

[About](#) [Projects](#) [Talks](#) [Blog](#) [Links](#)

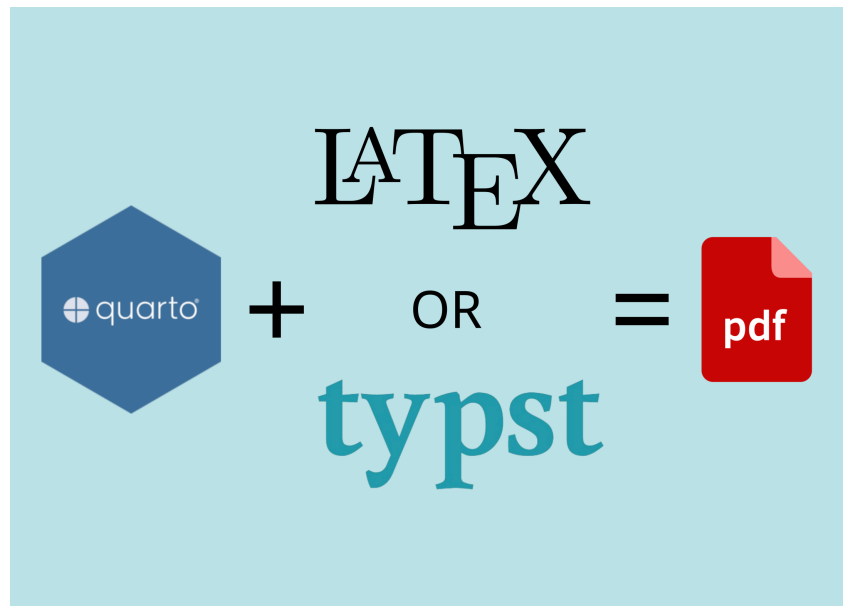
Making Pretty PDFs with Typst (and Quarto)

With the latest 1.4 release of Quarto, it's now possible to create PDF documents with Quarto using Typst. How does it compare to LaTeX, and is it actually easier to learn and use?

January 31, 2024

! Undefined control sequence - an error message you are probably unfortunate enough to be familiar with if you've ever tried to make a PDF using LaTeX. Until recently, if you've been creating reproducible PDFs using R Markdown or [Quarto](#), then your PDF documents have been most likely been created using LaTeX in the background. If you watch which files are generated when you render your document, you'll see that LaTeX files (.tex) are generated.

LaTeX is excellent for including mathematical notation in your documents, and gives users a lot of control over the formatting of documents. However, it can also be a steep learning curve if you want to implement more complex formatting, and the error messages are not always the most informative. Some people might ask, do you need LaTeX to make PDFs? And the answer is no. There are several other ways to create PDFs, including Typst.



What is Typst?

[Typst](#) is a *markup-based typesetting system that is designed to be as powerful as LaTeX while being much easier to learn and use* which has been around since late December 2022. In even more exciting news, with the [latest release of Quarto](#), it's now possible to create PDF documents with Quarto using Typst. To use Typst with Quarto, set the format in the YAML of your Quarto document:

Copy

```
1 ---
2 title: "Hello Typst!"
3 format: typst
4 ---
```

There are also some existing Quarto extensions which provide [custom Typst formats for Quarto](#) that you can use to add styling to your documents. So how easy is it create custom formats with Typst?

Custom formatting with Typst

Almost a year ago, I created a Quarto extension to add LaTeX styling to PDF outputs in order to make slightly nicer looking PDFs, with an [accompanying blog post](#) to explain the process. So it's only natural to come back a year later and try to recreate the same styling using Typst instead of LaTeX...

The Typst website provides some [guidance on creating templates](#), but as a newbie to Typst, I found it a little bit hard to follow. I started by looking at the [Department News Quarto Typst extension](#) because the PDF output looked somewhat similar to what I was trying to achieve - a coloured box in the background, an image in a specific place, and some formatted text. There are two files here we need:

- **typst-show.typ**: a file that calls the template. It maps Pandoc metadata to template function arguments.
- **typst-template.typ**: the core template file. It defines the styling of the document.

I didn't have to make too many changes to the `typst-show.typ` file - mapping to pandoc arguments is likely to be reasonably consistent across different style extensions. For example, the title will always map to the title, and the author will always map to the author.

The `typst-template.typ` is where it gets a little bit more complicated, as I tried to rewrite LaTeX styling with Typst. So I did what anyone would do - started deleting and editing lines of code... Then the error messages come along because you've broken it so you undo everything you just did. Eventually, my trial and error approach seemed to recreate something that worked (or at least the document rendered). I also used the [Typst documentation](#) to add in a few new lines.

