

# Paper Writing in Typst—How To

A subtitle is purely optional

Louisa Lambrecht  
University of Tübingen, Germany  
louisa.lambrecht@uni-tuebingen.de

## ABSTRACT

Start with a structured abstract as a tiny text. That should **not** become your final version of an abstract without alteration.

An abstract (for a short paper like yours) should comprise about 100-150 words. Please, write a minimum of 80 and a maximum of 200 words.

## KEYWORDS

Keywords, To, Increase, Discoverability

## 1 INTRODUCTION

Please find appropriate titles for your sections. *Introduction*, *Conclusion* are clearly present in the same position in every paper. Be creative here.

## 2 LAYOUT AND TYPESETTING

Correct layout and typesetting are essential. You must use the provided Typst template as the basis for your paper. This template defines the formatting, margins, spacing, and fonts, and these must not be altered under any circumstances. Paper submissions for conferences that do not adhere to the layout rules are usually desk-rejected, *i.e.*, they are rejected without even reading it.

The only exception to this rule is the so-called tt font (short for typewriter text). This font is typically used for code snippets, filenames, and terminal output. In Typst, it is accessed using the `backticks` command or with the `raw` environment. You may adjust the appearance of this font—for example, to improve legibility of your code—but please do so thoughtfully.

While the template provides a solid foundation, it does not take care of everything for you. You are responsible for the final look and readability of your paper. Pay close attention to the following issues:

- Hyphenation: Check your document for poor or missing hyphenation. Typst does a good job in most cases, but it is not perfect—especially with non-English or technical terms. Hyphenate manually if needed.
- Line and page breaks: Sometimes Typst may break lines or pages in awkward places such as between a heading and the following paragraph, or in the middle of a code block or figure. These situations often require manual adjustment sometimes by rewriting text to fit appropriate space. Do not use commands like `#pagebreak()` and `#colbreak()` lightly.
- Overflowing elements: Figures, tables, and code listings that extend into the margins or off the page are unacceptable.

Always review your compiled PDF and make sure everything fits cleanly within the page boundaries.

- Code listings: Pay special attention to how your code appears. Avoid breaking listings across pages when possible. If a long snippet must be broken, try to split it at a logical point and use captions or comments to help the reader follow along.

In short, use the template, but do not trust it blindly. A well-typeset paper reflects not only your technical knowledge, but also your care and professionalism. If something looks strange, fix it. If you are not sure how, ask. This is part of the writing process, and it is a skill you will carry forward in all your academic work.

### 2.1 Code Blocks

When creating code listings with Typst, it's important to ensure clarity, readability, and professional formatting. Common issues include missing syntax highlighting or language keywords. Syntax highlighting is usually quite good in Typst. If you need to define your own language, you can do so with `sublime-syntax` file. However, that is a bit tricky, let me know if you need help.

Enable line numbers to support referencing and discussions with the `show` rule in Line 30 of the Typst source file. You can change colors of the syntax highlighting with a `tmTheme` file. There are numerous options in some git repositories, *e.g.* [TextMate-Themes](#). Other options to style your code listings are available via the Typst `show` rules.

Avoid presenting large, uninterrupted blocks of code—split long listings into logical parts or use ellipses to skip unimportant sections. Lastly, include meaningful comments in the code itself to explain non-obvious logic or design choices; this makes the code more accessible.

Always include a clear and descriptive caption using the `caption` field and put the listing into a floating environment (*e.g.*, the `figure` environment). Ensure that code listings are placed near the relevant text. Always reference the listing in your text. A listing that is not referenced will not receive attention from the reader. Listing 1 is now referenced.

If I finished this text now, there would be a lot of white space here. Therefore, I should write some more sentences to fill this space. We sometimes have to adapt our text to create a truly appealing layout.

```
1 WITH RECURSIVE t AS (  
2   SELECT 1 AS x  
3 )  
4 SELECT *  
5 FROM   t  
6 WHERE  x > 0;
```

Listing 1: A code listing.



