typst

COMPOSE DOCUMENTS FASTER.

A quick start guide for users switching from LaTeX or Microsoft Word.

Welcome to typst !

Typst is a powerful, flexible, and friendly typesetting tool for the modern era. Using the built in functions and layouts, the ecosystem of external packages, and the powerful scripting tools, Typst can be used for any documents of your choosing, including research papers, theses, essays, reports, memos, and more.

Why Typst?

For anyone who has used LaTeX in the past, after learning some of the basics, you'll feel right at home with Typst. However, Typst is faster to write, easier to read and debug, renders nearly instantly, and allows for real time collaboration with peers.

Typst also benefits people who currently typeset documents using conventional visual editors like Microsoft Word or Google Docs. The powerful template system allows you focus on writing and lets Typst handle the formatting as you type. And like LaTeX, Typst outputs higher quality documents with better justification and layouts.

Who is this guide for?

Typst is an incredibly powerful piece of software with a rich featureset, limitless extensibility, and a plethora usecases. **However, the scope of this guide will be limited to basic features like simple typesetting and math/equations.** Someone who could find this document useful is a student or researcher who is frustrated with LaTeX and is looking for a better alternative for writing high quality documents.

Some of the topics this guide will cover include:

- Environment Setup
- Writing Basic Documents
- Math Mode
- Basic Scripting
- Third Party Packages
- Templates

Precautions and Safety Warning

Typst is still pre-release software. Not everything is working and there may be core functionality still missing from the software. Although I have had few issues with Typst, your mileage may vary. If you encounter issues or want to request new features, feel free to contact the developers on Discord or GitHub (links at the end of the document).

If you use the online version of Typst, you will be sending possibly personally identifiable information to the maintainers of Typst. Typst uses opt-out trackers on its website. Typst uses Microsoft Azure for content distribution and your information may be logged by Microsoft. Typst is compliant with the EU's GDPR and California's CCPA data privacy laws.

If you use the offline version of Typst, you will need to download possible unstable software onto your computer. By using an offline version of Typst, you assume all risks.

0 Environment Setup

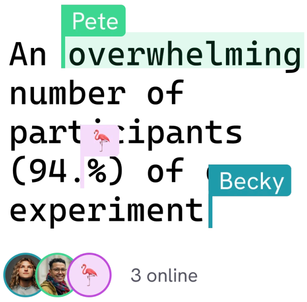
There are a few ways that you can get started with Typst.

Web Editor

For a hassle free experience, use the online editor!

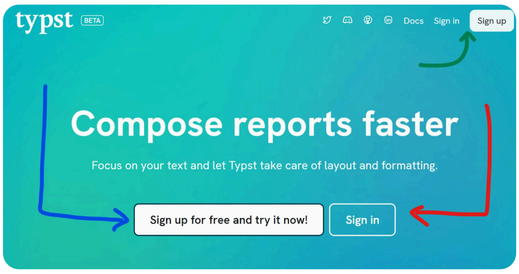
Pros of the Web Editor

- No hassle setup
- Automatic updates
- Live updating and recompiling
- Real time collaboration


An overwhelming number of participants (94.%) of experiment

Getting Started

Getting started on the web editor is as easy as going to typst.app and signing up for an account! Once you're there, you can work on projects, collaborate with teams, and create whatever you want with Typst!


Compose reports faster

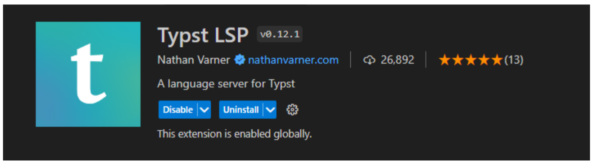
Focus on your text and let Typst take care of layout and formatting.

Visual Studio Code + Typst LSP

Using Vscode with the Typst LSP is as easy as the online editor with more flexibility!

