A Memo About Memos

Alejandro Revilla (@apr)

2024-07-24

2024-07-24

Contents

A Memo About Memos

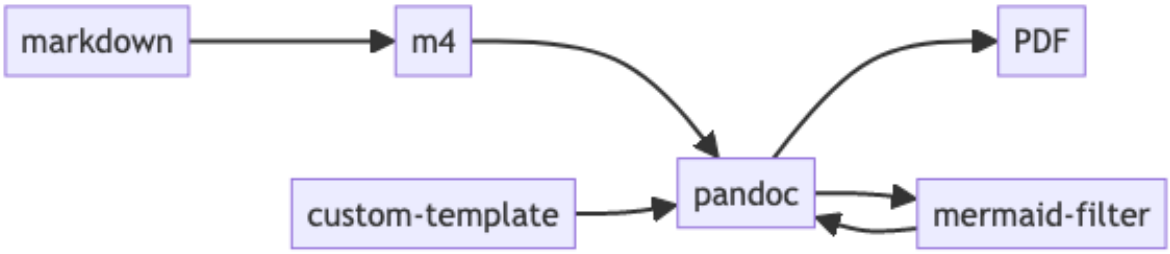
I am such a weak writer that I can't afford to add bad typography on top of my already flawed writing. I'm so challenged when it comes to aesthetics that, whenever I use a WYSIWYG editor—and I've used many since the early PageMaker days—I spend more time figuring out fonts and paragraph alignment than actually writing. Because I occasionally get to read beautifully typeset documents—and terrible ones most of the time—I can tell the difference and appreciate subtle details such as appropriate hyphenation, justification, kerning, leading, and the use of ligatures.

I write a lot of memos for different audiences, including technical, executive, internal, customer proposals, design analysis, and more. Over the years, I have used various tools to write these memos, such as YODL [1], TEX[2], LATEX[3], DocBook [4], DITA [5], AsciiDoc [6], AsciiDoctor [7], and of course, Markdown[1]. After decades of peregrination through different toolchains and editors, including high-end XML editors, nothing beats `vi` (or `emacs` if that's your religion) for fast editing, `Markdown` for ease of use, and LATEX for perfect typesetting.

While the awesome `pandoc` is the easiest way to convert `Markdown` to a properly typeset `PDF`, the default output, though better than that of many other Markdown editors, is great for occasional memos but doesn't look professional. I've been trying to write a nice `pandoc` template myself without much success, but I recently came across a beautiful one: Eisvogel.[2]

> This document could end right here: write Markdown and use 'pandoc' with the 'Eisvogel' template to produce beautiful documents. Period.

However, I believe I can contribute a small addition to the toolchain that will enhance your writing experience and content reuse.



[1] Disregard the bibliographic references and bookmarks like this; I'm adding them just to show how they look.

[2] https://github.com/Wandmalfarbe/pandoc-latex-template

Required tools

TexLive

TEXLive is a comprehensive distribution of the TEX typesetting system, offering a wide range of TEX-related software, packages, fonts, and utilities. Maintained by the TEX Users Group, it supports multiple operating systems, including Linux, macOS, and Windows. The live distribution can be found at: https://www.tug.org/texlive/

Pandoc

Pandoc is a powerful document converter that supports over a dozen input formats and more than thirty output formats, including Markdown, HTML, LATEX, PDF, and Word, making it a perfect fit for our memo toolchain and can be found at https://pandoc.org/

Mermaid-filter

Mermaid is a versatile tool for creating diagrams and charts from simple text descriptions. It generates flowcharts, sequence diagrams, Gantt charts, and more using an easy-to-learn syntax. Popular in technical documentation and project management, Mermaid integrates seamlessly with various platforms, enhancing documents with clear and visually appealing diagrams. We need the `mermaid-filter` package that gets called from `pandoc`, and can be installed with the command:

```
npm i -g mermaid-filter
```

M4

The `m4` macro processor, designed by Brian Kernighan and Dennis Ritchie and originally developed as part of the Unix operating system in the 1970s, is not technically required in this toolchain but is a perfect fit for splitting large documents into smaller pieces, reusing content snippets, and providing standard constants such as the Git revision of the document, date, time, filename, and more. Key features of `m4` include its ability to define macros, perform arithmetic operations, manipulate text with ease, and call the operating system[3]. It also supports file inclusion, conditional statements, and looping constructs.

As an example, to show the revision of the document, you can insert `__REVISION__`, which will be expanded in this case to ef22f21*. The same goes for other variables such as `__DATE__`, `__TIME__`, `__FILENAME__`, `__BRANCH__` and others that you may want to add to the `memo` script. If you're

[3] This poses a security risk when using third-party Markdown sources.

on Unix, `m4` is probably installed, verify that with the command `type m4`. On Windows, you can install it using Cygwin, or the Windows Subsystem for Linux (WSL).

Pandoc template

The Eisvogel template, located at https://github.com/Wandmalfarbe/pandoc-latex-template, has a fantastic set of examples and provides detailed instructions on how to install it. On macOS, with `pandoc` installed via `brew`, the location is `$HOME/.local/share/pandoc/templates` under the name `custom_eisvogel.latex`. It can, of course, be the pristine `eisvogel.latex` script. I suggest calling it `custom_eisvogel.latex` in case you want to add some additional LaTeX commands or configuration.

