

Universität Stuttgart

Thesis Title

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik der Universität Stuttgart zur Erlangung der Würde eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

Author Name

aus Geburtsort

Hauptberichter:	Prof. Dr. Eike Kraft
Mitberichter:	Prof. Dr. Luca Wind
Tag der mündlichen Prüfung:	unbekannt

Institut für Architektur von Anwendungssystemen

2026

Kurzfassung

Kurzfassung der Arbeit.

Abstract

Write an abstract for your work. Replace each of the points below with one sentence (two if you must) and you have your abstract. Write it when you finished your entire report.[1]

Introduction. In one sentence, what's the topic? Phrase it in a way that your reader will understand. If you're writing a PhD thesis, your readers are the examiners – assume they are familiar with the general field of research, so you need to tell them specifically what topic your thesis addresses. Same advice works for scientific papers – the readers are the peer reviewers, and later others in your field interested in your research, so again they know the background work, but want to know specifically what topic your paper covers.

State the problem you tackle. What's the key research question? Again, in one sentence. (Note: For a more general essay, I'd adjust this slightly to state the central question that you want to address) Remember, your first sentence introduced the overall topic, so now you can build on that, and focus on one key question within that topic. If you can't summarize your thesis/paper/essay in one key question, then you don't yet understand what you're trying to write about. Keep working at this step until you have a single, concise (and understandable) question.

Summarize (in one sentence) why nobody else has adequately answered the research question yet. For a PhD thesis, you'll have an entire chapter, covering the prior work in the literature. Here you have to boil that down to one sentence. But remember, the trick is not to try and cover all the ways in which people have tried and failed; the trick is to explain that there's this one particular approach that nobody else tried yet (hint: it's the thing that your research does). But

[1] https://www.easterbrook.ca/steve/2010/01/how-to-write-a-scientific-abstract-in-six-easy-steps/

here you're phrasing it in such a way that it's clear it's a gap in the literature. So use a phrase such as "previous work has failed to address…". (if you're writing a more general essay, you still need to summarize the source material you're drawing on, so you can pull the same trick – explain in one sentence what the general message in the source material is, but expressed in terms of what's missing)

Explain, in one sentence, how you tackled the research question. What's your big new idea? (Again for a more general essay, you might want to adapt this slightly: what's the new perspective you have adopted? Or, what's your overall view on the question you introduced in step 2?)

In one sentence, how did you go about doing the research that follows from your big idea. Did you run experiments? Build a piece of software? Carry out case studies? This tends to be the longest sentence, above all in a PhD thesis, which covers years of research. But don't overdo it – we're still looking for a sentence that you could read aloud without having to stop for breath. Remember, the word 'abstract' means a summary of the main ideas with most of the detail left out. So feel free to omit detail! (For those of you who got this far and are still insisting on writing an essay rather than signing up for a PhD, this sentence elaborates on sentence 4 – explore the consequences of your new perspective).

As a single sentence, what's the key impact of your research? Here we're not looking for the outcome of an experiment. We're looking for a summary of the implications. What's it all mean? Why should other people care? What can they do with your research. (Essay folks: all the same questions apply: what conclusions did you draw, and why would anyone care about them?)

Acknowledgments

I would like to thank the little green men from Mars. Without their probing equipment, the medical experiments on the monkeys described in this thesis would have been considerably more unpleasant for all parties involved.

Contents

1 Introduction 13
1.1 Context and Topic (SM1) 15
1.2 State of the Art (SM1) 15
1.3 Research Question (SM2) 16
1.4 Method or Approach (SM3, SM4) 17
1.5 Findings (SM5, SM6) 17

2 Background (SM1) 19

3 Problem Exposition (optional) 21
3.1 Context/Business Understanding (SM1) 21
3.2 Data Understanding (SM1) 21
3.3 Detailed Research Questions (SM2) 21
3.4 Detailed Method (SM3) 22

4 First Real Chapter addressing first Research Problem 23
4.1 First Sub-Problem 24
4.2 Second Sub-Problem 25

5 Second Real Chapter 27

6 Evaluation 29
6.1 Objective (SM2) 29
6.2 Setup (SM3) 29
6.3 Execution (SM4) 29
6.4 Results (SM5) 30
6.5 Discussion (SM6) 30
6.6 Threats to Validity 30

7 Writing in Markdown 33
7.1 Text . . . 33
7.2 Lists . . . 33
7.3 Tables and references . . . 34

8 LaTeX Hints 35
8.1 Handling of paragraphs . . . 35
8.2 Notes separated from the text . . . 37
8.3 Handling TODOs . . . 37
8.4 Hyphenation . . . 40
8.5 Typesetting Units . . . 40
8.6 Surrounding Text by Quotes . . . 41
8.7 Zref-clever examples . . . 41
8.8 Formulas and Equations . . . 42
8.9 Figures . . . 44
8.10 Sub Figures . . . 46
8.11 Figures with tikz . . . 47
8.12 Plots with pgfplots . . . 48
8.13 UML diagrams . . . 50
8.14 Tables . . . 54
8.15 Tables with tabularray . . . 58
8.16 Tables spanning multiple pages . . . 59
8.17 Source Code . . . 62
8.18 Itemization . . . 66
8.19 Writing in Markdown . . . 67
8.20 Abbreviations . . . 69
8.21 Other Features . . . 70
8.22 Zref-vario examples . . . 72
8.23 Citations . . . 72
8.24 Miscellaneous Examples . . . 73
8.24.1 Algorithms . . . 75
8.24.2 Definitions . . . 75
8.24.3 Enumerations . . . 75
8.24.4 The fquote environment . . . 77

9 Conclusion and Outlook 79

Bibliography 81

Appendix 82

A My first appendix 83

B Use of AI Tools 85

List of Figures 87

List of Tables 89

List of Listings 91

List of Definitions 93

List of Abbreviations 95

Chapter 1

Introduction

[TODO 1: The chapter structure of this template is only an example. Agree on the actual structure of your thesis with your supervisor.]

Purpose and scope of your entire report. The purpose of your entire report is to make a *scientific argument using the scientific method*. A scientific argument always has the following steps that all must come in this order.

SM1 *Explicate the assumptions and state of the art* on which you are going to conduct your research to investigate your research problem/test the hypothesis.

SM2 Clearly and precisely *formulate a research problem or hypothesis*.

SM3 *Describe the (research) method* that you followed to investigate the problem / to test the hypothesis in a way that *allows someone else to reproduce your steps*. The method must include steps and criteria for evaluating whether you answered your question successfully or not.

SM4 *Provide execution details* on how you followed the method in the given, specific situation.

SM5 *Report your results* by describing and summarizing your measurements. You must not interpret your results.

