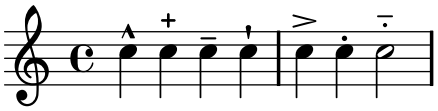
2.1.4 Articolazione e dinamica

Articolazioni

Glossario musicale: Sezione “articulation” in *Glossario Musicale*.

Le *articolazioni* di uso comune possono essere aggiunte a una nota con una lineetta `-` e un singolo carattere:

```
c4-^ c-+ c-- c-|
c4-> c-. c2-_
```
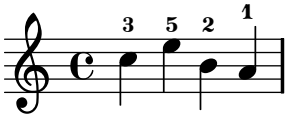


Diteggiature

Glossario musicale: Sezione “fingering” in *Glossario Musicale*.
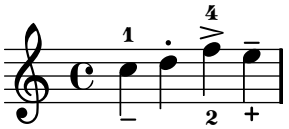
Analogalmente, le indicazioni di *diteggiatura* possono essere aggiunte a una nota con una lineetta (`-`) e il numero che si vuole visualizzare:

```
c4-3 e-5 b-2 a-1
```



Articolazioni e diteggiature solitamente sono posizionate verticalmente in automatico, ma si può specificarne la direzione sostituendo la lineetta (`-`) con `^` (su) o `_` (giù). Si possono usare anche articolazioni multiple sulla stessa nota. Tuttavia, nella maggior parte dei casi è meglio lasciare che sia LilyPond a determinare le direzioni delle articolazioni.

```
c4_-^1 d^. f^4_2-> e^-_+
```
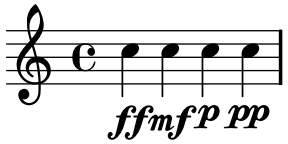


Dinamica

Glossario musicale: Sezione “dynamics” in *Glossario Musicale*, Sezione “crescendo” in *Glossario Musicale*, Sezione “decrescendo” in *Glossario Musicale*.
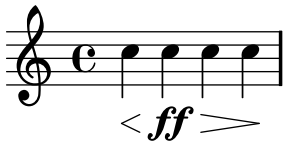
I segni di *dinamica* si ottengono aggiungendo alla nota i simboli (preceduti da un segno di barra invertita, o backslash):

```
c4\ff c\mf c\p c\pp
```



Crescendi e *decrescendi* iniziano coi comandi `\<` e `\>`. Il seguente segno di dinamica, ad esempio `\ff`, terminerà il (de)crescendo, oppure può essere usato il comando `\!`:

```
c4\< c\ff\> c c\!
```



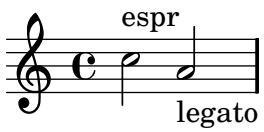
Vedi anche

Guida alla notazione: Sezione “Articulations and ornamentations” in *Guida alla Notazione*, Sezione “Fingering instructions” in *Guida alla Notazione*, Sezione “Dynamics” in *Guida alla Notazione*.

2.1.5 Aggiungere il testo

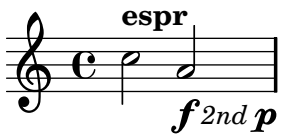
Puoi aggiungere del testo nei tuoi spartiti:

```
c2^"espr" a_"legato"
```



Per aggiungere delle formattazioni puoi usare il comando `\markup`:

```
c2^\markup{ \bold espr}
a2_\markup{
  \dynamic f \italic \small { 2nd } \hspace #0.1 \dynamic p
}
```



Vedi anche
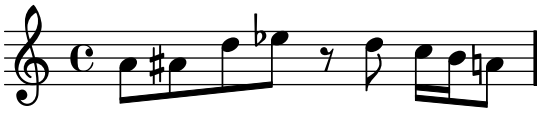
Guida alla notazione: Sezione “Writing text” in *Guida alla Notazione*.

2.1.6 Code automatiche e manuali

Glossario musicale: Sezione “beam” in *Glossario Musicale*.
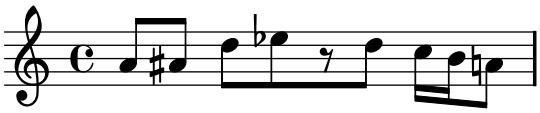
Tutte le *travature* vengono disegnate automaticamente:

```
a8 ais d ees r d c16 b a8
```
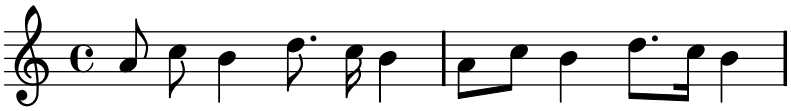


Se non ti piace il modo in cui vengono prodotte automaticamente le travature, è possibile sovrascriverle manualmente. Per correggere soltanto una singola travatura, indica la prima nota da raggruppare con `[` e l’ultima con `]`.

```
a8[ ais] d[ ees r d] c16 b a8
```



Se desideri disattivare il raggruppamento automatico, interamente o per un’ampia sezione del brano, usa il comando `\autoBeamOff` per disattivare il raggruppamento automatico e `\autoBeamOn` per riattivarlo.

```
\autoBeamOff
a8 c b4 d8. c16 b4 |
\autoBeamOn
a8 c b4 d8. c16 b4 |
```



Vedi anche

Guida alla notazione: Sezione “Automatic beams” in *Guida alla Notazione*, Sezione “Manual beams” in *Guida alla Notazione*.
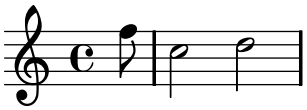
2.1.7 Comandi di tempo avanzati

Battuta parziale

Glossario musicale: Sezione “anacrusis” in *Glossario Musicale*.

Un’ *anacrusi*) può essere inserita con la parola chiave `\partial`. Questa è seguita da una durata: `\partial 4` è un’anacrusi di semiminima e `\partial 8` di croma.

```
\partial 8 f8 |
c2 d |
```

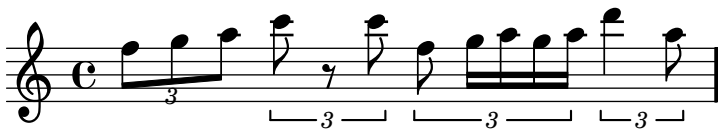


Gruppi irregolari

Glossario musicale: Sezione "note value" in *Glossario Musicale*, Sezione "triplet" in *Glossario Musicale*.

I *gruppi irregolari* vengono preceduti dalla parola chiave `\times`. Questa richiede due argomenti: una frazione e un frammento di musica. La durata del frammento viene moltiplicata per la frazione. Le terzine fanno sì che le note occupino 2/3 della loro durata, quindi una *terzina* ha 2/3 come frazione:

```
\times 2/3 { f8 g a }
\times 2/3 { c8 r c }
\times 2/3 { f,8 g16[ a g a] }
\times 2/3 { d4 a8 }
```
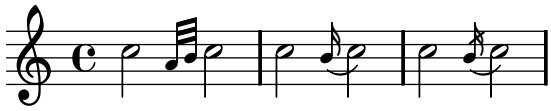


Abbellimenti

Glossario musicale: Sezione "grace notes" in *Glossario Musicale*, Sezione "acciaccatura" in *Glossario Musicale*, Sezione "appoggiatura" in *Glossario Musicale*.

Gli *abbellimenti* sono creati col comando `\grace`, ma possono essere creati anche ponendo davanti a un'espressione musicale le parole chiave `\appoggiatura` o `\acciaccatura`:

```
c2 \grace { a32[ b] } c2 |
c2 \appoggiatura b16 c2 |
c2 \acciaccatura b16 c2 |
```



Vedi anche

Guida alla notazione: Sezione "Grace notes" in *Guida alla Notazione*, Sezione "Tuplets" in *Guida alla Notazione*, Sezione "Upbeats" in *Guida alla Notazione*.

2.2 Note simultanee

Questa sezione spiega come inserire più note simultanee: molteplici strumenti, molteplici righi di pentagramma per un singolo strumento (es: piano), e accordi.

In musica per polifonia si intende la presenza di più di una voce in un brano. In LilyPond per polifonia si intende la presenza di più di una voce sullo stesso pentagramma.

2.2.1 Espressioni musicali
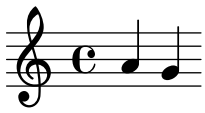
Nei file di input di LilyPond, la musica è rappresentata dalle *espressioni musicali*. Anche una singola nota è un'espressione musicale:

```
a4
```
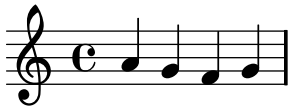


Se si racchiude una nota tra parentesi si crea un'*espressione musicale composta*. In questo esempio abbiamo creato un'espressione musicale composta da due note:

```
{ a4 g4 }
```



Se si mette un gruppo di espressioni musicali (es: note) tra parentesi, significa che sono in sequenza (ovvero, ciascuna espressione segue la precedente). Il risultato è un'altra espressione musicale:

```
{ { a4 g } f4 g }
```



Analogia: le espressioni matematiche

Questo meccanismo è analogo a quello delle formule matematiche: una grande formula può essere creata creando piccole formule. Tali formule sono chiamate espressioni, e possono contenere altre espressioni, così che sia possibile costruire a piacere espressioni grandi e complesse. Ad esempio,

```
1

1 + 2

(1 + 2) * 3

((1 + 2) * 3) / (4 * 5)
```

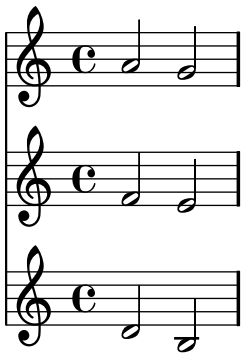
Questa è una sequenza di espressioni, dove ogni espressione è racchiusa in quella successiva (più grande). Le espressioni più semplici sono i numeri, e quelle più grandi si ottengono combinando le espressioni con gli operatori (come `+`, `*` e `/`) e le parentesi. Come le espressioni matematiche, le espressioni musicali possono essere annidate a qualsivoglia grado di profondità, e questo è indispensabile per musica complessa come le partiture polifoniche.

Espressioni musicali simultanee: righi multipli

Glossario musicale: Sezione "polyphony" in *Glossario Musicale*.

Questa tecnica è utile per la musica *polifonica*. Per inserire della musica che abbia più voci o più linee di pentagramma, basta combinare le espressioni in parallelo. Per indicare che le due voci devono suonare contemporaneamente, basta inserire una combinazione simultanea di espressioni musicali. Un' espressione musicale 'simultanea' si forma racchiudendo le espressioni all'interno di `<<` e `>>`. Nel seguente esempio, tre sequenze (tutte contenenti due note separate) vengono combinate in simultanea:

```
\relative c'' {
  <<
    { a2 g }
    { f2 e }
    { d2 b }
  >>
}
```
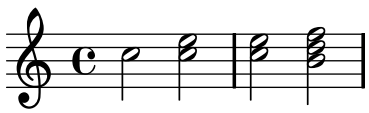


Si noti che abbiamo indentato ogni livello dell'input con una diversa quantità di spazi. LilyPond non si preoccupa di quanto spazio c'è all'inizio di una linea, tuttavia indentare il codice di LilyPond in questo modo lo rende molto più semplice da leggere per l'essere umano.

> **Nota:** ogni nota è relativa alla nota precedente nell'input, e non è relativa al `c''` nel comando `\relative` iniziale.

Espressioni musicali simultanee: rigo singolo

Per determinare il numero di pentagrammi in un brano, LilyPond guarda l'inizio della prima espressione. Se c'è una nota singola, ci sarà un pentagramma; se c'è un'espressione simultanea, ci saranno più pentagrammi. L'esempio seguente mostra un'espressione complessa, ma poiché inizia con una nota singola sarà impostata su un singolo rigo.

```
\relative c'' {
  c2 <<c e>> |
  << { e2 f } { c2 <<b d>> } >>
}
```
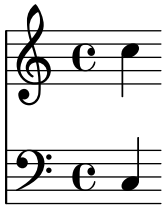


2.2.2 Righi multipli

Come abbiamo visto in Sezione 2.2.1 [Espressioni musicali], pagina 27, i file di input di LilyPond si costruiscono in base alle espressioni musicali. Se la partitura inizia con espressioni musicali simultanee, LilyPond crea più di un rigo musicale. Tuttavia, è più facile vedere quel che accade se creiamo ogni rigo musicale esplicitamente.

Per ottenere più di un rigo musicale, ogni brano musicale che costituisce un rigo è contrassegnato da `\new Staff`. Questi elementi `Staff` vengono poi combinati in parallelo con `<<` e `>>`:

```
\relative c'' {
  <<
    \new Staff { \clef "treble" c4 }
    \new Staff { \clef "bass" c,,4 }
  >>
}
```
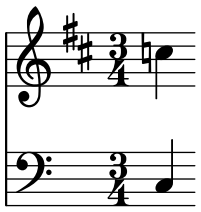


Il comando `\new` introduce un 'contesto di notazione.' Un contesto è un ambiente in cui vengono interpretati gli eventi musicali (come le note o i comandi `\clef`). Nel caso di brani semplici, tali contesti vengono creati automaticamente. Per brani più complessi, è meglio contrassegnare esplicitamente i contesti.

Ci sono vari tipi di contesto. `Score`, `Staff`, e `Voice` gestiscono la notazione melodica, mentre `Lyrics` imposta i testi e `ChordNames` visualizza i nomi degli accordi.

In termini di sintassi, se si inserisce `\new` prima di un'espressione musicale, si crea un'espressione musicale più grande. In questo modo assomiglia al segno di minore in matematica. La formula $(4 + 5)$ è un'espressione, quindi $-(4 + 5)$ è un'espressione più grande.

Le indicazioni di tempo inserite in un rigo musicale si estendono di default a tutti gli altri righi. L'armatura di chiave di un rigo, invece, *non* si estende agli altri. Questo diverso comportamento di default è dovuto al fatto che le partiture con strumenti traspositori sono molto più frequenti delle partiture poliritmiche.

```
\relative c'' {
  <<
    \new Staff { \clef "treble" \key d \major \time 3/4 c4 }
    \new Staff { \clef "bass" c,,4 }
  >>
}
```



2.2.3 Gruppi di pentagrammi
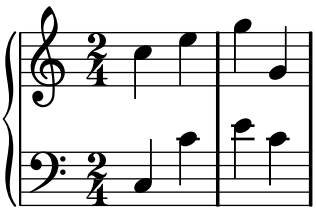
Glossario musicale: Sezione "brace" in *Glossario Musicale*, Sezione "staff" in *Glossario Musicale*, Sezione "system" in *Glossario Musicale*.

La musica per pianoforte viene stampata su due righi musicali collegati con una *graffa*. Produrre un pentagramma di questo tipo è simile all'esempio polifonico in ⟨undefined⟩ [Multiple staves], pagina ⟨undefined⟩. In questo caso, però, l'intera espressione è inserita all'interno di `PianoStaff`:

```
\new PianoStaff <<
  \new Staff ...
  \new Staff ...
>>
```

Ecco un piccolo esempio:

```
\relative c'' {
  \new PianoStaff <<
    \new Staff { \time 2/4 c4 e | g g, | }
    \new Staff { \clef "bass" c,,4 c' | e c | }
  >>
}
```



Altri raggruppamenti di pentagrammi vengono preceduti da `\new GrandStaff`, per le partiture orchestrali, e da `\new ChoirStaff`, per le partiture corali. Ognuno di questi gruppi di pentagramma costituiscono un altro tipo di contesto, un contesto che genera la graffa all'estremità sinistra di ogni sistema e controlla inoltre l'estensione delle linee della battuta.

Vedi anche
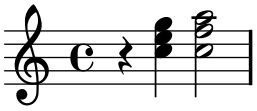
Guida alla notazione: Sezione "Keyboard and other multi-staff instruments" in *Guida alla Notazione*, Sezione "Displaying staves" in *Guida alla Notazione*.

2.2.4 Combinare le note negli accordi

Glossario musicale: Sezione "chord" in *Glossario Musicale*.

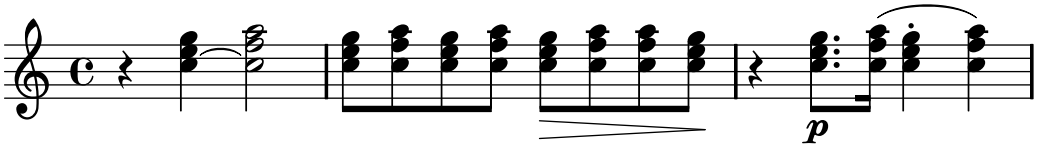
Abbiamo visto in precedenza come le note possano essere combinate in *accordi* racchiudendole tra parentesi a doppi angoli per indicare che sono simultanee. Tuttavia, il modo normale di indicare un accordo è quello di circondare le note con delle parentesi ad angolo *singolo*. Si noti che tutte le note in un accordo devono avere la stessa durata, e che la durata è posta dopo la parentesi chiusa.

```
r4 <c e g> <c f a>2
```



Pensa agli accordi come a qualcosa di equivalente alle note singole: quasi ogni cosa che puoi attaccare a una nota singola può essere attaccata a un accordo, e tutto questo deve stare *fuori* dalle parentesi angolari. Ad esempio, con gli accordi si possono combinare simboli come le travature e le legature di valore. Questi devono essere posti fuori dalle parentesi angolari.

```
r4 <c e g>~ <c f a>2 |
<c e g>8[ <c f a> <c e g> <c f a>] <c e g>\>[ <c f a> <c f a> <c e g>]\! |
r4 <c e g>8.\p <c f a>16( <c e g>4-. <c f a>) |
```



Vedi anche

Guida alla notazione: Sezione "Chorded notes" in *Guida alla Notazione*.

2.2.5 Polifonia su un singolo rigo

La musica polifonica in lilypond, per quanto non difficile, fa riferimento a concetti non ancora affrontati, quindi non la presenteremo subito. Saranno le sezioni successive a introdurre questi concetti e a spiegarli via via.

Vedi anche

Manuale di Apprendimento: Sezione 3.2 [Le voci contengono la musica], pagina 48.

Guida alla notazione: Sezione "Simultaneous notes" in *Guida alla Notazione*.

2.3 Canzoni
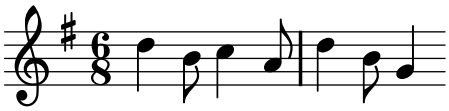
Questa sezione presenta la musica vocale e gli spartiti di semplici canzoni.

2.3.1 Impostare canzoni semplici

Glossario musicale: Sezione “lyrics” in *Glossario Musicale*.
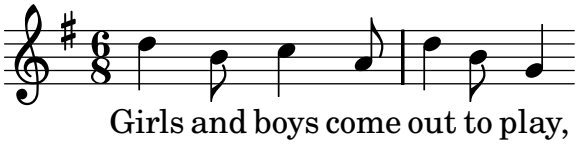
Questo è l’inizio della melodia di una filastrocca, *Girls and boys come out to play*:

```
\relative c'' {
  \key g \major
  \time 6/8
  d4 b8 c4 a8 | d4 b8 g4
}
```



I *testi* possono essere collegati a queste note, combinandoli a esse per mezzo della parola chiave `\addlyrics`. I testi si inseriscono separando ogni sillaba con uno spazio.

```
<<
  \relative c'' {
    \key g \major
    \time 6/8
    d4 b8 c4 a8 | d4 b8 g4
  }
  \addlyrics {
    Girls and boys come | out to play,
  }
>>
```



> **Nota:** È fondamentale che l’ultima sillaba sia dalla parentesi graffa che chiude con uno spazio o una nuova linea, altrimenti si presumerà che sia parte della sillaba, e questo causerà un errore poco chiaro, vedi Sezione “Apparent error in ../ly/init.ly” in *Uso del Programma*.

Si notino le doppie parentesi ad angolo `<< ... >>` che circondano l’intero brano per indicare che la musica e il testo devono trovarsi in simultanea.

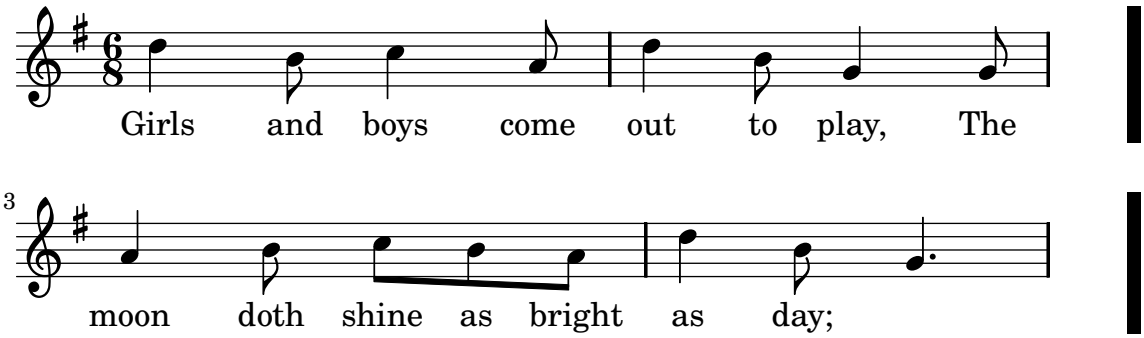
2.3.2 Allineare il testo alla melodia

Glossario musicale: Sezione “melisma” in *Glossario Musicale*, Sezione “extender line” in *Glossario Musicale*.

