List of Notes

Writing in Typst about a long, very scientific topic

Projektarbeit 1 (T3_2000)

As part of the examination:

Bachelor of Science (B. Sc.)

in Computer Science

at the
Baden-Württemberg Cooperative State University Karlsruhe

by

John Doe

Submission Date	31. März 2023
Processing Period	01.01.2023 - 31.03.2023
Matriculation Number, Course	0000000, TINF21B2
Training Company	SAP SE Dietmar-Hopp-Allee 16 69190 Walldorf, Deutschland
Supervisor of the Training Company	Christian Klein
Reviewer of the Duale Hochschule	Heinrich Braun

Declaration of Originality

I hereby declare that I have independently written my Projektarbeit 1 (T3_2000) with the topic:

Writing in Typst about a long, very scientific topic

in accordance with § 5 of the "Studien- und Prüfungsordnung DHBW Technik" dated September 29, 2017, and have not used any sources or aids other than those specified. The work has not been submitted to any other examining authority before and has not been published.

Furthermore, I confirm that the submitted electronic version corresponds to the printed version.

Karlsruhe, 01. January 1980

Signed: Doe, John

Confidentiality Clause

The following thesis contains confidential data from:

SAP SE
Dietmar-Hopp-Allee 16
69190 Walldorf
Deutschland

The content of this thesis must not be made accessible, either in its entirety or in excerpts, to individuals outside the examination process and evaluation procedure, unless there is explicit permission from the partner company.

Abstract

- Deutsch -

Dies ist eine kurze Zusammenfassung, um die Formatierung und den allgemeinen Stil der Vorlage zu zeigen

Es ist möglich, mehrere Abstracts in Unterschiedlichen Sprachen zu haben

Abstract

- English -

This is a short abstract to show the formatting and general style of the template

It is possible to have multiple abstracts in different languages

Table of Contents

List of Abbreviations

***NN* – Neural Network** 7

Table of Figures

Source Code Directory

1 Introduction

Typst is a versatile and powerful tool for creating well-structured and beautifully formatted documents, from technical reports to academic papers. This guide provides a detailed overview of using Typst, highlighting its key features, including basic formatting, embedding code blocks, and utilizing specialized functions for enhanced document design.

2 Basic Formatting

- **Headings and Subheadings:** Create headings by typing = followed by your heading text. Increase the number of `=` signs for subheadings, indicating their level within the document structure.
- **Emphasizing Text:** Use `_underscores_` to *italicize words or phrases*, adding emphasis where needed.
- **Lists:** Generate ordered lists with `+` and unordered lists with `-`. Indent lists for sub-items, creating a hierarchical structure.

Typst Code	Output

`= Main Heading`
`_This_ is a *level 1* heading.`
 	
`== Subheading`
`+ This is an ordered list.`
`+ It can contain multiple items.`
	ii.i. Main Heading
	This is a **level 1** heading.
	ii.i.i. Subheading
	1. This is an ordered list.
	2. It can contain multiple items.

3 Advanced Elements

3.1 Figures

Inserting figures and code blocks into your Typst document enhances its informational depth. When specifying a `caption` for a figure, the template will automatically generate a list of figures, making it easy to navigate your document.

Note When using “ieee” Sorting for bibliography, the sources for figures will be evaluated before the text. To prevent “false sorting”, you can use `#caption_with_source("Text", [@source])` instead. This will display the caption in outlines without source and will the source evaluate at the time the figure is displayed

3.1.1 Image Figures

Typst Code

```
#figure(
  image("assets/SAP-Logo.svg"),
  caption: "SAP Logo"
)
```

Output

Figure 1: SAP Logo

3.1.2 Code Snippets:

This template uses Codly for code snippets. Look at their documentation on how to further customize and control your code blocks.

Besides that the template provides two functions to create code snippet figures that get listed in a source code listing: `codefigure` and `codefigurefile`.

Use `codefigure` to display a code figure from the provided code.