SM6 *Now interpret your results* by contextualizing the measurements and drawing conclusion that lead to answering your research problem or defining further follow-up research problems.

This template marks the parts of the structure with SM1-SM6 to recall to you which step of a scientific argument is used and where.

Purpose and scope of Chapter 1. The introduction chapter is a summary of your work and your scientific argument that must be understandable to anyone in your scientific field, e.g., anyone in Data Science. A reader must be able to comprehend the problem, method, relevant execution details, results, and their interpretation by reading the introduction and the introduction alone. Section 1.1 introduces the general topic of your research. Section 1.2 discusses the state of the art and identifies a research gap. Section 1.3 then states the research problem to investigate. Section 3.4 explains the research method that was followed, possibly with execution details. Section 1.5 then presents the results and their interpretation. A reader should need to read further only if they are not convinced or need more details to reproduce your study. The individual chapters and sections provide the details for each of the steps in your scientific argument.

Write the introduction chapter *after* you wrote all other chapters, but you should keep on making notes for each of the sections as you write the later chapters.

Purpose and scope of the introduction paragraph to a chapter. The paragraph you are reading above is a typical introductory paragraph to a chapter. It is a high-level summary of the chapters' topic (SM1 and SM2). It gives the reader some guidance by breaking down the chapter topic into subtopics that are named (SM3) in the right order with forward references to the corresponding sections (SM4). It may close with announcing the result you obtain (SM6) but the opening paragraph of the introduction is not the place for it.

Two kinds of readers. Write your report so that it can be read from beginning to end: introduce every concept before you use it. Backward references are fine; forward references in the running text should be rare. At the same time, guide the reader who opens your report somewhere and has ten minutes: each chapter starts with an introductory paragraph as described above, which outlines the coming sections and links back to what the chapter builds on and forward to where its results are used. The book "Student Thesis Projects" [LLDS26] (open access) explains in detail how to organize, conduct, write, and present a thesis.

Before you hand in, work through the checklist in `docs/thesis-checklist.md` and remove all these hints.

1.1 Context and Topic (SM1)

Purpose and scope. You begin with providing the general scientific audience an introduction into the specific topic of your work. The aim of this section is to first introduce the *general subject of study* ("Giraffes are well-known animals and everyone's favorite"), the *specific topic of societal or scientific interest* to investigate ("Giraffes have blue tongues") and the *objective of society/science towards* this topic ("it is unknown at the moment how the blue color tone evolved"). It must be understandable by the general scientific public. Every *term* with a specific meaning must be highlighted and introduced in precise language/concepts that only builds on a general scientific background.

At the end of this section, you have explained and established a general goal that society/science universally agrees to be worth achieving ("knowing how everyone's favorite animal evolved the color of their tongue").

1.2 State of the Art (SM1)

Purpose and scope. You provide a more in-depth introduction into the research topic by contrasting the current state of the art in society/science in relation to the research topic you introduced in Section 1.1. This introduction has to

- present established facts, methods, and results that provide a deeper understanding of the research topic ("prior work on giraffe genomes, relevance of giraffes for societal well-being, giraffes being a model-animal for various other studies, etc.")
- discuss in which ways prior and recent ideas still fall short of reaching the general goal you explained in Section 1.1 ("prior work only sequenced the genome of one giraffe and did not consider genes of ancient ancestors")

You have to provide citations/literature references for each of the statements

and claims you are making. This section summarizes the related work discussion in Chapter 2.

At the end of this section, you have established a *knowledge gap* between the state of the art and the general objective you developed in Section 1.1. *Stating a (knowledge) gap between a status quo and a desired situation is the* first step *of a writing scientific argument.*

1.3 Research Question (SM2)

Purpose and scope. In this section you state in which way you will address the knowledge gap you identified at the end of Section 1.2. One work cannot address and resolve the entire knowledge gap. The purpose of this section is to detail the specific part of the knowledge that you will address. You thereby make all the assumptions explicit that underlie your work ("in this report we focus on genomes of female giraffes who lived in the years 1950–2000 in South Africa").

Your general research question states

- The starting point/assumptions you are making from which your research starts ("for the given 13 genomes of female giraffes…"), and
- the final objective/solution you want to reach ("…identify the genes involved in color expression of giraffe tongues…")
- and the evaluation criteria that will determine whether you are successful ("…that are present in at least 75% of the studied giraffes")

Break your general research question down into sub-research questions. You may do this here. The sub-research questions have to form a chain that take you in smaller steps from the starting point/assumptions of your general research question to your final objective and evaluation.

Number the research questions to reference them by their identifier, such as RQ1:

Research Question 1 (RQ1): *How can a program detect that another program modified a file it saves?*

Research Question 2 (RQ2): *How can users resolve conflicting modifications of a file?*

To answer RQ2, RQ1 ("How can a program detect that another program modified a file it saves?") has to be answered first.

1.4 Method or Approach (SM3, SM4)

Purpose and scope. In this section you outline the method that you applied to answer the research questions, or the new technical approach that you developed to answer it. It is a summary of the steps that someone else has to take in order to reproduce your steps. Mention here the data sets you had to obtain/gather/analyze, interviews with stakeholders you had to make to further develop the research questions, technical artifacts (programs, algorithms, models) you could apply or that you had to develop (and how they work).

The section is most readable if you give each of the steps in your method its own paragraph. In each paragraph you first briefly explain the concept of the step in your method (SM3, "we explored the data through visual analytics") and then provide details in execution (SM4, "we used tool X, we developed dashboard Y") include a forward reference to the respective chapter that provides more details.

1.5 Findings (SM5, SM6)

Purpose and scope. You close the introduction by stating the evaluation setup you designed to evaluate the success of your study regarding the research objective, which comes in two steps. It summarizes your evaluation in Chapter 6.

Results (SM5)

You state the evaluation method that is in line with your research question from Section 1.3 and summarize the measurements you obtained, but you do not interpret them, i.e., you only report the numbers, but you do not include judging statements.

Interpretation (SM6)

You summarize your interpretation of the results and draw conclusions. State whether, and to which degree, the research question from Section 1.3 has been answered.

Close by stating how much closer society and science have come in answering the general objective you outlined in Section 1.1.

Contribution. Make your own contribution explicit in one paragraph ("The contribution of this thesis is …"): the artifact, method, or insight that did not exist before, and what the reader can do with it.

Chapter 2

Background (SM1)

Purpose and scope. The background chapter has multiple roles.

