# Writing tunebooks with Lilypond

The Lilypond documentation is pretty complete but a little light on end-to-end tutorial material.This book shows you how to use the subset of its features required to write songs in lead sheet style. Normally it's a single melody line with lyrics.

# Writing the Score

This section goes through the creation of a simple score step by step.

## The minimum file

- From the **File** menu, choose **New.**

A simple notepad appears.

- The first line you should write is `\version` followed by whatever the current version of Lilypad is in quotes. For example:

```
\version "2.19.54"
```

If you don't know which version you're running, here's how to find out:

- From the **LilyPond** menu, choose **About.**

- Follow it with the notes of the score. The whole file should look like this:

```
\version "2.19.54"
{
     c d e f
}
```

## Save your file

- From the **File** menu, choose **Save.**

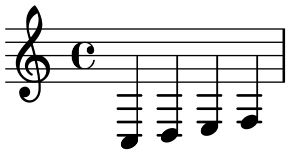
## Compile/typeset your file

Now compile your music file. *Compiling,* also called typesetting, converts the code you've written into beautifully engraved music.

- From the **Compile** menu, choose **Typeset** or press Command+R. (Save your file if you haven't already.)

Results:

```
\version "2.19.54"
{
    c d e f
}
```



## The Output window

Lilypond's output appears in a separate window. In normal operation you'll see many, many lines looking something like this:

```
Processing `/Users/tom/songbook/American Dance 1850-1900.ly'
Parsing...
Interpreting music...
Preprocessing graphical objects...
Calculating line breaks...
Drawing systems...
etc...
```

If your source file contains no errors, the concluding lines look something like this:

```
Layout output to `/var/folders/7k/2n4d6vn6k_mvr/T//lilypond-Ue4wmd'...
Converting to `American Dance 1850-1900.pdf'...
Deleting `/var/folders/7k/2n4d6vn6k_mvr/T//lilypond-Ue4wmd'...
Success: compilation successfully completed
```

If there's an error, for example, you forget a trailing } in a score, a large number of messages like these will scroll past:

```
Layout output to `/var/folders/7k/2n4d6vn6k_mvr/T//lilypond-Ue4wmd'...
'/Users/tom/songbook/American Dance 1850-1900.ly':206:1: error:
markup outside of text script or \lyricmode
'/Users/tom/songbook/American Dance 1850-1900.ly':228:1: error:
Spurious \line in \score
'/Users/tom/songbook/American Dance 1850-1900.ly':228:1: error:
markup outside of text script or \lyricmode
\markup '/Users/tom/songbook/American Dance 1850-1900.ly':1638:3:
error: syntax error, unexpected end of input, expecting '}'
Interpreting music...
Preprocessing graphical objects...
Calculating line breaks...
Drawing systems...
Finding the ideal number of pages...
Fitting music on 1 or 2 pages...
Drawing systems...
Layout output to `/var/folders/7k/2n4hyw3d6vn6k_mvr64mmk4h0000gp
/T//lilypond-01l3cQ'...
Converting to `American Dance 1850-1900.pdf'...
Deleting `/var/folders/7k/22n4d6vn6k_mvr/T//lilypond-Ue4wmd'...
fatal error: failed files: '/Users/tom/songbook/American Dance 1850-1900.ly
'/Users/tom/songbook/American Dance 1850-1900.ly
```

Running across such a giant list of error messages looks intimidating. You will find that it's invariably cascade of problems caused by just one thing. Usually that one thing is the absence of a curly brace, parenthesis, or quote mark.

In fact, let's take another look at the way your score is formatted to clarify how it's formatted.

Lilypond thinks of things in lists. Usually those lists are in between these things, which are sometimes called curly braces: `{ }` . The first score you did, which consisted of a list of four notes between the curly braces, looked like this:

```
\version "2.19.54"
{
    c d e f
}
```

It could ould also be written like this:

```
\version "2.19.54"
{  c d e f }
```

Or:

```
\version "2.19.54"
{
  c d e f }
```

A list just needs its delimiters, such as curly braces, and the items in the list, separated by spaces. The number of spaces between items in the list don't matter. And a carriage return (blank line) functions identically to a space. So does a tab character.

