`AAA-intro.ly`

Introduction

This document shows examples from the .

In the web version of this document, you can click on the file name or figure for each example to see the corresponding input file.

This document is for LilyPond version 2.11.28

'`adding-extra-fingering-with-scheme.ly`'

You can add various stuff to notes using `make-music`. In this example, an extra fingering is attached to a note.

In general, first do a `display` of the music you want to create, then write a function that will structure the music for you.
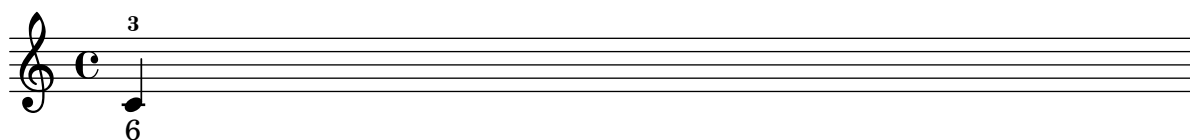
'`adding-staccato-dots.ly`'

Using `make-music`, you can add various stuff to notes. In this example staccato dots are added to the notes. For this simple case, it is not necessary to use scm constructs.

'`changing-properties-for-individual-grobs.ly`'

The `\applyOutput` command gives you the ability to tune any layout object, in any context. It requires a Scheme function with three arguments; advanced users can write it quite easily, whereas new users may want to use pre-defined functions such as this snippet, or the example in the manual.

'`creating-music-with-scheme.ly`'

This example shows prelude in C major of WTK1, but coded using Scheme functions to avoid typing work.
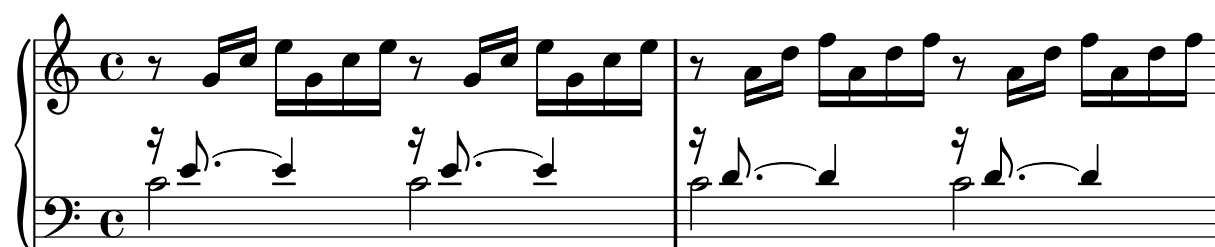
Thanks to its implementation of GUILE, LilyPond makes high level functionalities relatively easy to accomplish.

In this example, the title will mention the current version (i.e. the version the snippet was compiled with, regardless which \version was originally used).

You are running version 2.11.28
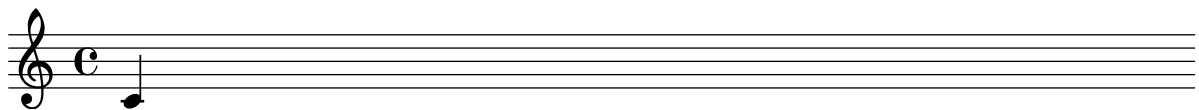
'engraving-music-using-scheme-expressions.ly'

You can engrave music using just Scheme expressions. Although those expressions reflect the inner mechanism of LilyPond, they are rather clumsy to use, so avoid them, if possible.
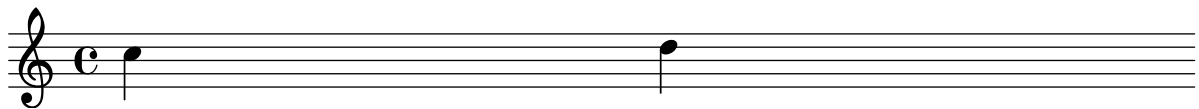
'engraving-symmetric-or-palindromical-music.ly'

Symmetric, or palindromical music can be produced, first, by printing some music, and second, by printing the same music applying a Scheme function to reverse the syntax.

'generating-random-notes.ly'

This Scheme-based snippet allows you to generate 256 random notes based on the current time (or any randomish number you might wish to specify instead, so you can obtain the same random notes each time): i.e. to get different random notes patterns, just change this number.

‘integrating-lilypond-expressions-inside-scheme-functions.ly’

It is possible to use fragments of LilyPond syntax inside Scheme, by putting them between #{ and #}. In this example, three functions are defined to apply different paddings on the TextScript markups, using native LilyPond commands such as "\override TextScript #'padding".

‘move-specific-text.ly’

Objects, like text, can be moved around by using some Scheme code.
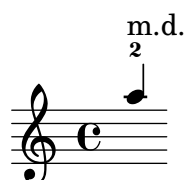
'`transpose-pitches-with-minimum-accidentals.ly`'

There is a way to enforce enharmonic modifications for notes in order to have the minimum number of accidentals. In that case, "Double accidentals should be removed, as well as E-sharp (-> F), bC (-> B), bF (-> E), B-sharp (-> C).", as proposed by a request for a new feature. In this manner, the most natural enharmonic notes are chosen in this example.