- *Preliminaries.* It has to provide all (and exactly the) information that is necessary to understand the methodological and technical parts of your work in the specific area of study. Assume as starting point another student in your degree who did not study the specific subject you are studying but has the task to understand your work. Which concepts, terms, definitions, etc. does the student have to know? Which formulas, symbols, etc. are standard in this topic? Only introduce definitions if you actually need them in any of the subsequent chapters.
- *Related Work.* It has to provide a comprehensive discussion of all prior work in the area on this subject. Your discussion has to summarize these prior works and has to explain in which way the research question you are solving (Section 1.3) has not been solved yet because prior work had more limiting assumptions, addressed a different angle, their results are not complete etc. Depending on the subject you are studying, the related work part can be larger and warrant an entire chapter on its own, or be fully concluded within Section 1.2.

  You can close the related work discussion by clarifying the positioning and formulation of your research question (SM2) in relation to all the prior work, making more explicit whether you address an existing research question under different premises or whether you work on a modified or completely new research question. While writing, mark a claim whose



source you still have to look up with `\CITE`, for example: giraffes sleep less than two hours a day [Source 1: Study on the sleep of giraffes].

Chapter 3

Problem Exposition (optional)

Purpose and scope. Introduce the problem context in more detail if Section 1.1 does not provide all necessary information about the problem to follow the rest of the report. This can include further details on the data you studied, context assumptions and requirements, etc.

If you have to expose the problem in more detail here, then this chapter should also provide a more detailed explanation of research question and the method you are applying, i.e., you can now provide more concrete sub-problems compared to Section 1.3 more details for the method Section 1.4 because you now have explained the problem much better. A typical structure can be.

3.1 Context/Business Understanding (SM1)

provide details

3.2 Data Understanding (SM1)

provide details

3.3 Detailed Research Questions (SM2)

provide details based on Sections 3.1 and 3.2

3.4 Detailed Method (SM3)

provide details based on Sections 3.1 and 3.2

Chapter 4

First Real Chapter addressing first Research Problem

Purpose and scope. After you stated research context (SM1), research problem (SM2), and research method (SM3) in Chapter 1 and possibly Chapter 3, the remainder of your entire report addresses execution (SM4), results (SM5), and interpretation (SM6). Do this by addressing the sub-problems through scientific arguments that follow the 6 steps SM1-SM6.

Have a short chapter introduction that recalls and explains the first research problem of your thesis. The problem has to show up in the introduction in Section 1.3 or in Section 3.3 already. This provides the background (SM1) for this chapter while the first research problem of the thesis becomes the research question/hypothesis (SM2) for this chapter.

Next, explain in the chapter intro how you solve the research problem in this chapter by breaking it down in further sub-problems. By this, you outline the method (SM3) through which you are going to solve the problem of this chapter. This is necessary to give the reader guidance of what's to come in this chapter and how it fits into the thesis as a whole. Explain that you will address the first sub-problem in Section 4.1 and the second sub-problem in Section 4.2, etc. The sections then provide the details for execution and results.

4.1 First Sub-Problem

The first paragraph describes the first sub-problem and develops the requirements a solution has to satisfy (SM2 for this section). The requirements have to be based on the knowledge and reasoning developing in the preceding chapters and sections. Try to use an example to illustrate the problem and the desired properties of the solution. Check that every term/concept you use here has already been defined already in a previous section. If you cannot describe your problem without defining new terms, you may have to add another section before this one that develops the terms and concepts you need to explain the problem.

The second paragraph describes the method/approach how you address the problem (SM3 for this section). Describe the method in a level of detail that allows another student to reproduce your steps. Make use of appendices if certain details take too much space.

The third, fourth, and following paragraph provides details on applying the method or developing a new approach, i.e., execution (SM4) and may explain results (SM5), i.e., details on the steps needed to reproduce the results.

Results (SM5) come in different forms, e.g., conceptual diagrams, algorithms, tables, charts, a list of articles from a literature research etc. You must reference them (e.g., "Figure 4.1 shows the runtime") and describe the results in text. If you use diagrams, tables, or charts, you cannot expect the reader to know what to you expect them to see in a diagram, table or chart. Describe to them how to read these, explain the meaning of particular elements, point out special observations. Describe the results, but do not interpret them. Make use of appendices if certain details take too much space.

After describing the results, you may interpret them (SM6). Here you can infer what a particular observation means (for you), how it can be applied, or what others can do with it. Do not interpret your results before you have described all of them. This is a common mistake done by most beginner writers. Get to the point: the final finding or interpretation. But you forget that your reader does not understand yet what you are interpreting – they do not know yet what you do know. Interpret the results only after describing all of them. The interpretation must be based on the written description only. Then you can

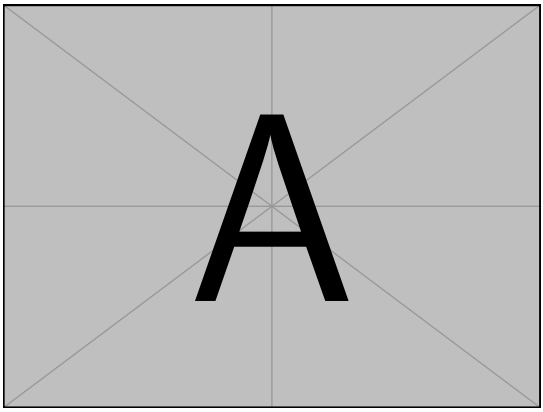


Figure 4.1: A scientific figure that has to be explained in the text

be sure that your readers can follow your interpretation and reach the same conclusions as you have.

Ideally, your interpretation leads to the next sub-problem in Section 4.2.

4.2 Second Sub-Problem

You now build on the solution to the first sub-problem of Section 4.1 (SM1) and recall second sub-problem (SM2, you detailed in the introduction of this chapter) and follow the same pattern as before (SM3-SM6).

Note that not all sections may not include all parts SM1-SM6 in all detail. Some sections need not restate the background (SM1) or the research problem (SM2) if a previous section already defined them. Sometimes, a section is only dedicated to describing the method (SM3) and execution (SM4) and does not contain any results or interpretations. Sometimes results (SM5) and interpretations (SM6) only come in the evaluation chapter.

What is important for you when you are writing a scientific argument is not to slavishly have SM1-SM6 in each section explicitly, but that you are always fully aware of the following:

- Which step of a scientific argument am I currently writing (SM1, SM2, ..., SM6)?
- Does the step that I am writing come in the right order, i.e., if you are writing about execution (SM4, e.g., details of building a model), is there a preceding paragraph or section

that describes the method (SM3) and is that one preceded by a clear statement of the (sub-)problem addressed (SM2)?

- Are you really *not* writing interpretation SM6 before SM5, SM4, or SM3?
- Is it clear to the reader which part of the scientific argument you are currently making?