La prossima riga della filastrocca è *The moon doth shine as bright as day*. Aggiungiamola:

```
<<
  \relative c'' {
    \key g \major
```

```
    \time 6/8
    d4 b8 c4 a8 | d4 b8 g4 g8 |
    a4 b8 c b a | d4 b8 g4. |
  }
  \addlyrics {
    Girls and boys come | out to play,
    The | moon doth shine as | bright as day; |
  }
>>
```
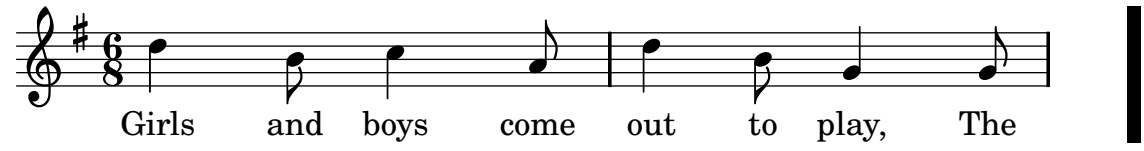


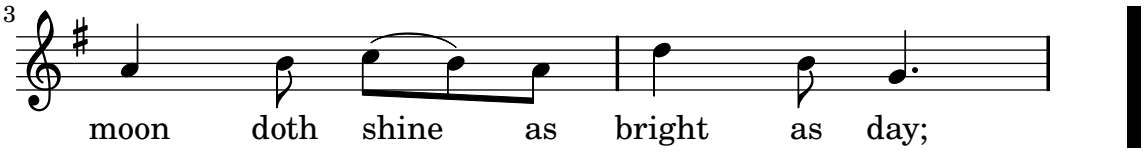
Se compili il codice dell'esempio precedente, dovresti vedere alcuni avvisi nell'output del terminale:

```
song.ly:12:29: warning: barcheck failed at: 5/8
    The | moon doth shine as
                              | bright as day; |
song.ly:12:46: warning: barcheck failed at: 3/8
    The | moon doth shine as | bright as day;
                                                |
```

Questo è un ottimo esempio dell'utilità dei controlli di battuta. Tornando alla musica, si può vedere che il testo aggiunto non risulta ben allineato alle note. La parola *shine* dovrebbe essere cantata su due note, non una. Questo si chiama *melisma*, una singola sillaba che viene cantata per più di una nota. Ci sono molti modi per estendere una sillaba su molteplici note, e il più semplice è aggiungere una legatura di portamento che le colleghi, per i dettagli si veda Sezione 2.1.3 [Legature di valore e di portamento], pagina 23:

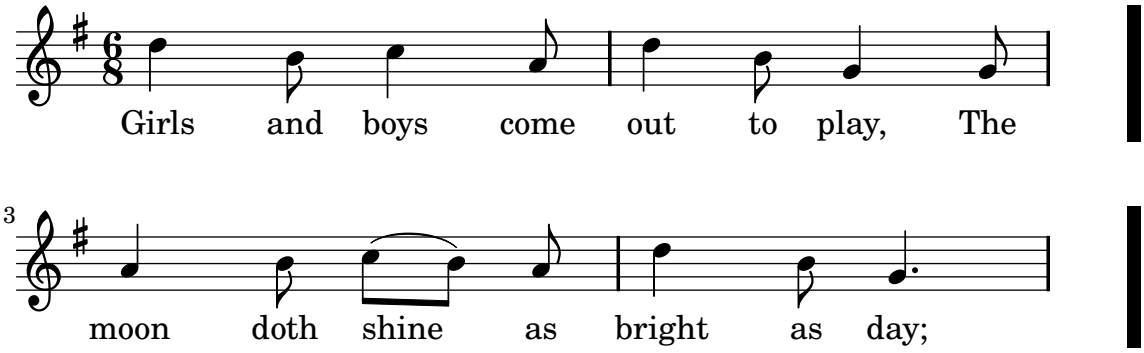
```
<<
  \relative c'' {
    \key g \major
    \time 6/8
    d4 b8 c4 a8 | d4 b8 g4 g8 |
    a4 b8 c( b) a | d4 b8 g4. |
  }
  \addlyrics {
    Girls and boys come | out to play,
    The | moon doth shine as | bright as day; |
  }
>>
```
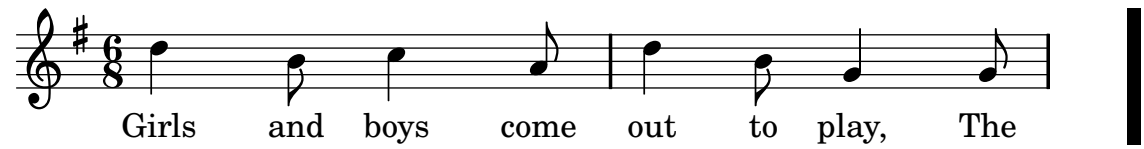




Adesso le parole sono allineate correttamente con le note, ma il raggruppamento automatico per le note che si trovano sopra *shine as* non ha un aspetto corretto. Possiamo correggerlo inserendo i comandi per il raggruppamento manuale così da scavalcare, in questo caso, il raggruppamento automatico; per i dettagli si veda Sezione 2.1.6 [Code automatiche e manuali], pagina 26.

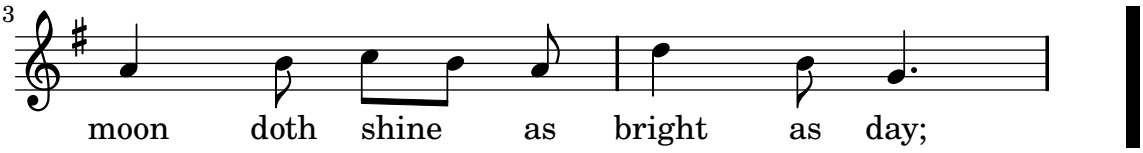
```
<<
  \relative c'' {
    \key g \major
    \time 6/8
    d4 b8 c4 a8 | d4 b8 g4 g8 |
    a4 b8 c([ b]) a | d4 b8 g4. |
  }
  \addlyrics {
    Girls and boys come | out to play,
    The | moon doth shine as | bright as day; |
  }
>>
```



Come alternativa all'uso delle legature di portamento, si possono indicare i melismi nel testo stesso, usando il trattino basso _ per ogni nota che si voglia includere nel melisma:
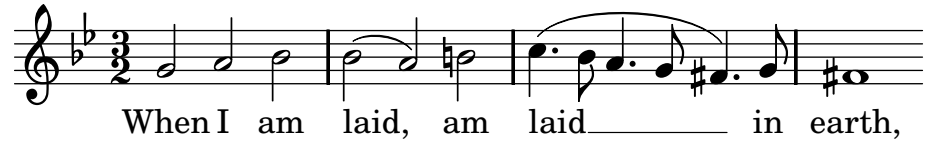
```
<<
  \relative c'' {
    \key g \major
    \time 6/8
    d4 b8 c4 a8 | d4 b8 g4 g8 |
    a4 b8 c[ b] a | d4 b8 g4. |
  }
  \addlyrics {
    Girls and boys come | out to play,
    The | moon doth shine _ as | bright as day; |
  }
>>
```
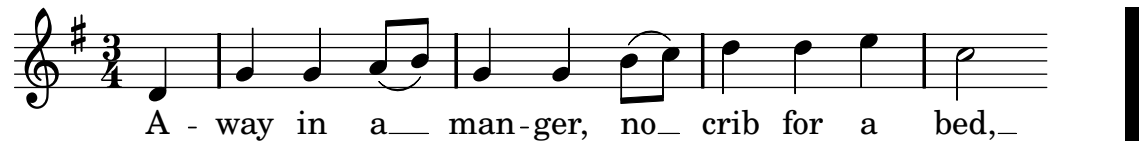




Se una sillaba si estende per molte note o per una singola nota molto lunga, solitamente viene disegnata una *linea di estensione* che va dalla sillaba e si estende per tutte le note comprese in quella sillaba. Si scrive con due trattini bassi `__`. Il seguente esempio è tratto dalle prime tre battute del *Lamento di Didone*, dal *Dido and Aeneas* di Purcell:

```
<<
  \relative c'' {
    \key g \minor
    \time 3/2
    g2 a bes | bes2( a) b2 |
    c4.( bes8 a4. g8 fis4.) g8 | fis1
  }
  \addlyrics {
    When I am | laid,
    am | laid __ in | earth,
  }
>>
```



Nessuno degli esempi visti finora ha comportato l'uso di parole aventi più di una sillaba. Tali parole vengono solitamente suddivise una sillaba per nota, con trattini posti tra le sillabe. Tali trattini vengono inseriti con due lineette, e producono un trattino centrato tra le sillabe. Ecco un esempio che, oltre a questo, mostra tutto quello che abbiamo imparato finora sull'allineamento del testo con le note.

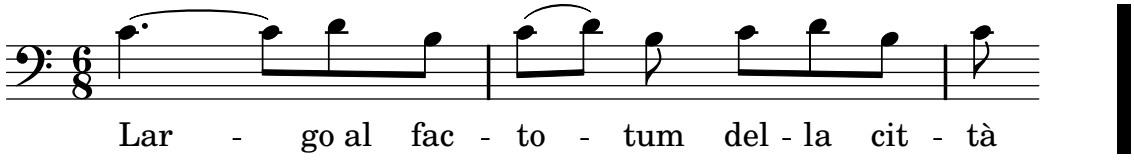
```
<<
  \relative c' {
    \key g \major
    \time 3/4
    \partial 4
    d4 | g4 g a8( b) | g4 g b8( c) |
    d4 d e | c2
  }
  \addlyrics {
    A -- | way in a __ | man -- ger,
    no __ | crib for a | bed, __
  }
>>
```



Alcuni testi, specialmente quelli in italiano, richiedono l'opposto: associare più di una sillaba a una singola nota. Questo può essere ottenuto collegando le sillabe insieme con un singolo trattino basso `_` (senza spazi), o includendole tra virgolette. Ecco un esempio tratto dal *Figaro*

di Rossini, dove *al* deve essere cantato sulla stessa nota del *go* di *Largo*, nell'aria di Figaro *Largo al factotum*:

```
<<
  \relative c' {
    \clef "bass"
    \key c \major
    \time 6/8
    c4.~ c8 d b | c8([ d]) b c d b | c8
  }
  \addlyrics {
    Lar -- go_al fac -- | to -- tum del -- la cit -- | tà
  }
>>
```
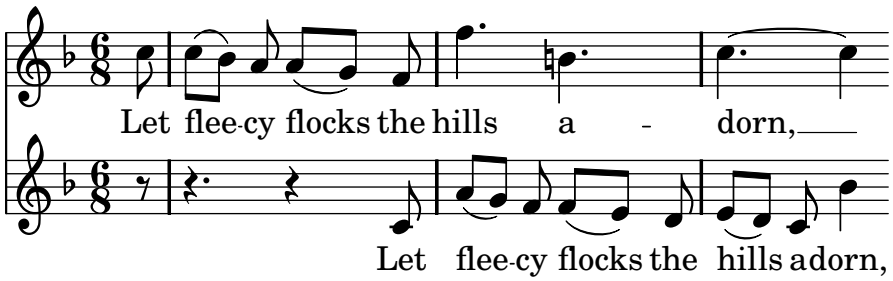


Vedi anche

Guida alla notazione: Sezione "Vocal music" in *Guida alla Notazione*.

2.3.3 Testo su più righi

Per porre lo stesso testo sotto più righi musicali, si può usare un approccio più semplice, ovvero `\addlyrics`. Ecco un esempio tratto dal *Judas Maccabæus* di Handel:

```
<<
  \relative c'' {
    \key f \major
    \time 6/8
    \partial 8
    c8 | c8([ bes]) a a([ g]) f | f'4. b, | c4.~ c4
  }
  \addlyrics {
    Let | flee -- cy flocks the | hills a -- | dorn, __
  }
  \relative c' {
    \key f \major
    \time 6/8
    \partial 8
    r8 | r4. r4 c8 | a'8([ g]) f f([ e]) d | e8([ d]) c bes'4
  }
  \addlyrics {
    Let | flee -- cy flocks the | hills a -- dorn,
  }
>>
```



Per scrivere partiture un po' più complesse di questi semplici esempi, è meglio separare la struttura dello spartito dalle note e dai testi mediante l'uso di variabili. Di queste si parla più approfonditamente in Sezione 2.4.1 [Organizzare i brani con le variabili], pagina 37.

Vedi anche

Guida alla notazione: Sezione "Vocal music" in *Guida alla Notazione*.

2.4 Ritocchi finali

Questa è la sezione finale della guida; spiega come dare gli ultimi ritocchi a semplici pezzi, e fornisce un'introduzione al resto del manuale.

2.4.1 Organizzare i brani con le variabili
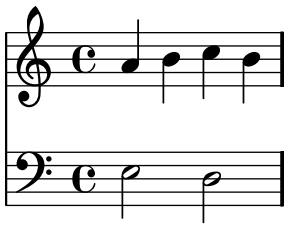
Quando tutti gli elementi discussi precedentemente vengono combinati insieme per produrre file di maggiori dimensioni, anche le espressioni musicali diventano molto più grandi. Nella musica polifonica con molti righi musicali, i file di input possono diventare molto confusi. Possiamo ridurre tale confusione attraverso l'uso delle *variabili*.

Con le variabili (conosciute anche come identificatori o macro), possiamo scomporre le espressioni musicali complesse. Una variabile viene assegnata nel seguente modo:

```
namedMusic = { ... }
```

I contenuti dell'espressione musicale `namedMusic` possono essere usati in seguito ponendo una barra inversa (backslash) di fronte al nome (`\namedMusic`, proprio come in un normale comando LilyPond).

```
violin = \new Staff {
  \relative c'' {
    a4 b c b
  }
}
cello = \new Staff {
  \relative c {
    \clef "bass"
    e2 d
  }
}
{
  <<
    \violin
    \cello
  >>
}
```
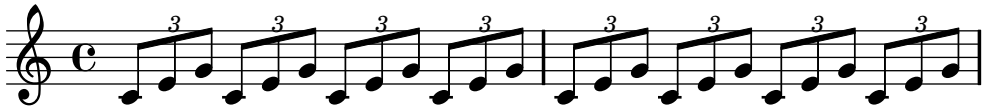


Il nome di una variabile deve contenere soltanto caratteri alfabetici, non può avere numeri, trattini bassi (underscore) o trattini di altro tipo.

Le variabili devono essere definite *prima* della principale espressione musicale, ma possono poi essere usate quante volte si vuole e ovunque, una volta definite. Possono essere usate anche all'interno della definizione di un'altra variabile successiva, dando la possibilità di accorciare l'input se una sezione della musica viene ripetuta molte volte.

```
tripletA = \times 2/3 { c,8 e g }
barA = { \tripletA \tripletA \tripletA \tripletA }

\relative c'' {
  \barA \barA
}
```



Le variabili possono essere utilizzate per molti tipi di oggetto nell'input. Ad esempio,

```
width = 4.5\cm
name = "Wendy"
aFivePaper = \paper { paperheight = 21.0 \cm }
```

A seconda dei suoi contesti, la variabile può essere usata in punti differenti. L'esempio seguente usa le variabili mostrate sopra:

```
\paper {
  \aFivePaper
  line-width = \width
}
{
  c4^\name
}
```

2.4.2 Aggiungere i titoli

Titolo, compositore, numero di opus e informazioni simili vengono inserite nel blocco `\header`. Questo si trova fuori dalla principale espressione musicale; il blocco `\header` viene solitamente posto sotto il numero di versione.

```
\version "2.14.1"
\header {
  title = "Symphony"
  composer = "Me"
  opus = "Op. 9"
}

{
  ... music ...
}
```
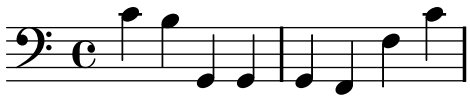
Quando il file viene elaborato, sopra la musica vengono visualizzati il titolo e il compositore. Si possono trovare maggiori informazioni sui titoli in Sezione “Creating titles” in *Guida alla Notazione*.

2.4.3 Nomi assoluti delle note

Finora abbiamo sempre usato `\relative` per definire le altezze. Questo è il modo più semplice per inserire gran parte della musica, ma esiste anche un altro modo per definire le altezze: il modo assoluto.

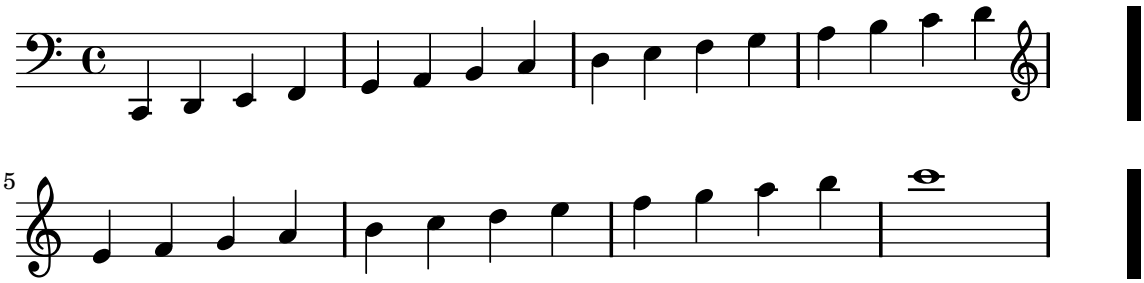
Se si omette `\relative`, LilyPond tratta tutte le altezze come valori assoluti. Un `c'` si riferirà sempre al Do centrale, un `b` si riferirà sempre alla nota che si trova un semitono sotto il Do centrale, e un `g,` indicherà sempre la nota sull’ultima riga della chiave di basso.

```
{
  \clef "bass"
  c'4 b g, g, |
  g,4 f, f c' |
}
```
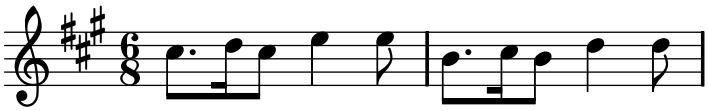


Ecco una scala con quattro ottave:

```
{
  \clef "bass"
  c,4 d, e, f, |
  g,4 a, b, c |
  d4 e f g |
  a4 b c' d' |
  \clef "treble"
  e'4 f' g' a' |
  b'4 c'' d'' e'' |
  f''4 g'' a'' b'' |
  c'''1 |
}
```
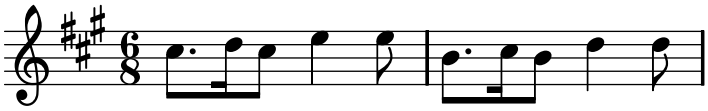


Come si può vedere, scrivere una melodia in chiave di violino richiede un ampio uso di virgolette `'`. Si consideri questo frammento tratto da Mozart:

```
{
  \key a \major
  \time 6/8
  cis''8. d''16 cis''8 e''4 e''8 |
  b'8. cis''16 b'8 d''4 d''8 |
}
```



Tutte queste virgolette rendono l'input meno leggibile e sono una fonte di errori. Usando `\relative`, l'esempio precedente è molto più semplice da leggere e scrivere:

```
\relative c'' {
  \key a \major
  \time 6/8
  cis8. d16 cis8 e4 e8
  b8. cis16 b8 d4 d8
}
```



Se si fa un errore con un segno di ottava (`'` o `,`) quando si lavora col modo `\relative`, è molto evidente – tante note saranno nell'ottava sbagliata. Quando si lavora in modo assoluto, un singolo errore non sarà tanto visibile, e non sarà così facile da trovare.

Tuttavia, il modo assoluto è utile per la musica che fa uso di ampi intervalli, e lo è ancora di più per i file LilyPond generati dal computer.

2.4.4 Dopo il tutorial

Dopo aver finito di leggere la guida, probabilmente dovresti cercare di scrivere uno o due brani. Puoi iniziare aggiungendo delle note a uno dei Appendice A [Templates], pagina 89. Se necessiti di una qualche notazione che non è stata coperta nel tutorial, dai un'occhiata alla Guida alla notazione, a cominciare da Sezione "Musical notation" in *Guida alla Notazione*. Se vuoi scrivere per un gruppo di strumenti che non è incluso nei template, dai un'occhiata a ⟨undefined⟩ [Extending the templates], pagina ⟨undefined⟩.

Una volta che hai scritto alcuni brevi pezzi, puoi proseguire la lettura del Manuale di Apprendimento (capitoli 3-5). Ovviamente non c'è niente di male nel leggerli subito! Però tieni conto che la parte restante del Manuale di Apprendimento parte dal presupposto che tu abbia già confidenza con l'input di LilyPond. Puoi saltare questi capitoli adesso, e tornare a leggerli quando ti sarai fatto un po' di esperienza.

