Template for Academic Theses at TH Köln

Bachelor's or Master's thesis to obtain the academic degree
Bachelor/Master of Science
in the degree program Computer Science
at the Faculty of Computer Science and Engineering Science
of TH Köln – University of Applied Sciences

Submitted by:	First Name Last Name
Matriculation no.:	123 456 789
Address:	Betzdorfer Str. 2
	50679 Cologne
Email:	name.surname@smail.th-koeln.de

Submitted to:	Prof. Dr. First Name Last Name
Second reviewer:	Prof. Dr. First Name Last Name

Cologne, 07.11.2026

Abstract

The abstract (if required) summarizes the presentation of the problem, the method(s) used and the most important result on about 1/2 to 1 page.

You can find out how to write a successful abstract in the seminars or by consulting the Center for Soft Skills Development[1].

Key words/key phrases: if necessary, provide 3 to 10 key words.

[1] https://www.th-koeln.de/schreibzentrum

Contents

List of Tables **IV**

List of Figures **V**

List of Listings **VI**

List of Algorithms **VII**

Glossary **VIII**

List of Abbreviations **IX**

List of Symbols **X**

1 Introduction **1**

2 Formal Structure **3**
2.1 Sequence . . . 3
2.2 Cover Page . . . 3
2.3 Table of Contents . . . 4
2.4 List of Figures and Tables . . . 4
2.5 List of Abbreviations . . . 4
2.6 Bibliography . . . 5
2.7 Spelling, Grammar . . . 5
2.8 Scope of the Thesis . . . 5

3 Layout: Typesetting with Quarto **6**
3.1 Under the Hood . . . 7
3.2 Headings . . . 7
3.3 Paragraphs . . . 8
3.4 Hyphenation . . . 8
3.5 Enumerations . . . 8
3.6 Illustrations . . . 8
3.6.1 Images . . . 9
3.6.2 Vector Graphics . . . 10

3.7 Tables . . . 11
3.8 Computations with marimo . . . 11
3.9 List of Figures and Tables . . . 14
3.10 Equations . . . 14
3.11 Theorems, Definitions and Proofs . . . 15
3.12 Source Code, Pseudocode . . . 17
3.13 Further Directories . . . 18
3.13.1 Creating a Glossary . . . 19
3.13.2 Creating a List of Abbreviations . . . 19
3.13.3 Creating a List of Symbols . . . 19
3.14 References . . . 20
3.15 Special Spaces and Signs . . . 20
3.16 Selecting the Basic Font . . . 21
3.17 Metadata for the PDF Viewer . . . 22
3.18 Switch Between Single and Double-sided Layout . . . 22
3.19 Compile . . . 22
3.20 This Template . . . 23
3.20.1 Differences from Quarto's Defaults . . . 24

4 Citations and References **25**
4.1 Citations . . . 25
4.1.1 References . . . 25
4.1.2 Ibid . . . 26
4.1.3 Secondary Quotations . . . 26
4.1.4 Citing Figures and Tables . . . 27
4.2 Citing with Quarto . . . 27
4.2.1 Cryptography: CryptoBib . . . 28
4.3 Bibliography . . . 28
4.3.1 Content and Order of the Bibliography . . . 28
4.3.2 References in the Bibliography . . . 29

Bibliography **30**

Appendix **33**

A Supplementary Material **33**

List of Tables

3.1 A simple table 11
3.2 Statistics of the measured values 13
3.3 Mathematical statements 16
3.4 Numbered elements and their references 20

4.1 Citation syntax in Markdown 27

List of Figures

3.1 Perhaps this is art? 9
3.2 A beautiful graphic generated in the source code! 10
3.3 The data of Figure 3.2, plotted with matplotlib 13

List of Listings

3.1 Quicksort in Python 17

List of Algorithms

3.1 Euclid 18

Glossary

Dog Hairy, four-legged mammal. Man's best friend 19

List of Abbreviations

SVM support vector machine 19

List of Symbols

$\vec{F}$ Force, vector quantity 19

1 Introduction

This document template can be used as a model for academic theses. It is based on guidelines created by Prof. Dr. Stephan Freichel as Chairman of the Examination Board for the B. Sc. Logistics and M. Sc. Supply Chain and Operations Management degree programs at the Faculty of Automotive Systems and Production.

The text in this template describes general formal requirements, in particular for the table of contents, the insertion of references and the compilation of a bibliography.

The template can be used interdisciplinary as a sample file for theses at TH Köln. However, when using it, you must make sure that it complies with the conventions of your degree program.

The content of the document was revised and modified by Maria-Anna Worth (retired) and Susanne Neuzerling (Department of Planning and Controlling) to make it suitable for use across all subjects. A further revision and update was carried out by Andreas Bissels (Writing Centre). Ms. Katharina Bata created the document template in LaTeX. The LaTeX template was revised in 2023 by André Ulrich and Jan Salmen and supplemented with various notes, specifically on design with LaTeX. The document was translated by Lynne Lörler (Language Learning Center). In 2026 it was reorganised as a Quarto template by Stavros Kousidis; the notes on typesetting were rewritten for Quarto (see Chapter 3). This Quarto version is maintained independently and is not an official TH Köln publication.

To prepare for your final thesis, we recommend that you take advantage of the services offered by the Writing Center of the Center for Soft Skills Development[1]; these include both counselling[2] and courses[3] on academic writing. The Writing Center is your point of contact at TH Köln for questions relating to academic writing. Additionally, the Language Learning Center[4] offers counselling[5] on writing in English.

Your thesis lives in a git repository that was created from this template. The history of the repository keeps the original template text available, so you can simply overwrite

[1] https://www.th-koeln.de/schreibzentrum
[2] https://www.th-koeln.de/schreibberatung
[3] https://www.th-koeln.de/schreibzentrum/seminare
[4] https://www.th-koeln.de/sprachlernzentrum
[5] https://www.th-koeln.de/internationales/schreibberatung-englisch-fuer-studierende_84724.php

the chapters with your own content. If you want to look something up later, the template repository itself remains available as well.

Please note: This template does not supersede the specific requirements of the respective examination boards. If there are special formal requirements in your subject, these apply.

Cologne, November 2026

2 Formal Structure

In this chapter you will find basic information on the formal structure of your thesis.

2.1 Sequence

A scientific paper usually consists of the following parts:

1. Cover sheet
2. Abstract (optional)
3. Table of contents
4. List of figures and tables (also common at the end)
5. List of abbreviations (also common at the end)
6. Introduction
7. Main part
8. Summary/conclusions
9. Bibliography
10. Appendices (optional)
11. Declaration

In this template, the cover sheet and the declaration are generated from the `thesis:` block in `_quarto.yml`, the lists are inserted automatically after the abstract, and the order of the chapters is defined in the `book:` block of the same file.

2.2 Cover Page

The cover page includes: title of the thesis, type of thesis, author, matriculation number, submission date, supervisor and second reviewer. For theses longer than 15 pages, the cover page is included in the number of pages but not numbered.