Pros of VSCode + Typst LSP

- Quick Setup
- Familiar Environment
- Fast, offline compilation
- Use local resources like fonts

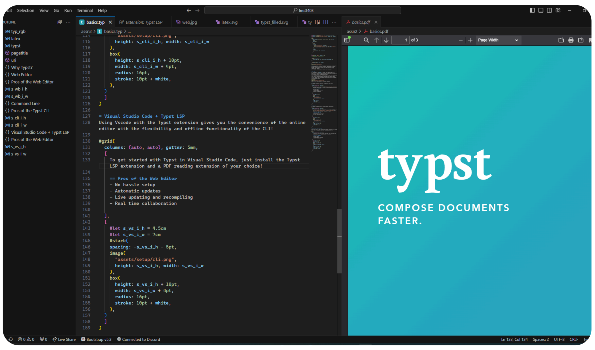

Typst LSP v0.12.1

Nathan Varner nathanvarner.com | 26,892 | ★★★★★(13)

A language server for Typst

This extension is enabled globally.

To get started with Typst in Visual Studio Code, just install the "Typst LSP" extension and a PDF extension of your choice. Then create a `.typ` file and start writing!


```
      "assets/setup/cli.png",
      height: s_cli_i_h, width: s_cli_i_w
    ),
    box(
      height: s_cli_i_h + 10pt,
      width: s_cli_i_w + 4pt,
      radius: 16pt,
      stroke: 10pt + white,
    ),
  )
  ]
)

= Visual Studio Code + Typst LSP
Using Vscode with the Typst extension gives you the convenience of the online editor with the flexibility and offline functionality of the CLI!

#grid(
  columns: (auto, auto), gutter: 5mm,
  [
    To get started with Typst in Visual Studio Code, just install the Typst LSP extension and a PDF reading extension of your choice!

    == Pros of the Web Editor
    - No hassle setup
    - Automatic updates
    - Live updating and recompiling
    - Real time collaboration

  ],
  [
    #let s_vs_i_h = 4.5cm
    #let s_vs_i_w = 7cm
    #stack(
    spacing: -s_vs_i_h - 5pt,
    image(
      "assets/setup/cli.png",
      height: s_vs_i_h, width: s_vs_i_w
    ),
    box(
      height: s_vs_i_h + 10pt,
      width: s_vs_i_w + 4pt,
      radius: 16pt,
      stroke: 10pt + white,
    ),
  )
  ]
)
```

Other Options

There are other options for setting up a development environment for typst as well!

Quarto, an open source scientific and technical publishing system, now supports typst as an input format.

Additionally, the Typst CLI (found at github.com/typst/typst) allows you to use typst with any text editor of your choosing!

1 The Basics

Basic text entry in Typst is just like typing in Word. If you type text into a blank typst file, it will render that text onto a blank document (like a pdf).

your_doc.typ

```
Hello! This is a blank typst document
with some text in it.
```

OUTPUT

Hello! This is a blank typst document with some text in it.

Text Formatting

For formatting, Typst uses formatting similar to **Markdown**, a lightweight markup language. If you've used markdown or other markdown like editors (usually found in messaging applications), you'll feel right at home with Typst. If not, there are a few simple rules for formatting in Typst.

The main ones to remember are:

- Use *s to make text **Bold**
- Use _ to make text *Oblique*
- Use = to create headers of different sizes!
- - to make Bulleted lists
- + to make Numbered Lists
- Wrap text in `s to make a code block!
  - Wrap text with 3 ```s and write the name of a language right after for syntax highlighting! (eg. ` ```rust /* rust code here */ ``` `)

your_doc.typ

```
Wrap text with \* to make it *Bold* \
Wrap text with \_ to make it _Oblique_\
Wrap with \` to make it `a code block`

= Large Header
// add '='s for smaller headers
== Smaller Header

- Bulleted List Item
+ Numbered List Item

Use the '\\' character for line breaks
```

OUTPUT

Wrap text with * to make it **Bold**
Wrap text with _ to make it *Oblique*
Wrap with ` to make it `a code block`

Large Header

Smaller Header

- Bulleted List Item

1. Numbered List Item

Use the '\' character for line breaks

2 Math Input

One of the major draws of Typst is it's strong but simple math input functionality. Users coming from Word may be familiar with how slow inputting math can be, and people switching from LaTeX are well aware of how messy TeX code can get with complex equations.