In questo tutorial e nel resto del Manuale di Apprendimento, alla fine di ogni sezione c'è un paragrafo **Vedi anche**, che contiene riferimenti incrociati ad altre sezioni: non dovresti seguire questi riferimenti quando leggi il manuale per la prima volta; quando avrai completato la lettura di tutto il Manuale di Apprendimento, potrai rileggere alcune sezioni e seguire i riferimenti incrociati per letture di approfondimento.

Se non lo hai già fatto, *ti consigliamo* di leggere Sezione 1.4.3 [Panoramica dei manuali], pagina 20. Contiene molte informazioni su LilyPond, ed è quindi utile per i nuovi utenti, che spesso non sanno dove cercare aiuto. Se dedichi cinque minuti all'attenta lettura di quella sezione, potresti risparmiarti ore di frustrazione sprecate a guardare in posti sbagliati!

3 Concetti fondamentali

Nel Tutorial abbiamo visto come produrre della bella musica a stampa da un semplice file di testo. Questa sezione presenta i concetti e le tecniche richiesti per produrre partiture ugualmente belle, ma più complesse.

3.1 Come funzionano i file di input di LilyPond

Il formato di input di LilyPond ha una struttura piuttosto libera, che dà agli utenti esperti una grande flessibilità nell'organizzare i file come preferiscono. Ma questa flessibilità può creare confusione nei nuovi utenti. Questa sezione spiegherà in parte questa struttura, ma sorvolerà su alcuni dettagli in favore della semplicità. Per una descrizione completa del formato di input, si veda Sezione "File structure" in *Guida alla Notazione*.

3.1.1 Introduzione alla struttura di un file di LilyPond

Un esempio basilare di un file di input di LilyPond è

```
\version "2.14.1"

\header { }

\score {
  ...compound music expression...  % all the music goes here!
  \layout { }
  \midi { }
}
```

Ci sono molte varianti a questo modello di base, ma questo esempio serve da utile punto di partenza.

Finora nessuno degli esempi che abbiamo visto ha usato il comando `\score{}`. Questo si spiega col fatto che LilyPond, quando elabora un input semplice, aggiunge automaticamente gli altri comandi necessari. LilyPond tratta un input come questo:

```
\relative c'' {
  c4 a d c
}
```

come forma abbreviata per questo:

```
\book {
  \score {
    \new Staff {
      \new Voice {
        \relative c'' {
          c4 a b c
        }
      }
    }
    \layout { }
  }
}
```

In altre parole, se l'input contiene un'espressione musicale singola, LilyPond interpreterà il file come se l'espressione musicale fosse racchiusa dentro i comandi mostrati sopra.

Attenzione! Molti esempi nella documentazione di LilyPond ometteranno i comandi `\new Staff` e `\new Voice`, lasciando che questi siano creati implicitamente. Per gli esempi semplici

questo metodo funziona bene, ma per quelli più complessi, soprattutto quando vengono usati ulteriori comandi, la creazione implicita dei contesti può dare risultati inattesi, ad esempio creando dei righi non voluti. Il modo per creare i contesti esplicitamente è spiegato in ⟨undefined⟩ [Contexts and engravers], pagina ⟨undefined⟩.

> **Nota:** Quando si inseriscono più di poche linee di musica, si consiglia di creare sempre esplicitamente i righi e le voci.

Ora però torniamo al primo esempio ed esaminiamo il comando `\score`, lasciando gli altri comandi secondo l'impostazione predefinita.

Un blocco `\score` deve sempre contenere una sola espressione musicale, e questa deve trovarsi subito dopo il comando `\score`. Ricorda che un'espressione musicale può essere qualsiasi cosa, da una singola nota a una grande espressione composta come

```
{
  \new StaffGroup <<
    ...insert the whole score of a Wagner opera in here...
  >>
}
```

Tutto quanto è compreso in `{ ... }` costituisce un'unica espressione musicale.

Come abbiamo detto prima, il blocco `\score` può contenere altri elementi, come ad esempio

```
\score {
  { c'4 a b c' }
  \header { }
  \layout { }
  \midi { }
}
```

Si noti che questi tre comandi – `\header`, `\layout` e `\midi` – sono speciali: diversamente da molti altri comandi che iniziano con un backslash (\), *non* sono espressioni musicali né fanno parte di alcuna espressione musicale. Dunque, possono essere collocati dentro o fuori da un blocco `\score`. Di solito questi comandi vengono posti fuori dal blocco `\score` – ad esempio, `\header` spesso viene messo sopra il comando `\score`, come mostra l'esempio all'inizio di questa sezione.

Altri due comandi che non hai incontrato finora sono `\layout { }` e `\midi {}`. Se questi appaiono come in figura, LilyPond creerà rispettivamente un output per la stampa e uno per il MIDI. Sono descritti dettagliatamente nella Guida alla notazione, in Sezione "Score layout" in *Guida alla Notazione*, e Sezione "Creating MIDI files" in *Guida alla Notazione*.

Puoi scrivere molteplici blocchi `\score`. Ciascuno verrà trattato come una partitura separata, ma saranno tutti combinati in un unico file di output. Non è necessario il comando `\book` – ne verrà creato uno automaticamente. Tuttavia, se si desiderano file di output separati da un file '`.ly`', allora si deve usare il comando `\book` per separare le diverse sezioni: ogni blocco `\book` produrrà un file di output separato.

In breve:

Ogni blocco `\book` crea un file di output separato (ovvero, un file PDF). Se non ne hai aggiunto uno esplicitamente, LilyPond racchiude implicitamente tutto il tuo codice di input in un blocco `\book`.

Ogni blocco `\score` è un pezzo di musica separato all'interno di un blocco `\book`.

Ogni blocco `\layout` influenza il blocco `\score` o `\book` in cui compare – ovvero, un blocco `\layout` che si trova dentro un blocco `\score` riguarda solo quel blocco `\score`, mentre un blocco `\layout` che si trova fuori da un blocco `\score` (e quindi in un blocco `\book`, esplicitamente o implicitamente) riguraderà ogni `\score` in quel `\book`.

Per maggiori dettagli si veda Sezione "Multiple scores in a book" in *Guida alla Notazione*.

Un'altra grande scorciatoia è la possibilità di definire variabili, come è spiegato in Sezione 2.4.1 [Organizzare i brani con le variabili], pagina 37). Tutti i modelli usano questa forma

```
melodia = \relative c' {
  c4 a b c
}

\score {
  \melodia
}
```

Quando LilyPond esamina questo file, prende il valore di `melodia` (tutto ciò che si trova dopo il segno di uguale) e lo inserisce ovunque si trovi `\melodia`. Non c'è una regola specifica per i nomi – il nome può essere `melodia`, `globale`, `tempo`, `manodestrapiano`, o qualsiasi altro nome. Ricordati che puoi usare quasi ogni nome che vuoi, purché esso contenga solo caratteri alfabetici e sia diverso dai nomi dei comandi di LilyPond. Le esatte limitazioni relative ai nomi delle variabili sono spiegate dettagliatamente in Sezione "File structure" in *Guida alla Notazione*.

Vedi anche

Per una definizione completa del formato di input, si veda Sezione "File structure" in *Guida alla Notazione*.

3.1.2 La partitura è una (singola) espressione musicale composta

Abbiamo visto l'organizzazione generale dei file di input di LilyPond nella sezione precedente, Sezione 3.1.1 [Introduzione alla struttura di un file di LilyPond], pagina 41. Ma sembra che abbiamo saltato la parte più importante: cosa dobbiamo scrivere dopo `\score`?

In realtà non l'abbiamo affatto dimenticato. Il grande mistero è, semplicemente, che *non c'è* alcun mistero. La seguente frase spiega tutto:

Un blocco `\score` *deve iniziare con un'espressione musicale composta.*

Per capire cosa si intende per espressione musicale e per espressione musicale composta, potrebbe esserti utile ripassare il tutorial, Sezione 2.2.1 [Espressioni musicali], pagina 27. In quella sezione, abbiamo visto come costruire grandi espressioni musicali a partire da piccoli brani – abbiamo iniziato con le note, poi gli accordi, etc. Adesso inzieremo da una grande espressione musicale e proseguiremo poi a spiegarne i dettagli. Per semplicità, nel nostro esempio useremo soltanto un canto e un pianoforte. Per questa formazione non abbiamo bisogno di `StaffGroup`, che non fa altro che raggruppare un insieme di righi con una parentesi graffa a sinistra, ma abbiamo comunque bisogno dei righi per il canto e per il pianoforte.

```
\score {
  { % questa parentesi inizia la principale espressione musicale composta
    \new StaffGroup <<
      ...inserisci qui la partitura completa di un'opera di Wagner...
    >>
  } % questa parentesi chiude la principale espressione musicale composta
  \layout { }
}
```

Un'intera opera di Wagner sarebbe probabilmente il doppio di questo manuale, quindi aggiungeremo solo un pianoforte e un cantante. Per questo ensemble non abbiamo bisongo di `StaffGroup`, che non fa altro che raggruppare un certo numero di righi con una parentesi a sinistra, dunque lo toglieremo. Tuttavia *abbiamo* bisogno di un pianoforte e di una voce.

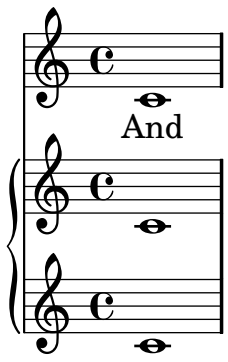
```
\score {
  <<
```

```
    \new Staff = "canto" <<
    >>
    \new PianoStaff = "pianoforte" <<
    >>
  >>
  \layout { }
}
```

In questo esempio abbiamo dato dei nomi ai righi – “canto” e “pianoforte”. Non è necessario in questo caso, ma è una buona abitudine da coltivare, perché ti permette di vedere a colpo d’occhio a cosa serve ciascun rigo.

Ricorda che si usano `<< ... >>` invece di `{ ... }` per indicare la musica simultanea. In questo modo la parte vocale e quella di pianoforte appaiono una sopra l’altra nello spartito. Il costrutto `<< ... >>` non sarebbe necessario per il rigo del cantante nell’esempio precedente se contenesse soltanto un’espressione musicale sequenziale, ma `<< ... >>` al posto delle parentesi è necessario se la musica sul rigo deve contenere due o più espressioni simultanee, ad esempio due voci simultanee, o una voce con del testo. Vogliamo avere una voce con del testo, dunque ci servono le parentesi ad angolo. Aggiungeremo la musica vera e propria in seguito; per adesso mettiamo soltanto delle semplici note e del testo. Se hai dimenticato come aggiungere del testo, potresti voler ripassare `\addlyrics` in Sezione 2.3.1 [Impostare canzoni semplici], pagina 32.

```
\score {
  <<
    \new Staff = "canto" <<
      \new Voice = "vocal" { c'1 }
      \addlyrics { And }
    >>
    \new PianoStaff = "piano" <<
      \new Staff = "upper" { c'1 }
      \new Staff = "lower" { c'1 }
    >>
  >>
  \layout { }
}
```



Ora abbiamo molti più dettagli. Abbiamo il rigo del cantante: esso contiene una `Voice` o voce (in LilyPond, questo termine si riferisce a un insieme di note, non necessariamente alle note della voce – ad esempio, un violino di solito costituisce una voce) e del testo. Abbiamo anche il rigo del pianoforte, che a sua volta comprende un rigo superiore (per la mano destra) e uno inferiore (per la mano sinistra), sebbene a quest’ultimo debba ancora essere assegnata una chiave di basso.

A questo punto possiamo iniziare ad inserire le note. All’interno delle parentesi graffe vicine a `\new Voice = "vocal"`, possiamo iniziare a scrivere
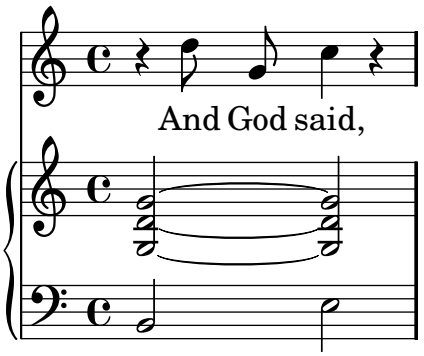
```
\relative c'' {
  r4 d8\noBeam g, c4 r
}
```

Ma se facessimo così, la sezione `\score` diventerebbe molto lunga, e sarebbe più difficile comprendere quel che accade. Usiamo quindi le variabili piuttosto. Queste sono state introdotte alla fine della sezione precedente, ricordi? Per far sì che i contenuti della variabile `text` siano interpretati come testo, li facciamo precedere da `\lyricmode`. Come in `\addlyrics`, questo comando trasforma la modalità di input in modalità testo. Senza di esso, LilyPond cercherebbe di interpretare i contenuti come se fossero note, e questo produrrebbe degli errori. (Sono disponibili molte altre modalità di input, si veda Sezione "Input modes" in *Guida alla Notazione*.)

Dunque se aggiungiamo un po' di note e una chiave di basso per la mano sinistra, otteniamo un brano musicale vero e proprio:

```
melody = \relative c'' { r4 d8\noBeam g, c4 r }
text   = \lyricmode { And God said, }
upper  = \relative c'' { <g d g,>2~ <g d g,> }
lower  = \relative c { b2 e }

\score {
  <<
    \new Staff = "canto" <<
      \new Voice = "vocal" { \melody }
      \addlyrics { \text }
    >>
    \new PianoStaff = "piano" <<
      \new Staff = "upper" { \upper }
      \new Staff = "lower" {
        \clef "bass"
        \lower
      }
    >>
  >>
  \layout { }
}
```



Quando scrivi (o leggi) una sezione `\score`, prenditela comoda e stai attento. Comincia dal livello più esterno, poi lavora su ogni livello più piccolo. È anche molto utile essere rigorosi nell'indentare l'input – ovvero fare attenzione che ogni elemento di uno stesso livello presente nell'editor di testo si trovi nella stessa posizione orizzontale.
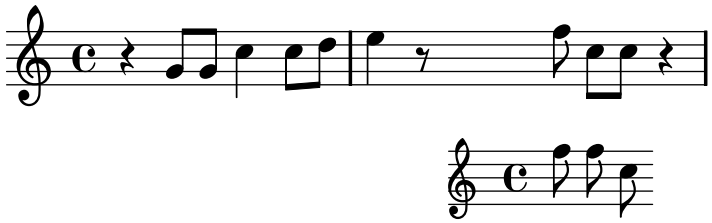
Vedi anche

Guida alla notazione: Sezione "Structure of a score" in *Guida alla Notazione*.

3.1.3 Annidare le espressioni musicali

Non è obbligatorio dichiarare tutti i righi fin dall'inizio; possono essere invece introdotti temporaneamente in ogni momento. Questo è utile in particolare per creare le sezioni ossia – si veda Sezione "ossia" in *Glossario Musicale*. Ecco un semplice esempio che mostra come inserire un nuovo rigo temporaneamente, per la durata di tre note:
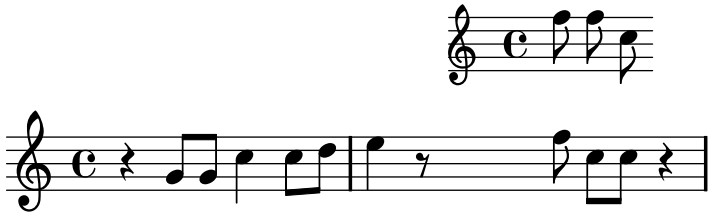
```
\new Staff {
  \relative g' {
    r4 g8 g c4 c8 d |
    e4 r8
    <<
      { f8 c c }
      \new Staff {
        f8 f c
      }
    >>
    r4 |
  }
}
```



Si noti che la dimensione della chiave è la stessa di una chiave che segue un cambio di chiave –ovvero leggermente più piccola della chiave all'inizio del rigo. Questo è utile per le chiavi che devono essere posizionate a metà di un rigo.

La sezione ossia può anche essere posta sopra il rigo nel seguente modo:

```
\new Staff = "main" {
  \relative g' {
    r4 g8 g c4 c8 d |
    e4 r8
    <<
      { f8 c c }
      \new Staff \with {
        alignAboveContext = #"main"
      } { f8 f c }
    >>
    r4 |
  }
}
```



Questo esempio usa `\with`, che verrà spiegato in modo più completo in seguito. È un modo per cambiare il comportamento predefinito di un singolo rigo. In questo caso, indica che il nuovo rigo deve essere posizionato sopra il rigo chiamato "main" invece che nella posizione predefinita, che è in basso.

Vedi anche

Gli ossia vengono spesso scritti senza armatura di chiave e senza tempo, e solitamente hanno un font più piccolo. Richiedono ulteriori comandi che non sono stati ancora presentati. Si veda ⟨undefined⟩ [Size of objects], pagina ⟨undefined⟩, e Sezione "Ossia staves" in *Guida alla Notazione*.

3.1.4 Sul non annidamento di parentesi e legature di valore

Abbiamo già incontrato vari tipi di parentesi e di costrutti che fanno uso di parentesi mentre scrivevamo il file di input di LilyPond. Ognuna obbedisce a diverse regole, e questo può generare confusione all'inizio. Rivediamo prima i diversi tipi di parentesi e di costrutti in parentesi.
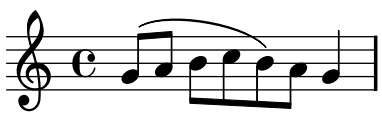
Tipo di parentesi	**Funzione**
`{ .. }`	Racchiude un frammento di musica sequenziale
`< .. >`	Racchiude le note di un accordo
`<< .. >>`	Racchiude le espressioni musicali simultanee
`( .. )`	Contrassegna l'inizio e la fine di una legatura di portamento
`\( .. \)`	Contrassegna l'inizio e la fine di una legatura di frase
`[ .. ]`	Contrassegna l'inizio e la fine di una cediglia impostata manualmente

A questi dovremmo aggiungere altri costrutti che generano linee tra e lungo le note: legature di valore (contrassegnate dal tilde, `~`), i gruppi irregolari scritti in questa forma `\times x/y {..}`, e gli abbellimenti, indicati con `\grace{..}`.

Fuori da LilyPond, l'uso convenzionale delle parentesi esige che i diversi tipi siano annidati adeguatamente, in questo modo, `<< [ { ( .. ) } ] >>`, dove le parentesi che chiudono si trovano esattamente nell'ordine opposto alle parentesi che aprono. Questo **è** un requisito per i tre tipi di parentesi descritti dal termine 'Racchiude' nella tabella precedente – devono annidarsi correttamente. Tuttavia, i restanti costrutti in parentesi, descritti dal termine 'Contrassegna' nella tabella, così come le legature e i gruppi irregolari, **non** devono annidarsi correttamente con alcuna delle parentesi o dei costrutti in parentesi. Infatti queste non sono parentesi nel senso che racchiudono qualcosa – sono semplicemente dei contrassegni che indicano dove qualcosa inizia e finisce.

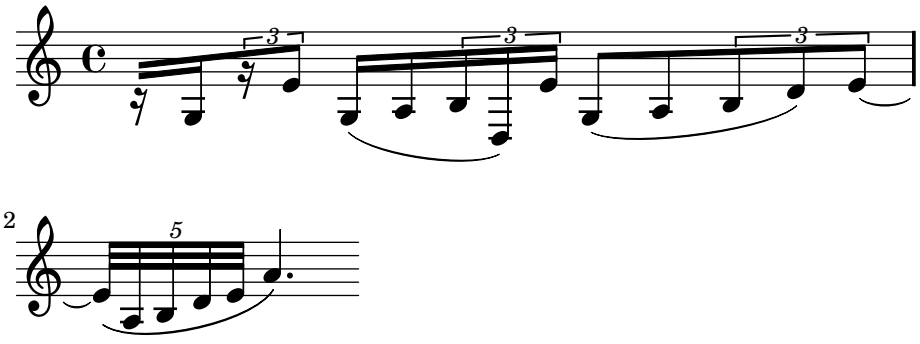
Quindi, ad esempio, una legatura di frase può iniziare prima di una cediglia inserita manualmente e finire prima della fine della cediglia – non molto musicale, forse, ma possibile:

```
{ g8\( a b[ c b\) a] g4 }
```



In generale, tipi diversi di parentesi, costrutti in parentesi e segni che riguardano gruppi irregolari, legature e abbellimenti possono essere combinati liberamente. L'esempio seguente mostra una cediglia che si estende su un gruppo irregolare (linea 1), una legatura di portamento che si estende su una terzina (linea 2), una cediglia e una legatura di portamento che si estendono su una terzina, una legatura di valore che attraversa due gruppi irregolari, e una legatura di frase che si estende fuori da un gruppo irregolare (linee 3 e 4).

```
{
  r16[ g \times 2/3 { r16 e'8] }
  g16( a \times 2/3 { b16 d) e' }
  g8[( a \times 2/3 { b8 d') e'~] } |
  \times 4/5 { e'32\( a b d' e' } a'4.\)
}
```



3.2 Le voci contengono la musica

Un cantante ha bisogno della voce per cantare, e lo stesso vale per LilyPond. La musica vera e propria per tutti gli strumenti di una partitura è contenuta nelle Voci – il più importante concetto di LilyPond.
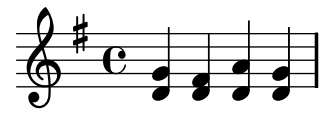
3.2.1 Sento le Voci

I livelli più profondi, più interni e più importanti di uno spartito di LilyPond sono chiamati 'Voice contexts' («Contesti della voce») o semplicemente 'Voices' («Voci»). In altri programmi di notazione le voci sono chiamate talvolta 'layers' («livelli»).

