# **Sing** *Happy Birthday*
# Musical typesetting with GNU LilyPond

*Henning Hraban Ramm*

## 1. What is LilyPond and how is it related to TEX?

GNU LilyPond is a

- command based
- Open Source (GPL)
- musical typesetting (engraving) system
- with integrated programming language (Scheme)
- and output to PDF, SVG and MIDI.

What does that mean?

- Command based: like TEX, the input is text based, more like programming, and there's no graphical user interface.
- Open Source: As a GNU project, LilyPond is under the GNU Public License (while TEX is in the public domain).
- Musical typesetting (engraving) system: Since works of music were traditionally printed from engraved copper plates – lead letters don't work very well for music –, LilyPond uses "engraving" rather than "typesetting".
- Integrated programming language: If you know Scheme (a LISP dialect), you can influence the output of LilyPond in most details.[1]
- Output to PDF, SVG and MIDI: LilyPond can write EPS (that get converted to PDF), SVG and MIDI files. Other formats can be made from that. MusicXML is becoming a viable choice for music conversion between different systems; it's not yet integrated in the LilyPond core, but e.g. in the Frescobaldi IDE.

The first version of LilyPond was built as a TEX package. That resulted in a few design choices that are familiar to TEX users.
Even if current versions of LilyPond don't depend on TEX any more, there are some similarities between ConTEXt and LilyPond:

- Both are unique Open Source software systems. There's no serious competition (any more – LinoSetting and NOTE are dead).
- Their commands start with a backslash and comments with a percent sign.
- Both stem from the Netherlands.
- Their default fonts were originally designed in MetaFont (Emmentaler[2] vs. Computer Modern). Nowadays OpenType fonts are usable.

[1] Taco remarked at the meeting, if they would have started a few years earlier, LISP would have been a good choice for enhancing TEX also.

[2] Named for its many holes.

- They have a similar font family concept (serif/sans/mono plus math/music).
- Their font glyphs are accompanied by rules (note stems, bar lines).
- Both use an embedded powerful programming language (Lua vs. Scheme).
- Both have an amazing international community.

Also similarly to TeX, there are several editors and IDEs for LilyPond:

- Frescobaldi (powerful LilyPond-only IDE, widely used in the community, active developers)
- Elysium (Eclipse plugin)
- Emacs (there's a LilyPond major mode)
- jEdit (LilyPondTool plugin, abandoned)
- at least two "live" online editors[3]

## 2. Basic Input

The basic music input codes of LilyPond are easy and comprehensible:

- Pitches are letters, `\relative`[4] to some reference note with octave marks: `c d e f g a b cis dis`…, `es bes`…, `r` is a rest. Default are Dutch note names, others like `do re mi fa sol` are just one `\include` away.
- Numbers are durations: `c4` (fourth), `c4.` (dotted fourth), `c16`…
- Chord names: `a b c` are major chords, there are several modifiers: `a:m b:7 c:maj7`. `s` is silence.

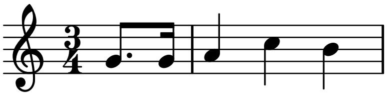
A lot of other codes and commands are also quite easy:

- Bar lines `|` are optional, but work as a debugging helper: LilyPond complains if note durations don't fit into the measure.
- Choose the necessary `\clef`, e.g. `bass`, `G_8`…
- Timing can change within the score, just put a new `\time` at the start of the measure.
- Pickup measures are marked as `\partial` and their length. You don't need to mark incomplete measures at the end.

```
\version "2.18.2" % always define the version of LilyPond used
\relative c'' {
       \clef "G"
       \time 3/4
       \partial 4
       g8. g16 |
       a4 c b |
}
```

[3] One is http://www.weblily.net, can't find another link…

[4] There's also `\absolute` mode, but then you must tell LilyPond the octave of every single note.

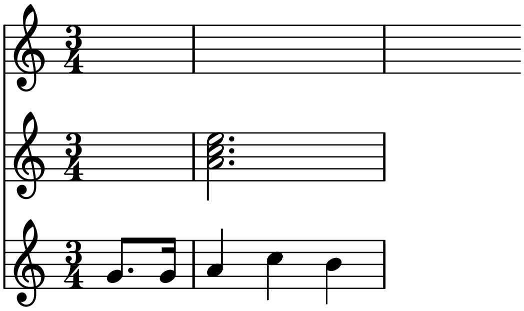


LilyPond complains if you leave out the `\version` mark. LilyPond's syntax changes over time, and if it knows the version of your score, the script `convert-ly` can do most of the conversion work for you automatically.
LilyPond uses always the nearest note, e.g. if you want to go *up* from `f` to `e`, you need the octave modifier `'` – downward is `,`. Try `c4 c, c' c,, c''`.

## 3. Putting it together

You can easily assign "anything" to a name and use it as a variable. Double angles keep different voices together.

