# An Awesome Project

Writing a project in Typst

Alice, Bob & Chad

Computer Science, CS-xx-DAT-y-zz, 2025

Semester Project

STUDENT REPORT

AALBORG UNIVERSITY

This report is typeset using the Typst system.

AALBORG UNIVERSITY
STUDENT REPORT

**Department of Computer Science**
Aalborg University
https://www.cs.aau.dk

**Title:**
An Awesome Project

**Theme:**
Writing a project in Typst

**Project Period:**
Fall Semester 2025

**Project Group:**
CS-xx-DAT-y-zz

**Participants:**
Alice
Bob
Chad

**Supervisor:**
John McClane

**Copies:** 1

**Number of Pages:** 13

**Date of Completion:**
21/08-2025

**Abstract:**

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.

AALBORG UNIVERSITET
STUDENTERRAPPORT

**Institut for Datalogi**
Aalborg Universitet
https://www.dat.aau.dk

**Titel:**
Et Fantastisk Projekt

**Tema:**
Et projekt i Typst

**Projektperiode:**
Efterårssemesteret 2025

**Projektgruppe:**
CS-xx-DAT-y-zz

**Deltagere:**
Alice
Bob
Chad

**Vejleder:**
John McClane

**Opsalgstal:** 1

**Sidetal:** 13

**Afleveringsdato:**
21/08-2025

**Resumé:**

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.

# Contents

# List of Todos

# Introduction

This is the introoduction, it has no chapter number, but is just a short introduction to the subject.

Is this word spelled correct?

Chapter 1

# Problem Analysis

It is easy to test a layout by providing blind text…

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.

## 1.1 Basic Syntax

A lot of the syntax is inspired by the Markdown syntax, like **bold** and *emphasised* text. The same goes with lists:

- Item 1
- Item 2
    - A sub-item

And enumerations:

1. Ordered item using a specified number
5. Another ordered item using a specified number
6. Ordered item using the previous number + 1

And even a terms list (like `description` in LaTeX):

**Term** Description

This text is a new paragraph and is not attached to the term.

**Another term** It is possible to have multiple paragraphs on the right-hand side of a paragraph though.

Just nest it in a content block using the `#[]` notation.

The documentation also has a guide for LaTeX users.

## 1.2 Citing

The citation style is determined by the `bibliography()` command in `main.typ`[1].

Citing works the same as referencing, by using `@`. [1] It's also possible to add a supplement to the source using `@source[supplement]` notation. [1, pp. 10]

[1] https://typst.app/docs/reference/model/bibliography/

We can also change the citation style using the `#cite()` command:
… This has been shown by A. Einstein [1].

## 1.3 Math

The math syntax is fundamentally different from its LaTeX counterpart. We can make inline math $a^2 + b^2 = c^2$.

And blocked equations by adding spaces in the `$` notation, as seen in Equation (1.1).

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2 \cdot a} \tag{1.1}$$

## 1.4 Figures & Labels

Unlike LaTeX, Typst does not have floats. All figures are placed immediately, or at the top or bottom of the page. The default is immediately, but can be changed with `placement: auto` (or specifically `top` or `bottom`)

Labels for figures and headings are denoted by the `<i-am-a-label>` syntax, and referenced with `@i-am-a-label`

In Section 1.4 we show how figures work, specifically Table 1.1.

| Name | Age |
|---|---|
| Alice | 25 |
| Bob | 28 |
| Chad | 24 |

**Table 1.1:** I am placed is put in the document, and may cause bad page breaks.

Include an image by using `image("../figures/your-image.png")` in place of the `table()` function. Typst will automatically infer the type of figure, and reference it accordingly.

### 1.4.1 Subfigures

Write subfigures using `#subfig()`, which uses the `subpar` package internally (see `setup/macros.typ`).

These can be individually referenced as Figure 1.1a, Figure 1.1b and Figure 1.1.

*First image goes here*

(a) The subcaption

*Second image goes here*

(b) The second subcaption

**Figure 1.1:** A figure composed of two sub figures.

# Chapter 2

# Custom Macros

## 2.1 Listings

Typst has bulitin support for raw code blocks with syntax-highlighting.