Il livello o contesto della voce è l'unico che può contenere la musica. Se un contesto della voce non è dichiarato esplicitamente, ne viene creato uno automaticamente, come abbiamo visto all'inizio di questo capitolo. Alcuni strumenti, come ad esempio un oboe, possono produrre una sola nota per volta. La musica scritta per tali strumenti è monofonica e necessita di una sola voce. Invece gli strumenti che possono produrre più di una nota contemporaneamente, come ad esempio il pianoforte, richiederanno spesso voci multiple per codificare le diverse note e ritmi simultanei che sono capaci di riprodurre.

Ovviamente, una singola voce può contenere molte note in un accordo, dunque quando l'uso delle voci multiple è davvero necessario? Si osservi questo esempio di quattro accordi:

```
\key g \major
<d g>4 <d fis> <d a'> <d g>
```
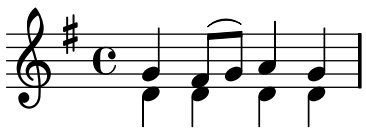


Questa musica può essere espressa usando soltanto i simboli dell'accordo, ovvero le parentesi angolari, `< ... >`, e una singola voce è sufficiente. Ma cosa accadrebbe se il Fa# fosse in realtà una nota di un ottavo seguita da un Sol di un ottavo, una nota di passaggio che porta al La? In questo caso abbiamo due note che iniziano nello stesso momento ma hanno durate diverse: il Re da un quarto e il Fa# da un ottavo. Come si possono scrivere queste note? Non possono essere scritte come un accordo perché tutte le note di un accordo devono avere la stessa durata. E non possono nemmeno essere scritte come due note in sequenza perché devono iniziare in contemporanea. Si tratta quindi di un caso in cui sono necessarie due voci.

Vediamo come ottenerle nella sintassi di input di LilyPond.

Il modo più semplice per inserire frammenti che utilizzino più di una voce su un rigo è scrivere ogni voce come una sequenza (con `{...}`), e poi combinarle in simultanea tramite le doppie parentesi angolari, `<<...>>`. Per collocarli in voci distinte, i frammenti devono essere separati da un doppio backslash, `\\`. Senza di esso, le note sarebbero inserite in un'unica voce, e questo normalmente causerebbe degli errori. Questa tecnica è particolarmente adatta ai brani che sono in gran parte monofonici ma con brevi e occasionali sezioni polifoniche.

Ecco come suddividere gli accordi precedenti in due voci e aggiungere sia la nota di passaggio che la legatura di portamento:
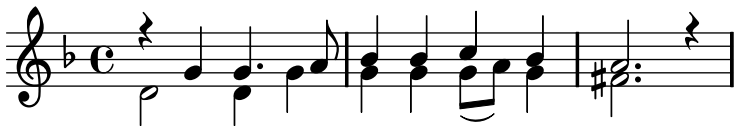
```
\key g \major
%    Voice "1"                Voice "2"
<< { g4 fis8( g) a4 g } \\ { d4 d d d }  >>
```



Si noti come i gambi della seconda voce adesso siano rivolti in basso.
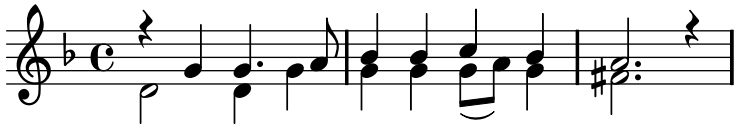
Ecco un altro semplice esempio:

```
\key d \minor
%    Voice "1"             Voice "2"
<< { r4 g g4. a8 }    \\ { d,2 d4 g }       >> |
<< { bes4 bes c bes } \\ { g4 g g8( a) g4 } >> |
<< { a2. r4 }         \\ { fis2. s4 }       >> |
```



Non è necessario usare un costrutto con `<< \\ >>` in ogni battuta. Per musiche che hanno poche note in ogni battuta questo layout può aiutare la leggibilità del codice, ma se ci sono molte note in ogni battuta è preferibile dividere ogni voce, così:

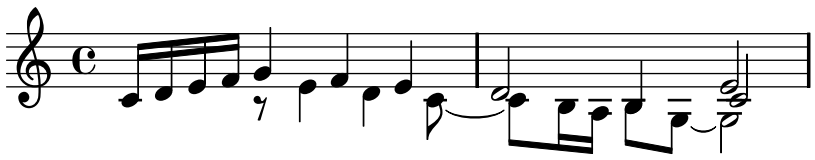
```
\key d \minor
<< {
  % Voice "1"
  r4 g g4. a8 |
  bes4 bes c bes |
  a2. r4 |
} \\ {
  % Voice "2"
  d,2 d4 g |
  g4 g g8( a) g4 |
  fis2. s4 |
} >>
```



Questo esempio ha solo due voci, ma si potrebbe usare lo stesso costrutto per scrivere tre o più voci aggiungendo più backslash.

I contesti della voce hanno i nomi "1", "2", etc. In ogni contesto, la direzione verticale di legature di portamento, gambi, legature di valore, dinamica, etc., è impostata correttamente.

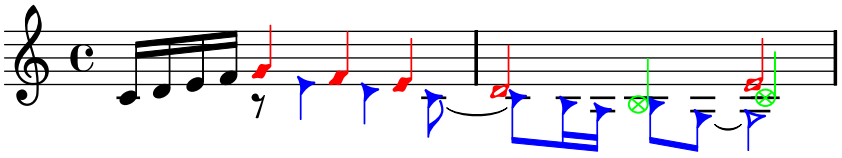
```
\new Staff \relative c' {
  % Main voice
  c16 d e f
  %    Voice "1"     Voice "2"                Voice "3"
  << { g4 f e } \\ { r8 e4 d c8~ } >> |
  << { d2 e }   \\ { c8 b16 a b8 g~ g2 } \\ { s4 b c2 } >> |
}
```



Tutte queste voci sono separate dalla voce principale che contiene le note e che si trova fuori dal costrutto `<< .. >>`, che chiameremo il *costrutto simultaneo*. Legature di portamento e di valore possono connettere solo note che fanno parte della stessa voce, quindi le legature non possono entrare in un costrutto simultaneo o uscirne. Viceversa, voci parallele appartenenti a costrutti simultanei distinti sullo stesso rigo sono la stessa voce. Anche altre caratteristiche della voce di riferimento sono trasferite ai costrutti simultanei. Ecco lo stesso esempio, ma con colori e teste delle note diversi per ogni voce. Si noti che i cambiamenti in una voce non interessano le altre voci, ma persistono sulla stessa voce in seguito. Si noti anche che le note legate possono essere divise sulle stesse voci in due costrutti, come mostra qui la voce con i triangoli blu.

```
\new Staff \relative c' {
  % Main voice
  c16 d e f
  <<  % Bar 1
    {
      \voiceOneStyle
      g4 f e
    }
  \\
    {
      \voiceTwoStyle
      r8 e4 d c8~
    }
  >> |
  <<  % Bar 2
     % Voice 1 continues
    { d2 e }
  \\
     % Voice 2 continues
    { c8 b16 a b8 g~ g2 }
  \\
    {
      \voiceThreeStyle
      s4 b c2
    }
```

```
  >> |
}
```



I comandi `\voiceXXXStyle` sono pensati soprattutto per documenti didattici come questo. Modificano il colore della testa, del gambo e delle cediglie, e lo stile della testa, così da rendere le voci facilmente distinguibili. La voce uno è impostata su rombi rossi, la voce due su triangoli blu, la voce tre su cerchi barrati verdi, e la voce quattro (non impiegata nell'esempio) su croci magenta; `\voiceNeutralStyle` (anch'esso non usato qui) riporta lo stile all'impostazione predefinita. Vedremo in seguito come l'utente possa creare comandi simili. Si veda ⟨undefined⟩ [Visibility and color of objects], pagina ⟨undefined⟩ e ⟨undefined⟩ [Using variables for tweaks], pagina ⟨undefined⟩.

La polifonia non cambia le relazioni tra le note all'interno di un blocco `\relative`. L'altezza di ogni nota continua a essere calcolata in rapporto a quella della nota che la precede, o della prima nota del precedente accordo. Dunque, in
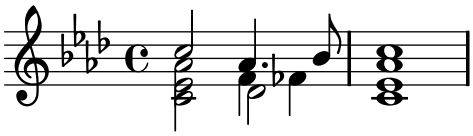
```
\relative c' { notaA << < notaB notaC > \\ notaD >> notaE }
```

`notaB` è relativa a `notaA`
`notaC` è relativa a `notaB`, non a `notaA`;
`notaD` è relativa a `notaB`, non a `notaA` o a `notaC`;
`notaE` è relativa a `notaD`, non a `notaA`.

Un metodo alternativo, che potrebbe essere più chiaro se le note nelle voci sono ampiamente separate, consiste nel porre un comando `\relative` all'inizio di ogni voce:

```
\relative c' { noteA ... }
<<
  \relative c'' { < noteB noteC > ... }
\\
  \relative g' { noteD ... }
>>
\relative c' { noteE ... }
```

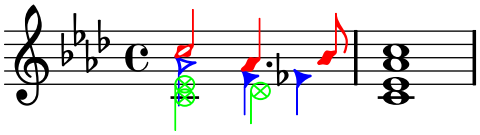
Analizziamo infine le voci in un brano più complesso. Queste note sono tratte dalle prime due battute del secondo dei due Notturni op. 32 di Chopin. Questo esempio verrà usato successivamente, in questo e nel prossimo capitolo, per illustrare varie tecniche di notazione, quindi per il momento ignora qualsiasi cosa del codice sottostante che ti sembra incomprensibile e concentrati solo sulla musica e sulle voci – le parti più complesse saranno spiegate tutte in sezioni successive.



La direzione dei gambi viene spesso usata per indicare la continuità di due linee melodiche simultanee. In questo esempio i gambi delle note più acute sono tutti rivolti in su mentre i gambi delle note più gravi sono tutti rivolti in giù. Questo è il primo indizio del fatto che è coinvolta più di una voce.
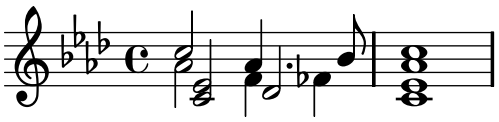
Ma è quando note che iniziano nello stesso momento hanno durate diverse che il ricorso a voci multiple diventa realmente indispensabile. Osserva le note che iniziano alla terza pulsazione

della prima battuta. Il La bemolle è una nota di tre ottavi, il Fa è una semiminima e il Re bemolle è una minima. Non possono essere scritte come un accordo perché tutte le note di un accordo devono avere la stessa durata. Né possono essere scritte come note in sequenza, dato che devono iniziare contemporaneamente. Questa sezione della battuta necessita di tre voci, e la normale pratica consiste nello scrivere l'intera battuta su tre voci, come mostrato sotto, dove abbiamo usato diverse teste e colori per le tre voci. Ancora una volta, il codice che sta dietro questo esempio verrà spiegato dopo, quindi ignora quel che non capisci.
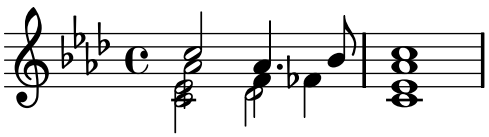


Proviamo a scrivere il codice di questa musica da zero. Come vedremo, questo pone alcune difficoltà. Come abbiamo imparato, iniziamo usando il costrutto `<< \\ >>` per inserire la musica della prima battuta in tre voci:

```
\new Staff \relative c'' {
  \key aes \major
  <<
    { c2 aes4. bes8 } \\ { aes2 f4 fes } \\ { <ees c>2 des2 }
  >>
  <c ees aes c>1
}
```



La direzione dei gambi è assegnata automaticamente: le voci dispari avranno i gambi in su e le voci pari i gambi in giù. I gambi per le voci 1 e 2 sono giusti, ma in questo brano i gambi della voce 3 dovrebbero essere in giù. Possiamo correggere semplicemente omettendo la voce tre e ponendo la musica nella voce quattro. Si può fare aggiungendo semplicemente un altro paio di `\\`:

```
\new Staff \relative c'' {
  \key aes \major
  << % Voice one
    { c2 aes4. bes8 }
  \\ % Voice two
    { aes2 f4 fes   }
  \\ % Omit Voice three
  \\ % Voice four
    { <ees c>2 des2 }
  >> |
  <c ees aes c>1 |
}
```



Possiamo vedere che questo corregge la direzione del gambo, ma espone a un problema talvolta presente con le voci multiple: i gambi delle note in una voce possono collidere con le teste delle

note di un'altra voce. Nel disporre le note, LilyPond permette alle note o agli accordi di due diverse voci di occupare la stessa colonna verticale della nota purché i gambi siano in direzioni opposte, ma le note della terza e quarta voce vengono spostate, se necessario, per evitare la collisione tra le teste. Questo di solito funziona bene, ma in questo esempio le note della voce più bassa, con le impostazioni predefinite, appaiono chiaramente disposte in una posizione non ottimale. LilyPond fornisce molti modi per aggiustare la collocazione orizzontale delle note. Per ora, non siamo ancora pronti a cercare di correggere questo problema, dunque dovremo aspettare fino a una prossima sezione – si veda la proprietà `force-hshift` in ⟨undefined⟩ [Fixing overlapping notation], pagina ⟨undefined⟩.

Vedi anche

Guida alla notazione: Sezione "Multiple voices" in *Guida alla Notazione*.

3.2.2 Definire esplicitamente le voci

I contesti della voce possono anche essere creati manualmente, all'interno di un blocco `<< >>` che crea musica polifonica, usando `\voiceOne` ... `\voiceFour` per indicare le direzioni desiderate per gambi, legature, etc. Nelle partiture più grandi questo metodo è più chiaro, perché fa sì che le voci possano essere separate e nominate in modo più descrittivo.
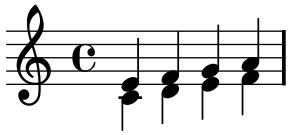
Nello specifico, il costrutto `<< \\ >>` usato nella sezione precedente:

```
\new Staff {
  \relative c' {
    << { e4 f g a } \\ { c,4 d e f } >>
  }
}
```

è equivalente a

```
\new Staff <<
  \new Voice = "1" { \voiceOne \relative c' { e4 f g a } }
  \new Voice = "2" { \voiceTwo \relative c' { c4 d e f } }
>>
```
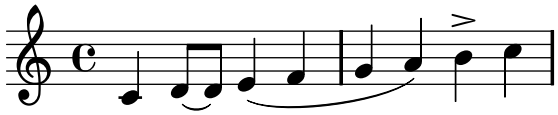
Entrambi hanno come risultato
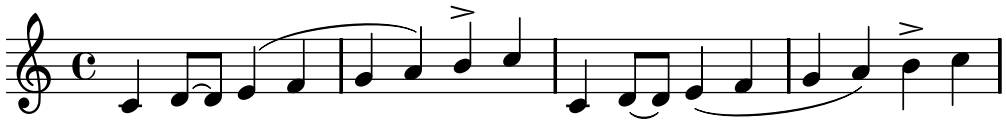


I comandi `\voiceXXX` impostano le direzioni di gambi, legature di portamento, legature di valore, articolazioni, annotazioni, punti di aumentazione di note puntate e diteggiature. `\voiceOne` e `\voiceThree` fanno sì che questi oggetti siano rivolti verso l'alto, mentre `\voiceTwo` e `\voiceFour` fanno sì che puntino verso il basso. Questi comandi producono anche uno spostamento orizzontale per ogni voce quando si crei la necessità di evitare collisioni tra le teste. Il comando `\oneVoice` riporta i valori alle normali impostazioni di una singola voce.

Vediamo tramite alcuni semplici esempi quali effetti esattamente `\oneVoice`, `\voiceOne` e `voiceTwo` hanno su markup, legature di valore, legature di portamento, e dinamica:
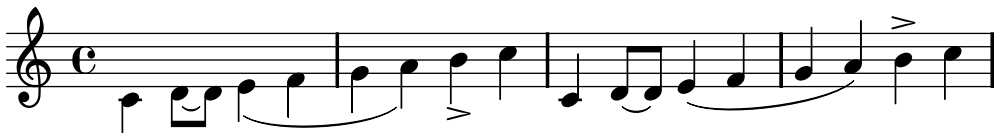
```
\relative c'{
  % Default behavior or behavior after \oneVoice
  c4 d8~ d e4( f | g4 a) b-> c |
}
```



```
\relative c' {
  \voiceOne
  c4 d8~ d e4( f | g4 a) b-> c |
  \oneVoice
  c,4 d8~ d e4( f | g4 a) b-> c |
}
```



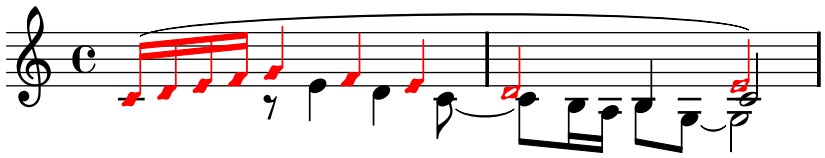
```
\relative c' {
  \voiceTwo
  c4 d8~ d e4( f | g4 a) b-> c |
  \oneVoice
  c,4 d8~ d e4( f | g4 a) b-> c |
}
```



Vediamo adesso, usando l'esempio della sezione precedente, tre modi diversi di scrivere uno stesso passo di musica polifonica, e i rispettivi vantaggi, a seconda delle circostanze.

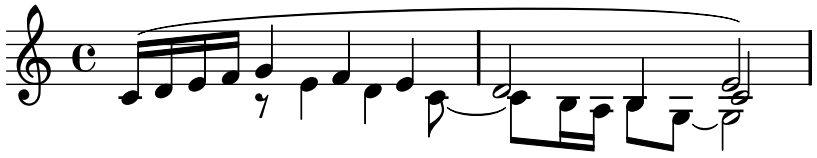
Un'espressione che appare direttamente in un << >> appartiene alla voce principale (ma, attenzione: **non** in un costrutto << \\ >>). Questo metodo è utile quando le altre voci entrano mentre la voce principale sta già suonando. Ecco una versione più corretta del nostro esempio. Le note a rombi rossi mostrano che la melodia principale si trova ora nel contesto di una voce singola, e questo fa sì che sia possibile disegnare una legatura di frase sopra di esse.

```
\new Staff \relative c' {
  \voiceOneStyle
  % The following notes are monophonic
  c16^( d e f
  % Start simultaneous section of three voices
  <<
    % Continue the main voice in parallel
    { g4 f e | d2 e) | }
    % Initiate second voice
    \new Voice {
      % Set stems, etc., down
      \voiceTwo
      r8 e4 d c8~ | c8 b16 a b8 g~ g2 |
    }
    % Initiate third voice
    \new Voice {
      % Set stems, etc, up
      \voiceThree
      s2. | s4 b c2 |
    }
  >>
}
```



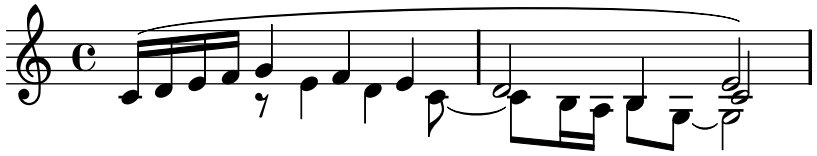
Sono possibili costrutti polifonici annidati più fittamente, e se una voce appare solo brevemente questo potrebbe essere un modo più semplice di scrivere lo spartito:

```
\new Staff \relative c' {
  c16^( d e f
  <<
    { g4 f e | d2 e) | }
    \new Voice {
      \voiceTwo
      r8 e4 d c8~ |
      <<
        { c8 b16 a b8 g~ g2 | }
        \new Voice {
          \voiceThree
          s4 b c2 |
        }
      >>
    }
  >>
}
```



Questo metodo di annidare nuove voci in breve è utile quando solo piccole parti della musica sono polifoniche, ma quando la polifonia è impiegata largamente in tutta la parte può essere più chiaro ricorrere a voci multiple, usando le note spaziatrici per saltare le parti in cui una delle voci è muta, come nel seguente esempio:

```
\new Staff \relative c' <<
  % Initiate first voice
  \new Voice {
    \voiceOne
    c16^( d e f g4 f e | d2 e) |
  }
  % Initiate second voice
  \new Voice {
    % Set stems, etc, down
    \voiceTwo
    s4 r8 e4 d c8~ | c8 b16 a b8 g~ g2 |
  }
  % Initiate third voice
  \new Voice {
    % Set stems, etc, up
    \voiceThree
    s1 | s4 b c2 |
  }
>>
```



Note columns

Le note maggiormente ravvicinate all'interno di un accordo o quelle che compaiono nello stesso momento in voci diverse sono disposte in due, e talvolta più, colonne, per impedire la sovrapposizione delle teste. Esse vengono chiamate colonne delle note. Le singole colonne di ognuna delle voci sono indipendenti, e lo scarto adottato nella voce in uso è determinato dal distanziamento delle colonne nei casi in cui altrimenti si determinerebbe una collisione. Si può vedere nell'esempio in basso. Nella seconda battuta il Do della seconda voce è spostato a destra del Re nella prima voce, e nell'ultimo accordo il Do nella terza voce è spostato anch'esso a destra delle altre note.