The cover page of this template shows placeholders ("Bachelor's or Master's thesis"). Set the type of your thesis and your degree in `_quarto.yml`:

```
thesis:
  type: master                  # or: bachelor
  degree: "Master of Science"   # or e.g. "Bachelor of Science"
```

The type `bachelor` or `master` gives “Bachelor’s thesis” or “Master’s thesis” on the cover page.

2.3 Table of Contents

We recommend a decimal structure as laid out in this document. If subheadings are used within a chapter, there must be at least two: where there is a 2.1, there must be a 2.2.

The table of contents always contains the page references for the listed items; however, it is not listed in the table of contents itself. The pages that the table of contents itself takes up can be counted in Roman numerals.

For a thesis, a structure of at least three levels is usually used. As a rule, only up to four levels are shown in the table of contents. However, you must find out about and consider the conventions in your subject.

2.4 List of Figures and Tables

Figures and tables are listed in corresponding directories. In this template, they appear directly after the table of contents. The corresponding pages can then be numbered in Roman numerals. However, the lists can also be placed at the end of the work before or after the bibliography. In this case, they are given regular page numbers.

2.5 List of Abbreviations

The number of abbreviations should be kept to a minimum. The list of abbreviations only contains important subject-specific abbreviations in alphabetical order, in particular abbreviations of organisations, associations or laws. Common abbreviations such as i. a., e. g., etc. are not included.

For the technical implementation, see also Section 3.13.

2.6 Bibliography

The bibliography is organized alphabetically by author name. It contains all sources cited in the text – and only these. Several writings by the same person are organized by year of publication. You must distinguish between writings by the same person from the same year of publication. In engineering, a number or author abbreviation is often placed before the name in square brackets. You can learn more about this and other important rules of citation in the Writing Centre's e-learning courses[1].

Appropriate software tools such as Citavi or Zotero, which are compatible with various word processing programs, are suitable for managing the literature used. Zotero, for example, can export your library as a `.bib` file that is updated automatically (Better BibTeX plugin).

2.7 Spelling, Grammar

When submitting your work, make sure that your English is flawless. If errors impair readability, this can have a negative impact on your grade. It is therefore essential that you use a spell checker, even if it does not detect all errors. Most editors used with Quarto (VS Code, Positron, RStudio) offer one.

For those who feel unsure about this topic, we recommend the Writing Centre's e-learning courses[2]. If necessary, please also contact the representative for students with disabilities[3].

2.8 Scope of the Thesis

All subjects specify binding upper and lower limits that must generally be adhered to. Indexes and appendices are generally not counted. In individual cases – especially in the case of empirical work – deviating agreements can be made with the supervisor.

[1] https://ilu.th-koeln.de/goto.php?target=cat_52109&client_id=thkilu
[2] https://ilu.th-koeln.de/goto.php?target=cat_52109&client_id=thkilu
[3] https://www.th-koeln.de/studium/studieren-mit-beeintraechtigung_169.php

3 Layout: Typesetting with Quarto

This template uses Quarto, an open-source publishing system. You write your text in *Markdown*, a plain-text format with a very small set of formatting rules. Quarto converts it with Pandoc into LaTeX and LaTeX produces the PDF. You thus get the typographic quality of LaTeX, which is based on TeX, developed by Donald Knuth [Knu86], without having to write LaTeX yourself. Wherever Markdown is not sufficient, you can still use LaTeX directly. Some further advantages over conventional word processing are:

Free/platform-independent Quarto and LaTeX are free software. No proprietary editor is required; the documents can be edited on *any* computer with *any* editor. VS Code, Positron and RStudio have Quarto support with a live preview.

Pure text format The source text – the `.qmd` files – is pure text. This makes the documents ideal for version control with git. This in turn enables efficient collaboration between you and your supervisor, e.g. via issues and pull requests.

Splitting large documents Every chapter lives in its own file in the folder `chapters/`. The order of the chapters is defined in `_quarto.yml`.

Separating layout/content The layout of the entire document is defined once, in the Quarto extension in `_extensions/thk-thesis/`. You can concentrate fully on writing.

Professional result A document typeset with LaTeX looks more professional than a corresponding document created with word processing. This is especially true for math-heavy documents, but other documents also benefit from a uniform layout, consistent gray value of the continuous text, more consistent page breaks and high-quality vector graphics.

Executable Python code in marimo cells runs while rendering and inserts the resulting figures and tables. Your evaluation then stays reproducible, see Section 3.8.

The following Section 3.1 to Section 3.20 deal with various aspects that you should consider when creating your document.

3.1 Under the Hood

In your `.qmd` files, you define what your content is (text with structure, images, tables, references, …) and how it should roughly look in each case (e.g. placement of images in *floating environments*, see Section 3.6).

When creating the final document, LaTeX applies a whole host of rules "under the hood" that determine how everything can be implemented in the best possible way. These rules concern, for example, the proportion of text and images per page, spacing within lines, but also special cases such as avoiding individual lines of a section alone on a page (so-called "orphans" or "widows").

As a result, for example, your illustrations might "jump" while you are working on your text. This may well be correct and can be optimized at the end if necessary.

In this context, avoid interfering with the design process, e.g. by manually inserting line breaks or spacing. Please only make exceptions in justified cases.

3.2 Headings

In this template, the chapter (`#`) is the highest level of structure. Each chapter file starts with exactly one of them. This is followed by sections (`##`) and subsections (`###`). These three levels are numbered and appear in the table of contents. If you want to structure your text further, there is also `####`. A heading gets an identifier for cross-references with `{#sec-name}`, an unnumbered heading with `{.unnumbered}`:

```
# Formal Structure {#sec-formal}

## Sequence {#sec-sequence}

# Abstract {.unnumbered}
```

Please note that two headings should never directly follow each other in the text. A heading should always be followed by some text (see, for example, the beginnings of the chapters here on page 3 and page 6). For further information, see Section 2.3.

3.3 Paragraphs

A new paragraph begins after a blank line in the source text. A single line break in the source text has no effect, so you can – and should – start a new line for each sentence or after about 80 characters. This makes changes easier to follow in git. The exact appearance of paragraphs in the finished document is determined by the `parskip` option of the document class – more on this in Section 3.20.

3.4 Hyphenation

The automatic hyphenation in LaTeX works well in principle. However, there can always be minor problems and undesired results. If you want to influence the hyphenation for a specific word, add a `\hyphenation{...}` command to `tex/macros.tex` to specify the permitted hyphenation points. This also allows you to ensure that certain words are never hyphenated, for example proper names.

For example, words containing a hyphen are *only* separated there, which can lead to lines not being set correctly, resulting in a warning (see Section 3.19). In such cases, you would have to specify additional hyphenation points manually.

3.5 Enumerations

Markdown provides

- bulleted lists with `*` or `-`,
- numbered lists with `1.`, `2.`, …,
- definition lists (a term in one line, followed by `: definition`, as at the beginning of this chapter).

The exact appearance is set globally for the document.