In Typst, there are two types of math blocks. Inline and Blocks. Both are created by wrapping text with `$`'s. However, blocks have whitespace between the `$` and the start of the math code.

your_doc.typ

```
There's *inline math*:
$nabla dot bold(E) = rho/epsilon_0$ \

$ bold("Block Math") "is centered."\
nabla times bold(B) = mu_0 (bold(J) +
epsilon_0 frac(diff bold(E), diff t)) $
```

OUTPUT

There's **inline math**: $\nabla \cdot \boldsymbol{E} = \frac{\rho}{\varepsilon_0}$

$$\textbf{Block Math} \text{ is centered.}$$

$$\nabla \times \boldsymbol{B} = \mu_0 \left(\boldsymbol{J} + \varepsilon_0 \frac{\partial \boldsymbol{E}}{\partial t} \right)$$

Additional Syntax

Compared to Word and LaTeX, writing math code in Typst is a little different, but after learning some of the basic syntax, you'll be on your way to typesetting math in typst like a pro!

Sub / Superscripts

`$x^u_d$` $\to \quad x_d^u$

Fractions

`$1/x$` $\to \frac{1}{x}$

Greek Symbols

`$sigma$ $Sigma$` $\to \quad \sigma \; \Sigma$

Some Shorthands

`$ => != >=$` $\to \; \Rightarrow \; \neq \; \geq$

Special Characters (See Docs for more)

`$arrow.l.curve integral.cont$` $\to \quad \hookleftarrow \quad \oint$

Text in Math

`$x_0 "Text!" x_1$` $\to x_0 \text{ Text! } x_1$

Math Functions

`$floor(x), frac(X, z*y), sqrt(9)$` $\to \lfloor x \rfloor, \frac{X}{z * y}, \sqrt{9}$

Aligning

Use the `&` character to align math on different lines! This is good for if you're showing many different steps on different lines.

your_doc.typ

```
Use & to align lines in block math:
$
nabla times D &= & &rho_v \
nabla times H &= &(diff D)/(diff t) + &J
$
```

OUTPUT

Use & to align lines in block math:

$$\begin{aligned} \nabla \times D &= && \rho_v \\ \nabla \times H &= \frac{\partial D}{\partial t} &&+ J \end{aligned}$$

3 Scripting

One of Typst's headlining features is the built in scripting features. Typst's scripting features allow you to automate repetitive tasks like layouting. Typst has lots of built in functions for layouting and visualization. Some of these include `text()` for rendering special text, `align()` for aligning content, `page()` for making special pages in your document, and many more!

Typst also includes functionality specifically for scripting and calculations. Some of these include control flow constructs like loops, conditionals, and others! Typst also has lots of built-in functions like `range()`, `zip()`, and `map()` which behaves like Python's range, zip, and map functions.

One sample usecase for the scripting is working on a tedious mathematical problem. You can write code that both does all the calculations for the problem while also generating code that shows all of your work!

your_doc.typ

```
#let factorial(n) = {
  let curr = 1
  for val in range(1, n+1) {
    let prod = curr * val
    [ Round #val: $curr*val=prod$ \ ]
    curr = prod
  }
}
#factorial(8)
```

OUTPUT

Round 1: $1 * 1 = 1$
Round 2: $1 * 2 = 2$
Round 3: $2 * 3 = 6$
Round 4: $6 * 4 = 24$
Round 5: $24 * 5 = 120$
Round 6: $120 * 6 = 720$
Round 7: $720 * 7 = 5040$
Round 8: $5040 * 8 = 40320$

Learn more about scripting in Typst at typst.app/docs/reference/scripting

❹ Templates

“Templates” include a few different features within Typst.

You can create templates using scripting functions that apply certain styles and layouts automatically. This allows you to focus on writing and lets you make changes to the template that apply across the whole document.