Chapter 5

Second Real Chapter

Have a short chapter introduction that recalls what you already achieved in Chapter 4 and explain the second research problem of your thesis. The problem has to show up in the introduction in Section 1.3 or in Section 3.3 already. Continue likewise for further research problems.

Chapter 6

Evaluation

Purpose and scope. The evaluation chapter should be the most formal and rigorously structured chapter of your thesis as the validity of your evaluation argument depends on it.

6.1 Objective (SM2)

State what you want to evaluate and what you want to measure. The Goal Question Metric approach [BCR94] derives the measurements from the goal: state the goal, ask the questions that tell whether it is reached, and define a metric that answers each question.

6.2 Setup (SM3)

State which data, participants, tools, etc. you chose and why. State how you measure outcomes and how you compare them to baselines, reference groups, etc.

6.3 Execution (SM4)

Provide all details on the execution that are necessary to allows another person to reproduce your results at a later point.

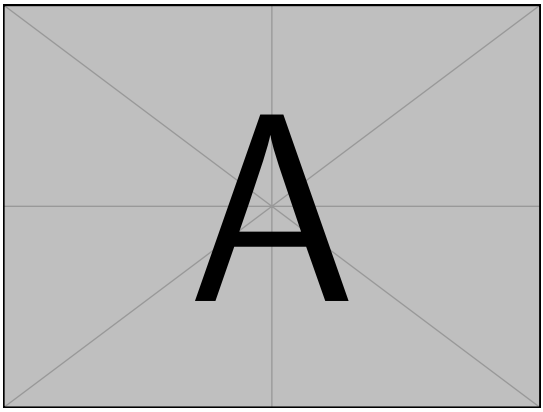


Figure 6.1: Another scientific figure that has to be explained in the text

6.4 Results (SM5)

You only report the measurements. You must present and reference them (e.g., "Figure 6.1 shows the runtime") and describe the results in text. If you use diagrams, tables, or charts, you cannot expect the reader to know what to you expect them to see in a diagram, table or chart. Describe to them how to read these, explain the meaning of particular elements, point out special observations. Describe the results, but do not interpret them. Make use of appendices if certain details take too much space.

6.5 Discussion (SM6)

Interpret the results only after describing all of them. The interpretation must be based on the written description in Section 6.4 only. Then you can be sure that your readers can follow your interpretation and reach the same conclusions as you have.

6.6 Threats to Validity

Discuss what limits the validity of your results and how you mitigated it. Typical questions are:

Construct validity Do your measurements capture what you intended to evaluate (Section 6.1)?

Internal validity Are the results caused by what you varied, or by other factors?

External validity To which other settings, data, or participants do the results generalize?

Reliability Would another person obtain the same results and conclusions following Section 6.3?

Runeson and Höst [RH09] give a checklist for these threats in case studies in software engineering.

Chapter 7

Writing in Markdown

This chapter uses plain Markdown, not LaTeX. Its source is a Markdown file in the `chapters-en` directory. The main file includes it with the `markdownInput` command of the `markdown` package[2], and LaTeX takes care of the typesetting. This needs LuaLaTeX.

7.1 Text

Text can be *emphasized* or **bold**, and a monospaced font shows `code`. A citation looks like this [Sch17], and here is a footnote.[3]

> A quote is a paragraph starting with a greater-than sign.

7.2 Lists

- first point
- second point
  - sub-point

1. firstly
2. secondly

[2] `https://ctan.org/pkg/markdown`
[3] This is an inline footnote.

Table 7.1: A table written in Markdown

Right	Left	Centered
12	12	12
123	123	123

7.3 Tables and references

Table 7.1 shows a table with caption. Headings and tables get an identifier in curly braces; a reference puts it in angle brackets. For instance, Chapter 8 presents hints on writing LaTeX.

Chapter 8

LaTeX Hints

This chapter contains hints on writing LaTeX. It focuses on minimal examples, which can be directly adapted to the content

8.1 Handling of paragraphs

One sentence per line. This rule is important for the usage of version control systems. A new line is generated with a blank line. As you would do in Word: New paragraphs are generated by pressing enter. In LaTeX, this does not lead to a new paragraph as LaTeX joins subsequent lines. To start a new paragraph, press enter twice. This leads to an empty line. In word, there is the functionality to press shift and enter. This leads to a hard line break. The text starts at the beginning of a new line. In LaTeX, you can do that by using two backslashes (\\).
Running text seldom needs a hard line break.

Please do *not* use two backslashes for new paragraphs. For instance, this sentence belongs to the same paragraph, whereas the last one started a new one. Andrew Stacey gives a long motivation for that in the section “Line Wrapping” of his *Document Revision System*[4].

[4] https://loopspace.mathforge.org/HowDidIDoThat/TeX/VCS/

Corresponding LaTeX code of `80 - latex hints.tex`

```
One sentence per line.
This rule is important for the usage of version control systems.
A new line is generated with a blank line.
As you would do in Word:
New paragraphs are generated by pressing enter.
In LaTeX, this does not lead to a new paragraph as LaTeX joins subsequent lines.
To start a new paragraph, press enter twice.
This leads to an empty line.
In word, there is the functionality to press shift and enter.
This leads to a hard line break.
The text starts at the beginning of a new line.
In LaTeX, you can do that by using two backslashes (\textbackslash\textbackslash).
\\
Running text seldom needs a hard line break.

Please do \textit{not} use two backslashes for new paragraphs.
For instance, this sentence belongs to the same paragraph, whereas the last one started a new one.
Andrew Stacey gives a long motivation for that in the section \enquote{Line Wrapping} of his \href{https://loopspace.mathforge.org/HowDidIDoThat/TeX/VCS/#section.3}{\emph{Document Revision System}}\footnote{\url{https://loopspace.mathforge.org/HowDidIDoThat/TeX/VCS/}}.
```

Once the text is final, single paragraphs can be polished. If the last line of a paragraph holds only a word or two, or is nearly full, let TeX set the paragraph one line shorter or longer: write `\looseness=-1` (or `=1`) as the first thing in that paragraph. TeX resets `\looseness` at the end of every paragraph, so only that one paragraph is affected. With LuaLaTeX, a lone short word as the last line is avoided automatically (package impnattypo).

8.2 Notes separated from the text

The package mindflow enables writing down notes and annotations in a way so that they are separated from the main text.

This is a small note.

Corresponding LaTeX code of `80 - latex hints.tex`

```
45 \begin{mindflow}
46 This is a small note.
47 \end{mindflow}
```

8.3 Handling TODOs

Marked text.

Corresponding LaTeX code of `80 - latex hints.tex`

```
53 \textmarker{Marked text.}
```

With `\textmarker`, only the text color is changed, since this also works well for individual words.