3.6 Illustrations

When someone looks at your finished thesis, it may well be that they first leaf through it roughly, hardly reading the text but looking at the illustrations. Hence the "rule" that you should be able to understand the most important points of the thesis in this way.

Illustrations never stand alone but are described by the *caption*. It should contain everything that is necessary to understand the illustration. Only in exceptional cases must the caption refer to the text. Conversely, each figure must be referenced at least once in the text, see also Section 3.14.

In the following two sections, a distinction is made between images (in Section 3.6.1) and vector graphics (in Section 3.6.2), as these are completely different techniques that should be used appropriately in each case.

Remember that not everyone can see all colors equally well. Around 10 % of men in Germany, for example, are affected by red-green color-blindness. Your thesis may also be printed in black and white. Therefore, you should ideally design illustrations in such a way that they are understandable even without colors.

3.6.1 Images

You include an image with `![Caption](path){#fig-name}`. The identifier starting with `fig-` makes it a numbered figure that floats to an optimal position, see Figure 3.1.



Figure 3.1: Perhaps this is art?

```
![Perhaps this is art?](/images/ThkKunst.png){#fig-kunst
↪  width=35%}
```

Where a figure may be placed is controlled by `fig-pos` (in this template `tbh`): `t` stands for the top of the page, `b` for the bottom and `h` for "here", which means the position within the text. If you need to save space, `t` and `b` are preferable.

Attention: Many contents such as formulas, code, diagrams, visualization of data, etc. should not be inserted as an image, but in a suitable form. See the following section on vector graphics.

3.6.2 Vector Graphics

You do not need to insert simple illustrations (e.g. coordinate systems, flowcharts, etc.) as an image. Instead, you can generate them directly in the source code. The powerful TikZ package is ideal for this: Figure 3.2 is drawn by `figures/fleuret.tex` from the data in `data/pointsFleuret.csv`.

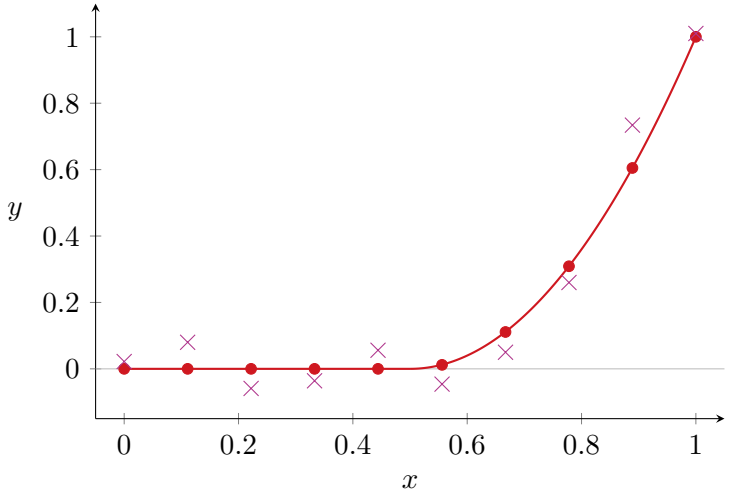

Figure 3.2: A beautiful graphic generated in the source code!

```
::: {#fig-fleuret}
```{=latex}
\input{figures/fleuret.tex}
```

A beautiful graphic generated in the source code!
:::
```

One advantage is that your document remains smaller. Another is that the illustrations usually look more attractive. This is especially true when viewing on a screen, as vector graphics can be scaled to any size. You can find lots of examples of TikZ on the Internet. There is also an up-to-date book [Kot23].

Alternatively, you can create plots from your data with Python while rendering, see Section 3.8.

3.7 Tables

Tables are written as *pipe tables*. The caption follows after the table, introduced by a colon, together with an identifier starting with `tbl-`.

Table 3.1: A simple table

Heading left	Heading right
1	2222
10	222
100	22

```
| Heading left | Heading right |
|-------------:|--------------:|
|            1 |          2222 |

: A simple table {#tbl-example}
```

The colons in the second line define the alignment of the columns. For more complex tables (merged cells, colored rows), you can use grid tables or raw LaTeX, e.g. with the `booktabs` and `tabularx` packages.

3.8 Computations with marimo

If your thesis contains measurements, simulations or other evaluations, you should not paste their results into the text by hand. Instead, put the Python code that produces figures and tables directly into the chapter. Quarto runs it on every render with marimo, a reactive Python notebook, so text, figures and data cannot get out of sync.

A marimo cell is a code block with the class `.marimo`. A chapter with such cells needs two entries in its front matter (at the very top of the file, as in this chapter): `engine: marimo` and the Python packages it uses:

```
---
engine: marimo
pyproject: |
  requires-python = "==3.13.*"
```

```
  dependencies = [
    "marimo==0.25.2",
    "matplotlib==3.11.2",
    "pandas==3.1.1",
    "tabulate==0.10.0",
  ]
---
```

The packages are installed automatically into an isolated environment by uv, which you need to install once (`make check` tells you whether it is there). Pin exact versions (`==`) as above: computed results in a thesis must be reproducible. The packages these depend on are frozen by a date, `UV_EXCLUDE_NEWER` in the file `_environment`: uv uses no release published after it. Move the date forward only on purpose (`make python-date`), and keep it fixed once you submit. All cells of a chapter form one marimo app, i.e. a later cell can use the variables of an earlier one. Code is run in the folder of the project, so paths such as `data/...` start there. By default, only the output of a cell appears in the PDF; with `#| echo: true` the code is shown as well, as in the following cells.

The first cell loads the data of Figure 3.2 and defines a small helper `figure()`. It saves a matplotlib figure as vector PDF and inserts it as a numbered figure with a caption and the identifier `fig-<label>`:

```
import marimo as mo
import matplotlib.pyplot as plt
import pandas as pd
from pathlib import Path

data = pd.read_csv("data/pointsFleuret.csv", names=["x", "y",
 ↪  "z"])

def figure(fig, label, caption, width="70%"):
    Path("_generated").mkdir(exist_ok=True)
    fig.savefig(f"_generated/{label}.pdf", bbox_inches="tight")
    plt.close(fig)
    return mo.md(f"![{caption}](/_generated/{label}.pdf)"
                 f"{{#fig-{label} width={width}}}")
```

The next cell creates Figure 3.3:

```
fig, ax = plt.subplots(figsize=(5, 3))
ax.plot(data.x, data.y, "o", color="#CF1820", label="$y$")
ax.plot(data.x, data.z, "x", color="#AF368C", label="$z$")
ax.set_xlabel("$x$")
ax.legend()
figure(fig, "measurements", "The data of @fig-fleuret, plotted
↪  with matplotlib")
```