In that `pandoc/templates` directory, you may also want to add `ieee-with-url.csl` from https://github.com/citation-style-language/styles/blob/master/ieee-with-url.csl or another citation style of your choice.

The memo script

The memo script can be located at https://github.com/ar/memo/blob/main/bin/memo and has the following options:

```
Usage: memo [options] file.md [single|double]

Options:
  --info          Verbose logging to <file>_processing.log instead of /dev/null.
  --tex           Force TeX intermediate even if no glossary/nomenclature.

  --glossary     Build LaTeX glossaries (makeglossaries + two extra XeLaTeX runs).
  --nomenclature  Build LaTeX nomenclature (makeindex  + two extra XeLaTeX runs).

  --open          Automatically open the generated PDF (macOS only).

  --keep          Keep all temporary / intermediate files (no cleanup at exit).
  --clean        Delete temporary / intermediate files and exit without building.

  -DNAME=VALUE    m4 define passed through to preprocessing (repeatable).
```

If called without options, it converts the Markdown document directly to PDF. For documents that include glossary or nomenclature entries—both of which require multiple LaTeX passes—the `--tex` switch can be used to generate an intermediate `.tex` file. This can then be processed manually, optionally using the `--glossary` and `--nomenclature` switches if those features are needed.

Frontmatter

This is a sample frontmatter for this document:

```yaml
---
title: A Memo About Memos
subject: Memo Typesetting
author: Alejandro Revilla (\@apr)
date: 2024-07-24
keywords: [Typography]
titlepage: true
# titlepage-logo: jpos.jpg
lang: "en"
toc: true
toc-own-page: true
footer-center: jPOS.org (ef22f21*)
titlepage-rule-color: "360049"
titlepage-text-color: "FFFFFF"
titlepage-rule-color: "360049"
titlepage-rule-height: 0
titlepage-background: "backgrounds/background.pdf"
footnotes-pretty: true
header-includes:
  - \usepackage{multicol}
  - \usepackage{mdframed}
  - \usepackage{graphicx}
  - \usepackage{xcolor}
  - \usepackage{pgfplots}
  - \usepackage{amsmath}
  - \pgfplotsset{compat=1.17}
  - \usepackage{longtable}


---
```

> Please note that the front matter has been included in the document. Refer to memomemo.md for details.

Note that some of the added packages are used in my custom template, just as a PoC of different features. You can safely remove that `header-includes` section.

Bibliography

You can add `.bib` file, with the same name as your main `.md` file with the following format:

```bibtex
@article{kubat1997yodl,
  title = {YODL or Yet Oneother Document Language},
  author = {Kubat, Karel},
  journal = {Linux Journal},
```

```
  year = {1997},
  month = {November},
  number = {2089},
  url = {https://www.linuxjournal.com/article/2089}
}

@manual{docbook,
  title = {DocBook 5.0: The Transition Guide},
  author = {{OASIS DocBook Technical Committee}},
  year = {2010},
  note = {Available online at \url{https://www.docbook.org/tdg5/en/html
     /docbook.html}},
  url = {https://www.docbook.org/tdg5/en/html/docbook.html}
}

@manual{dita,
  title = {DITA 1.3 Specification},
  author = {{OASIS DITA Technical Committee}},
  year = {2015},
  note = {Available online at \url{https://docs.oasis-open.org/dita/
     dita/v1.3/os/}},
  url = {https://docs.oasis-open.org/dita/dita/v1.3/os/}
}
```

Revision History

Date	Message	Author	Hash
2024-11-01	Generate PDF	A.Revilla	ef22f21
2024-11-01	ignore temp *_preprocessed.md file	A.Revilla	ddaa199
2024-11-01	add revision history and better handling of dirty repos	A.Revilla	100cdf4
2024-08-29	Add link to source	A.Revilla	bcec67a
2024-08-29	Add PDF output	A.Revilla	7fb65ab
2024-08-29	A memo about memos	A.Revilla	748a83f
2024-08-29	Initial commit	A.Revilla	afed625

References

[1] K. Kubat, “YODL or yet oneother document language,” *Linux Journal*, no. 2089, 1997, Available: https://www.linuxjournal.com/article/2089

[2] D. E. Knuth, *The TeXbook*. in Computers & typesetting, volume a. Reading, Massachusetts: Addison-Wesley, 1984.

[3] L. Lamport, *LaTeX: A document preparation system*, 2nd ed. Reading, Massachusetts: Addison-Wesley, 1994.

[4] OASIS DocBook Technical Committee, *DocBook 5.0: The transition guide*. 2010. Available: https://www.docbook.org/tdg5/en/html/docbook.html

[5] OASIS DITA Technical Committee, *DITA 1.3 specification*. 2015. Available: https://docs.oasis-open.org/dita/dita/v1.3/os/

[6] E. Stuart, *AsciiDoc user guide*. 2002. Available: http://www.methods.co.nz/asciidoc/

[7] D. Allen, *Asciidoctor documentation*. 2021. Available: https://asciidoctor.org/docs/