Marked text. [TODO 2: A comment on it.]



Corresponding LaTeX code of `80 - latex hints.tex`

```
59 \textcomment{Marked text.}{A comment on it.}
```

Manual marking for text that has been changed since the last version.

Corresponding LaTeX code of `80 - latex hints.tex`

```
63 \modified{Manual marking for text that has been changed since the last
   ↪ version.}
```

This is some text. FL1: text adjusted → Changed text.

Corresponding LaTeX code of 80 - latex hints.tex

```
67 This is some text.
68 \change{FL1: text adjusted}{Changed text}.
```

1 Here is a comment[Comment 1: Comment].

Corresponding LaTeX code of 80 - latex hints.tex

```
72 Here is a comment\sidecomment{Comment}.
```

3 [Todo 3: A lot of text still needs to be produced here]

Corresponding LaTeX code of 80 - latex hints.tex

```
76 \todo{A lot of text still needs to be produced here}
```

\TODO, \FIXME, and \CITE work with every package for notes the template offers: \TODO marks an open task, \FIXME an important one, and \CITE a claim that still needs a citation.

4 [Todo 4: Add the evaluation]

1 [Fixme 1: The numbers are outdated]

2 Most scientific texts in computer science are written with LaTeX [Source 2: Survey on the tools used for writing].

Corresponding LaTeX code of 80 - latex hints.tex

```
82 \TODO{Add the evaluation}
83
84 \FIXME{The numbers are outdated}
85
86 Most scientific texts in computer science are written with
   ↪ \LaTeX{}~\CITE{Survey on the tools used for writing}.
```

With zebra, \todo takes the responsible person as optional argument; further note types are \note, \comment, \fixed, \placeholder, \fixme, and \source

(defined by the template, used by `\FIXME` and `\CITE`). All notes are listed at the end of the document.

[Todo 5@FA: Add a citation]
[Note 1: Shorten this paragraph?]
[Comment 2@FA: Is this claim backed by the evaluation?]
[Fixed 1: Typo corrected]
[Placeholder 1: Table with the results]



Corresponding LaTeX code of `80 - latex hints.tex`

```
93  \todo[FA]{Add a citation}
94
95  \note{Shorten this paragraph?}
96
97  \comment[FA]{Is this claim backed by the evaluation?}
98
99  \fixed{Typo corrected}
100
101 \placeholder{Table with the results}
```

Co-authors can mark their edits and comments with the phfcc package; the initials of the author appear once per page in the margin. Define one command per author in the preamble (`\phfMakeCommentingCommand`); `\phfDisableCommentingCommands` hides everything for the final version.

Quantum mechanics developed within one decade thanks to the works of Schrödinger, Heisenberg, and Dirac, who drew inspiration from earlier ideas by Planck and Einstein [Einstein must not be missing here.] ~~This sentence is to be removed.~~ **!!! This is important. !!!** FA

Corresponding LaTeX code of `80 - latex hints.tex`

```
109 Quantum mechanics developed within one decade thanks to the works of
    ↪ Schrödinger, Heisenberg, and Dirac,
110 who drew inspiration from earlier ideas by Planck \FA{and Einstein}
    ↪ \FA[Einstein must not be missing here.]
111 \FA*{This sentence is to be removed.} \FA!{This is important.}
```

8.4 Hyphenation

LaTeX automatically hyphenates words. When using microtype, there should be fewer hyphenations than in other settings. You might still need to tweak the hyphenation. Here are some hints:

In case you write “application-specific”, then the word will only be hyphenated at the dash. You can also insert `\allowbreak{}` into the word, as in applica tion-specific, but this is much more effort.

You can now write words containing hyphens which are hyphenated at other places in the word. For that, write "= instead of -, as in application”=specific. This is enabled by an additional configuration of the babel package.

Corresponding LaTeX code of `80 - latex hints.tex`

```
122 In case you write \enquote{application-specific}, then the word will
    ↪ only be hyphenated at the dash.
123 You can also insert \verb1\allowbreak{}1 into the word, as in
    ↪ applica\allowbreak{}tion-specific, but this is much more effort.
124
125 You can now write words containing hyphens which are hyphenated at
    ↪ other places in the word.
126 For that, write \verb1"=1 instead of \verb1-1, as in
    ↪ application"=specific. % chktex 18
127 This is enabled by an additional configuration of the babel package.
```

8.5 Typesetting Units

Numbers can be written plain text (such as 100), by using the siunitx package as follows: 100 $\frac{\mathrm{km}}{\mathrm{h}}$, or by using plain LaTeX (and math mode): $100 \frac{\mathit{km}}{h}$.

Corresponding LaTeX code of `80 - latex hints.tex`

```
133 Numbers can be written plain text (such as 100), by using the
    ↪ \href{https://ctan.org/pkg/siunitx}{siunitx} package as follows:
134 \SI{100}{\km\per\hour},
135 or by using plain \LaTeX{} (and math mode):
136 $100 \frac{\mathit{km}}{h}$.
```

5 % of 10 kg

Corresponding LaTeX code of `80 - latex hints.tex`

```
\SI{5}{\percent} of \SI{10}{kg}
```

Numbers are automatically grouped: 123 456.

Corresponding LaTeX code of `80 - latex hints.tex`

```
Numbers are automatically grouped: \num{123456}.
```

8.6 Surrounding Text by Quotes

Please use the “enquote command” to quote something. Quoting with “quote” or “quote” also works.

Corresponding LaTeX code of `80 - latex hints.tex`

```
Please use the \enquote{enquote command} to quote something.
Quoting with "`quote"' or ``quote'' also works. % chktex 18

```

8.7 Zref-clever examples

Demonstration of zref-clever: Zcref (short for zcref[S]) at the beginning of a sentence, zcref in all other cases.

Figure 8.1 shows a simple fact, although Figure 8.1 could also show something else.

Table 8.1: Example table for zcref demo

Heading1	Heading2
One	Two
Three	Four

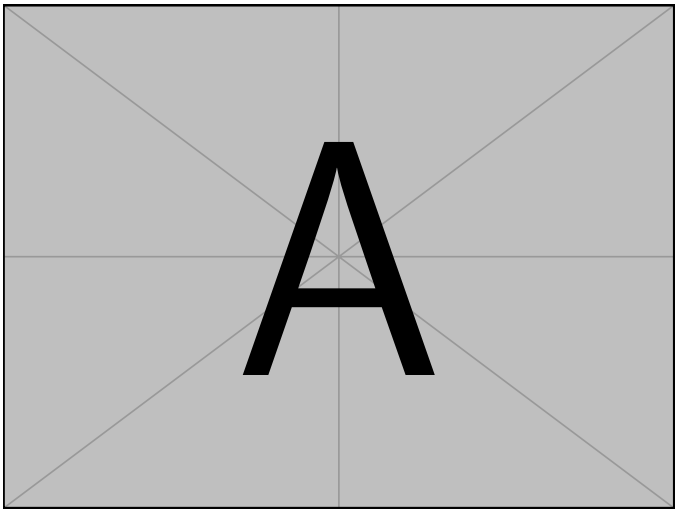


Figure 8.1: Example figure for zcref demo

Table 8.1 shows a simple fact, although Table 8.1 could also show something else.