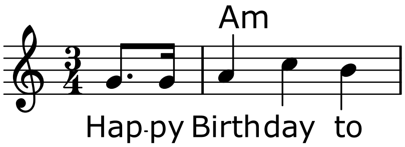
```
Tune = \relative c'' { \time 3/4 \partial 4 g8. g16 | a4 c b | }
Chords = \chordmode { \partial 4 s4 | a2.:m | }
Lyrics = \lyricmode { Hap -- py birth -- day to you! }

<<
       \Chords
       \Tune
       \Lyrics
>>
```



Hm, our chords are displayed as note-chords, but I wanted chord names!
And where are the lyrics?
LilyPond needs a bit more context information.

## 4. Contexts in LilyPond

```
\score{
<<
       \context ChordNames {
              \Chords
       }
       \context Voice = "first" {
              \Tune
       }
       \lyricsto "first" \new Lyrics { \Lyrics }
>>
}
```



That looks better.

Now we can typeset ("engrave") simple songs. If you need more, please read LilyPond's great and extensive documentation![5] There's a learning manual, a notation reference, a multilingual glossary, an extension manual (for programmers) etc.

But LilyPond's *text* typesetting facilities are quite limited and uncomfortable. Usually you would integrate LilyPond code in LaTeX documents and use `lilypond-book` as a preprocessor. But how can I include LilyPond scores in my ConTeXt code?

## 5. LilyPond in ConTeXt

Minimal example:

```
\usemodule[filter]
\def\readPDFfile#1{\externalfigure[#1]}
\defineexternalfilter[lilypond]
  [continue=yes,
   readcommand=\readPDFfile,
   directory=./, % directory for LilyPond's files
   output={\externalfilterbasefile.pdf},
   filtercommand={lilypond -dbackend=eps -o"\externalfilterbasefile"\
                                         "\externalfilterinputfile"}]

\startlilypond[name=some]
\relative c' { c d e f g }
\stoplilypond
```

You could call LilyPond directly via `\write18` or LUATEX's `os.executecommand`, but the most comfortable way is the filter module.

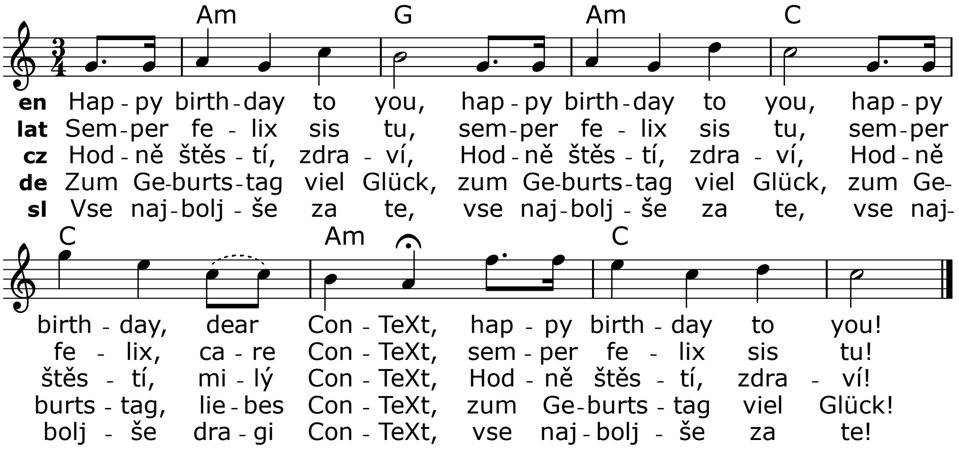
The shown simple setup creates full page LilyPond scores and includes them as external figures. It makes more sense to use single, trimmed note systems and let ConTeXt do the page breaking.

You need some obscure code from `lilypond-book`'s LilyPond preamble to achieve that, and some Lua code to find and include all the systems. You can copy the code from our wiki![6]

[5] http://lilypond.org/manuals.html

[6] http://wiki.contextgarden.net/LilyPond

## 6. Happy Birthday, ConTEXt!


Am G Am C
en Hap - py birth - day to you, hap - py birth - day to you, hap - py
lat Sem - per fe - lix sis tu, sem - per fe - lix sis tu, sem - per
cz Hod - ně štěs - tí, zdra - ví, Hod - ně štěs - tí, zdra - ví, Hod - ně
de Zum Ge-burts-tag viel Glück, zum Ge-burts-tag viel Glück, zum Ge-
sl Vse naj - bolj - še za te, vse naj - bolj - še za te, vse naj-
C Am C
birth - day, dear Con - TeXt, hap - py birth - day to you!
fe - lix, ca - re Con - TeXt, sem - per fe - lix sis tu!
štěs - tí, mi - lý Con - TeXt, Hod - ně štěs - tí, zdra - ví!
burts - tag, lie - bes Con - TeXt, zum Ge - burts - tag viel Glück!
bolj - še dra - gi Con - TeXt, vse naj - bolj - še za te!

By the way: This song is *not* under copyright! Warner/Chappell never had the rights to the song, except one special piano arrangement.[7] Same story with a lot of folk songs.

[7] http://www.latimes.com/local/lanow/la-me-ln-happy-birthday-song-lawsuit-decision-20150922-story.html