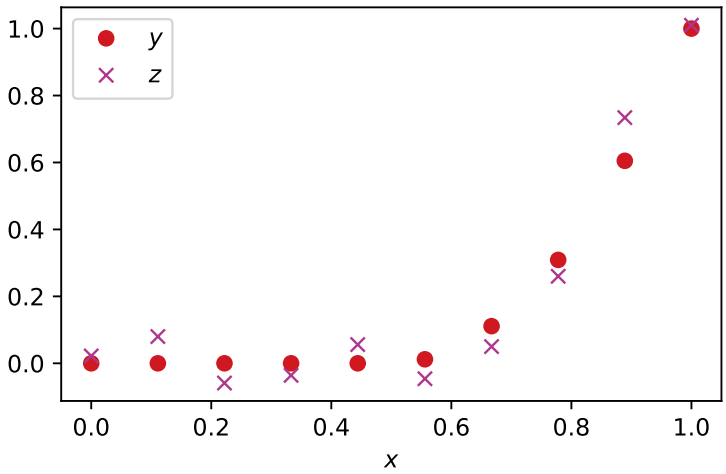

Figure 3.3: The data of Figure 3.2, plotted with matplotlib

A table is created as Markdown text with `mo.md()`; the caption and the identifier are added by a surrounding div, as for any other content (Table 3.2):

```
stats = data.describe().loc[["mean", "std", "min", "max"]]
mo.md(stats.round(3).to_markdown())
```

Table 3.2: Statistics of the measured values

	x	y	z
mean	0.5	0.204	0.207
std	0.337	0.343	0.368
min	0	0	-0.059
max	1	1	1.01

The figures are written to the folder `_generated/`, which is not part of the repository. Rendering a chapter with marimo cells takes a few seconds longer; the first render also downloads the packages. For larger evaluations, it is often more convenient to develop the code in a separate marimo notebook (`uvx marimo edit --sandbox notebooks/analysis.py`) and to move only the final plotting and table code into the chapter.

3.9 List of Figures and Tables

Quarto creates the list of figures and the list of tables automatically. All figures and tables with an identifier and a caption appear there (see Section 3.6.1 and Section 3.7).

The caption is used by default for these lists. This text is often too long. You can therefore specify a shorter version with the attribute `fig-scap` (for figures) or `tbl-scap` (for tables). See also Section 4.1.4.

3.10 Equations

One of the greatest strengths of LaTeX is that you have many possibilities to write down formulas simply and beautifully. You can use LaTeX math directly in Markdown: inline between single dollar signs, displayed between double dollar signs, followed by an identifier starting with `eq-`. Here is just a small example (Equation (3.1)):

$$\begin{aligned} \sum_{i=1}^{n} i &= 1 + 2 + \ldots + (n-1) + n \\ &= (1+n) + (2 + (n-1)) + \ldots \\ &= \frac{n}{2} \cdot (n+1) \end{aligned} \tag{3.1}$$

However, simple formulas in the text such as $x \in \mathbb{N}$ are of course also possible. A common mistake here is that the "math mode" is forgotten in the text: We are talking about the x-value and *not* the x-value.

The error is just as common the other way round, i.e. text is written in math mode: $a_{wrong} = 42$, but $a_{correct} = 42$, perhaps also $a_{\text{correct}} = 42$.

Functions such as sin are automatically displayed well, as in

$$\sin\alpha = \left(\frac{a}{c}\right) \tag{3.2}$$

The brackets have only been inserted here to demonstrate the corresponding mechanism: automatic scaling of brackets!

Only equations with an identifier get a number, so number exactly those you refer to. A reference `@eq-gauss` gives "Equation (3.1)"; `-@eq-gauss` gives only the number in parentheses, e.g. "by (3.1) and (3.2)". This is a customization of the template, see Section 3.20.1.

Sometimes you want to use your own "operator" that should look the same. This is exactly what the `\DeclareMathOperator` command is for, allowing you to add missing functions such as $\operatorname{sgn}(x)$.

It is advisable to give all mathematical symbols that you use in your work meaningful names in the source code; the easiest way to do this is with the `\newcommand` command in `tex/macros.tex`. You can then customize how you want to display the weight vector $\mathbf{w}$ in the entire document or what the imaginary unit i with $\mathrm{i}^2 = -1$ should look like. Of course, this assumes that you use the designations consistently. However, this also makes your source code easier to read!

3.11 Theorems, Definitions and Proofs

Mathematical statements are written as divs whose identifier starts with the type of the statement. An optional heading inside gives the statement a name. A proof follows in a div with the class `.proof`; it ends with a box.

Definition 3.1 (Triangular number)**.** For $n \in \mathbb{N}$, the n-th *triangular number* is $T_n = \sum_{i=1}^{n} i$.

Theorem 3.2 (Gauss)**.** *For every $n \in \mathbb{N}$, $T_n = \frac{n(n+1)}{2}$.*

Proof. Pair the first and the last summand, the second and the second to last, and so on, as in Equation (3.1). Each of the $\frac{n}{2}$ pairs sums to $n+1$; for odd n, the middle summand is half a pair. □

Lemma 3.3. *For every $n \geq 2$, $T_n + T_{n-1} = n^2$.*

Proof. By Theorem 3.2, $T_n + T_{n-1} = \frac{n(n+1)}{2} + \frac{(n-1)n}{2} = n^2$. □

Corollary 3.4. *For every $n \in \mathbb{N}$, the product $n(n+1)$ is even.*

Proposition 3.5 (Nicomachus)**.** *For every $n \in \mathbb{N}$, $\sum_{i=1}^{n} i^3 = T_n^2$.*

Example 3.6. $T_4 = 1 + 2 + 3 + 4 = 10 = \frac{4 \cdot 5}{2}$.

Remark 3.7. According to a well-known anecdote, Gauss found Theorem 3.2 as a schoolboy.

The source of Definition 3.1, Theorem 3.2 and its proof:

```
::: {#def-triangular}
## Triangular number
For $n \in \mathbb{N}$, the $n$-th *triangular number* is
$T_n = \sum_{i=1}^{n} i$.
:::

::: {#thm-gauss}
## Gauss
For every $n \in \mathbb{N}$, $T_n = \frac{n(n+1)}{2}$.
:::

::: {.proof}
Pair the first and the last summand, ...
:::
```

All statements of a chapter share one counter: Definition 3.1 is followed by Theorem 3.2, Lemma 3.3 and so on, which makes each number easy to find (a customization, see Section 3.20.1). Theorems, lemmas, corollaries and propositions are set in italics; definitions and examples upright; remarks upright with an italic heading.

Table 3.3: Mathematical statements

Statement	Identifier	Reference gives
Definition	`#def-...`	Definition 3.1
Theorem	`#thm-...`	Theorem 3.2
Lemma	`#lem-...`	Lemma 3.3
Corollary	`#cor-...`	Corollary 3.4
Proposition	`#prp-...`	Proposition 3.5
Example	`#exm-...`	Example 3.6

Statement	Identifier	Reference gives
Remark	`#rem-...`	Remark 3.7

Further types are conjecture (`#cnj-`), exercise (`#exr-`) and solution (`#sol-`). A statement without an identifier, e.g. `::: {.theorem}`, is not numbered.