Section 8.7 shows a simple fact, although Section 8.7 could also show something else.

Corresponding LaTeX code of `80 - latex hints.tex`

```
182 \Zcref{fig:ex:cref} shows a simple fact, although \zcref{fig:ex:cref}
    ↪ could also show something else.
183
184 \Zcref{tab:ex:cref} shows a simple fact, although \zcref{tab:ex:cref}
    ↪ could also show something else.
185
186 \Zcref{sec:ex:cref} shows a simple fact, although \zcref{sec:ex:cref}
    ↪ could also show something else.
```

8.8 Formulas and Equations

Equations $f(x) = x$ inside the text can be provided.

Corresponding LaTeX code of `80 - latex hints.tex`

```
192 Equations $f(x)=x$ inside the text can be provided.
```

A list with all available mathematical symbols is provided at `http://texdoc.net/pkg/symbols-a4`.

As an example, the set of natural numbers is given by $\mathbb{N}$.

Corresponding LaTeX code of `80 - latex hints.tex`

```
198 As an example, the set of natural numbers is given by $\mathbb{N}$.
```

For the documentation of editing mathematical formulas, read the package documentation of `amsmath`[5].

Equation (8.1) is numbered and can be referenced in the text:

$$x = y \tag{8.1}$$

Corresponding LaTeX code of `80 - latex hints.tex`

```
205 \begin{align}
206   \label{eq:test}
207   x = y
208 \end{align}
```

The following equation is not numbered because of using `\align*` as environment.

$$x = y$$

Corresponding LaTeX code of `80 - latex hints.tex`

```
213 \begin{align*}
214   x = y
215 \end{align*}
```

[5] `http://texdoc.net/pkg/amsmath`

The template offers `\abs` to enable the bars to scale well at the absolute value: $|X|$.

Corresponding LaTeX code of `80 - latex hints.tex`

```
220 $\abs{X}$.
```

The documentation available at `http://www.ctan.org/tex-archive/help/Catalogue/entries/voss-mathmode.html` provides more details about mathematical environments.

8.9 Figures

Figure 8.2 shows something interesting.

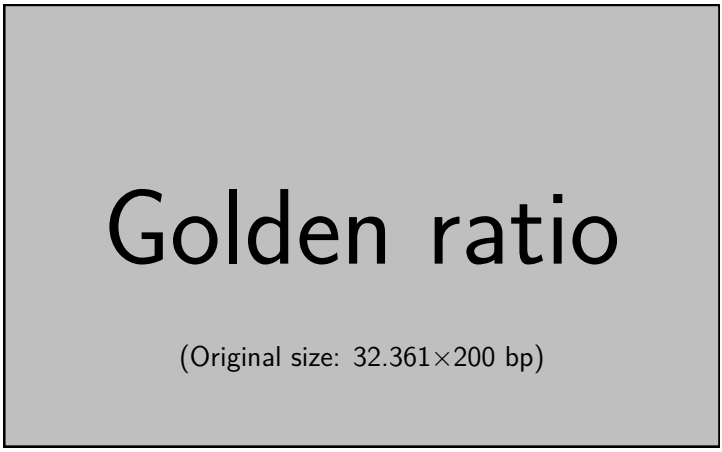


Figure 8.2: Simple Figure. Based on Scharrer [Sch17].



Figure 8.3: Ulm Minster. Photo: wkrauth, CC0, via Wikimedia Commons.

Corresponding LaTeX code of `80 - latex hints.tex`

```
228 \Zcref{fig:label} shows something interesting.
229
230 \begin{figure}
231   \centering
232   \includegraphics[width=.8\linewidth]{example-image-golden}
233   \caption[Simple Figure]{
234     Simple Figure.
235     Based on \citet{mwe}.
236   }
237   \label{fig:label}
238 \end{figure}
```

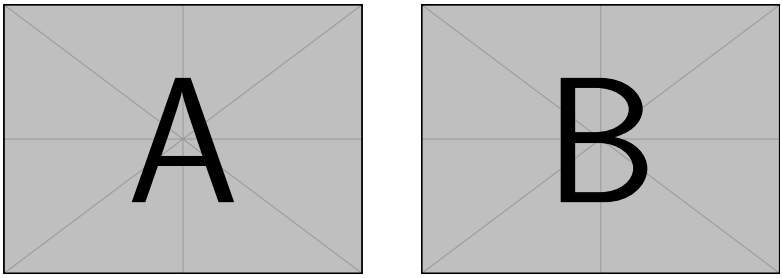
Photos in JPEG or PNG format are included the same way, as shown in Figure 8.3.

Corresponding LaTeX code of `80 - latex hints.tex`

```
244 \begin{figure}
245   \centering
246   \includegraphics[width=.5\linewidth]{figures/ulm-minster.jpg}
247   \caption{Ulm Minster. Photo: wkrauth, CC0, via Wikimedia Commons.}
248   \label{fig:jpg}
249 \end{figure}
```

8.10 Sub Figures

An example of two sub figures is shown in Figure 8.4.



(a) Case I **(b)** Case II

Figure 8.4: Example figure with two sub figures.

Corresponding LaTeX code of `80 - latex hints.tex`

```
\begin{figure}[!b]
  \centering
  \subfloat[Case I]{\includegraphics[width=.4\linewidth]{example-image-a}%
    \label{fig:first_case}}
  \hfil
  \subfloat[Case II]{\includegraphics[width=.4\linewidth]{example-image-b}%
    \label{fig:second_case}}
  \caption{Example figure with two sub figures.}
  \label{fig:two_sub_figures}
\end{figure}
```

8.11 Figures with tikz

The tikz package creates graphics programmatically. This package generates grids and other regular structures.

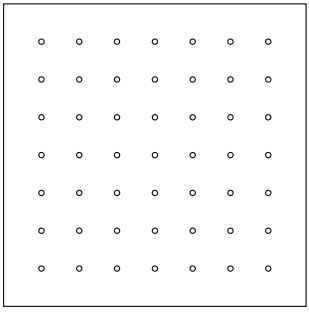


Figure 8.5: A regular grid generated easily with two for loops.