I comandi `\shiftOn`, `\shiftOnn`, `\shiftOnnn`, e `\shiftOff` specificano il grado con cui le note e gli accordi della voce debbano essere spostati in caso di collisione. L'impostazione predefinita prevede che le voci esterne (di norma le voci uno e due) abbiano `\shiftOff` attivato, mentre le voci interne (terza e quarta) abbiano `\shiftOn` attivato. Quando uno spostamento viene applicato, le voci una e tre vengono spostate a destra e le voci due e quattro a sinistra.

`\shiftOnn` e `\shiftOnnn` definiscono ulteriori livelli di scarto che possono essere temporaneamente specificati per risolvere le collisioni in situazioni complesse – si veda ⟨undefined⟩ [Real music example], pagina ⟨undefined⟩.

Una colonna di note può contenere soltanto una nota (o accordo) di una voce con gambi in su e una nota (o accordo) di una voce con gambi in giù. Se note di due voci che hanno i gambi nella stessa direzione sono poste nella stessa posizione ed entrambe le voci non hanno uno spostamento specificato oppure ne hanno uno dello stesso tipo, si produrrà il messaggio di errore "Too many clashing note columns".

Vedi anche
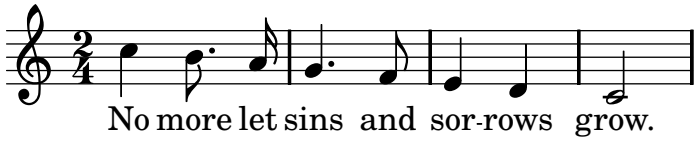
Guida alla notazione: Sezione "Multiple voices" in *Guida alla Notazione*.

3.2.3 Voci e musica vocale

La musica vocale presenta una difficoltà in più: occorre combinare due espressioni – note e testo.

Abbiamo già visto il comando `\addlyrics{}`, che ben si comporta con le partiture semplici. Tuttavia, questa tecnica è piuttosto limitata. Per musica più complessa, occorre introdurre il testo in un contesto `Lyrics` usando `\new Lyrics` e collegando esplicitamente il testo alle note con `\lyricsto{}`, tramite il nome assegnato alla voce.

```
<<
  \new Voice = "one" {
    \relative c'' {
      \autoBeamOff
      \time 2/4
      c4 b8. a16 | g4. f8 | e4 d | c2 |
    }
  }
  \new Lyrics \lyricsto "one" {
    No more let | sins and | sor -- rows | grow. |
  }
>>
```



Si noti che il testo deve essere collegato a un contesto `Voice`, *non* a un contesto `Staff`. Questo è un caso in cui è necessario creare esplicitamente contesti `Staff` e `Voice`.
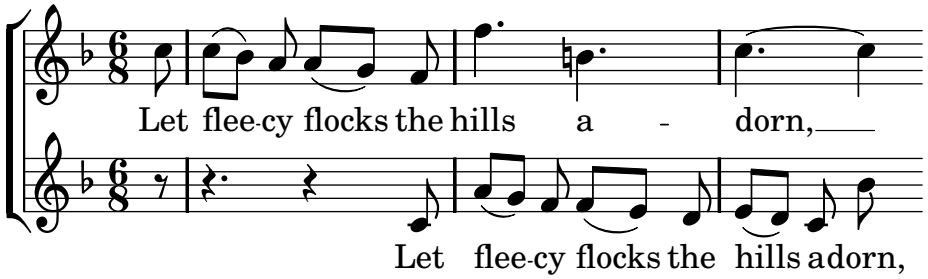
La disposizione automatica delle travature predefinita di LilyPond funziona bene per la musica strumentale, ma non altrettanto per la musica con testi, dove le travature o non sono usate affatto o servono a indicare la presenza di melismi nel testo. Nell'esempio precedente usiamo il comando `\autoBeamOff` per disattivare la travatura automatica.

Riprendiamo l'esempio precedente di Judas Maccabæus per presentare questa tecnica più flessibile. Innanzitutto lo rimaneggiamo per usare delle variabili per mezzo delle quali la musica e il testo possano essere separate dalla struttura del rigo. Inseriamo anche una parentesi ChoirStaff. Il testo deve essere introdotto da `\lyricmode` per assicurare che siano interpretati come testo invece che come musica.

```
global = { \key f \major \time 6/8 \partial 8 }

SopOneMusic = \relative c'' {
  c8 | c8([ bes)] a a([ g)] f | f'4. b, | c4.~ c4
}
SopOneLyrics = \lyricmode {
  Let | flee -- cy flocks the | hills a -- dorn, __
}
SopTwoMusic = \relative c' {
  r8 | r4. r4 c8 | a'8([ g)] f f([ e)] d | e8([ d)] c bes'
}
SopTwoLyrics = \lyricmode {
  Let | flee -- cy flocks the | hills a -- dorn,
}

\score {
  \new ChoirStaff <<
    \new Staff <<
      \new Voice = "SopOne" {
        \global
        \SopOneMusic
      }
      \new Lyrics \lyricsto "SopOne" {
        \SopOneLyrics
      }
    >>
    \new Staff <<
      \new Voice = "SopTwo" {
        \global
        \SopTwoMusic
      }
      \new Lyrics \lyricsto "SopTwo" {
        \SopTwoLyrics
      }
    >>
  >>
}
```



Questa è la struttura di base di tutte le partiture vocali. Si possono aggiungere più righi, più voci in ogni rigo, più versi nei testi, e le variabili contenenti la musica possono essere poste in file separati se dovessero diventare troppo lunghe.
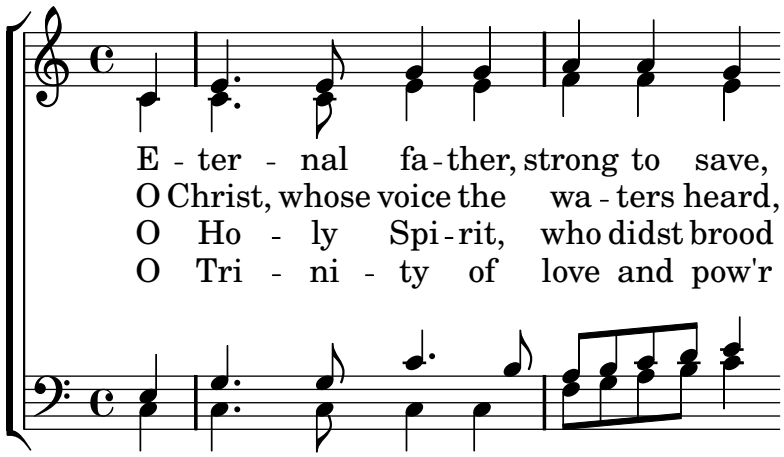
Ecco un esempio della prima linea di un inno con quattro strofe, impostate su SATB. In questo caso le parole per tutte e quattro le parti sono le stesse. Si noti l'uso delle variabili per separare la notazione musicale e le parole dalla struttura del rigo. Si veda anche come una variabile, che abbiamo deciso di chiamare 'keyTime', venga usata per avere vari comandi a disposizione all'interno dei due righi. In altri esempi questo viene spesso chiamato 'global'.

```
keyTime = { \key c \major \time 4/4 \partial 4 }

SopMusic   = \relative c' { c4 | e4. e8 g4  g    | a4   a   g  }
AltoMusic  = \relative c' { c4 | c4. c8 e4  e    | f4   f   e  }
TenorMusic = \relative c  { e4 | g4. g8 c4.   b8 | a8 b c d e4 }
BassMusic  = \relative c  { c4 | c4. c8 c4  c    | f8 g a b c4 }

VerseOne =
  \lyricmode { E -- | ter -- nal fa -- ther, | strong to save, }
VerseTwo   =
  \lyricmode { O | Christ, whose voice the | wa -- ters heard, }
VerseThree =
  \lyricmode { O | Ho -- ly Spi -- rit, | who didst brood }
VerseFour  =
  \lyricmode { O | Tri -- ni -- ty of | love and pow'r }

\score {
  \new ChoirStaff <<
    \new Staff <<
      \clef "treble"
      \new Voice = "Sop"  { \voiceOne \keyTime \SopMusic }
      \new Voice = "Alto" { \voiceTwo \AltoMusic }
      \new Lyrics \lyricsto "Sop" { \VerseOne   }
      \new Lyrics \lyricsto "Sop" { \VerseTwo   }
      \new Lyrics \lyricsto "Sop" { \VerseThree }
      \new Lyrics \lyricsto "Sop" { \VerseFour  }
    >>
    \new Staff <<
      \clef "bass"
      \new Voice = "Tenor" { \voiceOne \keyTime \TenorMusic }
      \new Voice = "Bass"  { \voiceTwo \BassMusic }
    >>
  >>
}
```



Vedi anche

Guida alla notazione: Sezione "Vocal music" in *Guida alla Notazione*.
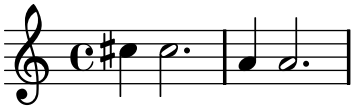
3.3 Contesti e incisori

I contesti e gli incisori sono stati menzionati in modo informale nelle sezioni precedenti; ora dobbiamo approfondire questi concetti, perché sono importanti nell'ottimizzazione dell'output di LilyPond.

3.3.1 I contesti

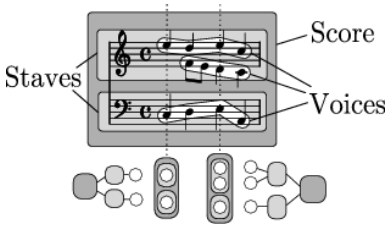
Quando la musica viene elaborata, molti elementi notazionali che non compaiono esplicitamente nel file di input devono essere aggiunti nell'output. Ad esempio, si confrontino l'input e l'output del seguente esempio:

```
cis4 cis2. | a4 a2. |
```



L'input è alquanto essenziale, ma nell'output sono stati aggiunti stanghette, alterazioni, l'armatura di chiave e il tempo. Quando LilyPond *interpreta* l'input l'informazione musicale viene analizzata da sinistra a destra, proprio come un musicista legge uno spartito. Mentre legge l'input, il programma ricorda dove si trovano i confini della misura, e quali altezze richiedono espliciti accidenti. Questa informazione deve essere conservata a diversi livelli. Ad esempio, un accidente influisce solo su un singolo rigo, mentre una stanghetta deve essere sincronizzata lungo l'intera partitura.

All'interno di LilyPond, queste regole e pezzi di informazione sono raggruppati nei *Contesti*. Abbiamo già presentato il contesto `Voice`. Altri contesti sono `Staff` e `Score`. I contesti sono strutturati gerarchicicamente per riflettere la natura gerarchica di una partitura musicale. Ad esempio: un contesto `Staff` può contenere molti contesti `Voice`, e un contesto `Score` può contenere molti contesti `Staff`.



Ogni contesto è responsabile di far rispettare alcune regole di notazione, creare alcuni oggetti della notazione e conservare le proprietà associate. Ad esempio, il contesto `Voice` può introdurre

un'alterazione e poi il contesto `Staff` mantiene la regola per mostrare o sopprimere l'alterazione per il resto della misura.

Un altro esempio: la sincronizzazione delle stanghette è gestita, per impostazione predefinita, nel contesto `Score`. Tuttavia, in alcune forme musicali potremmo non volere che le stanghette siano sincronizzate – si consideri una partitura polimetrica in 4/4 e 3/4. In questi casi, dobbiamo modificare le impostazioni predefinite dei contesti `Score` e `Staff`.

Per spartiti molto semplici, i contesti vengono creati implicitamente, e non è necessario occuparsene. Ma per brani più ampi, come qualsiasi cosa abbia più di un rigo, devono essere creati esplicitamente per essere sicuri di avere tutti i righi necessari, e che questi siano nel giusto ordine. Per scrivere brani che utilizzano una notazione speciale, di solito si modificano i contesti esistenti, o addirittura se ne creano di completamente nuovi.

Oltre ai contesti `Score`, `Staff` e `Voice` ci sono contesti che stanno tra i livelli della partitura (score) e del rigo (staff) per controllare i gruppi di pentagrammi, come i contesti `PianoStaff` e `ChoirStaff`. Ci sono anche contesti alternativi per il rigo e la voce, e contesti per il testo, le percussioni, la tastiera, il basso continuo, etc.

I nomi di tutti i tipi di contesto sono formati da una o più parole, e ogni parola viene unita immediatamente alla parola precedente senza trattini o underscore bensì con la prima lettera maiuscola: ad esempio, `GregorianTranscriptionStaff`.

Vedi anche

Guida alla notazione: Sezione "Contexts explained" in *Guida alla Notazione*.

3.3.2 Creare i contesti

In un file di input un blocco della partitura, introdotto dal comando `\score`, contiene un'espressione musicale singola e una definizione di output associata (o un blocco `\layout` o un blocco `\midi`). Di solito si lascia che il contesto `Score` sia creato automaticamente quando inizia l'interpretazione di quell'espressione musicale.

Per le partiture che hanno una sola voce e un solo rigo, si può lasciare che i contesti `Voice` e `Staff` siano creati automaticamente, ma per le partiture più complesse è necessario crearli manualmente. Il comando più semplice per farlo è `\new`. Viene posto prima di un'espressione musicale, ad esempio

```
\new tipo espressione-musicale
```

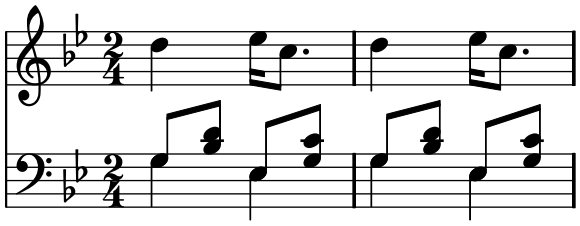
dove *tipo* è il nome di un contesto (come `Staff` o `Voice`). Questo comando crea un nuovo contesto, e inizia a interpretare *espressione-musicale* all'interno di quel contesto.

(Si noti che normalmente non è necessario il comando `\new Score`, perché il fondamentale contesto di livello superiore `Score` viene creato automaticamente quando l'espressione musicale all'interno del blocco `\score` viene interpretata. L'unica ragione per creare un contesto `Score` esplicitamente con `\new Score` è di inserire un blocco `\with` dove si possono specificare uno o più valori delle proprietà di contesto, predefiniti per tutto lo spartito.. Le informazioni su come usare i blocchi `\with` si trovano sotto il titolo "Setting context properties with `\\with`" in ⟨undefined⟩ [Modifying context properties], pagina ⟨undefined⟩.)

> **Nota:** `\new Score` non dovrebbe essere usato perché il fondamentale contesto di livello superiore `Score` viene creato automaticamente quando l'espressione musicale all'interno del blocco `\score` viene interpretata. I valori predefiniti delle proprietà di contesto validi per tutta la partitura possono essere modificati nel blocco `\layout`. Vedi ⟨undefined⟩ [Modifying context properties], pagina ⟨undefined⟩

Nelle sezioni precedenti hai già visto molti esempi pratici della creazione di nuovi contesti `Staff` e `Voice`, ma per ricordarti come questi comandi vengano usati in pratica, ecco un esempio di musica vera e propria:

```
\score {  % start of single compound music expression
  <<  % start of simultaneous staves section
    \time 2/4
    \new Staff {  % create RH staff
      \clef "treble"
      \key g \minor
      \new Voice {  % create voice for RH notes
        \relative c'' {  % start of RH notes
          d4 ees16 c8. |
          d4 ees16 c8. |
        }  % end of RH notes
      }  % end of RH voice
    }  % end of RH staff
    \new Staff <<  % create LH staff; needs two simultaneous voices
      \clef "bass"
      \key g \minor
      \new Voice {  % create LH voice one
        \voiceOne
        \relative g {  % start of LH voice one notes
          g8 <bes d> ees, <g c> |
          g8 <bes d> ees, <g c> |
        }  % end of LH voice one notes
      }  % end of LH voice one
      \new Voice {  % create LH voice two
        \voiceTwo
        \relative g {  % start of LH voice two notes
          g4 ees |
          g4 ees |
        }  % end of LH voice two notes
      }  % end of LH voice two
    >>  % end of LH staff
  >>  % end of simultaneous staves section
}  % end of single compound music expression
```



(Si noti che tutte le asserzioni che aprono un blocco o con delle parentesi graffe, `{`, o con delle parentesi angolari doppie, `<<`, sono indentate di due spazi, e la parentesi di chiusura corrispondente è indentata esattamente dello stesso spazio. Pur non essendo un requisito indispensabile, seguire questa pratica ridurrà enormemente il numero di errori da 'parentesi spaiate', ed è quindi fortemente consigliato. Permette di vedere a colpo d'occhio la struttura della musica, e qualsiasi parentesi spaiata sarà facilmente riconoscibile. Si noti anche come il rigo LH faccia uso di parentesi angolari doppie perché richiede due voci, mentre il rigo RH è costituito da una singola espressione musicale compresa tra parentesi graffe perché richiede una sola voce.)

Il comando `\new` può anche dare un nome identificativo al contesto per distinguerlo da altri contesti dello stesso tipo,

```
\new tipo = id espressione-musicale
```

Si noti la distinzione tra il nome del tipo di contesto, `Staff`, `Voice`, etc, e il nome identificativo di un particolare esempio di quel tipo, che può essere qualsiasi sequenza di lettere inventata dall'utente. Nel nome identificativo si possono usare anche numeri e spazi, ma in questo caso deve essere compreso tra virgolette, ovvero `\new Staff = "MioPentagramma 1" espressione-musicale`. Il nome identificativo viene utilizzato per riportare a quel particolare esempio di un contesto. Abbiamo visto questo utilizzo nella sezione sul testo, si veda Sezione 3.2.3 [Voci e musica vocale], pagina 56.

Vedi anche

Guida alla notazione: Sezione "Creating contexts" in *Guida alla Notazione*.

3.3.3 Gli incisori

Ogni segno presente nell'output di una partitura realizzata con LilyPond è prodotto da un `Engraver` (incisore). Dunque c'è un incisore per creare i righi, uno per le teste delle note, uno per i gambi, uno per le cediglie, etc, etc. In totale ci sono più di 120 incisori! Fortunatamente, per gran parte delle partiture è necessario conoscerne pochi soltanto, e per partiture semplici non occorre conoscerne alcuno.

Gli incisori vivono ed operano all'interno dei Contesti. Incisori come il `Metronome_mark_engraver`, la cui azione e il cui output si applicano alla partitura nel suo complesso, operano nel contesto di livello più superiore – il contesto `Score`.

Gli incisori `Clef_engraver` e `Key_engraver` devono invece trovarsi in ogni contesto `Staff`, poiché righi diversi potrebbero richiedere diverse chiavi e tonalità.

Gli incisori `Note_heads_engraver` e `Stem_engraver` abitano ogni contesto `Voice`, il contesto che si trova al livello più basso di tutti.

Ogni incisore elabora gli oggetti specifici associati alla sua funzione, e gestisce le proprietà che a quella funzione si riferiscono. Queste proprietà, come le proprietà associate ai contesti, possono essere modificate per cambiare il funzionamento dell'incisore o l'aspetto di quegli elementi nella partitura.

Gli incisori hanno tutti dei nomi composti formati da parole che descrivono la loro funzione. Solo la prima parola inizia con una maiuscola, e il resto è collegato insieme con dei trattini bassi. Quindi l'incisore `Staff_symbol_engraver` ha il compito di creare le linee del pentagramma, il `Clef_engraver` determina e definisce il punto di riferimento dell'altezza sul rigo disegnando il simbolo di una chiave.

Ecco alcuni degli incisori più utilizzati insieme alla loro funzione. Vedrai che di solito è facile indovinare la funzione a partire dal nome, o viceversa.

Incisore	**Funzione**
Accidental_engraver	Crea le alterazioni, le alterazioni di precauzione e di cortesia
Beam_engraver	Incide le cediglie
Clef_engraver	Incide le chiavi
Completion_heads_engraver	Separa le note che attraversano le stanghette
New_dynamic_engraver	Crea le forcelle e i testi relativi alla dinamica
Forbid_line_break_engraver	Impedisce l'a capo se un elemento musicale è ancora attivo
Key_engraver	Crea l'armatura di chiave
Metronome_mark_engraver	Incide il tempo metronomico
Note_heads_engraver	Incide le teste delle note
Rest_engraver	Incide le pause

Staff_symbol_engraver	Incide le cinque linee (predefinite) del pentagramma
Stem_engraver	Crea i gambi e i tremoli su singoli gambi
Time_signature_engraver	Crea l'indicazione di tempo

Vedremo in seguito come cambiare l'output di Lilypond modificando il comportamento degli incisori.