3.12 Source Code, Pseudocode

If a section of source code is to be displayed in the final thesis, the obvious idea is to simply take a screenshot of it and display it as an image. However, this is not a good idea, as the text pixelates when the finished document is zoomed in. Instead, use a *fenced code block* with the name of the programming language; Quarto then highlights the syntax automatically. With an identifier starting with `lst-` and a caption, the code becomes a numbered listing, see Listing 3.1.

Listing 3.1 Quicksort in Python

```python
def quicksort(xs):
    if len(xs) <= 1:
        return xs
    pivot, *rest = xs
    smaller = [x for x in rest if x < pivot]
    larger = [x for x in rest if x >= pivot]
    return quicksort(smaller) + [pivot] + quicksort(larger)
```

The source of this listing looks like this:

````
```{#lst-quicksort .py lst-cap="Quicksort in Python"}
def quicksort(xs):
    ...
```
````

The language is given by its name or file extension (`.py`, `.c`, `.java`, `.rust`, …). In chapters with marimo cells (Section 3.8), always write `.py` for Python listings: a block marked `.python` is treated as a marimo cell there and stops the render.

The colors are controlled by the option `highlight-style` in `_quarto.yml`. To show a file from your project, use the include shortcode inside a code block, e.g. `{{< include ../src/main.c >}}`.

Pseudocode is written as an algorithm, one of Quarto's statement types: a div whose identifier starts with `alg-`, with an optional name as heading and the pseudocode in a raw LaTeX block (Algorithm 3.1). The commands (`\Procedure`, `\While`, `\State`, `\Return`, …) are those of the package `algpseudocodex`. Algorithms are numbered separately from theorems and are listed in the List of Algorithms.

Algorithm 3.1 (Euclid)**.**

procedure EUCLID(a, b)
  while $b \neq 0$ ***do***
    $(a, b) \leftarrow (b, a \bmod b)$
  return a

````
::: {#alg-euclid}
## Euclid

```{=latex}
\begin{algorithmic}[1]
\Procedure{Euclid}{$a, b$}
  \While{$b \neq 0$}
    \State $(a, b) \gets (b, a \bmod b)$
  \EndWhile
  \State \Return $a$
\EndProcedure
\end{algorithmic}
```
:::
````

3.13 Further Directories

Additional directories are created using the LaTeX `glossaries` package. A glossary and lists of abbreviations and symbols are already set up. All entries are defined in the file `tex/glossary.tex`; only entries that are actually used in the text appear in the lists. The lists are inserted after the table of contents and listed there. Which lists are printed is set in the `thesis: lists:` block of `_quarto.yml`.

The template uses the `\makenoidxglossaries` mode of the package, so no external program is needed to sort the entries.

3.13.1 Creating a Glossary

An entry in the glossary is defined in `tex/glossary.tex`:

```
\newglossaryentry{dog}{name={Dog}, description={Hairy,
 ↪  four-legged mammal}}
```

The term is then displayed in the text with `\gls{dog}`. Want an example? The Dog is man's best friend.

3.13.2 Creating a List of Abbreviations

An abbreviation is defined in `tex/glossary.tex` with the label, the abbreviation and the long form:

```
\newacronym{svm}{SVM}{support vector machine}
```

It is then used in the text with `\gls{svm}`. LaTeX automatically ensures that the abbreviation is written out the first time it is mentioned in the text. For all subsequent mentions, only the abbreviation is then displayed. Want an example? This is a support vector machine (SVM). And here is another SVM. Use `\glspl{svm}` for the plural.

3.13.3 Creating a List of Symbols

A symbol is defined like a glossary entry, but with the additional key `type=symbols`. The command `\gls{label}` can then be used to display the symbol in the text – also in formulas. Need an example? The force $\vec{F}$ is defined according to the following equation:

$$\vec{F} = m \cdot \vec{a}$$

3.14 References

Everything numbered gets an identifier; `@identifier` in the text then gives the name and the number, and the PDF links to it. Table 3.4 lists all numbered elements of this template.

Table 3.4: Numbered elements and their references

Element	Identifier	Example	Reference gives
Chapter, section	`{#sec-...}`	Section 3.2	Section 3.2
Figure	`{#fig-...}`	Section 3.6.1	Figure 3.1
Table	`{#tbl-...}`	Section 3.7	Table 3.1
Equation	`{#eq-...}`	Section 3.10	Equation (3.1)
Statement (theorem, …)	`#thm-...` etc.	Section 3.11	Theorem 3.2
Listing	`#lst-...`	Section 3.12	Listing 3.1
Algorithm	`#alg-...`	Section 3.12	Algorithm 3.1

Sections, figures, tables, listings, equations and algorithms each have their own counter; all mathematical statements share one. If you only want the number, write `-@fig-kunst` (3.1); for the page, use `\pageref{fig-kunst}` (page 9). Several references at once: `[@thm-gauss; @lem-consecutive]` gives Theorem 3.2, Lemma 3.3.

Every figure and every table must be referenced in the text; no numbered environments may simply "hang in the air". In contrast, sections and equations do not all have to be explicitly referenced. However, you can help your readers by thinking about meaningful references.

3.15 Special Spaces and Signs

A line break (or even a page break) can always occur at spaces. This should be avoided in some cases, partly because lines should not start with numbers. A typical example is "Figure 3". Here you need a so-called protected space after "Figure", which is created in Markdown with a backslash followed by a space (`Figure\ 3`). Cross-references such as `@fig-kunst` get one automatically.

Horizontal spacing between words and sentences is dynamically adjusted in each line. This template uses `\frenchspacing`, so the space after the dot in abbreviations such as "e.g." is not too large.

As with spaces, there are also different lengths for horizontal dashes. For dashes – here, for example – or if "to" is meant (as in 14:00–16:00), the single hyphen (minus sign) is not sufficient, the appropriate dash is created by a double minus (`--`).

Only a few characters have a special function in Markdown, e.g. `*`, `_`, `#`, `$`, `@`, `[` and `]`, as well as `<` and `\`. If you want to output them literally, put a backslash in front of them: `\$` gives $, `\@` gives @. Characters such as % and & that are problematic in LaTeX can be typed directly.

Quotation marks are set automatically in the correct style of the document language when you type straight quotes ("…" or '…').

3.16 Selecting the Basic Font

By default, LaTeX uses a serif font (Latin Modern) for continuous text. Serif fonts are advantageous for printed work because the serifs emphasize the baseline and thus help the eye to jump back to the beginning of the next line at the end of the line. In addition, the different stroke widths lead to clearer word images and thus support the reading process.

However, if such a document is viewed on an old monitor with a low resolution, the fine serifs may no longer be displayed properly. In such a case, it may be advantageous to use a sans serif font. It may also be that sans serif fonts are preferred for reasons of accessibility.

As this template uses `lualatex`, you can use any font installed on your computer. Add the following to the `thk-thesis-pdf` format in `_quarto.yml`:

```
format:
  thk-thesis-pdf:
    mainfont: "Arial"     # or e.g. "TeX Gyre Pagella", "Source
 ↪  Serif 4"
    sansfont: "Arial"
```

3.17 Metadata for the PDF Viewer

Some PDF viewers can display additional metadata, such as the name of the author, the title of the document (also different from the name of the file) or keywords. This template takes them from `book: title`, `book: author` and `keywords` in `_quarto.yml`.

