TypXdian

A modern Typst template inspired by Obsidian

SUPERVISORS
Prof. Mario Rossi
Prof. Giuseppe Verdi

CANDIDATE
Angelo Nazzaro

Faculty of Science
Department of Computer Science
Master's Degree in Computer Science

Academic year 2024/2025

"***Dreams***. *Each man longs to pursue his dream. Each man is tortured by this dream, but the dream gives meaning to his life. Even if the dream ruins his life, man cannot allow himself to leave it behind. In this world, is man ever able to possess anything more solid, than a dream?*"

~ Kentaro Miura

Abstract

This is Typst template inspired by Obsidian callouts and color palette. The template is designed for note taking and thesis/reports writing. This document serves both as a showcase example and as documentation on how to use the template.

Table of Contents

List of Figures

List of Definitions

List of Theorems

Glossary

WIP	Work In Progress
AI	Artifical Intelligence

1 Introduction

The **TypXdian template** offers a clean, modern structure for academic documents. It is heavily inspired by Obsidian's design principles, while the style of headers, headings, and citations takes inspiration from [Sca24]. This chapter introduces the general layout of the template and demonstrates how it can be configured through parameters.

1.1 Front Matter

1.1.1 Cover Page

The cover page supports two display modes:

- **Authors-only view**: for reports, notes, or shorter projects;
- **Supervisors + authors view**: for theses or dissertations.

You can switch between them by setting the `supervisors` parameter. If you pass an empty list (the default), the supervisors section will be hidden.

Additional metadata parameters include:

- `university`: name of the university;
- `logo`: university logo (defaults to `assets/figures/logo.svg`);
- `academic-year`: the academic year, e.g. **2024/2025** (defaults to `none`);
- `faculty`: the faculty name (defaults to `none`);
- `department`: the department name (defaults to `none`);
- `degree`: the degree program (defaults to `none`);
- `is-thesis`: if `true`, the "AUTHOR(S)" label changes to "CANDIDATE(S)" (defaults to `false`).

1.1.2 Abstract & Citation

By setting the `abstract` and `citation` parameters, the template adds two additional pages **before** the table of contents. The abstract page summarizes the work, while the citation page provides an optional epigraph or quotation. The citation page does **not** appear in the table of contents.

1.1.3 Table of Contents

The table of contents automatically detects headings up to level 3. It also supports dynamically generated:

- List of Figures,
- List of Tables,
- List of Definitions (Work In Progress (WIP)),
- List of Theorems (WIP).

1.1.4 Glossary

Glossary terms can be referenced with explicit citation functions from the `glossarium` package:

- `#gls("ai")` ⟶ renders the glossary entry for Artifical Intelligence (AI).
- `#glspl("AI")` ⟶ renders the plural form (if defined).

Currently, direct referencing with `@ref` is not supported.

1.1.5 Additional Content

You may inject content directly via parameters:

- `before-content`: material inserted between the abstract/citation and the table of contents (defaults to `none`);
- `after-content`: material added immediately after the table of contents.

These are useful for dedicatory pages, acknowledgements, or institutional notices.

1.2 Main Matter

Each top-level heading (=) starts on a fresh page, separating major chapters. The heading and header styles are designed for clarity and are inspired by [Sca24].

1.3 Back Matter

The back matter includes automatically generated references and a credits page. You can disable the credits page by setting the `include-credits` parameter.

2 | Figures, Equations, and Paragraphs

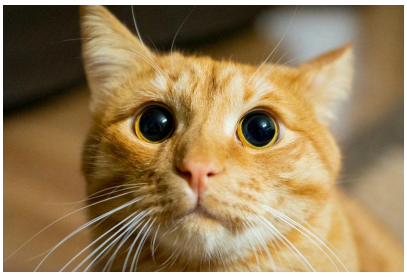
Figures (images, tables and custom environments), equations, and paragraphs are all numbered within their section to ensure consistent cross-referencing.

2.1 Subfigures

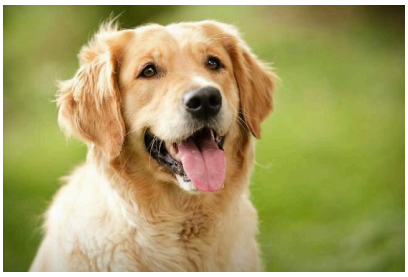
The template provides a wrapper around the `subpar` package to simplify subfigure handling. Use the `subfigure` function instead of `subpar.grid` to maintain numbering consistency. For example, this code:

```
#subfigure(
  columns: (1fr, 1fr),
  figure(
    image("assets/figures/cat.jpg"),
    caption: [This is a cat.],
  ),
  figure(
    image("assets/figures/dog.jpg"),
    caption: [This is a dog.],
  ),
  caption: [A cat and a dog.]
)
```

outputs:



(a) This is a cat.



(b) This is a dog.

Figure 2.1: A cat and a dog.

2.2 Paragraphs

The `paragraph` function enables you to create paragraphs similar to LaTeX's `\paragraph{}{}` command. For instance, this code:

```
#paragraph([Paragraph test], [I am the body of this paragraph])
<par-test>
```

outputs:

Paragraph test. I am the body of this paragraph

Each labeled paragraph can be referenced like any other element, e.g. Paragraph 2.1. Keep in mind that the paragraph functions adds vertical space before and after the content.

3 | Callouts, Definitions and Theorems

3.1 Callouts

Obisidian-like callouts are available through the following functions: `info`, `danger`, `tip`, `success` and `faq`.

Info

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.

FAQ

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.

Tip

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.

Danger

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.

Success

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.

Each callout type is referenceable, just like any other figure environment. For instance, if I were to add the `<info>` label to the first info callout, I would be able to reference it as Info. 3.1.

3.2 Definitions and Theorems

Definitions and theorems are inspired by [Sca24]. You can access either of them through the `definition` and `theorem` functions.

An example of definition would be the following:

Definition 3.1 (Machine Learning) *Machine Learning is a subfield of AI concerned with the development of algorithms able to learn from data.*

An example of theorem would be the follwoing:

Theorem 3.1 (Universal Approximation Theorem) *The universal approximation theorems state that neural networks with a certain structure can, in principle, approximate any continuous function to any desired degree of accuracy.*

Each environment is also referenceable, e.g. Definition. 3.1, Theorem 3.1.

Bibliography

[Sca24] S. Scardapane, “Alice's Adventures in a Differentiable Wonderland – Volume I, A Tour of the Land.” [Online]. Available: https://arxiv.org/abs/2404.17625

This document was written using *TypXidian*, a Typst template inspired by Obsidian.
If you'd like to contribute to the project, please open an issue or a pull request on our github repository.

Github Repository.