

## Coding Annoyances

---

### Pandoc, beamer, MAKE workflow

01 August 2014 • Rubén Hernández-Murillo •

---

The goal is to produce a nicely formatted beamer presentation in pdf format without losing focus away from the slides content by constantly meddling with the latex styles.

The solution is to write the presentation core content in `markdown`, then process the file with `pandoc` to obtain the pdf using a `beamer` template, automating the process with `make`.

We require the following tools. (I won't discuss how to install them in this post.)

1. Pandoc (<http://johnmacfarlane.net/pandoc/>) is a tool used to convert files between different formats. In particular, it is great for converting markdown input files into LaTeX, html, pdf, or Word format, including beamer latex presentations.
2. Texlive (<https://www.tug.org/texlive/>) is one of the many LaTeX system distributions, available for Linux and Windows.
3. Make (<http://www.gnu.org/software/make/>) is a "tool which controls the generation of executables and other non-source files of a program from the program's source files." In our case, we will use it to automate the generation of pdf files from a markdown source using texlive as the engine to compile the latex files.
4. Sed (<http://www.gnu.org/software/sed/>) is a stream editor which operates on files. We will use it to automatically change strings of text using regular expressions.

While `make` and `sed` are common tools in a Linux environment, the GnuWin32 project (<http://gnuwin32.sourceforge.net/>) provides alternatives for Windows.

## Steps

1. The core of the presentation is written in markdown. See example ([http://rubenhm.org/downloads/blog/2014-08-01-pandoc-beamer-make-workflow/example\\_pres.md](http://rubenhm.org/downloads/blog/2014-08-01-pandoc-beamer-make-workflow/example_pres.md)).
2. This file is run through pandoc to generate a `tex` file ([http://rubenhm.org/downloads/blog/2014-08-01-pandoc-beamer-make-workflow/example\\_pres.tex](http://rubenhm.org/downloads/blog/2014-08-01-pandoc-beamer-make-workflow/example_pres.tex)).
3. The above tex file is embedded as input in a beamer presentation template ([http://rubenhm.org/downloads/blog/2014-08-01-pandoc-beamer-make-workflow/presentation\\_example.tex](http://rubenhm.org/downloads/blog/2014-08-01-pandoc-beamer-make-workflow/presentation_example.tex)).
4. The presentation template file is finally processed by `xelatex` to generate a pdf

presentation ([http://rubenhm.org/downloads/blog/2014-08-01-pandoc-beamer-make-workflow/presentation\\_example.pdf](http://rubenhm.org/downloads/blog/2014-08-01-pandoc-beamer-make-workflow/presentation_example.pdf)). We use `xelatex` because in Windows we can use TrueType fonts, for example `Calibri`. The example uses `Times New Roman`.

5. The entire process is automated with `make` (with optional use of `sed`).

Running the following command in the command line will produce the final presentation pdf file and will clean up any auxiliary files, except for the intermediate tex files and the original markdown source.

```
1. make FILE=example
```

To automate the process, the `make` command uses the following set of instructions from a file named `makefile` (<http://rubenhm.org/downloads/blog/2014-08-01-pandoc-beamer-make-workflow/makefile>).

```
1. FILE=
2. MDFILE=$(FILE)_pres
3. TEXFILE=presentation_$(FILE)
4. INPUTS=
5.
6. all: pdf; make clean
7.
8. .PHONY: all clean
9.
10. pdf:
11.     # Use pandoc to convert the main markdown source to latex
12.     pandoc $(MDFILE).md --slide-level 2 -t beamer -o $(MDFILE).tex
13.     # Use sed to replace a string in the latex file that sets spaces in itemize environment
14.     sed -i "s/\\itemsep1pt\\parskip0pt\\parsep0pt/\\itemsep1.6pt\\parskip1.6pt\\parsep0pt/g" $(MDFILE).tex
15.     # Apply xelatex twice to obtain the pdf
16.     xelatex $(TEXFILE).tex
17.     xelatex $(TEXFILE).tex
18.
19. # Remove unnecessary files
20. clean:
21.     -rm -f $(TEXFILE).log $(TEXFILE).aux $(TEXFILE).toc $(TEXFILE).snm $(TEXFILE).nav
22.     $(TEXFILE).out $(TEXFILE).fdb_latemk $(TEXFILE).synctex.gz $(TEXFILE).fls
23.     -rm -f sed*.*
```

The `--slide-level 2` option indicates we can use `##` in the markdown file to generate new slides, and `###` to create subheadings inside a slide. A single `#` creates sections.

## Attributions

This post is my implementation (really a simplified version) of this post in Jeromy Anglim's blog (<http://jeromyanglim.blogspot.com/2012/07/beamer-pandoc-markdown.html>).

Pictures used in the example are attributed to [placekitten.com](http://placekitten.com) (<http://placekitten.com/attribution.html>).

---

Follow for updates.

 (<http://rubenhm.org/feed.xml>)  @rubenhm\_org