Vedi anche

Guida al funzionamento interno: Sezione "Engravers and Performers" in *Guida al Funzionamento Interno*.

3.3.4 Modificare le proprietà di contesto

I contesti si occupano di tenere i valori di un certo numero di *proprietà* del contesto. Molte di queste possono essere cambiate per influenzare l'interpretazione dell'input e quindi modificare l'aspetto dell'output. Per cambiarle si usa il comando `\set`. Questo assume la seguente forma

```
\set NomeDelContesto.nomeDellaProprietà = #valore
```

Dove il *NomeDelContesto* è di solito `Score`, `Staff` o `Voice`. Può essere omesso, e in questo caso viene considerato il contesto attuale (solitamente `Voice`).

I nomi delle proprietà del contesto consistono in parole unite insieme e senza lineette o trattini bassi, e solo la prima parola inizia con una lettera maiuscola. Ecco alcuni esempi di quelle più usate comunemente. Ma ne esistono molte altre.
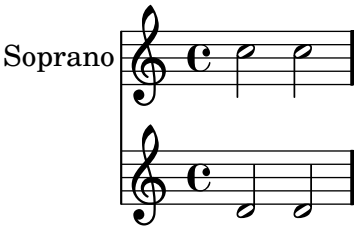
nomeDellaProprietà	**Tipo**	**Funzione**	**Valore di esempio**
extraNatural	Booleano	Se è vero, mette i segni di bequadro prima degli accidenti	`#t`, `#f`
currentBarNumber	Intero	Imposta il numero della battuta corrente	`50`
doubleSlurs	Booleano	Se è vero, stampa le legature di portamento sia sopra che sotto le note	`#t`, `#f`
instrumentName	Testo	Imposta il nome da inserire all'inizio del rigo	`"Cello I"`
fontSize	Reale	Aumenta o riduce la dimensione del carattere	`2.4`
stanza	Testo	Imposta il testo da stampare prima dell'inizio di una strofa	`"2"`

mentre un Booleano è o Vero (`#t`) o Falso (`#f`), un Intero è un numero intero positivo, un Reale è un numero decimale positivo o negativo, e il testo è racchiuso tra virgolette. Si noti la presenza del segno cancelletto, (`#`), in due punti diversi – come parte del valore Booleano prima di `t` o `f`, e prima del *valore* nell'affermazione `\set`. Dunque quando si inserisce un valore Booleano bisogna scrivere due cancelletti, ad esempio `##t`.

Prima di poter impostare una qualsiasi di queste proprietà dobbiamo sapere in quale contesto esse operino. Talvolta questo è ovvio, ma talvolta può risultare complicato. Se viene specificato il contesto sbagliato, non viene generato alcun messaggio di errore, ma l'azione desiderata non avrà luogo. Ad esempio, la proprietà `instrumentName` risiede ovviamente nel contesto `Staff`, poiché è il pentagramma a dover essere nominato. Nell'esempio seguente viene etichettato il primo pentagramma, ma non il secondo, perché abbiamo omesso il nome del contesto.

```
<<
  \new Staff \relative c'' {
    \set Staff.instrumentName = #"Soprano"
```

```
    c2 c
 }
  \new Staff \relative c' {
    \set instrumentName = #"Alto"  % Wrong!
    d2 d
 }
>>
```



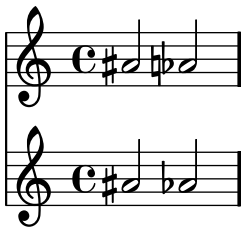
Ricorda che il nome di contesto predefinito è `Voice`, dunque il secondo comando `\set` imposta la proprietà `instrumentName` nel contesto `Voice` col valore "Alto", ma dato che LilyPond non trova tale proprietà nel contesto `Voice`, non ha avuto luogo alcuna altra azione. Questo non è un errore, e nessun messaggio di errore viene riportato nel file di log.

Analogamente, se il nome della proprietà è stato scritto male, non viene generato alcun messaggio di errore, e ovviamente l'azione desiderata non può essere eseguita. Infatti, col comando `\set` puoi impostare qualsiasi 'proprietà' (anche inventata) usando qualsiasi nome che ti piaccia in qualsiasi contesto esistente. Ma se LilyPond non conosce il nome, allora non verrà intrapresa alcuna azione. Alcuni editor di testo che hanno uno speciale supporto per i file di input di LilyPond presentano i nomi delle proprietà con una lista scorrevole al passaggio del mouse, come JEdit col plugin LilyPondTool, oppure evidenziano in modo diverso i nomi delle proprietà non conosciuti, come fa ConTEXT. Se non usi un editor con queste funzionalità, è consigliabile controllare il nome delle proprietà nella Guida al funzionamento interno: si veda Sezione "Tunable context properties" in *Guida al Funzionamento Interno*, o Sezione "Contexts" in *Guida al Funzionamento Interno*.

La proprietà `instrumentName` funzionerà soltanto se inserita nel contesto `Staff`, ma alcune proprietà possono essere collocate in più di un contesto. Ad esempio, la proprietà `extraNatural` è impostata di default sul valore ##t (vero) su tutti i righi. Se viene impostata su ##f (falso) in uno specifico contesto `Staff` si applica solo alle alterazioni presenti su quel rigo. Se impostata su falso nel contesto `Score` si applica a tutti i righi.

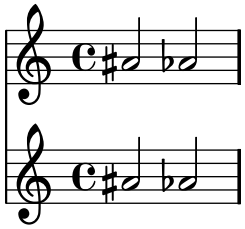
Quindi in questo modo si disattiva il bequadro su un rigo:

```
<<
  \new Staff \relative c'' {
    ais2 aes
 }
  \new Staff \relative c'' {
    \set Staff.extraNatural = ##f
    ais2 aes
 }
>>
```



e in questo modo si disattiva in tutti i righi:

```
<<
  \new Staff \relative c'' {
    ais2 aes
 }
  \new Staff \relative c'' {
    \set Score.extraNatural = ##f
    ais2 aes
 }
>>
```
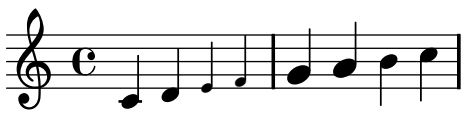


Un altro esempio: se la proprietà `clefOctavation` viene posta nel contesto `Score`, cambia immediatamente il valore dell'ottavazione in tutti i righi presenti e imposta un nuovo valore predefinito che sarà applicato a tutti i righi.

Il comando opposto, `\unset`, di fatto rimuove la proprietà dal contesto, e questo fa sì che molte proprietà tornino al valore predefinito. Solitamente `\unset` non è necessario dal momento che un nuovo comando `\set` permetterà di ottenere quanto si desidera.

I comandi `\set` e `\unset` possono trovarsi in qualsiasi punto del file di input e avranno effetto dal momento in cui si incontrano fino alla fine della partitura o finché la proprietà non viene attivata (`\set`) o disattivata (`\unset`) di nuovo. Proviamo a cambiare varie volte la dimensione del font, che influisce (tra le altre cose) sulla dimensione delle teste delle note. La modifica è relativa al valore predefinito, non all'ultimo valore impostato.

```
c4 d
% make note heads smaller
\set fontSize = #-4
e4 f |
% make note heads larger
\set fontSize = #2.5
g4 a
% return to default size
\unset fontSize
b4 c |
```



Abbiamo visto come impostare i valori di diversi tipi di proprietà. Si note che gli interi e i numeri sono sempre preceduti da un segno di cancelletto, `#`, mentre un valore vero o falso è

specificato con `##t` e `##f`, con due cancelletti. Una proprietà testuale dovrebbe essere racchiusa tra virgolette, come abbiamo visto prima, sebbene vedremo in seguito che in realtà il testo può essere specificato in un modo molto più generale usando il potente comando `\markup`.
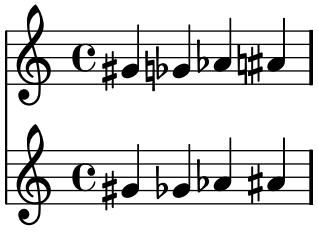
Impostare le proprietà di contesto con `\with`

Il valore predefinito delle proprietà di contesto possono essere impostate anche nel momento in cui il contesto viene creato. Talvolta questo è un modo più chiaro per specificare il valore della proprietà se questa deve rimanere fissa per la durata del contesto. Un contesto creato col comando `\new` può essere immediatamente seguito da un blocco `\with { .. }` dove vengono impostati i valori predefiniti della proprietà. Ad esempio, se vogliamo eliminare la stampa del bequadro per l'estensione di un rigo possiamo scrivere:

```
\new Staff \with { extraNatural = ##f }
```

così:

```
<<
  \new Staff {
    \relative c'' {
      gis4 ges aes ais
    }
  }
  \new Staff \with { extraNatural = ##f } {
    \relative c'' {
      gis4 ges aes ais
    }
  }
>>
```



Si possono ancora cambiare dinamicamente le proprietà impostate in questo modo usando `\set`, mentre con `\unset` si possono riportare al valore predefinito impostato nel blocco `\with`.
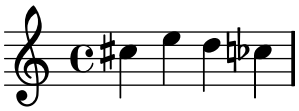
Quindi se la proprietà `fontSize` viene inserita in una proposizione `\with`, imposta il valore predefinito della dimensione del font. Se viene in seguito modificato con `\set`, questo nuovo valore predefinito può essere recuperato col comando `\unset fontSize`.

Impostare le proprietà di contesto con `\context`

I valori delle proprietà di un contesto possono essere impostate in *tutti* i contesti di un particolare tipo, così come in tutti i contesti `Staff`, con un solo comando. Il tipo di contesto viene identificato attraverso il suo nome, come `Staff`, preceduto da una barra inversa (backslash): `\Staff`. La dichiarazione che imposta il valore della proprietà è la stessa che abbiamo visto nel blocco `\with`, introdotto prima. Viene posta in un blocco `\context` all'interno di un blocco `\layout`. Ogni blocco `\context` avrà effetto su tutti i contesti del tipo specificato nel blocco `\score` o `\book` nel quale il blocco `\layout` si trova. Ecco un esempio per mostrare la struttura:

```
\score {
  \new Staff {
    \relative c'' {
```

```
      cis4 e d ces
    }
  }
  \layout {
    \context {
      \Staff
      extraNatural = ##t
    }
  }
}
```



Le proprietà di contesto impostate in questo modo possono essere sovrascritte per alcuni particolari contesti attraverso asserzioni in un blocco `\with`, e tramite comandi `\set` incorporati nelle asserzioni musicali.

Vedi anche

Guida alla notazione: Sezione “Changing context default settings” in *Guida alla Notazione*, Sezione “The set command” in *Guida alla Notazione*.

Guida al funzionamento interno: Sezione “Contexts” in *Guida al Funzionamento Interno*, Sezione “Tunable context properties” in *Guida al Funzionamento Interno*.

3.3.5 Aggiungere e togliere gli incisori

Abbiamo visto che ciascuno dei contesti contiene vari incisori, ognuno dei quali ha il compito di produrre una parte specifica dell’output, come stanghette, righi, teste, gambi, etc.. Se un incisore viene rimosso da un contesto, non può più produrre il suo output. Si tratta di un metodo sbrigativo per modificare l’output, e talvolta può essere utile.

Cambiare un singolo contesto

Per rimuovere un incisore da un singolo contesto usiamo il comando `\with` posto subito dopo il comando di creazione del contesto, come esposto nella sezione precedente.

A titolo di esempio, ripetiamo un esempio della sezione precedente rimuovendo le linee del rigo. Ricorda che le linee del rigo vengono create dall’incisore `Staff_symbol_engraver`.

```
\new Staff \with {
  \remove Staff_symbol_engraver
}
\relative c' {
  c4 d
  \set fontSize = #-4  % make note heads smaller
  e4 f |
  \set fontSize = #2.5  % make note heads larger
  g4 a
  \unset fontSize  % return to default size
  b4 c |
}
```
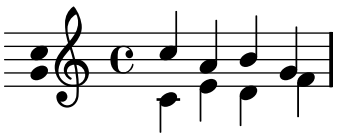


Gli incisori possono essere aggiunti anche a contesti individuali. Il comando per farlo è

`\consists` *`Engraver_name`*,
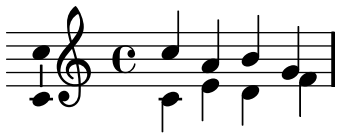
posto all'interno di un blocco `\with`. Alcune partiture vocali hanno un ambitus collocato all'inizio del pentagramma per indicare l'intervallo tra la nota più grave e quella più acuta in quella parte – si veda Sezione "ambitus" in *Glossario Musicale*. L'ambitus viene creato dall'incisore `Ambitus_engraver`, che di norma non è incluso in alcun contesto. Se lo aggiungiamo al contesto `Voice`, calcola l'intervallo di quella voce soltanto:

```
\new Staff <<
  \new Voice \with {
    \consists Ambitus_engraver
  } {
    \relative c'' {
      \voiceOne
      c4 a b g
    }
  }
  \new Voice {
    \relative c' {
      \voiceTwo
      c4 e d f
    }
  }
>>
```



ma se aggiungiamo l'incisore ambitus al contesto `Staff`, calcola l'intervallo di tutte le note in tutte le voci di quel pentagramma:

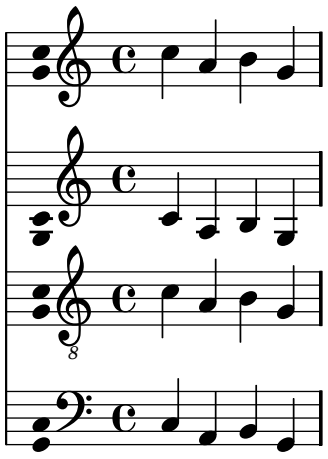
```
\new Staff \with {
  \consists Ambitus_engraver
}
<<
  \new Voice {
    \relative c'' {
      \voiceOne
      c4 a b g
    }
  }
  \new Voice {
    \relative c' {
      \voiceTwo
      c4 e d f
    }
  }
>>
```



Modificare tutti i contesti dello stesso tipo

Gli esempi precedenti mostrano come rimuovere o aggiungere degli incisori a contesti individuali. È anche possibile rimuovere o aggiungere gli incisori per ogni contesto di un particolare tipo inserendo i comandi nel contesto appropriato in un blocco `\layout`. Ad esempio, se volessimo mostrare un ambitus per ogni pentagramma in una partitura di quattro pentagrammi, potremmo scrivere

```
\score {
  <<
    \new Staff {
      \relative c'' {
        c4 a b g
      }
    }
    \new Staff {
      \relative c' {
        c4 a b g
      }
    }
    \new Staff {
      \clef "G_8"
      \relative c' {
        c4 a b g
      }
    }
    \new Staff {
      \clef "bass"
      \relative c {
        c4 a b g
      }
    }
  >>
  \layout {
    \context {
      \Staff
      \consists Ambitus_engraver
    }
  }
}
```



I valori delle proprietà di un contesto possono essere impostati anche per tutti i contesti di un particolare tipo includendo il comando `\set` in un blocco `\context` nello stesso modo.

Vedi anche

Guida alla notazione: Sezione “Modifying context plug-ins” in *Guida alla Notazione*, Sezione “Changing context default settings” in *Guida alla Notazione*.

3.4 Estendere i modelli

Hai letto il tutorial, sai come scrivere la musica, comprendi i concetti fondamentali. Ma come puoi ottenere i pentagrammi che desideri? Puoi trovare molti modelli (vedi Appendice A [Templates], pagina 89) da cui partire. Ma se ti serve qualcosa che non è presente lì? Continua a leggere.

3.4.1 Soprano e violoncello

Parti dal modello che sembra più simile a ciò che vuoi ottenere. Diciamo che vuoi scrivere qualcosa per soprano e violoncello. In questo caso, inizieremmo col modello ‘Note e testo’ (per la parte di soprano).

```
\version "2.14.1"

melodia = \relative c' {
  \clef "treble"
  \key c \major
  \time 4/4
  a4 b c d
}

testo = \lyricmode {
  Aaa Bee Cee Dee
}

\score {
  <<
    \new Voice = "one" {
      \autoBeamOff
      \melodia
    }
    \new Lyrics \lyricsto "one" \testo
  >>
  \layout { }
  \midi { }
}
```

Ora vogliamo aggiungere una parte per violoncello. Vediamo l'esempio 'Solo note':

```
\version "2.14.1"

melodia = \relative c' {
  \clef "treble"
  \key c \major
  \time 4/4
  a4 b c d
}

\score {
  \new Staff \melodia
  \layout { }
  \midi { }
}
```

Non ci servono due comandi `\version`. Ci servirà invece la sezione `melodia`. Non vogliamo due sezioni `\score` – se avessimo due `\score`, le due parti sarebbero separate. Ma le vogliamo insieme, come un duetto. All'interno della sezione `\score`, non ci servono due `\layout` o due `\midi`.

Se semplicemente tagliamo e incolliamo la sezione `melodia`, finiremo con l'avere due definizioni `melodia`. Questo non produrrebbe un errore, ma la seconda definizione sarebbe usata per entrambe le melodie. Dunque rinominiamole per distinguerle. Chiameremo la sezione per il soprano `musicaSoprano` e la sezione per il violoncello `musicaVioloncello`. Se facciamo questo, rinominiamo anche `testo` in `testoSoprano`. Ricorda di rinominare entrambe le occorrenze di tutti questi nomi – sia la definizione iniziale (la parte `melody = \relative c' {`) sia l'uso del nome (nella sezione `\score`).

Cambiamo anche il pentagramma della parte per violoncello – che normalmente usa la chiave di basso. Daremo anche al violoncello delle note diverse.

```
\version "2.14.1"

musicaSoprano = \relative c' {
  \clef "treble"
  \key c \major
  \time 4/4
  a4 b c d
}

testoSoprano = \lyricmode {
  Aaa Bee Cee Dee
}

musicaVioloncello = \relative c {
  \clef "bass"
  \key c \major
  \time 4/4
  d4 g fis8 e d4
}

\score {
  <<
    \new Voice = "one" {
```

```
      \autoBeamOff
      \musicaSoprano
    }
    \new Lyrics \lyricsto "one" \testoSoprano
  >>
  \layout { }
  \midi { }
}
```

L'aspetto è promettente, ma la parte del violoncello non apparirà nella partitura – perché non l'abbiamo inserita nella sezione `\score`. Se vogliamo che la parte del violoncello compaia sotto quella del soprano, dobbiamo aggiungere

```
\new Staff \musicaVioloncello
```

sotto la parte del soprano. Dobbiamo anche aggiungere `<<` e `>>` intorno alla musica – che dice a LilyPond che c'è più di una cosa (in questo caso, due `Staff`) che sono simultanei. Lo `\score` ora appare così:

```
\score {
  <<
  <<
    \new Voice = "one" {
      \autoBeamOff
      \musicaSoprano
    }
    \new Lyrics \lyricsto "one" \testoSoprano
  >>
  \new Staff \musicaVioloncello
  >>
  \layout { }
  \midi { }
}
```

L'aspetto è un po' confuso; l'indentazione non è pulita. Ma si può correggere facilmente. Ecco il template completo per soprano e violoncello.