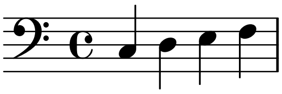
### Always write the opening and closing list characters first

The most important thing to remember about writing any kind of list in Lilypond is to make sure you write the opening and closing characters (for example, `{` and `}` before you write anything else, because the most common errors are when you forget to write a closing list delimiter or quote mark.

## Changing the clef

That's probably not how you want these notes represented. The clef is assumed to be treble. You can specify the clef easily using `\clef` .

```
\version "2.19.54"
{
    \clef bass
    {
         c d e f
    }
}
```

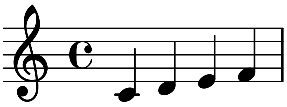


It's not just bass and treble. Lillypond has a huge variety of clefs available. More common clefs include `tenor` , `alto` , and `percussion` , but you'll see a host of others at http://lilypond.org/doc/Documentation/notation/clef-styles

## Specifying a starting pitch

You can specify a relative starting pitch and Lilypond will use it to put the whole sequence of notes in context. Repeat the last example but surround with with a `relative 'c { }` notation:

```
\version "2.19.54"
{
    \relative c' {
         c d e f
    }
}
```

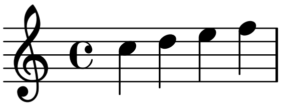


You're not restricted to `c'` . It could be any note, like `g'` or `d'` .

### Raise relative pitch one octave per '

To go up an octave, try it again but with double `''` marks like this, once again surrounding the whole sequence in curly braces:
`relative c'' { }:`

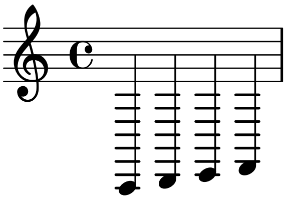
```
\version "2.19.54"
{
    \relative c'' {
         c d e f
    }
}
```



### Lower relative pitch one octave per ,

To go down an octave, append a comma `,` like this:

```
\version "2.19.54"
{
    \relative c, {
         c d e f
    }
}
```

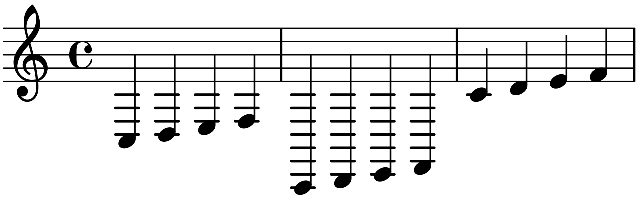


And of course they can be mixed:

```
\version "2.19.54"
{
    \relative c {
         c d e f
    }

    \relative c, {
         c d e f

    }
    \relative c' {
         c d e f
    }
}
```

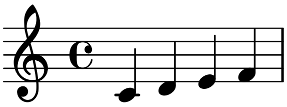


See http://lilypond.org/doc/Documentation/notation/writing-pitches for more details.

## Comments: adding notes to yourself in the Lilypad source

Suppose you want to include a message to yourself in the score but you don't want it to appear in the final output. You can do that with comments. Single-line comments start with a `%` . Copy this, then compile it. The comments do not appear in the final output. That's a feature, not a bug. Comments that go to your Lilypond source, stay in your Lilypond source. They won't be seen in the printed score:

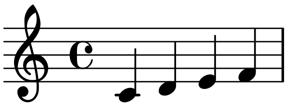
```
\version "2.19.54"
% Have someone double-check this passage
{
    \relative c' {
         c d e f
    }
}
```



### Multi-line comments

Sometimes your comments need many more than one or two lines. There is a second comment type for this purpose. Multi-line comments start with `{%` and end with `%}` .