Compare the two below where we set the page margins and position a light purple coloured rectangle on the right hand side of the page:

- [Typst](#)
- [LaTeX](#)

Copy

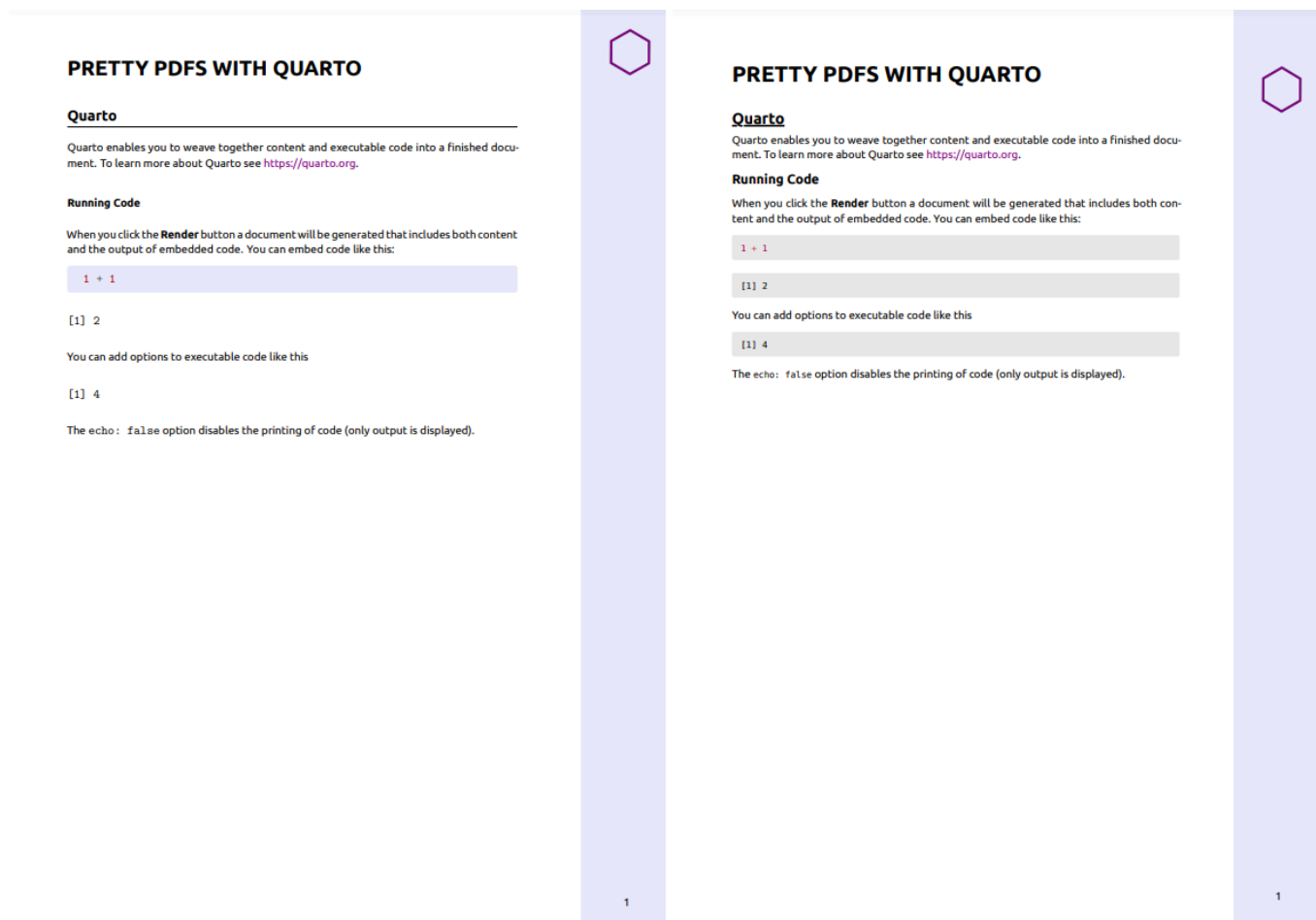
```
1 // Configure pages.
2 set page(
3   margin: (left: 2cm, right: 1.5cm, top: 2cm, bottom: 2cm),
4   background: place(right + top, rect(
5     fill: rgb("#E6E6FA"),
6     height: 100%,
7     width: 3cm,
8   ))
9 )
```

Copy

```
1 \usepackage{geometry}
2 \usepackage{eso-pic}
3 \usepackage{xcolor}
4 \geometry{a4paper, total={170mm,257mm}, left=20mm, top=20mm, bottom=20mm, right=50mm}
5 \definecolor{light}{HTML}{E6E6FA}
6 \AddToShipoutPicture{%
7   \AtPageLowerLeft{%
8     \put(\LenToUnit{\dimexpr\paperwidth-3cm},0){%
9       \color{light}\rule{3cm}{\LenToUnit\paperheight}%
10      }%
11   }%
12 }
```

I think you'll agree that the Typst code is much easier to read and understand what's going on. The learning curve for Typst has been much less steep compared to LaTeX. The outputs look fairly similar, though not

quite identical.