Here’s a basic example of a template!

your_doc.typ

```
// Example from typst docs
#let amazed(term) = box[✨ #term ✨]

You are #amazed[beautiful]!
```

OUTPUT

You are ✨ beautiful ✨!

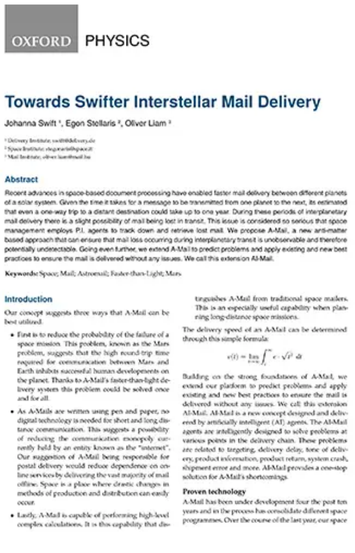
Templates can do as little or as much as you want them to! The only limitations on what you can do are the Typst scripting language and ✨ Your Imagination ✨

Separate Files

You can also create separate files for templates that can define the visual style of your document, how the text flows on the page, different graphic elements on the page, page numbers, and more! One place this would be useful would be if you need to follow a specific format for a document. Examples of this would be if a specific conference needs a paper to include certain visual elements, or if you need to follow an organizational template.

Towards Swifter Interstellar Mail Delivery

IEEE Format



OXFORD PHYSICS

Towards Swifter Interstellar Mail Delivery

Oxford Format

Towards Swifter Interstellar Mail Delivery

MDPI Format

Typst maintains a small repository of provided templates here github.com/typst/templates

Learn more about templates at typst.app/docs/tutorial/making-a-template.

5 Third Party Packages

Typst has a strong ecosystem of 3rd party packages that extend the core functionality of Typst with even more features. You can browse and learn how to install packages at typst.app/docs/packages.

John Gosset also maintains a github repository containing a list of cool and helpful typst tools! Some things on there include chatbots for rendering typst code, typst extensions for other editors, templates for other formats and conferences, resume templates, formatting libraries, and even solutions for leetcode problems written with the typst scripting language! Check out the repo here: github.com/qjcq/awesome-typst

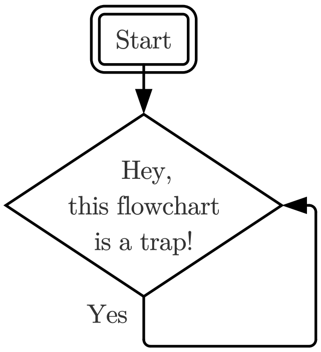
A package that students, engineers, and scientists alike may find useful is **Fletcher**, a package for making charts and diagrams. Here's an example of using fletcher:

your_doc.typ

```
#import "@preview/fletcher:0.4.2" as fletcher: node, edge

// https://xkcd.com/1195/
#import fletcher.shapes: diamond
#set text(font: "Comic Neue", weight: 600)
$
#fletcher.diagram(
  node-stroke: 1pt,
  edge-stroke: 1pt,
  node((0,0), [Start], corner-radius: 2pt, extrude: (0, 3)),
  edge("-|>"),
  node((0,1), align(center)[
    Hey,\ this flowchart\ is a trap!
  ], shape: diamond),
  edge("d,r,u,l", "-|>", [Yes], label-pos: 0.1)
)
$
```

OUTPUT



This example was borrowed from the documentation for fletcher: github.com/Jollywatt/typst-fletcher

WHAT'S NEXT FOR typst ?

Typst is still in beta and is getting new updates as we speak! To learn more about Typst, find more packages, or learn more about the features or Syntax of typst, visit the **Typst Docs** at typst.app/docs!

There's also a community Discord Server and a GitHub page if you want to speak to the developers! For example, if you're facing an issue with Typst, have a question on how to accomplish a task, or want to request a feature, join the open Typst Discord Server or leave an issue on the Typst GitHub page.

