The math engine, in your browser

Every formula on this site is static SVG, produced at build time by a pure-Swift engine. No MathJax, no web font, no client-side JavaScript. The server only hands over files.

But the engine is just Swift, and Swift compiles to WebAssembly. So the exact same parser, box-and-glue layout, and glyph-outline renderer that runs during the build can also run in your browser.

Try it

Open the live playground

Type a TeX formula and it renders to SVG, live, with no server doing the work. It is the same code path as the build, running in a WebAssembly sandbox in your tab.

There is a companion **PDF playground** too: the MarkdownPDF engine compiled to WebAssembly, turning Markdown into a real PDF entirely in your browser. Same idea, a different Swift engine.

Why this is a footnote, not the feature

The playground downloads about 18 MB of gzipped WebAssembly: the Swift runtime, Foundation, and the engine. That is heavy, and it is exactly the point. You would never ship that to readers. This site renders math to a few kilobytes of static SVG at build time precisely so the browser never loads an engine. The playground is the opposite choice, on purpose, so you can poke the engine directly.

What actually runs in the sandbox

The same pipeline as the build, with nothing stripped out:

1. parse the TeX into a typed tree
2. lay it out with a box-and-glue engine driven by the font's OpenType `MATH` table
3. pull glyph outlines from the font with a hand-written CFF Type2 charstring reader
4. emit a self-contained `<svg>` of `<path>` outlines and vector rules

The engine is the standalone MathTypeset package, shared with the MarkdownPDF generator, so one Swift codebase covers build-time web, in-browser web, and print. The WebAssembly build is served from a branch so the binary never lands in the main repository.