In addition, links in the document can be highlighted as colored text. For a printed thesis this is usually not wanted, so it is switched off. If you prefer colored links on screen, set `colorlinks: true` in the format options.

3.18 Switch Between Single and Double-sided Layout

This template is optimized for a one-sided layout. The left and right margins are the same size on all pages and new chapters start immediately on the next page. When the finished PDF document is viewed on the computer, it looks just right. However, if the document is to be printed double-sided, the layout can be adjusted slightly. Typically, the inner margins are then slightly narrower than the outer margins. New chapters always start on a new, right-hand page – which can lead to individual blank pages between the chapters. This looks better for double-sided documents.

To activate the double-sided layout, add the following lines to the `thk-thesis-pdf` format in `_quarto.yml`:

```
    classoption: [BCOR=8mm, headings=normal, parskip=half,
↪  headsepline,
                  automark, listof=totoc, bibliography=totoc,
                  twoside, open=right]
```

3.19 Compile

You create the PDF with `make` (or `quarto render`) in the folder of your thesis; it ends up in `_book/thesis.pdf`. During writing, `make preview` (or `quarto preview`) is more convenient: the PDF is updated every time you save a file. Your editor (VS Code, Positron, RStudio) has buttons for both. `make help` lists all available commands.

Rendering a large document takes some time. As you usually work on only a few chapters at a time, you can comment out the remaining chapters in `_quarto.yml` with `#`.

Basically, the aim should be for your document to compile without warnings. The best thing to do is to regularly take care of rectifying any problems.

A typical warning is about an unresolved cross-reference or citation, shown as `?@fig-xyz` in the PDF. In this case, check the identifier for typos.

Boxes that are too full (a line sticks out into the margin) or too empty (a line with large gaps) are listed by `make warnings`: it renders the thesis, keeps the LaTeX log and shows each warning with the matching lines of the generated `thesis.tex`, so you can find the place in your text. To avoid lines that are too long, it sometimes helps to mark hyphenation points (see Section 3.4). Otherwise, a sentence may have to be minimally reworded.

Every push to the repository on GitHub also renders the thesis automatically; the PDF can be downloaded from the *Actions* tab. Your supervisor therefore always sees the current state.

3.20 This Template

The layout is defined by the Quarto extension in `_extensions/thk-thesis/`. It uses KOMA-Script[1], a "collection of classes and packages for LaTeX", with carefully chosen typographic defaults. Please do not change the files in the extension; your own LaTeX commands belong in `tex/macros.tex`, your glossary entries in `tex/glossary.tex`. Improvements of the template can then be installed with `make update-template`.

The document class options (`classoption` in the extension) determine, among other things, the font size, the binding correction (`BCOR`), the size of the headings (`headings`) and the spacing between paragraphs (`parskip`). In principle, you can also influence the size of the margins with the `DIV` option, but this is not recommended. The margins are automatically set so that lines have a length that is easy to read.

The header shows the current chapter heading at the top right of the page; the page number is at the bottom right. The most important packages are:

`babel` Selects English (from `lang` in `_quarto.yml`) and its hyphenation patterns.
`selnolig` Ensures that ligatures are set correctly (with `lualatex`).
`microtype` Optimizes the appearance of the text (type area).

[1] https://komascript.de

`csquotes` Typesets quotation marks correctly.
`tikz` **and** `pgfplots` Generate images directly in the source code, see Section 3.6.2.
`hyperref` Clickable links and bookmarks in the PDF.
`fvextra` Wraps long lines in code blocks instead of letting them run into the margin.
`biblatex` Citations and bibliography, processed with `biber`, see Chapter 4.
`amsmath`, `amssymb` **and** `mathtools` Great expansion of the possibilities for displaying mathematical content.
`glossaries` Glossary, list of abbreviations and list of symbols, see Section 3.13.
`amsthm` **and** `algpseudocodex` Theorems, definitions and proofs (Section 3.11); pseudocode in algorithms (Section 3.12).

3.20.1 Differences from Quarto's Defaults

Everything in this template is standard Quarto, except for a few customizations in the extension `_extensions/thk-thesis/`. If you read Quarto's documentation, these are the places where the template behaves differently:

Shared counter for statements Quarto numbers each type of statement separately (Definition 3.1, Theorem 3.1, Lemma 3.1). Here, all statements of a chapter share one counter (Definition 3.1, Theorem 3.2, Lemma 3.3), see Section 3.11. Algorithms keep their own counter.
Equation references Quarto refers to an equation numbered (3.1) as "Equation 3.1". Here, `@eq-...` gives "Equation (3.1)" and `-@eq-...` gives "(3.1)", matching the number next to the equation.
List of Algorithms Algorithms (`#alg-...`) are typeset as in Quarto, but are also collected in a List of Algorithms after the table of contents.
Front matter and back matter Cover page, abstract, the lists after the abstract (table of contents, figures, tables, listings, algorithms, glossary, abbreviations, symbols) with Roman page numbers, the position of the bibliography before the appendix, and the declaration are generated from the `thesis:` block in `_quarto.yml`.
Layout KOMA-Script class `scrreprt` with header and page numbers as in the TH Köln LaTeX template, biblatex instead of Quarto's default bibliography processing, and code blocks that wrap long lines. The titles "Contents" and "Appendix" (Quarto: "Table of contents", "Appendices") are set under `language:` in `_quarto.yml`.

The template's own Lua filter (`thk-thesis.lua`) and LaTeX files (`thk-header.tex`, `partials/`) contain all of this; their comments mark each customization.

4 Citations and References

Third party ideas must always be identified. Above all, they must be verifiable and findable. The technique of citing and referencing serves this purpose.

Different disciplines and degree programs follow specific citation conventions. Important information on common citation systems and styles can be found in the Writing Center's e-learning courses[1].

The sections Section 4.1 and Section 4.3 give general advice on citing, which applies whatever tool you use. Boxes titled *In this template* say what the template does by default and how to change it. Section 4.2 explains the commands.

4.1 Citations

Quotation marks ("…") are used to indicate text passages that have been copied verbatim.

If your quote already contains a quotation, you must replace the "double quotation marks" in the text with 'single quotation marks'. See also Section 3.15.

4.1.1 References

A source reference must be created for all citations. The source reference is a short notation in the citation style which refers to the complete reference in the bibliography. References can be made *either* in the continuous text (Anglo-American citation style or Harvard style) *or* via a footnote at the bottom of the page *or* in an endnote at the end of the entire text. Different professional conventions apply here, which must be observed.

If you opt for short references in the text flow or for endnotes, the footnote area can be used for comments and for references to passages in your own text.

Note the positions of the superscript and punctuation marks:

[1] https://ilu.th-koeln.de/goto.php?target=cat_52109&client_id=thkilu

If the literal quotation itself ends with a period, this is placed before the closing quotation mark. The superscript then follows directly afterwards without a space. A period is then omitted for the sentence itself.