Read the docs
typst.app/docs

Join the Discord
discord.gg/2uDybryKPe

Check out the GitHub
github.com/typst

Awesome Typst Projects!
github.com/qjcq/awesome-typst

Source code for this doc
github.com/Liusef/3403-tech-doc

typst
Compose Documents Faster.

References

- Typst GmbH. (2024). Typst. Typst. https://typst.app
- Typst GmbH. (2024). The Typst Tutorial. Typst. https://typst.app/docs/tutorial
- The LaTeX Project. (2024). LaTeX. The LaTeX Project. https://www.latex-project.org/
- Microsoft Corp. (2024). Microsoft 365. Microsoft Corp. https://microsoft365.com/
- Alphabet Inc. (2024). Google Docs. Alphabet Inc. https://docs.google.com
- Microsoft Corp. (2024). Microsoft Azure. Microsoft Corp. https://azure.microsoft.com/
- European Union. (2018). General Data Protection Reg. European Union. https://gdpr.eu/
- State of California. (2018). California Consumer Privacy Act. California Department of Justice, Office of the Attorney General. https://www.oag.ca.gov/privacy/ccpa
- Typst GmbH. (2024). Typst Live Collaboration Graphic. Typst. https://typst.app
- Microsoft Corp. (2024). GitHub. Microsoft Corp. https://github.com/
- Typst GmbH. (2024). Typst Homepage. Typst. https://typst.app/home
- Microsoft Corp. (2024). Visual Studio Code. Microsoft Corp. https://code.visualstudio.com/
- Microsoft Corp. (2024). Microsoft Windows. Microsoft Corp. https://windows.com/
- Nathan Varner. (2024). Typst Language Server Protocol. Nathan Varner. https://github.com/nvarner/typst-lsp
- Microsoft Corp. (2024). Visual Studio Marketplace. Microsoft Corp. https://marketplace.visualstudio.com/items?itemName=nvarner.typst-lsp
- Quarto. (2024). Quarto. Quarto. https://quarto.org/
- Rust Foundation. (2024). Rust Lang. Rust Foundation. https://rust-lang.org/
- Python Software Foundation. (2024). the Python programming language. Python Software Foundation. https://python.org/
- Typst GmbH. (2024). Typst Package Repository. Typst. https://github.com/typst/packages
- John Gosset. (2024). Awesome Typst. John Gosset. https://github.com/qjcg/awesome-typst
- Joseph Wilson. (2024). Fletcher. Jollywatt. https://github.com/Jollywatt/typst-fletcher
- Randall Munroe. (2024). XKCD. Randall Munroe. https://xkcd.com/
- Typst Gmbh. (2024). Typst Templates for IEEE, Oxford, MDPI. Typst. https://typst.app/
- Institute of Electrical and Electronics Engineers. (2024). IEEE. Institute of Electrical and Electronics Engineers. https://www.ieee.org
- Oxford University. (2024). Oxford University Press. Oxford University. https://corp.oup.com
- Multidisciplinary Digital Publishing Institute. (2024). MDPI. Multidisciplinary Digital Publishing Institute. https://mdpi.com/
- Typst GmbH. (2024). Typst Documentation. Typst. https://typst.app/docs
- Typst GmbH. (2024). Typst Privacy Policy. Typst. https://typst.app/privacy/
- Discord Inc. (2024). Discord. Discord Inc. https://discord.com/
- Microsoft Corp. (2021). Cascadia Code. Microsoft Corp.
- Microsoft Corp. (2021). Fluent Emojis. Microsoft Corp.
- Microsoft Corp. (2024). Fluent Design System. Microsoft Corp.
- Danillo De Marco. (2017). The Aganè Font Family. Danillo De Marco.
- Adrian Frutiger. (2015). Avenir Next. Adrian Frutiger.
- Adrian Frutiger. (2015). Neue Frutiger. Adrian Frutiger.
- Libertine Open Fonts Project. (2012). Linux Libertine. Libertine Open Fonts Project.

typst

Compose Documents Faster.