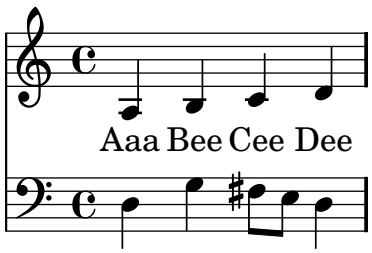
```
\version "2.14.1"

musicaSoprano = \relative c' {
  \clef "treble"
  \key c \major
  \time 4/4
  a4 b c d
}

testoSoprano = \lyricmode {
  Aaa Bee Cee Dee
}

musicaVioloncello = \relative c {
  \clef "bass"
  \key c \major
  \time 4/4
  d4 g fis8 e d4
}
```

```
\score {
  <<
    <<
      \new Voice = "one" {
        \autoBeamOff
        \musicaSoprano
      }
      \new Lyrics \lyricsto "one" \testoSoprano
    >>
    \new Staff \musicaVioloncello
  >>
  \layout { }
  \midi { }
}
```

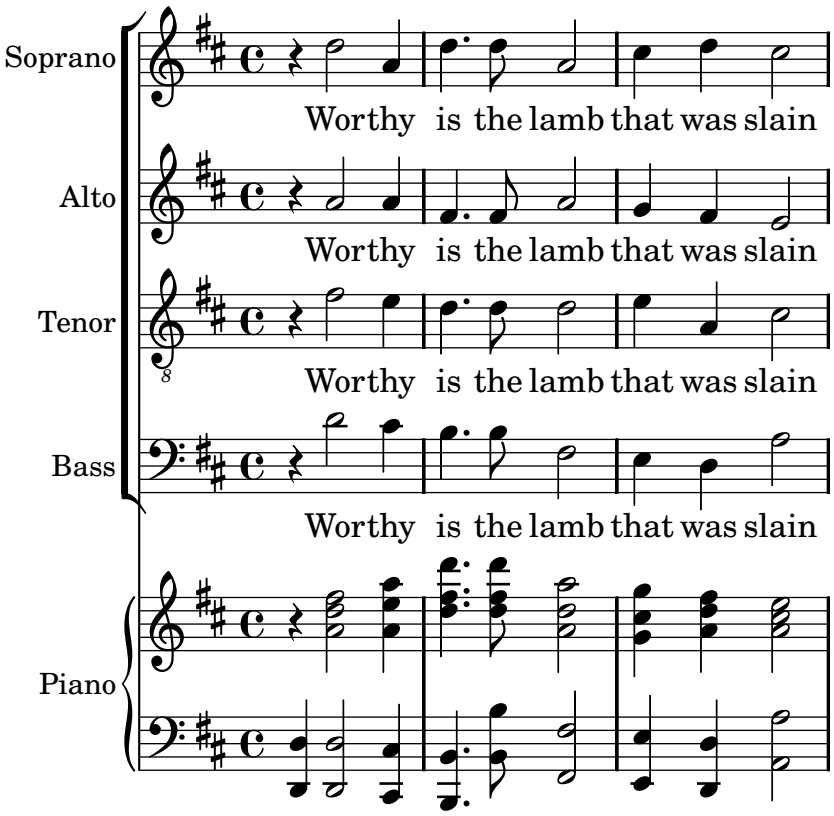


Vedi anche

I modelli da cui partire si trovano nell'appendice 'Modelli', si veda ⟨undefined⟩ [Single staff], pagina ⟨undefined⟩.

3.4.2 Partitura vocale a quattro parti SATB

Gran parte delle partiture vocali di musica scritta per coro misto a quattro voci con accompagnamento di orchestra, come l'Eliah di Mendelssohn o il Messiah di Handel, hanno la musica corale e le parole su quattro righi, una per ognuna delle voci SATB, con in basso una riduzione per pianoforte dell'accompagnamento orchestrale. Ecco un esempio tratto dal Messiah di Handel:



Nessun modello fornisce esattamente questo assetto. Quello più simile è 'Partitura vocale SATB e riduzione automatica per pianoforte' – si veda ⟨undefined⟩ [Vocal ensembles], pagina ⟨undefined⟩ – ma abbiamo bisogno di modificare l'assetto e aggiungere un accompagnamento per pianoforte che non sia derivato automaticamente dalle parti vocali. Le variabili che si riferiscono alla musica e alle parole per le parti vocali vanno bene, ma dovremo aggiungere le variabili per la riduzione per pianoforte.

L'ordine con cui i contesti appaiono nel ChoirStaff del modello non corrisponde all'ordine della partitura vocale mostrata sopra. Dobbiamo risistemarli in modo che ci siano quattro righi con le parole scritte direttamente sotto le note di ogni parte. Tutte le voci devono essere `\voiceOne`, che è l'impostazione predefinita, quindi i comandi `\voiceXXX` devono essere tolti. Dobbiamo anche specificare la chiave di tenore per i tenori. Il modo in cui il testo viene specificato nel modello non è stato ancora esaminato quindi dovremo usare un metodo con cui siamo già familiari. Aggiungeremo anche i nomi per ogni rigo.

In questo modo il nostro ChoirStaff avrà questo aspetto:

```
\new ChoirStaff <<
  \new Staff = "sopranos" <<
    \set Staff.instrumentName = #"Soprano"
    \new Voice = "sopranos" {
      \global
      \sopranoMusic
    }
  >>
  \new Lyrics \lyricsto "sopranos" {
    \sopranoWords
  }
  \new Staff = "altos" <<
    \set Staff.instrumentName = #"Alto"
    \new Voice = "altos" {
      \global
      \altoMusic
```

```
    }
  >>
  \new Lyrics \lyricsto "altos" {
    \altoWords
  }
  \new Staff = "tenors" <<
    \set Staff.instrumentName = #"Tenor"
    \new Voice = "tenors" {
      \global
      \tenorMusic
    }
  >>
  \new Lyrics \lyricsto "tenors" {
    \tenorWords
  }
  \new Staff = "basses" <<
    \set Staff.instrumentName = #"Bass"
    \new Voice = "basses" {
      \global
      \bassMusic
    }
  >>
  \new Lyrics \lyricsto "basses" {
    \bassWords
  }
>>  % end ChoirStaff
```

Poi dobbiamo lavorare sulla parte per pianoforte. Questo è facile - basta prendere la parte per pianoforte dal modello 'Pianoforte solista':

```
\new PianoStaff <<
  \set PianoStaff.instrumentName = #"Piano  "
  \new Staff = "upper" \upper
  \new Staff = "lower" \lower
>>
```

e aggiungere le definizioni delle variabili `upper` e `lower`.

ChoirStaff e PianoStaff devono essere uniti attraverso parentesi angolari, perché vogliamo che siano impilati uno sopra l'altro:

```
<<  % combine ChoirStaff and PianoStaff one above the other
  \new ChoirStaff <<
    \new Staff = "sopranos" <<
      \new Voice = "sopranos" {
        \global
        \sopranoMusic
      }
    >>
    \new Lyrics \lyricsto "sopranos" {
      \sopranoWords
     }
    \new Staff = "altos" <<
      \new Voice = "altos" {
        \global
        \altoMusic
```

```
      }
    >>
    \new Lyrics \lyricsto "altos" {
      \altoWords
    }
    \new Staff = "tenors" <<
      \clef "G_8"  % tenor clef
      \new Voice = "tenors" {
        \global
        \tenorMusic
      }
    >>
    \new Lyrics \lyricsto "tenors" {
      \tenorWords
    }
    \new Staff = "basses" <<
      \clef "bass"
      \new Voice = "basses" {
        \global
        \bassMusic
      }
    >>
    \new Lyrics \lyricsto "basses" {
      \bassWords
    }
  >>  % end ChoirStaff

  \new PianoStaff <<
    \set PianoStaff.instrumentName = #"Piano"
    \new Staff = "upper" \upper
    \new Staff = "lower" \lower
  >>
>>
```

Unendo tutto questo e aggiungendo la musica per le tre battute dell'esempio precedente, otteniamo:

```
\version "2.14.1"

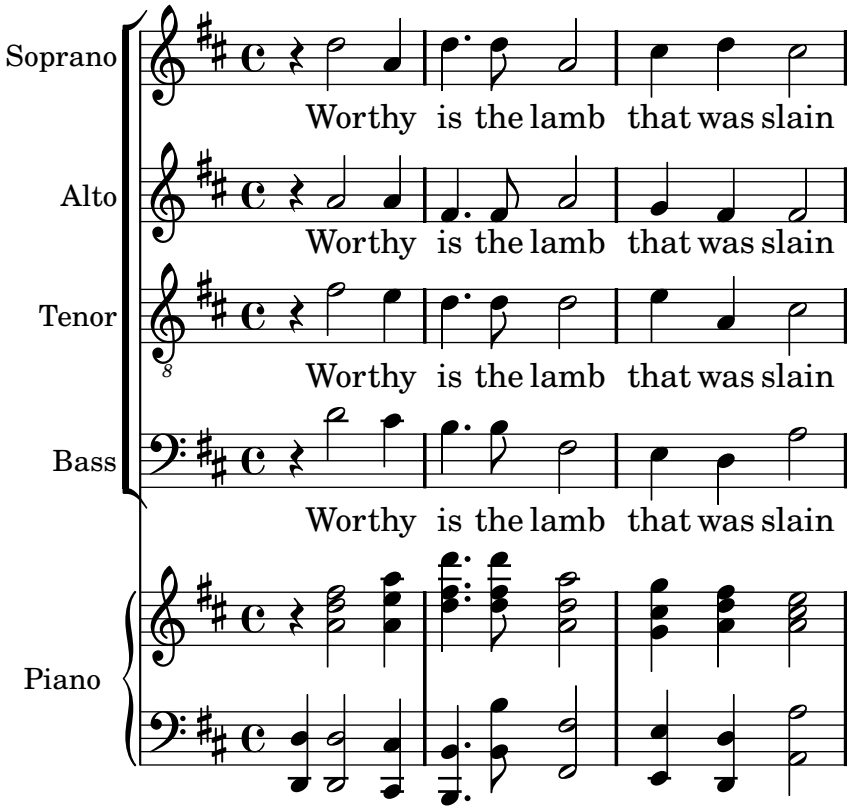
global = { \key d \major \time 4/4 }
sopranoMusic = \relative c'' {
  \clef "treble"
  r4 d2 a4 | d4. d8 a2 | cis4 d cis2 |
}
sopranoWords = \lyricmode {
  Wor -- thy | is the lamb | that was slain |
}
altoMusic = \relative a' {
  \clef "treble"
  r4 a2 a4 | fis4. fis8 a2 | g4 fis fis2 |
}
altoWords = \sopranoWords
tenorMusic = \relative c' {
  \clef "G_8"
```

```
  r4 fis2 e4 | d4. d8 d2 | e4 a, cis2 |
}
tenorWords = \sopranoWords
bassMusic = \relative c' {
  \clef "bass"
  r4 d2 cis4 | b4. b8 fis2 | e4 d a'2 |
}
bassWords = \sopranoWords
upper = \relative a' {
  \clef "treble"
  \global
  r4 <a d fis>2 <a e' a>4 |
  <d fis d'>4. <d fis d'>8 <a d a'>2 |
  <g cis g'>4 <a d fis> <a cis e>2 |
}
lower = \relative c, {
  \clef "bass"
  \global
  <d d'>4 <d d'>2 <cis cis'>4 |
  <b b'>4. <b' b'>8 <fis fis'>2 |
  <e e'>4 <d d'> <a' a'>2 |
}

\score {
  <<  % combine ChoirStaff and PianoStaff in parallel
    \new ChoirStaff <<
      \new Staff = "sopranos" <<
        \set Staff.instrumentName = #"Soprano"
        \new Voice = "sopranos" {
          \global
          \sopranoMusic
        }
      >>
      \new Lyrics \lyricsto "sopranos" {
        \sopranoWords
      }
      \new Staff = "altos" <<
        \set Staff.instrumentName = #"Alto"
        \new Voice = "altos" {
          \global
          \altoMusic
        }
      >>
      \new Lyrics \lyricsto "altos" {
        \altoWords
      }
      \new Staff = "tenors" <<
        \set Staff.instrumentName = #"Tenor"
        \new Voice = "tenors" {
          \global
          \tenorMusic
        }
```

```
      >>
      \new Lyrics \lyricsto "tenors" {
        \tenorWords
      }
      \new Staff = "basses" <<
        \set Staff.instrumentName = #"Bass"
        \new Voice = "basses" {
          \global
          \bassMusic
        }
      >>
      \new Lyrics \lyricsto "basses" {
        \bassWords
      }
    >>  % end ChoirStaff

    \new PianoStaff <<
      \set PianoStaff.instrumentName = #"Piano  "
      \new Staff = "upper" \upper
      \new Staff = "lower" \lower
    >>
  >>
}
```



3.4.3 Scrivere una partitura da zero

Dopo aver acquisito un po' di confidenza nella scrittura del codice LilyPond, potresti scoprire che è più facile costruire una partitura da zero piuttosto che modificare uno dei modelli. In

questo modo puoi anche sviluppare il tuo stile per adattarlo al tipo di musica che vuoi. Come esempio, vediamo come mettere insieme la partitura di un preludio per organo.

Cominciamo con una sezione d'intestazione. Qui va il titolo, il nome del compositore, etc, poi vengono le varie definizioni, e infine il blocco della partitura. Spieghiamo questi prima a grandi linee e in seguito aggiungeremo i dettagli.

Useremo le prime due battute del preludio di Bach basato su *Jesu, meine Freude*, che è scritto per organo con due manuali e pedaliera. Puoi vedere queste due battute in fondo a questa sezione. La parte per il manuale superiore ha due voci, quella inferiore e la pedaliera ne hanno una. Abbiamo quindi bisogno di quattro definizioni musicali e di una definizione per stabilire il tempo e la tonalità:

```
\version "2.14.1"
\header {
  title = "Jesu, meine Freude"
  composer = "J S Bach"
}
keyTime = { \key c \minor \time 4/4 }
ManualOneVoiceOneMusic = { s1 }
ManualOneVoiceTwoMusic = { s1 }
ManualTwoMusic = { s1 }
PedalOrganMusic = { s1 }

\score {
}
```

Per ora abbiamo usato soltanto una nota spaziatrice, `s1`, invece di musica vera e propria. La aggiungeremo in seguito.

Ora vediamo cosa dovrebbe andare nel blocco della partitura. Mostriamo soltanto la struttura del pentagramma che vogliamo. La musica per organo di solito viene scritta su tre righi, uno per ogni mano e uno per i pedali. I righi della tastiera dovrebbero essere raggruppati insieme con una graffa, dunque dobbiamo usare PianoStaff per loro. La parte del primo manuale ha bisogno di due voci mentre la parte per il secondo manuale di una soltanto.

```
\new PianoStaff <<
  \new Staff = "ManualOne" <<
    \new Voice {
      \ManualOneVoiceOneMusic
    }
    \new Voice {
      \ManualOneVoiceTwoMusic
    }
  >>  % end ManualOne Staff context
  \new Staff = "ManualTwo" <<
    \new Voice {
      \ManualTwoMusic
    }
  >>  % end ManualTwo Staff context
>>  % end PianoStaff context
```

Poi dobbiamo aggiungere un rigo per i pedali. Questo va sotto il PianoStaff, ma deve svolgersi simultaneamente a quest'ultimo, quindi abbiamo bisogno delle parentesi angolari intorno ai due gruppi. Altrimenti, verrà generato un errore nel file di log. È un errore comune che farai prima o poi! Prova a copiare l'esempio finale alla fine di questa sezione, togli le parentesi angolari, e compilalo per vedere quali errori genera.

```
<<  % PianoStaff and Pedal Staff must be simultaneous
  \new PianoStaff <<
    \new Staff = "ManualOne" <<
      \new Voice {
        \ManualOneVoiceOneMusic
      }
      \new Voice {
        \ManualOneVoiceTwoMusic
      }
    >>  % end ManualOne Staff context
    \new Staff = "ManualTwo" <<
      \new Voice {
        \ManualTwoMusic
      }
    >>  % end ManualTwo Staff context
  >>  % end PianoStaff context
  \new Staff = "PedalOrgan" <<
    \new Voice {
      \PedalOrganMusic
    }
  >>
>>
```

Non è necessario usare il costrutto simultaneo `<< .. >>` per il rigo del secondo manuale e per quello della pedaliera, poiché contengono solo una espressione musicale, ma non è male usarlo comunque; usare sempre le parentesi angolari dopo `\new Staff` è una buona abitudine da coltivare nel caso ci sia più di una voce. Per le Voci, è vero l'opposto: queste devono essere seguite regolarmente da parentesi graffe `{ .. }` nel caso in cui la musica sia composta da diverse variabili che devono essere eseguite consecutivamente.

Aggiungiamo questa struttura al blocco della partitura, e aggiustiamo l'indentazione. Aggiungiamo anche le chiavi appropriate, controlliamo che i gambi, le legature di portamento e quelle di valore in ogni voce del rigo superiore puntino nella direzione giusta usando `\voiceOne` e `\voiceTwo`, e inseriamo l'armatura di chiave e il tempo per ogni rigo attraverso la nostra variabile predefinita, `\keyTime`.

```
\score {
  <<  % PianoStaff and Pedal Staff must be simultaneous
    \new PianoStaff <<
      \new Staff = "ManualOne" <<
        \keyTime  % set key and time signature
        \clef "treble"
        \new Voice {
          \voiceOne
          \ManualOneVoiceOneMusic
        }
        \new Voice {
          \voiceTwo
          \ManualOneVoiceTwoMusic
        }
      >>  % end ManualOne Staff context
      \new Staff = "ManualTwo" <<
        \keyTime
        \clef "bass"
```

```
        \new Voice {
          \ManualTwoMusic
        }
      >>  % end ManualTwo Staff context
    >>  % end PianoStaff context
    \new Staff = "PedalOrgan" <<
      \keyTime
      \clef "bass"
      \new Voice {
        \PedalOrganMusic
      }
    >>  % end PedalOrgan Staff
  >>
}  % end Score context
```

L'aspetto dei righi dell'organo mostrati sopra è quasi perfetto; tuttavia c'è un piccolo difetto che non è visibile se si guarda un sistema singolo soltanto: La distanza tra il rigo della pedaliera e il rigo della mano sinistra dovrebbe essere all'incirca la stessa distanza tra il rigo della mano destra e quello della mano sinistra. In particolare, l'allungabilità dei righi in un contesto `PianoStaff` è limitata (in modo che la distanza tra i righi della mano destra e sinistra non possa diventare eccessiva), e il rigo della pedaliera dovrebbe comportarsi allo stesso modo.

L'allungabilità dei righi può essere controllata con la proprietà `staff-staff-spacing` dell' 'oggetto grafico' `VerticalAxisGroup` (gli oggetti grafici vengono comunemente chiamati 'grob' nella documentazione di lilypond) – non preoccuparti dei dettagli in questo momento; in seguito verrà fornita una spiegazione approfondita. I curiosi possono dare un'occhiata a Sezione "Overview of modifying properties" in *Guida alla Notazione*. In questo caso, vogliamo modificare soltanto la sottoproprietà `allungabilità`. Di nuovo, chi è curioso può trovare i valori predefiniti per la proprietà staff-staff-spacing nel file '`scm/define-grobs.scm`' guardando la definizione del grob `VerticalAxisGroup`. Il valore dell' `allungabilità` viene preso dalla definizione del contesto `PianoStaff` (nel file '`ly/engraver-init.ly`') così che i valori siano identici.

```
\score {
  <<  % PianoStaff e Pedal Staff devono essere simultanei
    \new PianoStaff <<
      \new Staff = "ManualOne" <<
        \keyTime  % imposta l'armatura di chiave e il tempo
        \clef "treble"
        \new Voice {
          \voiceOne
          \ManualOneVoiceOneMusic
        }
        \new Voice {
          \voiceTwo
          \ManualOneVoiceTwoMusic
        }
      >>  % fine del contesto ManualOne Staff
      \new Staff = "ManualTwo" \with {
        \override VerticalAxisGroup
          #'staff-staff-spacing  #'stretchability = 5
      } <<
        \keyTime
        \clef "bass"
        \new Voice {
```

```
          \ManualTwoMusic
        }
      >>  % fine del contesto ManualTwo Staff
    >>  % fine del contesto PianoStaff
    \new Staff = "PedalOrgan" <<
      \keyTime
      \clef "bass"
      \new Voice {
        \PedalOrganMusic
      }
    >>  % fine di PedalOrgan Staff
  >>
}  % fine del contesto Score
```

Questo completa la struttura. Qualsiasi musica per organo a tre righi avrà una struttura simile, sebbene il numero delle voci possa variare. Tutto ciò che resta da fare ora è aggiungere la musica, e combinare tutte le parti insieme.

```
\version "2.14.1"
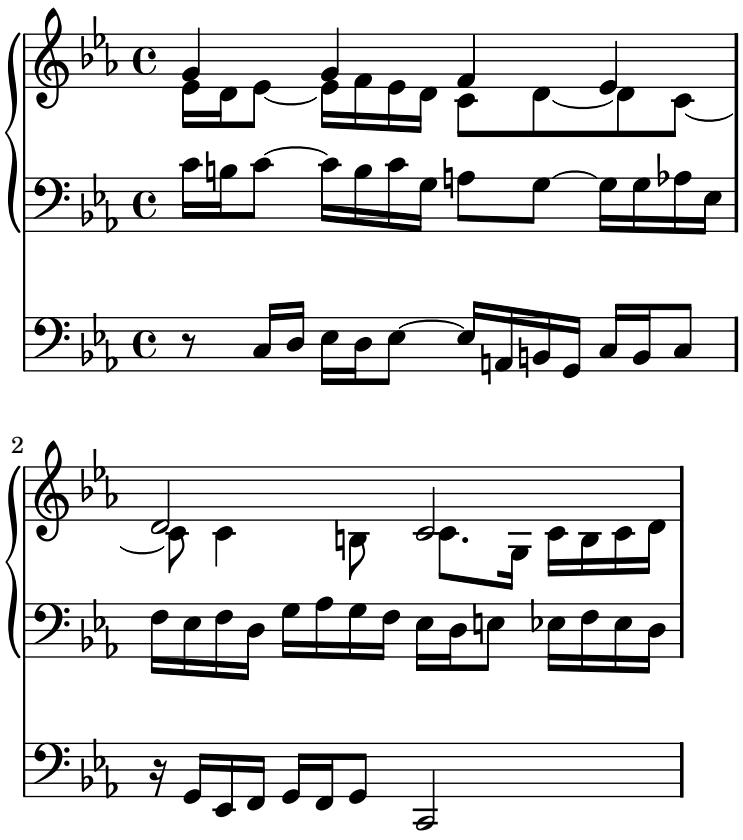
\header {
  title = "Jesu, meine Freude"
  composer = "J S Bach"
}
keyTime = { \key c \minor \time 4/4 }
ManualOneVoiceOneMusic = \relative g' {
  g4 g f ees |
  d2 c |
}
ManualOneVoiceTwoMusic = \relative c' {
  ees16 d ees8~ ees16 f ees d c8 d~ d c~ |
  c8 c4 b8 c8. g16 c b c d |
}
ManualTwoMusic = \relative c' {
  c16 b c8~ c16 b c g a8 g~ g16 g aes ees |
  f16 ees f d g aes g f ees d e8~ ees16 f ees d |
}
PedalOrganMusic = \relative c {
  r8 c16 d ees d ees8~ ees16 a, b g c b c8 |
  r16 g ees f g f g8 c,2 |
}

\score {
  <<  % PianoStaff and Pedal Staff must be simultaneous
    \new PianoStaff <<
      \new Staff = "ManualOne" <<
        \keyTime  % set key and time signature
        \clef "treble"
        \new Voice {
          \voiceOne
          \ManualOneVoiceOneMusic
        }
        \new Voice {
          \voiceTwo
```

```
          \ManualOneVoiceTwoMusic
        }
      >>  % end ManualOne Staff context
      \new Staff = "ManualTwo" \with {
        \override VerticalAxisGroup
          #'staff-staff-spacing #'stretchability = 5
      } <<
        \keyTime
        \clef "bass"
        \new Voice {
          \ManualTwoMusic
        }
      >>  % end ManualTwo Staff context
    >>  % end PianoStaff context
    \new Staff = "PedalOrgan" <<
      \keyTime
      \clef "bass"
      \new Voice {
        \PedalOrganMusic
      }
    >>  % end PedalOrgan Staff context
  >>
}  % end Score context
```

Jesu, meine Freude

J S Bach
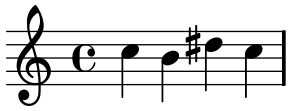


Vedi anche

Glossario musicale: Sezione “system” in *Glossario Musicale*.