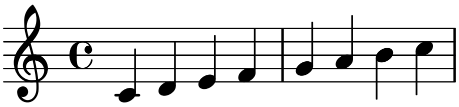
```
\version "2.19.54"
{% Have someone double-check this passage
I've been at this 24 hours.
I'm not thinking straight
And I still have 7 more sections to write %}
{
    \relative c' {
         c d e f
    }
}
```



## Measures are marked automatically

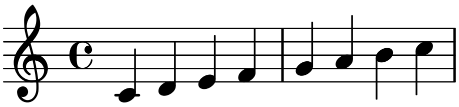
LilyPond keeps track of measures for you automatically.

```
\version "2.19.54"
{
    \relative c' {
         c d e f
         g a b c
    }
}
```



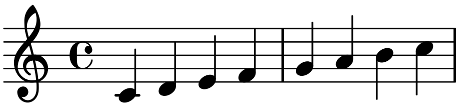
For clarity these examples separate the measures visually, but it's not necessary:

```
\version "2.19.54"
{
    \relative c' {
         c d e f g a b c
    }
}
```



You can add the pipe character ( `|` ) into your markup as a visual aid to remind you where measure breaks occur. If you put it in the wrong place, you'll get a warning that says something like `warning: barcheck failed at: 1/2 c d | e f g a b c` . Lilypond refuses to place a measure line where it wouldn't fit the score:

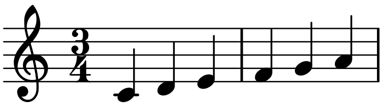
```
\version "2.19.54"
{
    \relative c' {
         % Causes warning message
         c d | e f g a b c
    }
}
```



## Changing the time signature

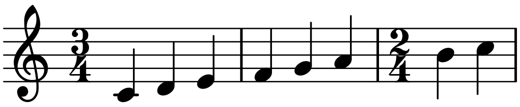
LilyPond defaults to 4 beats per measure but of course that's easily changed using `time` :

```
\version "2.19.54"
{
    \relative c' {
    \time 3/4
         c d e
         f g a
    }
}
```



You can change time signatures at any point.

```
\version "2.19.54"
{
    \relative c' {
    \time 3/4
         c d e
         f g a
    \time 2/4
         b c
    }
}
```



## Accidentals: adding sharp and flat signs

Form accidentals by appending them directly to the note letter. Make sure there is no space after the note letter.

### To make a note flat, append `es`

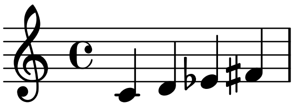
Append `es` to a note name to give it a flat sign.

### To make a note sharp, append `is`

Append `is` to a note name to give it a sharp sign.

In the example below, to make the F an F sharp use the notation `fis` , and to make the E an E flat use `ees` .

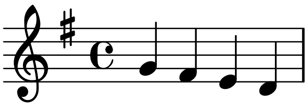
```
\version "2.19.54"
{
    \relative c'
    {
         c d ees fis
    }
}
```



## Changing the key signature

Define the key signature using `\key` , followed by its letter, followed by `\major` or `\minor` .

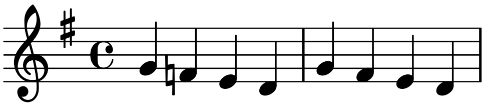
```
\version "2.19.54"
{
    \relative g'
    \key g \major
    {
         g fis e d
    }
}
```



## Note values are absolute and ignore key signature

In the previous example you may have noticed something unexpected. It was the fact that the F sharp was explicitly specified as `fis` even though the key was G. You'd think that you could just use `f` because of the key signature. **Lilypond requires that no matter what the key signature is, each note value must written out completely.** Let's look at an example.

```
\version "2.19.54"
{
    \relative g'
    \key g \major
        % WRONG for the key of G!
        g f e d
        g fis e d
}
```

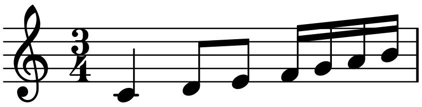


Intuitively, you would think that because the key signature is G major, the first notation example `g f e d` would be sufficient, that LilyPond would understand from context that the sharp sign is implied. Unfortunately that's not the case, so no matter what key you're in you need to write out each note in full, as if it were in the key of C.

