
Demo of the Pandoc Typst Template

Just another Pandoc template

Andreas Burri

01.11.2025

Demo of the Pandoc Typst Template

Until now, trying to make a beautiful, consistent document from Markdown using Pandoc has often meant spending hours fiddling with YAML headers, LaTeX templates, and obscure configuration options.

By default, Pandoc gives you a ton of power — you can output to PDF, HTML, DOCX, and more — but the default styling can be... let's say, “utilitarian.” If you want your document to look professionally typeset, you usually end up writing your own LaTeX template or resorting to manual tweaking.

People often ask us things like:

Why does my PDF look so plain? How do I make Pandoc output something that looks designed?

We hear you. You probably don't want to dive into a full LaTeX preamble just to get decent margins or a better heading style. And you definitely don't want to use the default Times New Roman look. What you really want is a document that looks *awesome*, not awful.

That's exactly what this Typst template aims to do — give you what you *actually* want: a clean, elegant design for Markdown-to-PDF conversion, without touching a single LaTeX file.

Just tell Pandoc to use this template, and you'll instantly get a well-balanced, beautiful document:

```
pandoc input.md -o output.pdf --template=typst.template
```

For more information about how to use this template and what it supports, see the documentation.

What to expect from here on out

What follows is a sample document meant to showcase how various Markdown elements render using this Typst template.

You'll see examples of **bold text**, *italics*, lists, code blocks, block quotes, tables, and more.

The goal is to make sure everything looks good out of the box — because when you're writing, you should be focusing on *content*, not kerning.

Here's a list of reasons we made this demo:

1. To test every typographic element.

2. To make sure spacing and sizing feel natural.
3. Because lists with three items always feel more satisfying.

Now, let's move on to another section.

Typography should just work

If we've done our job, this heading looks nicely balanced with the text that follows it.

Someone once said:

Good typography is invisible — you only notice it when it's bad.

Images should also look good by default:



Abbildung 1: Contrary to popular belief, Lorem Ipsum is not simply random text.

Now here's an unordered list:

- First item in the list.
- Second item, short and sweet.

- Third one, because three just feels right.

And that's the end of this section.

What if we stack headings?

This should look fine too.

Sometimes headings appear directly after one another — this template adjusts the spacing so they look intentional, not awkward.

When a heading comes after a paragraph ...

You should see a bit more breathing room above it.

Let's also test how complex lists look.

- **Heading inside a list item**

This is a list item with its own heading and a couple of paragraphs.

Getting this spacing right in a typeset document is surprisingly tricky, but we've tuned it carefully.

- **Here's another item**

A list wouldn't be much of a list with only one entry.

Notice how the paragraph spacing and indentation feel natural.

- **And finally, a third**

Mostly here for visual balance — but hey, it's nice to look at.

After this, it's good practice to have a closing paragraph. It helps the rhythm of the page.

Code should look great too

Pandoc can render fenced code blocks in many languages, and this Typst template styles them cleanly by default.

Here's an example configuration file:

```
from: markdown
to: pdf
template: typst-template.typ
variables:
```

```
color: "blue"
fontsize: 11pt
```

Hopefully that looks as good as it reads.

Nested lists

Nested lists are hard to make look good, but we've tried our best.

1. Try not to overdo it.

- Deep nesting makes things hard to read.
- Flat is almost always better.

2. But we support it anyway.

- Two levels deep looks okay.
- Three? You're on your own.

And now, back to a normal paragraph.

Description lists

These are useful for FAQs, glossaries, or term explanations — and yes, they're styled too.

What is Typst? A modern, fast typesetting system designed to make writing beautiful documents easy. What is Pandoc? The universal document converter — it turns your Markdown into just about anything. Why combine them? Because together, they let you go from plain text to publication-ready PDFs with minimal effort.

Other elements worth checking

We also have link styles — like this one to the Typst website — that blend neatly into the text.

Tables are supported as well:

Tabelle 1: Demo Table

Language	Purpose	Example Command
Markdown	Input format	<code>pandoc input.md</code>
Typst	Output format	<code>--template=typst-template.typ</code>
PDF	Final document	<code>-o output.pdf</code>

Inline code looks like this: `#let title = "Demo"`, and we even handle code inside headings.

Sometimes I even put code in headings

Probably not best practice, but hey — it's good to know it works.

We haven't used an h4 yet

Now we have. You probably don't need to go deeper — most documents only need a few heading levels anyway.

Let's finish with a paragraph that's long enough to show line wrapping, spacing, and rhythm. A good document template shouldn't just look good — it should make your writing feel comfortable to read. If you've made it this far, congratulations — your Typst template is ready for real work.