3.4.4 Ridurre l’input grazie a variabili e funzioni

Finora hai visto questo tipo di cose:

```
hornNotes = \relative c'' { c4 b dis c }
\score {
  {
    \hornNotes
  }
}
```
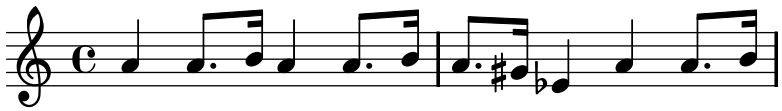


Potresti anche essere accorto che questo può essere utile nella musica minimalista:

```
fragmentA = \relative c'' { a4 a8. b16 }
fragmentB = \relative c'' { a8. gis16 ees4 }

violin = \new Staff {
  \fragmentA \fragmentA |
  \fragmentB \fragmentA |
}

\score {
  {
    \violin
  }
}
```



Tuttavia, puoi usare queste variabili (note anche come macro, o comandi definiti dall’utente) anche per le modifiche manuali:
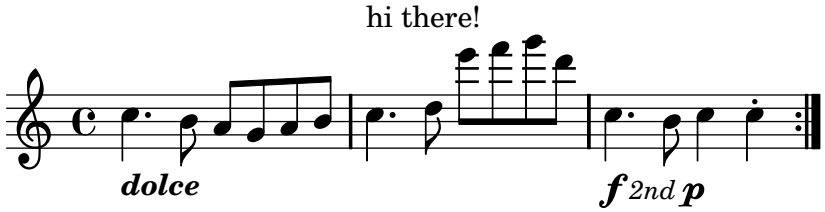
```
dolce = \markup { \italic \bold dolce }

padText = { \once \override TextScript #'padding = #5.0 }
fthenp =_\markup {
  \dynamic f \italic \small { 2nd } \hspace #0.1 \dynamic p
}

violin = \relative c'' {
  \repeat volta 2 {
    c4._\dolce b8 a8 g a b |
    \padText
    c4.^"hi there!" d8 e' f g d |
    c,4.\fthenp b8 c4 c-. |
  }
```

```
}

\score {
  {
    \violin
  }
  \layout { ragged-right = ##t }
}
```



Chiaramente queste variabili sono utili per ridurre la quantità di testo da scrivere. Ma vale la pena tenerle in considerazione anche se le usi una volta sola – perché riducono la complessità. Vediamo l'esempio precedente senza alcuna variabile. È molto difficile da leggere, soprattutto l'ultima linea.
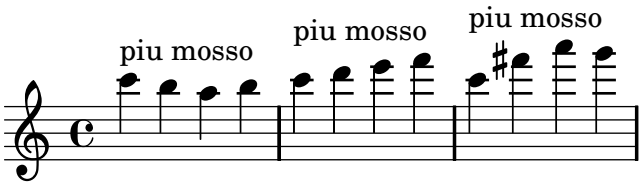
```
violin = \relative c'' {
  \repeat volta 2 {
    c4._\markup { \italic \bold dolce } b8 a8 g a b |
    \once \override TextScript #'padding = #5.0
    c4.^"hi there!" d8 e' f g d |
    c,4.\markup {
      \dynamic f \italic \small { 2nd } \hspace #0.1 \dynamic p
    }
    b8 c4 c-. |
  }
}
```

Finora abbiamo visto la sostituzione statica – quando LilyPond vede `\padText`, lo sostituisce con quel che noi abbiamo definito che sia (ovvero tutto ciò che sta a destra di `padtext=`).

LilyPond può gestire anche la sostituzione non statica (la puoi immaginare come una funzione).

```
padText =
#(define-music-function
     (parser location padding)
     (number?)
   #{
     \once \override TextScript #'padding = $padding
   #})

\relative c''' {
  c4^"piu mosso" b a b |
  \padText #1.8
  c4^"piu mosso" d e f |
  \padText #2.6
  c4^"piu mosso" fis a g |
}
```



L'uso di variabili è anche un buon modo per ridurre il lavoro quando la sintassi di input di LilyPond cambia (vedi Sezione "Updating files with convert-ly" in *Uso del Programma*). Se si ha un'unica definizione (come `\dolce`) per tutti i file di input (vedi ⟨undefined⟩ [Style sheets], pagina ⟨undefined⟩), allora se la sintassi cambia bisogna aggiornare soltanto la singola definizione `\dolce`, invece di dover modificare tutti i file `.ly`.

3.4.5 Partiture e parti

Nella musica orchestrale, tutte le note vengono stampate due volte. Una volta nella parte per i musicisti e una volta nella partitura completa per il direttore. Le variabili sono utili se si vuole evitare il doppio lavoro. La musica viene inserita una volta e salvata all'interno di una variabile. I contenuti di quella variabile vengono poi usati per generare sia la parte che l'intera partitura.

Conviene definire le note in un file speciale. Ad esempio, supponiamo che il file '`horn-music.ly`' contenga la seguente parte di un duetto per corno e fagotto

```
hornNotes = \relative c {
  \time 2/4
  r4 f8 a | cis4 f | e4 d |
}
```

Poi una parte individuale si ottiene inserendo il seguente codice in un file

```
\include "horn-music.ly"
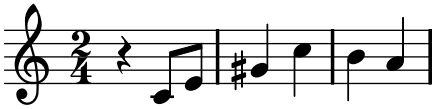
\header {
  instrument = "Horn in F"
}

{
 \transpose f c' \hornNotes
}
```

La linea

```
\include "horn-music.ly"
```

sostituisce i contenuti di '`horn-music.ly`' in quella posizione del file, quindi `hornNotes` è definito dopo. Il comando `\transpose f c'` indica che l'argomento, ovvero `\hornNotes`, deve essere trasposto di una quinta ascendente. La tonalità `f` è indicata dalla nota `c'`, che corrisponde all'accordatura di un normale corno francese in Fa. La trasposizione può essere vista nel seguente output
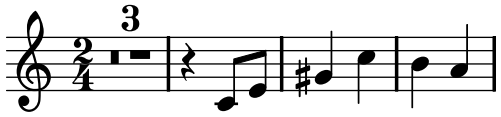


Nei brani di insieme, una delle voci non suona per molte misure. Questo viene indicato da una pausa speciale, la pausa multi-misura. Si inserisce con una `R` maiuscola seguita da una durata (`1` per una semibreve, `2` per una minima, etc.). Moltiplicando la durata, si possono costruire pause più lunghe. Ad esempio, questa pausa prende 3 misure in un tempo di 2/4

```
R2*3
```

Quando la parte viene stampata, le pause multiple devono essere condensate. Si può fare impostando una variabile run-time

```
\set Score.skipBars = ##t
```

Questo comando imposta la proprietà `skipBars` nel contesto `Score` su vero (`##t`). Aggiungendo la pausa e questa opzione alla musica precedente, si arriva al seguente risultato

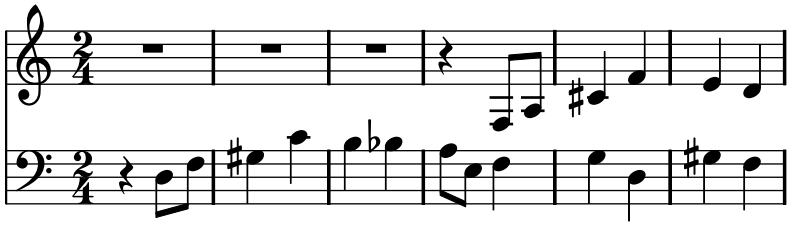


Lo spartito si forma combinando tutta la musica insieme. Assumendo che l'altra voce si trovi in `bassoonNotes` nel file '`bassoon-music.ly`', lo spartito sarà

```
\include "bassoon-music.ly"
\include "horn-music.ly"

<<
  \new Staff \hornNotes
  \new Staff \bassoonNotes
>>
```

ovvero



4 Tweaking output

Appendice A Templates

Appendice B GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
http://fsf.org/

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

   You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

   You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

   If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both

covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its

Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

   However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

   Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with. . . Texts." line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Appendice C Indice di LilyPond

!

! 25

%

% 18
%{ ... %} 18

,

' 14

(

(...) 23

,

, ... 14

.

... . 18

<

< 25, 31
< ... > 31
<< 28, 31
<< ... >> 28
<< ... \\ ... >> 31
<< \\ >> 49

>

> 25, 31
>> 28, 31

[

[..... 26
[...] 26

]

] 26

^

^ 24

_

_ 24

\

\! 25
\(... \) 23
\< 25
\> 25
\\ 31, 49
\acciaccatura 27
\addlyrics 32
\appoggiatura 27
\autoBeamOff 26, 57
\autoBeamOn 26
\book 41, 42
\clef 17
\consists 67
\context 66
\f 25
\ff 25
\grace 27
\header 38, 42
\key 22
\layout 42, 69
\lyricmode 57
\lyricsto 56
\major 22
\markup 25
\mf 25
\midi 42
\minor 22
\mp 25
\new 29, 60
\new ChoirStaff 57
\new Lyrics 56
\new Staff 29
\new Voice 53
\oneVoice 53
\p 25
\partial 26
\pp 25
\relative 14
\remove 67
\score 41, 43
\set 63
\shiftOff 56
\shiftOn 56
\shiftOnn 56
\shiftOnnn 56
\tempo 16
\time 16
\times 27
\unset 63
\version 18
\voiceFour 53
\voiceFourStyle 51
\voiceNeutralStyle 51
\voiceOne 53
\voiceOneStyle 51
\voiceThree 53
\voiceThreeStyle 51
\voiceTwo 53
\voiceTwoStyle 51
\with 66

~

~ 23

A

abbellimenti 27
accento 24
`acciaccatura` 27
accollatura 30
accordi 31
accordi vs. voci 48
`addlyrics` 32
aggiungere gli incisori 67
aggiungere testo 25
allineare il testo 32
allungabilità dei righi 81
alterazioni e armature di chiave 21, 22
alterazioni e modo relativo 14
alterazioni, accidenti 21
altezze 14
altezze, valori assoluti 39
ambitus, incisore dell' 68
anacrusi 26
annidare i costrutti simultanei 55
annidare le espressioni musicali 55
annidare le voci 55
apostrofo 14
appoggiatura 27
armatura di chiave 22
articolazione 24
assegnare le variabili 37
`autoBeamOff` 26, 57
`autoBeamOn` 26

B

basso 17
battuta parziale 26
bemolle 21
bemolle, doppio 21
blocco dell'intestazione 38
blocco, commento di 18
`book` 41, 42
book (libro) 41

C

canzoni 32
caratteri permessi nelle variabili 37
chiave 17
chiave di violino 17
chiave, armatura di, impostare 21
`ChoirStaff` 30, 57
`ChordNames` 29
`clef` 17
cliccabili, esempi 19
code automatiche 26
code manuali 26
code, a mano 26
code, automatiche 26
code, manuali 26
collisioni di note 56
colonna delle note 56
comandi di spostamento 56
combinare le espressioni in parallelo 28
come leggere il manuale 19
commenti 18
commento di blocco 18
commento di linea 18
compilazione 1
composta, espressione musicale 43
consigli su come costruire i file 19
`consists` 67
contenuto di un blocco score 43
contenuto vs. layout 22
contesti della voce, creazione dei 53
contesti impliciti 41
contesti, creazione di 60
contesti, dare un nome 61
contesti, impliciti 41
contesti, spiegazione dei 59
contesto 29
contesto della notazione 29
contesto Voice (voce) 48
contesto, notazione 29
contesto, proprietà del 63
contesto, proprietà del, impostare con \context 66
contesto, proprietà del, impostare con \with 66
contesto, proprietà del, modificare 63
`context` 66
contralto 17
coro, pentagramma per 30
costruire i file, consigli 19
creazione di contesti 60
crescendo 25

D

dare un nome ai contesti 61
decrescendo 25
diesis 21
diesis, doppio 21
dinamica 25
diteggiatura 24
doppio bemolle 21
doppio diesis 21
durate 15
durate della nota 15
durate delle note negli accordi 31

E

`Easier editing` 2, 6
`Easier editing` 13
`Editing facilitato` 1
editor di testo 1
errori comuni 19
`es` 21
eseguire LilyPond in MacOS X 2
eseguire LilyPond in Unix 13
eseguire LilyPond in Windows 6
esempi cliccabili 19
esempio di scrittura di una partitura 78
esempio iniziale 1
esempio, iniziale 1
`eses` 21
espressione musicale 27
espressione musicale composta 27, 43

espressione musicale, composta ... 27
espressione, musicale ... 27
espressioni ... 18
espressioni parallele ... 28
espressioni parallele e note relative ... 28

F

file, consigli per costruirli ... 19
fontSize, impostazione predefinita e impostazione manuale ... 66
formato di input ... 41

G

gambi e direzione delle voci ... 52
gambo in giù ... 52
gambo in su ... 52
`grace` ... 27
graffe, parentesi ... 18
grafici, oggetti ... 81
`GrandStaff` ... 30
grob ... 81
gruppi irregolari ... 27
gruppo di pentagrammi ... 30

H

`header` ... 38, 42

I

identificatori ... 37
implicito, blocco del libro ... 42
impostare le proprietà all'interno dei contesti ... 63
incisori ... 62
incisori, aggiungere ... 67
incisori, rimuovere ... 67
insensibile agli spazi ... 18
intestazioni ... 38
`is` ... 21
`isis` ... 21

K

`key` ... 22

L

`layout` ... 42, 69
layout vs. contenuto ... 22
layout, effetto della posizione del blocco ... 42
legatura di fraseggio ... 23
legatura di portamento ... 23
legatura di portamento, fraseggio ... 23
legatura di valore ... 23
legature che attraversano le parentesi ... 50
legature di portamento vs. legature di valore ... 24
leggere il manuale ... 19
libro ... 41
libro, blocco implicito ... 42
linea di estensione ... 32
linea, commento di ... 18
livelli ... 48
`lyricmode` ... 57
`Lyrics` ... 29, 56
Lyrics, creazione di un contesto ... 56
`lyricsto` ... 56

M

MacOS X, eseguire LilyPond ... 2
macro ... 37
maggiore ... 22
maiuscole, sensibile alle ... 1
`major` ... 22
manuale, leggere ... 19
`Manuals` ... 1
`markup` ... 25
melisma ... 32
metronome marks ... 16
`midi` ... 42
minima ... 15
`minor` ... 22
minore ... 22
modelli ... 19
modelli, SATB ... 73
modello, modificare ... 70
modificare i modelli ... 70
modificare le proprietà del contesto ... 63
modo assoluto ... 39
modo relativo ... 14
modo relativo e polifonia ... 51
modo relativo, e alterazioni ... 14
multipli, righi ... 28, 29
musica simultanea ... 48
musica simultanea e note relative ... 28
musica sincrona ... 48

N

`new` ... 29, 60
`new Staff` ... 29
nomi assoluti delle note ... 39
nomi delle note ... 39
nomi delle note, assoluti ... 39
nota puntata ... 15
notazione delle durate ... 15
notazione delle pause ... 16
notazione semplice ... 13
notazione, semplice ... 13
note relative e espressioni parallele ... 28
note relative e musica simultanea ... 28
note spaziatrici ... 55
note, collisioni di ... 56
numero di versione ... 18
nuovi contesti ... 60

O

oggetti grafici ... 81
`oneVoice` ... 53
ossia ... 46

P

parallele, espressioni ... 28
parentesi graffe ... 18

parentesi, annidare ... 47
parentesi, racchiudere vs. contrassegnare ... 47
parole con sillabe multiple nel testo ... 32
`partial` ... 26
partitura ... 41, 43
partitura, esempio di scrittura ... 78
partiture, multiple ... 42
pausa ... 16
pausa spaziatrice ... 31
PDF file ... 1
pentagramma per coro ... 30
pentagramma per piano ... 30
piano, pentagramma per ... 30
`PianoStaff` ... 30
polifonia ... 28, 31, 48
polifonia e modo relativo ... 51
polifonia su un singolo rigo ... 31
proprietà che operano nei contesti ... 63
proprietà, sottoproprietà ... 81

R

raggruppamento ... 26
`relative` ... 14
`remove` ... 67
righi multipli ... 28, 29
righi multipli e testo ... 36
righi temporanei ... 46
righi, allungabilità ... 81
rigo per un coro ... 57
rigo, posizionamento del ... 46
rimuovere gli incisori ... 67
ritmi ... 15

S

SATB, modelli ... 73
SATB, struttura di ... 58
`score` ... 41, 43
`Score` ... 29
score (partitura) ... 41
score, contenuto del blocco ... 43
scrivere una partitura, esempio ... 78
semibreve ... 15
semiminima ... 15
sensibile alle maiuscole ... 18
`set` ... 63
`shiftOff` ... 56
`shiftOn` ... 56
`shiftOnn` ... 56
`shiftOnnn` ... 56
simultanea, musica ... 48
sincrona, musica ... 48
sottoproprietà ... 81
spaziatrice, pausa ... 31
staccato ... 24
`Staff` ... 29
strofe, vocali, varie ... 58
struttura del file ... 41
struttura di un inno ... 58
struttura di una partitura vocale ... 57

T

template, scrivere il tuo ... 78
`tempo` ... 16
tempo marks ... 16
tempo, indicazione di ... 16
temporanei, righi ... 46
tenore ... 17
terzine ... 27
testi ... 32
testo e righi multipli ... 36
testo e travatura ... 57
testo, aggiungere ... 25
testo, allineare ... 32
testo, collegare a una voce ... 56
testo, creazione di un contesto ... 56
testo, parole polisillabiche ... 32
`time` ... 16
`times` ... 27
tipi di parentesi ... 47
titolo ... 38
tornare alla voce singola ... 53
trattini ... 32
trattino basso ... 32
travatura e testo ... 57
troubleshooting ... 19

U

Unix, eseguire LilyPond ... 13
`unset` ... 63
usare le variabili ... 37

V

valori assoluti per le altezze ... 39
variabili ... 37, 43, 84
variabili, caratteri permessi nelle ... 37
variabili, definire ... 37
variabili, usare ... 37
varie strofe vocali ... 58
varie voci ... 48
vedere la musica ... 1
versionamento ... 18
versione ... 18
virgola ... 14
vocale, partitura, varie strofe ... 58
voci che attraversano le parentesi ... 50
voci e direzione dei gambi ... 52
voci multiple ... 31
voci temporanee ... 55
voci vs. accordi ... 48
voci, annidare ... 55
voci, nome delle ... 50
voci, più su un rigo ... 31
voci, tornare alla voce singola ... 53
voci, varie ... 48
`Voice` ... 29
Voice (voce), contesto di ... 48
`voiceFour` ... 53
`voiceOne` ... 53
`voiceThree` ... 53
`voiceTwo` ... 53

W

Windows, eseguire LilyPond ... 6
`with` ... 66