## Changing note durations

Lilypond assumes you are using quarter notes. Append `\1` to make it a whole note, `\2` to make it a half note, `\4` for a quarter, `\8` for an eighth... you see the pattern. If the next note is the same duration, you don't have to repeat its duration marking.

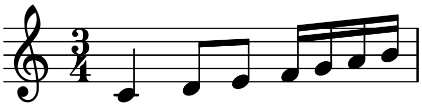
```
\version "2.19.54"
{
    \relative c'
    \time 3/4
    {
         c d8 e8 f16 g16 a16 b16
    }
}
```



### Duration marks apply until the duration changes

If the next note is the same duration, you don't have to repeat its duration marking. This is a simplified version of the previous example.

```
\version "2.19.54"
{
    \relative c'
    \time 3/4
    {
         c d8 e f16 g a b
    }
}
```

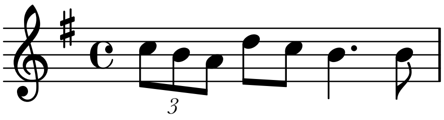


For more, see http://lilypond.org/doc/Documentation/notation/writing-rhythms#durations

## Triplets

Triplets use the `\tuplet 3/2` notation. The `3` is the number of notes to be written and the `2` is the number of notes that would normally be written there.

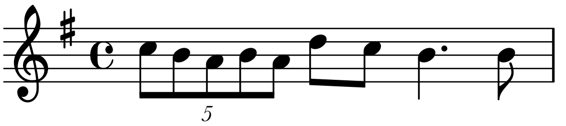
```
\version "2.19.54"
{
    \relative g' {
    \key g \major
    \clef treble
    \time 4/4
         \tuplet 3/2 { c8 b a } d c b4. b8
    }
}
```



### Figures other than triplets

The `\tuplet` idea may have sounded more general to you than use for mere triplets. If so, you were right. The fraction-style notation can be used for ther figures. For example, if you wanted 5 beats for the quarter note duration, you would use `\tuplet 5/2` as shown next.

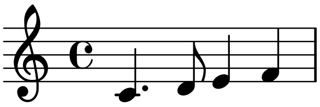
```
\version "2.19.54"
{
    \relative g' {
    \key g \major
    \clef treble
    \time 4/4
         \tuplet 5/2 { c8 b a b a } d c b4. b8
    }
}
```



## Dotted notes

Dotted notes are easy. Just add a dot, like this: **.**

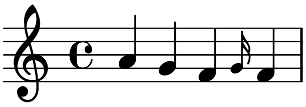
```
\version "2.19.54"
{
    \relative c'
    {
         c. d8 e4 f
    }
}
```



## Grace notes

To add grace notes (which aren't included in the note count), use `\grace` followed by the grace note. In this case the `\grace g16` isn't applied to the measure count.

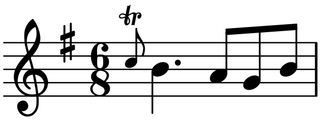
```
\version "2.19.54"
{
    \relative a'
    {
         a4 g f \grace g16 f4
    }
}
```



## Trills

It is common to add a trill to a grace note. Use \trill directly after the note for this purposes.

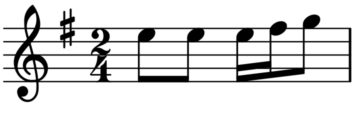
```
\version "2.19.54"
{
    \relative g'
    \key g \major
    \clef treble
    \time 6/8
    {
         \grace c8\trill b4. a8 g b
    }
}
```



## Beams

Lilypond does a good job guessing when notes need beams. Let's start with this example.

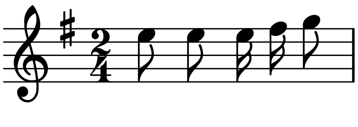
```
\version "2.19.54"
{
 \relative e'' {
  \key e \minor
  \time 2/4
  e8 e e16 fis16 g8
 }
}
```



This is no accident. Just as LilyPond comes preset to 4/4 time and the key of C, it starts in a state called \autoBeamOn . As you've already guessed, that means there's also \autoBeamOff .