Similar PDF styling implemented with LaTeX (left) and Typst (right)

I added the Typst styling as an additional format contributed by the PrettyPDF LaTeX extension I created a year ago, instead of creating a new separate extension. You can view the updated PrettyPDF extension on [GitHub](#).

If you want to try out the PrettyPDF extension for yourself, install it using:

Copy

```
1 quarto use template nrennie/PrettyPDF
```

Note: the default template still uses LaTeX PDF, so you'll need to update the YAML in the .qmd file to format: PrettyPDF-typst, and ensure you have Quarto 1.4 installed if you want to use the Typst version.

Typst vs LaTeX

Let's see how Typst compares to LaTeX when it comes to making PDF documents in Quarto.

What do I like more about Typst?

- It's fast. Much faster than LaTeX. Typst is written in [Rust](#), and uses incremental compilation giving faster compile times. Even with simple single page documents, it's noticeably quicker than LaTeX.
- The error messages are more informative. When I was debugging error messages in the style files, it was easier to find the source of error.
- No extra installation. To create PDFs with LaTeX using Quarto, you also need to install a recent distribution of TeX. After installing Quarto 1.4, Typst was all set up and ready to go!

What do I like more about LaTeX?

- Although Typst does provide functionality for mathematical formulae typesetting, LaTeX still feels easier to use for mathematical notation at this point. Perhaps this is simply because I'm a lot more familiar with LaTeX, or because Typst is still relatively young, but LaTeX still has the edge here.
- Sizing of images is a little bit odd, and I had to manually specify image widths. This is a known problem that Typst, pandoc and Quarto are actively working to fix so hopefully this will be fixed soon!

So, will I be using Typst if I'm creating PDF documents with Quarto? Yes, (unless I have to write a lot of equations...)

Further resources

If you're thinking about trying out Typst and you're already a LaTeX user, you might find the [official Typst Guide for LaTeX Users](#) useful.

The [Quarto documentation on how to use Typst](#) to create PDFs is a nice starting point - just make sure you've downloaded and installed at least version 1.4 of Quarto first. There are some existing extensions which provide [custom Typst formats for Quarto](#) that you can use directly, or use as inspiration to create your own custom format!

Now you're ready to go and create some beautiful looking PDFs with Quarto - either using LaTeX or Typst!



Image: [giphy.com](#)

For attribution, please cite this work as:

Making Pretty PDFs with Typst (and Quarto).

Nicola Rennie. January 31, 2024.

nrennie.rbind.io/blog/making-pdf-with-quarto-typst-latex

Licence: creativecommons.org/licenses/by/4.0

5 Comments - powered by [utteranc.es](#)

rainer-rq-koelle commented 3 months ago

Hi Nicola, thanks that was a useful resource. I am currently looking into using Typst/Quarto for reporting at work. One of the style "requirements" is to have a line separating the header and footer. Would you have an idea how this is done/can be done? I somehow fail to find the answer in the typst documentation.



1

nrennie commented 3 months ago

Owner

I don't think there's a built-in option to do it, but as long as you know the height of the header/footer, you can use the line function to draw a line: <https://typst.app/docs/reference/visualize/line/>

butames commented 2 months ago

Thank you for introducing me to this resource. I used LaTeX sometime ago when I was in graduate school. After doing my data analysis, writing out my work using LaTeX felt like an additional chore. I will look into Typst, it certainly looks more accessible. Thank you again.



1

castedo commented 2 weeks ago

A third approach is to not use LaTeX nor Typst at all and instead output HTML that can be rendered into a PDF using WeasyPrint. I can email you more details if you are interested. Quarto 1.4 by default outputs HTML with CSS that does not work ([quarto-dev/quarto-cli#9510](#)), but custom CSS that is print-responsive could work similar to what is described here: [quarto-dev/quarto-cli#9505](#)

An interesting blog post might be contrasting the pros and cons of generating PDFs from print-responsive HTML vs using LaTeX/Typst.

nrennie commented 2 weeks ago

Owner

Thanks for the suggestion - I'm aware that you can convert from HTML to PDF (this post was focusing on PDF as a primary output rather than secondary). Personally, I normally use pagedjs rather than WeasyPrint (paged.js works out of the box with R Markdown via pagedown so I can use the code I already have from that).

See also this GitHub issue: [quarto-dev/quarto-cli#6401](#)



1

Write

Preview

Sign in to comment

 Styling with Markdown is supported

Sign in with GitHub

[← Four ways to streamline your R workflows](#) [Answering some 'Forecasting with GAMs in R' questions](#) [→](#)

© 2024 Nicola Rennie. Made with [Hugo Apéro](#).