These can be placed as is:

```
This is `typst` source code.

This means that *bold* is _highlighted_, and
#show: block
are also highlighted as expected
```

Bob: I'm manually placed on the left

Alice: Notice the blank line in the raw code block.

Or inside a figure:

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

**Listing 2.1:** A piece of Rust code.

## 2.2 Revisioning

In the `setup/macros.typ` file you'll find macros to control revisions. Use them if you want, or delete / ignore them. The revision can be set in `main.typ` using `#set-revision(n)`.

This is removed content for this revision and is highlighted as such. Doing both at once is also easy. How is this great!

Removed content in a future revision is not shown as removed yet. ~~And removed content in an old revision is completely gone!~~ I was added in a previous iteration, and show up normally.

These macros can be wrapped in pretty much anything, including chapters, figures and tables.

Alice: Does not flow well, rewrite later.

Bob: I disagree

## 2.3 Acronyms

In `main.typ` you can initialize the acronyms / glossary.

Invoke an acronym via the label syntax, "`@LTS`": Labelled Transition System (LTS). Subsequent uses are then short: LTS, unless used with the `:long` or `:both` suffix.

To pluralize an acronym, use the `:pl` suffix (can be combined with the previous). This is show in Table 2.1.

| Suffix | Result |
|---|---|
| `:pl` | LTSs |
| `:long` | Labelled Transition System |
| `:both` | Labelled Transition System (LTS) |
| `:long:pl` | Labelled Transition Systems |
| `:both:pl` | Labelled Transition Systems (LTSs) |

**Table 2.1:** Basic syntax of invoking acronyms.

The World Wide Web (WWW) acronym is explicitly defined to not have its key as the shorthand. Problem Based Learning (PBL) is just another acronym.

## 2.4 Subfiling / Scoping

It is possible to sub-file the report any way you so desire (a file per chapter / section), but please note that the online app is limited to 100 files per project (free tier) as of writing. Also note that the `setup/macros.typ` file must be imported everywhere you wish to use a macro.

Furthermore, set and show rules are limited to their current scope, which includes files (setting a show rule in a parent file *will* apply to the child, but not vice versa).

To scope a set or show rule within a file, use the `#{}` or `#[]` notation.

Alice: I am scoped

foobar foo foo bar foo bar foo

Alice: End of scope

## 2.5 A note on `todonotes`

The todo macros attempts to mimic the `todonotes` package for LaTeX. It is not perfect, but it works in most scenarios.

Please note that 4 or more notes in a row will give a compiler warning (if they overlap), as the Typst compiler iteratively tries to move the notes and eventually gives up. You can either ignore the warning (at some point the notes will just be placed immediately and overlap), or move the notes around using the `side: (left | right | auto)` argument.

# Conclusion

If you find any bugs in the theme (or have a great suggestion for improvement), please submit it on https://github.com/Tinggaard/classic-aau-report/

# List of Acronyms

| | |
|---|---|
| **LTS**. Labelled Transition System | 5, 6 |
| **PBL**. Problem Based Learning | 6 |
| **WWW**. World Wide Web | 6 |

# References

[1] A. Einstein, “Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies],” *Annalen der Physik*, vol. 322, no. 10, pp. 891–921, 1905, doi: 10.1002/andp.19053221004.

# Appendix A

# Scripting

Generally, the docs are really great. For scripting specifically, see https://typst.app/docs/reference/scripting/

There is also a table guide as they are really extendible.

## A.1 Loading Data

Typst can also read files and put them in the document. It even includes parsers for a few data-file formats, including CSV, JSON, TOML and YAML[2].

If you are not limited by the amount of files in your project, you can use this to load your data-findings directly into a table or figure.

```
#let results = csv("example.csv")

#table(
  columns: 2,
  [*Condition*], [*Result*],
  ..results.flatten(),
)
```

**Listing A.1:** Loading arbitrary data into a table.

The glossary package also supports this type of loading, if you would like to manage your glossary in an external file.

Oh yeah and a single backslash is a shorthand for the `#linebreak()` function (but it does not indent).

```
#show: init-glossary.with(yaml("glossary.yaml"))
```

[2] https://typst.app/docs/reference/data-loading/