Typst Code

~~~
#codefigure(caption: "My Code")[```rust
fn main() {
  println!("Hello World!");
}
```]
~~~

Output

```
fn main() {
  println!("Hello World!");
}
```

Code 1: My Code

Note You can also provide custom syntax (`.sublime-syntax`) files for code highlighting. The template already includes a syntax file for CDS, so you can use the `cds` language in your code blocks.

Use `codefigurefile` to create a code snippet figure from the content of a file. Note that the provided file is searched relative to `./`.

Typst Code

```
#codefigurefile(
  "assets/example-code.typ",
  caption: "My Code from a file"
)
```

Output

~~~
#codefigure(caption: "My Code")[```rust
fn main() {
  println!("Hello World!");
}
```]
~~~

Code 2: My Code from a file

3.2 Math

The math syntax is a loose interpretation of LaTeX, allowing you to create complex mathematical equations with ease. See the Typst documentation for a detailed overview of the math syntax.

Typst Code

```
$ sum_(k=0)^n k
    &= 1 + ... + n \
    &= (n(n+1)) / 2 $
```

Output

$$\begin{aligned}\sum_{k=0}^{n} k &= 1 + \ldots + n \\ &= \frac{n(n+1)}{2}\end{aligned}$$

3.3 Block Quotes

Typst Code	Output

`#quote(attribution: [Frankling D. Roosevelt @fdr_inaugural_address])[`
`  The only thing we have to fear is fear itself.`
`]`
	The only thing we have to fear is fear itself.
	— Frankling D. Roosevelt [1]

3.4 Notes

This template uses Drafting for notes. Using `margin-note` you can add notes to the margin of your document.

Anywhere in your document!

You can also add inline notes to your document with `inline-note`

Check out their documentation for more advanced use casese.

You might have noticed the notes listing on the first page of this document. This listing reminds you of the notes still present in your document. Once you remove all notes, the listing will disapear.

4 References and Citations

4.1 Local Elements

You can reference local elements like figures, code blocks, and sections using the `ref()` function. You can also use the synax sugar `<ref>` to define and `@<ref>` to reference references.

Typst Code	Output
= Section 1 <section-1> Some important text = Section 2 More important text, just like @section-1	**iv.ii. Important Section** Some important text **iv.iii. Other Section** More important text, just like Section iv.ii

4.4 Code Blocks

If you use the provided `codefigure` function, you can specify a reference name via the `reference` parameter. This allows you to reference the code block later in the document.

Typst Code	Output

#codefigure(caption: "Code w/ Ref", reference: "my-rust-code")[
```rust
fn main() {
  panic!("Hilfe!");
}
```
]
 	
Look at my code in @my-rust-code!
	fn main() {
	  panic!("Hilfe!");
	}
	Code 3: Code w/ Ref
	Look at my code in Code 3!

4.5 Bibliography

Typst supports references to external sources, such as books, articles, and websites. You can include a `.yaml` or `.bib` file with your references and use the `cite()` function to reference them in your document. Again, you can use the same syntax sugar `@<ref>` to cite.

Typst Code	Output

`#quote[`
`  Just Because [...] Whip Your Wands Out For Everything! @harry`
`  _\~ Molly Weasley_`
`]`
	Just Because You're Allowed To Use Magic Now Does Not Mean You Have To Whip Your Wands Out For Everything! [2] ~ *Molly Weasley*

4.6 Acronyms

Specify acronyms in the project configuration to use them throughout your document. You can use the `@acr` function to reference acronyms and the `@acr:pl` function to reference their plural form.

Typst Code	Output

`#quote[`
`  I don't understand @NN or @NN:pl and never will.`
`  _\~ Me_`
`]`
	I don't understand *Neural Network (NN)* or *NNs* and never will understand.

On the first usage, the full form of the acronym is displayed, and on subsequent usages, only the acronym is shown.

5 Conclusion

Happy writing with Typst! If you have any questions, open an issue :)

Bibliography

[1] "Franklin D. Roosevelt's First Inaugural Address." [Online]. Available: https://avalon.law.yale.edu/20th_century/froos1.asp

[2] J. K. Rowling, *Harry Potter and the Order of the Phoenix*, vol. 5. 2003.

Index of Appendices

1 Appendix: Titel 1

2 This is some more source code

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.

3 Appendix: Titel 2

4 Even more!

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.