

LaTeX-based Static Website

Kyler Cain

2026-07-24

Abstract

This site uses a small static publishing system in which every post is written as a standalone $\text{\LaTeX}$ document and compiled to both a PDF and HTML version.

1 Objectives

I wanted to be able to write articles in a familiar format and publish them both to the web and as a PDF (for printing, sharing, or offline reading) without maintaining having to think about both formats separately. I also wanted to keep the site implementation small and simple without a database or application server.

This site is intended to be a personal blog and notes site, not a general-purpose static-site generator, so I didn't have to think about too many use cases.

2 Challenges

My main difficulty is that $\text{\LaTeX}$ was never intended to produce html output, so I took a page from arxiv.org[1] and used LaTeXXML[2] to handle the conversion.

After getting the basic conversion working, I also wanted to add a tagging system, a site index, and some basic navigation. This was fairly easy using Jinja[3] templates and a small Python 3[4] script to read the $\text{\LaTeX}$ sources and generate the additional HTML pages.

3 Implementation

Each post is a standalone document whose file name is also the URL slug to the post directory (where I also place the generated PDF). The source files contain a title, author, date, and abstract that I extract for metadata. The tags are declared using a comment like `% tags: webdev`.

The builder normalizes each tag into a URL-safe slug and generates a page for each post and tag. It also generates an index page listing all posts in reverse chronological order.

The toolchain is packaged in Docker[5], and Makefile[6] targets handle the build and deployment. The hosting is done using GitHub Pages, which serves the generated static files directly from CI/CD build artifacts.

4 Results

I have a working static site that is easy to maintain and should be fairly easy to extend in the future. Adding posts doesn't really require any knowledge of the publishing system and building/deploying is entirely automated. Hosting via GitHub Pages is free and simple and the site is fast and reliable thanks to the simplicity of the static files.

5 Future Work

The biggest shortcoming of the current system is that it requires static artifacts for each post, which means I can't make it build dependencies of posts (such as figures or output from other scripts).

There is also no paging so you would load the entire index or all of the posts for a tag. At the scale I am planning to use this site, this isn't a problem (especially since I'm not incurring any kind of hosting cost), but it may make the site less usable as the number of posts grows.

References

- [1] arXiv.org, “Accessible HTML,” https://arxiv.org/help/web_accessibility
- [2] Bruce R. Miller, “LaTeXML: A LaTeX to XML/HTML/MathML Converter,” <https://math.nist.gov/~BMiller/LaTeXML/>
- [3] Pallets, “Jinja Documentation,” <https://jinja.palletsprojects.com/en/stable/>
- [4] Python Software Foundation, “Python 3 Documentation,” <https://docs.python.org/3/>
- [5] Docker, Inc., “Docker,” <https://docker.com/>
- [6] Free Software Foundation, “GNU Make Manual,” <https://www.gnu.org/software/make/manual/make.html>