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Table 1: A table also needs a caption.

	tol = $u_{\text{single}}$			tol = $u_{\text{double}}$		
	<i>mv</i>	Rel. err	Time	<i>mv</i>	Rel. err	Time
trigmv	11034	1.3e-7	3.9	15846	2.7e-11	5.6
trigexpmv	21952	1.3e-7	6.2	31516	2.7e-11	8.8
trigblock	15883	5.2e-8	7.1	32023	1.1e-11	1.4e1
expleja	11180	8.0e-9	4.3	17348	1.5e-11	6.6

2.2 Tables

Tables are a good way of presenting lots of numbers in a paper. In the written form, the reader has enough time to take in all the content. If possible and with the appropriate meaning, plot data.

#figure is the floating environment for tabular data, too. Table captions belong above the table. A Typst show rules (Line 45) takes care of that. Tables should also always be referenced (see Table 1).

Always use **booktabs** style:

- Booktabs only uses horizontal lines, no vertical lines.
- Booktabs has thick horizontal lines in the beginning and end of the table that frame the table vertically (top and bottom rules).
- A thinner midrule separates the header from the table content.
- Additional even thinner c rules (latex lingo) can separate the header further.

2.3 Figures

Figures can contain arbitrary graphics like data plots, diagrams (fletcher, cetz), images, screenshots *etc.* All of these contents should be referenced as “Figure x”. A label will do that automatically in Typst. Figure 1 shows a simple fletcher diagram.

3 LANGUAGE

When writing a scientific paper, it is essential to use language that is clear, precise, and formal. Use the past tense when describing methods and results, as these refer to actions that have already been completed. Present tense is appropriate when discussing established knowledge or explaining the implications of your findings.

In computer science, we prefer the active voice because it is more direct and easier to understand than the passive voice, which can make sentences unnecessarily complex.

Avoid using synonyms for key terms; consistency in terminology helps prevent confusion.

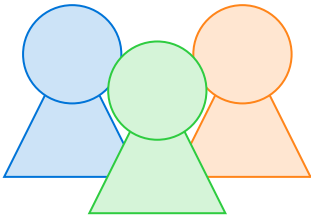


Figure 1: An example caption.

Abbreviations should always be spelled out in full the first time they appear, followed by the abbreviation in parentheses—for example, scanning electron microscope (SEM). After that, you may use the abbreviation alone.

Finally, do not use contractions such as “it’s” or “won’t”; scientific writing requires a formal tone, and these short forms are not appropriate.

4 CITATIONS AND BIBLIOGRAPHY

Proper citation and bibliography management are essential for academic integrity and clarity. In general, always ensure that citations are accurate, consistent, and correspond to the correct sources. In LaTeX and Typst, the recommended way to handle references is by using Bib(TeX) which allows for flexible and standardized formatting. The template will take care of the most important aspects. However, you should never blindly copy BibTeX entries from the internet, especially from sources like Google Scholar, which often contain incomplete, incorrect, or poorly formatted metadata. Instead, always check the official publisher’s website (not arXiv!), which typically provides more reliable BibTeX entries—though these may include excessive fields or information not needed for our bibliography style. It’s good practice to clean and simplify these entries (*e.g.*, trimming unnecessary fields) to maintain a consistent and professional bibliography. Author (all authors!—lastname, firstnames), title, year and venue/conference/journal including volume (keywords in a BibTeX file may be: book-title, publisher, journal, volume) should always be present. Please omit details like doi, pages, exact dates, isbn *etc.*

When citing websites, be cautious and selective—use them only when they are credible, stable, and relevant (*e.g.*, official documentation, white papers, or standards bodies). In BibTeX, websites can be cited using the @misc or @online entry type (depending on the bibliography package), and must include at minimum the title, author or organization, access date, and URL. Never cite vague or transient sources like random blog posts, unless they are from recognized experts or institutions. When a website was *used* or data downloaded rather than actually having citable information read there, it may be sufficient to add a footnote with a URL without creating a full BibTeX entry. [1]

5 AI AND PLAGIARISM

Feel free to use AI-based tools such as Grammarly and DeepL to aid you with writing better English. We strongly encourage that.

Any use of generative AI (ChatGPT, Copilot, Llama, Gemini, Mixtral *etc.*) counts as plagiarism. AIs often invent “facts” or may give direct citations from relevant sources. Direct citations (direct copy of text or translated text) whether marked or unmarked is plagiarism.

Any form of plagiarism will lead to immediate failure of the course.

REFERENCES

[1] Nemo Nobody. 2006. My Secret Article, in *Proceedings of My Secret Conference*, 2006.