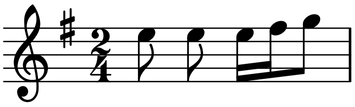
Let's separate notes using \autoBeamOff .

```
\version "2.19.54"
{
 \relative e'' {
  \key e \minor
  \time 2/4
  \autoBeamOff
  e8 e e16 fis16 g8
 }
}
```



And you can mix and match at will, even within the same measure.

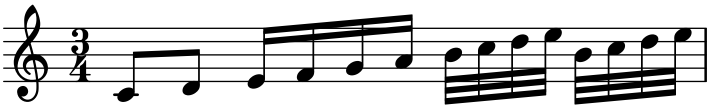
```
\version "2.19.54"
{
 \relative e'' {
  \key e \minor
  \time 2/4
  \autoBeamOff
  e8 e
  \autoBeamOff
  e16 fis16 g8
 }
}
```



### Adding beams: Using `[` and `]` to replace flags with beams

The last note of this measure is an example where one might wish to override `autoBeamOn` . No `autoBeamOff` directive is necessary. For the second half of the note, just use surround the second through fourth notes with square brackets, like this: `b32 [c32 d32 e32]` .

```
\version "2.19.54"
{
 \relative c' {
  \time 3/4
  c8 d
  e16 f g a
  b32 c32 d32 e32
  b32 [c32 d32 e32]
 }
}
```

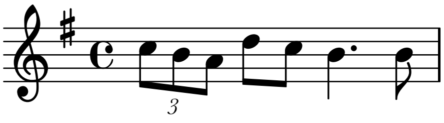


## Phrase marks

To add a phrase mark, surround **all but the first note in the phrase** in parentheses: `( )` .

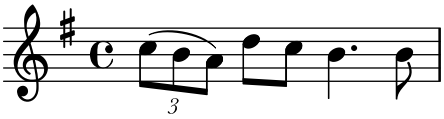
In this example, we want the triplet to use a phrase mark. Here it is without the phrase mark.

```
\version "2.19.54"
\relative g' {
    \key g \major
    \clef treble
    \time 4/4
    \tuplet 3/2 {c8 b a} d c b4. b8
}
```



Note the parentheses around the `(b a)` only; not the `(c8 b a)` as you might suppose.

```
\version "2.19.54"
\relative g' {
    \key g \major
    \clef treble
    \time 4/4
    \tuplet 3/2 {c8 (b a)} d c b4. b8
}
```



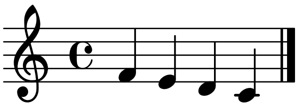
## Bar lines ending a phrase or piece

Lilypond has every kind of bar line you need to end the piece or a part of it.

### Ending with a double bar

Use `\bar` and the bar style to end a piece.

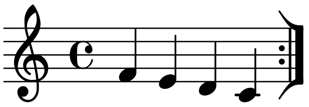
```
\version "2.19.54"
{
    \relative c'
    {
         f e d c \bar "|."
    }
}
```



### Ending with a repeat

Use `\bar ":|]"` to repeat a phrase.

```
\version "2.19.54"
{
    \relative c'
    {
         f e d c \bar ":|]"
    }
}
```

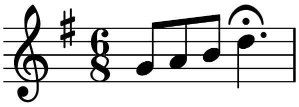


Lilypond has more bar options than most sane composers have ever contemplated. See http://lilypond.org/doc/Documentation/notation/bars for more details.

## Fermatas

To add a fermata over a note, simply append `\fermata` to it with no space.

```
\version "2.19.54"
{
 \relative g' {
  \key g \major
  \clef treble
  \time 6/8
  g8 a b d4.\fermata
 }
}
```



See
http://lilypond.org/doc/v2.19/Documentation/notation/list-of-articulations#fermata-scripts for more fermata options.