Corresponding LaTeX code of `80 - latex hints.tex`

```
\begin{figure}[ht]
  \centering
  \begin{tikzpicture}
    \draw(0,0) rectangle (4,4);
    \foreach \x in {0.5,1,1.5,2,2.5,3,3.5}
    \foreach \y in {0.5,1,1.5,2,2.5,3,3.5}
    \draw(\x,\y) circle (1pt);
  \end{tikzpicture}
  \caption{A regular grid generated easily with two for
  ↪ loops.}\label{fig:tikz_example}
\end{figure}
```

8.12 Plots with pgfplots

The pgfplots package plots functions directly in LaTeX, similar to MATLAB or gnuplot. Some visual examples are available in the package documentation[6].

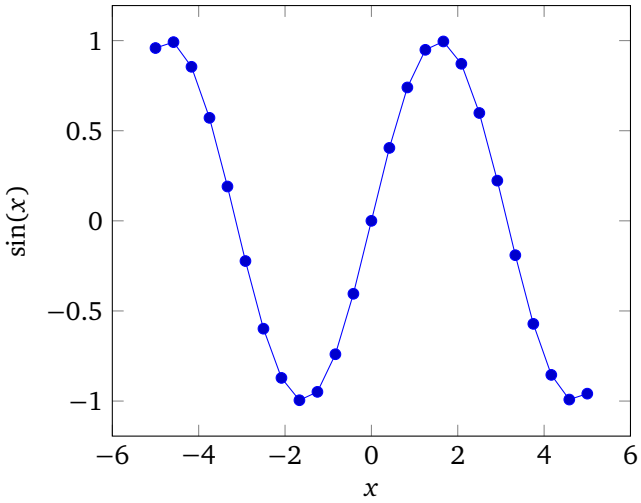


Figure 8.6: Plot of $\sin(x)$ directly inside the figure environment with pgfplots.

[6] `http://texdoc.net/pkg/pgfplots`

Corresponding LaTeX code of `80 - latex hints.tex`

```
\begin{figure}[ht]
  \centering
  \begin{tikzpicture}
    \begin{axis}[xlabel=$x$, ylabel=$\sin(x)$]
      \addplot {sin(deg(x))};  % Plot the sine function
    \end{axis}
  \end{tikzpicture}
  \caption{Plot of $\sin(x)$ directly inside the figure environment
  ↪ with pgfplots.}
  \label{fig:pgfplots_example}
\end{figure}
```

Coordinates can also be given explicitly instead of as a function:

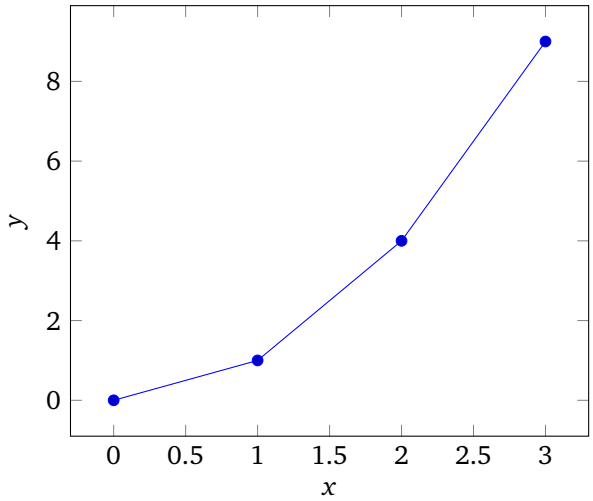


Figure 8.7: Explicit coordinates plotted with pgfplots.

Corresponding LaTeX code of `80 - latex hints.tex`

```
307 \begin{figure}[ht]
308   \centering
309   \begin{tikzpicture}
310     \begin{axis}[xlabel=$x$, ylabel=$y$]
311       \addplot coordinates {(0,0) (1,1) (2,4) (3,9)};  % Plot explicit
          ↪ coordinates
312     \end{axis}
313   \end{tikzpicture}
314   \caption{Explicit coordinates plotted with pgfplots.}
315   \label{fig:pgfplots_coords}
316 \end{figure}
```

8.13 UML diagrams

UML diagrams can be written as text next to the LaTeX source and are typeset in the fonts of the document. MetaUML draws class, component, and further diagrams. It is MetaPost code, which `lualatex` runs itself via the luamplib package; no further tools are needed. Thus, the MetaUML diagrams are shown only when compiling with `lualatex`. The pgf-umlsd package draws sequence diagrams with TikZ and works with every engine. A comparison of the packages for UML diagrams is given at `https://github.com/latextemplates/gadr-latex-packages/blob/main/uml-diagrams.md`.

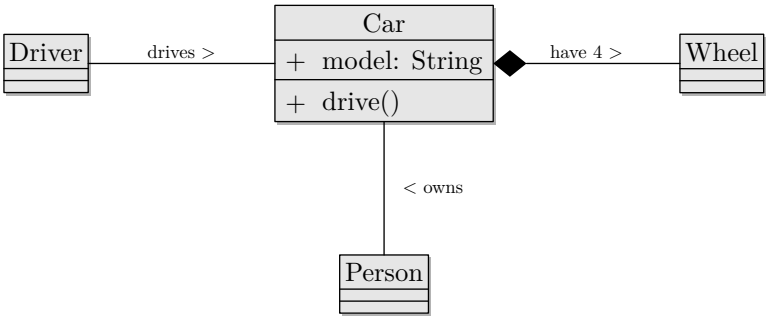


Figure 8.8: Class diagram drawn with MetaUML

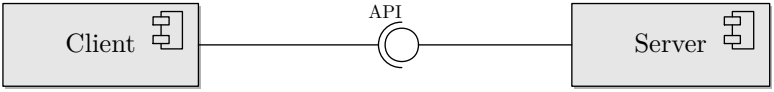


Figure 8.9: Component diagram drawn with MetaUML

Corresponding LaTeX code of `80 - latex hints.tex`

```
330  \begin{figure}
331    \centering
332    \sffamily
333    \begin{mplibcode}
334      input metauml;
335      beginfig(1);
336      Class.Driver("Driver")()();
337      Class.Car("Car")("+model: String")("+drive()");
338      Class.Wheel("Wheel")()();
339      Class.Person("Person")()();
340      % distances between the boxes in PostScript points (1/72 inch)
341      leftToRight(70)(Driver, Car, Wheel);
342      topToBottom(50)(Car, Person);
343      drawObjects(Driver, Car, Wheel, Person);
344      link(association)(Driver.e -- Car.w);
345      item(iAssoc)("drives >")(obj.s = .5[Driver.e, Car.w]);
346      % the diamond is drawn at the end of the path
347      link(composition)(Wheel.w -- Car.e);
348      item(iAssoc)("have 4 >")(obj.s = .5[Car.e, Wheel.w]);
349      link(association)(Car.s -- Person.n);
350      item(iAssoc)("< owns")(obj.w = .5[Car.s, Person.n] + (5, 0));
351      endfig;
352    \end{mplibcode}
353    \caption{Class diagram drawn with MetaUML}
354    \label{fig:uml-class}
355  \end{figure}
```