If the literal quotation does not end with a period, there are two cases: If the quotation is in the middle of your own sentence, the superscript follows directly after the quotation marks. If the quotation is at the end of your own sentence, first add the quotation marks, then your own period and only then the superscript.

To ensure that the source references remain unambiguous even if several short titles or year numbers of an author have the same name, the year numbers are given an additional small letter.

> **In this template**
>
> References are short labels in square brackets in the text, e.g. [Knu86] (biblatex style `alphabetic`). There are no footnote citations, and the superscript rules above do not apply. Labels that would be equal are told apart automatically (e.g. [BF01a] and [BF01b]). For author-year references in the text, set `biblio-style: authoryear`; for references in footnotes, add `autocite=footnote` to `biblatexoptions` as well (both in `_quarto.yml`, under `format: thk-thesis-pdf:`). Agree on the style with your supervisor.

4.1.2 Ibid

Today, reference managers make referencing via *ibid* superfluous, as you are no longer forced to type out the same information repeatedly. If you nevertheless decide to use this method (e.g. on the advice of your examiner), we strongly recommend that you only insert ibid. in the final editorial phase, as the references are only clear then.

> **In this template**
>
> Never type *ibid.* by hand. If it is required, use a biblatex style that inserts it automatically, e.g. `biblio-style: authoryear-ibid` or `verbose-ibid`.

4.1.3 Secondary Quotations

Secondary quotations are quotations from an author without having checked the original source for the context. Secondary citations are only permitted if the original source cannot be obtained. In the case of generally accessible scientific literature,

secondary quotations should be avoided at all costs. If it is not possible to compare a quotation with the original text, then you should note *cited by* (followed by the source from which the quotation was taken) or *cited in*. This occurs, for example, when quoting facts from textbooks on the history of science.

4.1.4 Citing Figures and Tables

The same guidelines apply here as for text quotations, i.e. there are also verbatim and modified reproductions. In the case of a verbatim reproduction, you can use an illustration e.g. in PowerPoint or a drawing/vector program. Copyright must be taken into account when scanning; this is only permitted if the author or publisher has granted permission – usually for a fee. In this case, the source reference must be made in the same way as for a literal quotation. For your own changes, an addition must be added to the source reference (e.g. *with minor changes*, *with own calculations*, etc.).

Because graphics are often copied, it is advisable to mark your own graphics or illustrations as such.

In table and figure captions, the source is always given directly below the figure and not in a footnote or endnote. However, the source information does not belong in the list of figures and tables. See Section 3.9.

4.2 Citing with Quarto

All sources are stored in BibTeX files in the folder `bib/`. Your own entries go into `bib/references.bib`; you can maintain this file by hand or export it from a reference manager such as Zotero. In the text, you cite with the key of the entry:

Table 4.1: Citation syntax in Markdown

Markdown	Result
`[@knuth:tex]`	[Knu86]
`[@knuth:tex, p. 33]`	[Knu86, p. 33]
`[@knuth:tex; @kottwitz:tikz]`	[Knu86; Kot23]
`@knuth:tex says`	Knuth [Knu86] says

The citation style is a biblatex style and is set with `biblio-style` in `_quarto.yml`. This template uses `alphabetic`, which is common in computer science and mathematics; `numeric`, `authoryear` or `ieee` are other choices. Clarify with your supervisor

which style to use. A page number goes after a comma: `[@knuth:tex, p. 33]`; biblatex then typesets "p." itself.

Every citation key must exist in one of the bibliography files. Before each render, the script that prepares CryptoBib (Section 4.2.1) also lists keys that it cannot find, with file and line; LaTeX then shows the key in bold instead of a reference.

4.2.1 Cryptography: CryptoBib

For papers on cryptography, you do not need to type the entries yourself. CryptoBib contains BibTeX entries for all papers of the IACR conferences and journals, many other security conferences and the Cryptology ePrint Archive, with consistent keys. The key consists of the venue, the first letters of the authors' names and the year, e.g. `C:BonFra01` for the identity-based encryption scheme of Boneh and Franklin at CRYPTO 2001 and `EPRINT:BonFra01` for its full version on ePrint. You cite them like any other entry: `[@C:BonFra01]`. You can find the key of a paper with the search on the CryptoBib website.

Run `make cryptobib` once to add CryptoBib to your repository as a git submodule in `bib/cryptobib/`. As the database is very large (more than 90 000 entries), the script `scripts/cryptobib-subset.ts` copies only the entries you actually cite into `bib/cryptobib.bib` every time you render. You never have to edit this file. To get the latest version of CryptoBib, run `make cryptobib-update`.

4.3 Bibliography

The structure of the mandatory bibliography is explained below.

4.3.1 Content and Order of the Bibliography

In the bibliography, all publications used are listed in alphabetical order by author's name. The bibliography must include all sources cited in the text, but no additional sources. If works are published not only in printed form but also electronically, the bibliography should follow the printed version, unless the digital document has a fixed document number (DOI). If a letter abbreviation is used in the references in the text section instead of the title, this should also be indicated in the bibliography.

Several works published by one author within one year must be differentiated. For the short titles in the text, the years must therefore be distinguished by appending

lower case letters. However, the year in the source citation remains unchanged without appended letters. The order of a, b, c etc. depends on the order of the source references. Whether first names are abbreviated or written out in full depends on the conventions of your subject. However, please remain consistent.

> **In this template**
>
> biblatex lists exactly the cited sources, sorted alphabetically, and adds the letters itself. The one exception is the reading list at the end of this template: it comes from `nocite` in `_quarto.yml`, which adds uncited entries on purpose. Remove `nocite` and the note in `references.qmd` for your thesis.

4.3.2 References in the Bibliography

The order in the notation of the bibliography depends on the document type and subject-specific conventions. It is therefore different for monographs than for journal articles, and again different in chemistry than in the humanities or mathematics. There are also no uniform standards for punctuation.

The data for a bibliography can be found on the title page of a book. This is not the book cover, but a printed sheet at the beginning of the book with the most important information identifying the work and book. First, the author or editor is named together with the title, followed by publication and printing information such as the publisher, place and year of publication. Although there is a DIN rule for title lettering, it is interpreted very differently.

In the e-learning courses of the Writing Center[2] you will find further information on the bibliographic information for different types of publications. The following bibliography is just one example of a possible format.

[2] https://ilu.th-koeln.de/goto.php?target=cat_52109&client_id=thkilu

Bibliography

As explained in Section 4.3.1, only the sources that are actually referenced in a paper are listed in the bibliography. Therefore, please always check the bibliography of your paper to ensure that all sources cited – and only these – have been included. This does not apply to the following list; most of the sources listed here are not cited in this template (see `nocite` in `_quarto.yml`). This is merely an example of a bibliography with recommended reading for academic writing.

