

[More quick guides]

A quick guide to musical notation in Prince



Prince is a HTML-to-PDF-via-CSS converter which offers advanced layout capabilities. Prince excels at formatting text, images and figures into PDF documents. Prince does not support musical notation natively, but will happily read SVG files with musical scores. In this quick guide, you will learn how to embed musical notation in HTML files and generate the necessary SVG files behind the scene. This tutorial requires some knowledge of linux and how to install software on your own machine. With the tools in place, you will be able to generate the [PDF version](#) of this document.

The purpose of this guide is to show how small musical snippets can be included in your documents. For example, if you are writing an article about a musician, you may want to include a few bar of music to remind the readers of a catchy tune.

Expressing music

Our language of choice for expressing musical notation is [Lilypond](#). Lilypond is also the name of a computer program that interprets this musical notation and converts it into SVG, which Prince can include. Here is a simple Lilypond example:

```
\relative { c' d e( f) g2 a2 }
```

When processed by the Lilypond program, the above code is nicely rendered into an SVG file which Prince and browser can show:



You can find the Lilypond source file in the links on the far right.

Musical notation can be complex, and Lilypond has many advanced features which allows it to generate beautiful musical “engravings”. You can learn more about the Lilypond syntax in their [documentation](#).

Complete example

The small code example above is complete, but it will generate a full sheet and not the small snippets we use in this guide.

In order to create small snippets, one must add some headers:

```
\version "2.22.1"

\paper {
  page-breaking = #ly:one-page-breaking
  paper-width = 170
  indent = 0
  top-margin = 0
  bottom-margin = 0
  left-margin = 0
  right-margin = 0
}

\header {
  tagline = ""
}
```

```
\relative { c' d e( f) g2 a2 }
```

The first header declares the version of Lilypond the code is written for. The code in the `\paper` section turns off page breaking, then sets the output width to 170 millimeters, and removes indentation and margins around the example. In the `\header` section, the default promotional tagline is removed.

Installing tools

One goal of this tutorial is to embed the musical notation inside the HTML code. Having only one file is simpler. However, in order to present the musical notation to Lilypond for processing, it must be extracted from the HTML file. We will use the `htmlq` processor for this, and you will need to install `htmlq` on your own computer to follow this tutorial. (Or, you must find some other way to extract the Lilypond code; copy and paste will also work.)

If you use `snap`, this command should work:

```
$ snap install htmlq
```

Also, we need LilyPond. On a common Linux computer, Lilypond can be installed with:

```
$ sudo apt-get install lilypond
```

The third and final tool needed is `wget`. It's probably already installed on your machine.

Generating the PDF

To generate the PDF version of the document you are reading, you will first need to download the HTML source file. That's easy:

```
$ wget https://css4.pub/2024/music/index.html
```

You should now have a file called `index.html` in your directory. (You may want to create a new directory for this experiment.)

The next step is to run Prince:

```
$ prince --javascript index.html
```

In the console, Prince will complain about missing SVG files. That's ok. It will also show you the command to generate these SVG files:

```
htmlq -t .ly#crh < index.html > crh.ly; lilypond --svg crh.ly; htmlq -t .ly#c < index.html > c.ly; lilypond --svg c.ly; htmlq -t .ly#cr < index.html > cr.ly; lilypond --svg cr.ly; htmlq -t .ly#dv < index.html > dv.ly; lilypond --svg dv.ly; prince index.html
```

It's a bit cryptic, but you will not need to study it. The command is generated by a small script inside the HTML file; the script looks for elements with `class=ly` and an `id` attribute; this is where we hide the Lilypond code:

```
<pre class=ly id=dv>
\version "2.22.1"
...
</pre>
```

When running the command, `htmlq` will extract the Lilypond code and copy it into a separate file. Thereafter, Lilypond will be called to convert the file into SVG. Finally, `prince` is called again to process the document now that the SVG files are available. So, Prince is called twice: first to print out a command, and then to generate the final PDF file.

Linux has an elegant way of handling all this in a one-line command:

```
$ eval "$(prince -j index.html)"
```

If all goes well, you will now have a file called `index.pdf` in your directory.

Adding lyrics

Here's a catchy tune by Händel, with words added underneath:

```
<<
```

```

\relative {
  g'2 e4. f8 g2 c, d8 e f g f4 e4 d1 e8( f) g a g4 g c2 g f4 e d4. c8
c1
}
\addlyrics {
  Deg væ -- re æ -- re Her -- re ov -- er død -- ens makt. E -- vig
skal dø -- den væ -- re Kris -- tus un -- der -- lagt.
}
>>

```

The image shows two staves of musical notation in treble clef with a common time signature (C). The first staff contains the melody for the first line of lyrics: "Deg være ære Herre over dødens makt." The second staff contains the melody for the second line: "Ewig skal døden være Kristus underlagt." The lyrics are written below the notes, with hyphens indicating syllables that span across multiple notes.

When text is introduced, the questions of fonts come up.

Fonts

The Lilypond program will only find fonts that are installed on your computer. You can make Lilypond use specific font names instead of its default generic font families “roman”, “sans”, and “typewriter”. In this code, all three are replaced with the not-so-suitable “Comic Sans MS”:

```

#(define fonts
  (make-pango-font-tree "Comic Sans MS"
    "Comic Sans MS"
    "Comic Sans MS"
    (/ staff-height pt 20)))

```

The result can be seen in this excerpt from the *Christmas Oratorio*:

auf, frei - set die Ta - ge!

Font size

The font size can also be set to an absolute point size:

```
\addlyrics {  
  \markup \abs-fontsize #20 { auf, frei "- set" die Ta "- ge!" }  
}
```

auf, frei - set die Ta - ge!