[Ahr18] Volker Ahrens. *Abschlussarbeiten richtig gliedern in Naturwissenschaften, Technik und Wirtschaft*. 2nd edition. Zürich: vdf, 2018.

[Ber10] Doris Berger. *Wissenschaftliche Arbeiten in den Wirtschafts- und Sozialwissenschaften. Hilfreiche Tipps und praktische Beispiele*. Wiesbaden: Gabler, 2010.

[BH21] Georg Bruhn and Gertrude Hirsch Hadorn. *Textanalyse in den Wissenschaften. Inhalte und Argumente analysieren und verstehen*. 4th, revised and updated edition. Zürich: vdf Hochschulverlag, 2021.

[DSN14] Uhrs Dahinden, Sabrina Sturzenegger, and Alessia Neuroni. *Wissenschaftliches Arbeiten in der Kommunikationswissenschaft*. 2nd, corrected and updated edition. Bern: Haupt, 2014.

[EB09] Hans Ebel and Claus Bliefert. *Bachelor-, Master- und Doktorarbeit. Anleitungen für den naturwissenschaftlich-technischen Nachwuchs*. 4th edition. Weinheim: Wiley-VCH, 2009.

[Eco10] Umberto Eco. *Wie man eine wissenschaftliche Abschlußarbeit schreibt*. 13th edition. Wien: facultas, 2010.

[Ess17] Helga Esselborn-Krumbiegel. *Von der Idee zum Text. Eine Anleitung zum wissenschaftlichen Schreiben*. 5th, updated edition. Paderborn: Schöningh, 2017.

[FHL07] Andrea Frank, Stefanie Haacke, and Swantje Lahm. *Schlüsselkompetenzen: Schreiben in Studium und Beruf*. Studtgart, Weimar: Metzler, 2007.

[FHS17] Melanie Fröhlich, Christiane Henkel, and Anna Surmann. *Zusammen schreibt man weniger allein. (Gruppen-)Schreibprojekte gemeinsam meistern*. Opladen, Toronto: Budrich, 2017.

[Göt02] Karl-Heinz Göttert. *Kleine Schreibschule für Studierende*. 2nd edition. München: Fink, 2002.

[HS16] Andreas Hirsch-Weber and Stefan Scherer. *Kleine Schreibschule für Studierende*. Stuttgart: Eugen Ulmer, 2016.

[KLR18] Ulrike Kipman, Ulrike Leopold-Wildburger, and Thomas Reiter. *Wissenschaftliches Arbeiten 4.0. Vortragen und Verfassen leicht gemacht*. 3rd edition. Berlin: Springer, 2018.

[Knu86] Donald E. Knuth. *The TEX Book*. Addison-Wesley Professional, 1986.

[Kot23] Stefan Kottwitz. *LaTeX Graphics with TikZ*. Packt, 2023.

[KR19] Matthias Karmasin and Rainer Ribing. *Die Gestaltung wissenschaftlicher Arbeiten. Ein Leitfaden für Facharbeit/VWA, Seminararbeiten, Bachelor-, Master-, Magister- und Diplomarbeiten sowie Dissertationen*. 10th, revised and updated edition. Wien: facultas, 2019.

[Kre18] Bruno P. Kremer. *Vom Referat bis zur Abschlussarbeit. Naturwissenschaftliche Texte perfekt produzieren, präsentieren und publizieren*. 5th, updated and expanded edition. Berlin: Springer Spektrum, 2018.

[Kru18] Otto Kruse. *Lesen und Schreiben. Der richtige Umgang mit Texten im Studium*. 3rd, revised and expanded edition. Konstanz, München: UKV/Lucius, 2018.

[Küh21] Stefan Kühtz. *Wissenschaftlich formulieren. Tipps und Textbausteine für Studium und Schule*. 6th, updated and expanded edition. Paderborn: Schöningh, 2021.

[Lan18] Ulrike Lange. *Fachtexte lesen, verstehen, wiedergeben*. 2nd, revised edition. Paderborn: Schöningh, 2018.

[ND19] Jürg Niederhauser and Dudenredaktion. *Die schriftliche Arbeit. Für Schule, Hochschule und Universität*. 3rd edition. Berlin: Dudenverlag, 2019.

[Nor19] Frank Norbert. *Handbuch Wissenschaftliches Schreiben. Eine Anleitung von A bis Z*. Paderborn: Schöningh, 2019.

[OST14] Monika Oertner, Ilona St. John, and Gabriele Thelen. *Wissenschaftlich Schreiben. Ein Praxisbuch für Schreibtrainer und Studierende*. Paderborn: Schöningh, 2014.

[Pre19] Lydia Prexl. *Mit digitalen Quellen arbeiten. Richtig zitieren aus Datenbanken, E- Books, YouTube und Co*. 3rd updated and revised edition. Paderborn: Schöningh, 2019.

[Pre21] Lydia Prexl. *Alles, was Ingenieur:innen über Deutsch wissen müssen*. München: UKV, 2021.

[Sch11] Kirsten Schindler. *Klausur, Protokoll, Essay. Kleine Texte optimal verfassen*. 3rd, revised edition. Paderborn u. a.: Schöning, 2011.

[Sch16] Ulrike Scheuermann. *Schreibdenken. Schreiben als Denk- und Lernwerkzeug nutzen und vermitteln*. 3rd, revised edition. Opladen, Toronto: Budrich, 2016.

[Sch20] Jan Schaller. *Papierlos studieren. Wissenschaftlich arbeiten in digitalen Zeiten*. Opladen, Toronto: Budrich, 2020.

[Sti15] Mona Stier. "Jenseits der Hausarbeit: Die vielen kleinen Textsorten des Hochschulalltags." In: *JoSch. Journal der Schreibberatung* 9 (2015), pp. 82–87.

[Vos22] Rödiger Voss. *Wissenschaftliches Arbeiten … leicht verständlich*. 8th, revised and expanded edition. München: UKV, 2022.

[Wol16] Judith Wolfsberger. *Frei geschrieben. Mut, Freiheit & Strategie für wissenschaftliche Abschlussarbeiten*. 4th, edited edition. Wien, Köln, Weimer: Böhlau Verlag, 2016.

[Wym21] Christian Wymann. *Der Schreibzeitplan: Zeitmanagement für Schreibende*. 2nd, revised edition. Opladen, Berlin, Toronto: Budrich, 2021.

A Supplementary Material

Appendices contain material that would interrupt the flow of the main text, e.g. detailed proofs, additional measurements, questionnaires or longer listings. They are listed in `_quarto.yml` under `book: appendices:` and are numbered with capital letters.

The declaration of authorship follows on the next page. It is generated from `_quarto.yml` (`thesis: declaration:`). In some degree programs, the declaration must be placed directly after the cover page instead; set `declaration: after-cover` in this case.

Declaration

I declare that I have written this thesis independently. I have acknowledged all passages taken verbatim or in spirit from published or unpublished works of others or from the author himself/herself. All sources and aids that I have used for the thesis are indicated. The thesis has not been submitted to any other examination authority with the same content or in essential parts.

Place, date

Signature