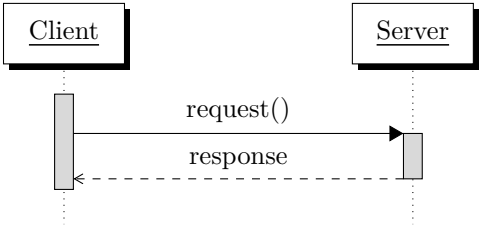


Figure 8.10: Sequence diagram drawn with pgf-umlsd

Corresponding LaTeX code of `80 - latex hints.tex`

```
359  \begin{figure}
360    \centering
361    \sffamily
362    \begin{mplibcode}
363      input metauml;
364      beginfig(1);
365      Component.Client("Client")();
366      Component.Server("Server")();
367      % MetaUML has no automatic layout: each object is placed relative
         ↪ to another one;
368      % here, the left edge of Server is 140 PostScript points (1/72
         ↪ inch, about 5 cm) right of the right edge of Client
369      Server.w = Client.e + (140, 0);
370      drawObjects(Client, Server);
371      % the interface: Server provides it (ball), Client requires it
         ↪ (socket)
372      pair iface;
373      iface = .5[Client.e, Server.w];
374      link(providedInterface)(Server.w -- iface);
375      link(requiredInterface)(Client.e -- iface);
376      item(iAssoc)("API")(obj.s = iface + (0, 8));
377      endfig;
378    \end{mplibcode}
379    \caption{Component diagram drawn with MetaUML}
380    \label{fig:uml-component}
381  \end{figure}
```

Corresponding LaTeX code of `80 - latex hints.tex`

```latex
\begin{figure}
  \centering
  \sffamily
  \begin{sequencediagram}
    \newthread{client}{Client}
    % [3]: horizontal gap (in cm) to the box of the previous instance;
    ↪  the default (0.2) leaves no room for the message labels
    \newinst[3]{server}{Server}
    \begin{call}{client}{request()}{server}{response} % chktex 36
    \end{call}
  \end{sequencediagram}
  \caption{Sequence diagram drawn with pgf-umlsd}
  \label{fig:uml-sequence}
\end{figure}
```

Table 8.2: Simple Table

Heading1	Heading2
One	Two
Three	Four

Table 8.3: Table with colored cells

Test	Result
A	pass
B	fail

8.14 Tables

Corresponding LaTeX code of `80 - latex hints.tex`

```latex
\begin{table}
  \caption{Simple Table}
  \label{tab:simple}
  \centering
  \begin{tabular}{ll}
    \toprule
    Heading1 & Heading2 \\
    \midrule
    One      & Two      \\
    Three    & Four     \\
    \bottomrule
  \end{tabular}
\end{table}
```

A table caption goes above the table (`\caption` before the `tabular`), a figure caption below the figure.

Rows and cells can be colored with `\rowcolor` and `\cellcolor` from the colortbl package, which xcolor loads with its option `table`.

Table 8.4: Table with a footnote

Heading1	Heading2[7]
One	Two

Corresponding LaTeX code of `80 - latex hints.tex`

```
424 \begin{table}
425   \caption{Table with colored cells}
426   \label{tab:colored}
427   \centering
428   \begin{tabular}{ll}
429     \toprule
430     \rowcolor{gray!20}
431     Test & Result                    \\
432     \midrule
433     A    & \cellcolor{green!20}pass \\
434     B    & \cellcolor{red!20}fail   \\
435     \bottomrule
436   \end{tabular}
437 \end{table}
```

A `\footnote` inside a floating table is lost. Use `\tablefootnote` from the tablefootnote package instead: it is set as a normal footnote at the bottom of the page. If the notes should appear directly below the table instead (common in journals), use the threeparttable package.

[7] A footnote inside a table.

Table 8.5: Wide table scaled to the line width

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1.11	2.22	3.33	4.44	5.55	6.66	7.77	8.88	9.99	10.10	11.11	12.12
Rain	Snow	Wind	Sun	Fog	Hail	Heat	Storm	Mist	Frost	Sleet	Calm

Corresponding LaTeX code of `80 - latex hints.tex`

```
445 \begin{table}
446   \caption{Table with a footnote}
447   \label{tab:footnote}
448   \centering
449   \begin{tabular}{ll}
450     \toprule
451     Heading1 & Heading2\tablefootnote{A footnote inside a table.} \\
452     \midrule
453     One      & Two                                                \\
454     \bottomrule
455   \end{tabular}
456 \end{table}
```

A table wider than the text runs into the margin. `\begin{adjustbox}{max width=\linewidth}` from the adjustbox package scales it down to the line width (the column width in a two-column layout), and only if it is too wide. Scaling shrinks the font, so for a table with long text in its cells, prefer wrapping the text with the column type `X` of the tabularx package, and for a table wider than the page, a landscape page (see `pdflscape`).

Table 8.6: Table with diagonal line

Column / Row	A	B
1	x	y

Corresponding LaTeX code of `80 - latex hints.tex`

```
\begin{table}
  \caption{Wide table scaled to the line width}
  \label{tab:wide}
  \centering
  \begin{adjustbox}{max width=\linewidth}
    \begin{tabular}{lllllllllll}
      \toprule
      Jan  & Feb  & Mar  & Apr  & May  & Jun  & Jul  & Aug   & Sep
       ↪  & Oct   & Nov   & Dec   \\
      \midrule
      1.11 & 2.22 & 3.33 & 4.44 & 5.55 & 6.66 & 7.77 & 8.88  & 9.99
       ↪  & 10.10 & 11.11 & 12.12 \\
      Rain & Snow & Wind & Sun  & Fog  & Hail & Heat & Storm & Mist
       ↪  & Frost & Sleet & Calm  \\
      \bottomrule
    \end{tabular}
  \end{adjustbox}
\end{table}
```

Corresponding LaTeX code of `80 - latex hints.tex`

```latex
% Adapted from https://tex.stackexchange.com/a/468994/9075
\begin{table}
  \caption{Table with diagonal line}
  \label{tab:diag}
  \centering
  \begin{tabular}{|l|c|c|} % chktex 44
    \hline % chktex 44
    \diagbox{Row}{Column} & A & B \\
    \hline % chktex 44
    1                     & x & y \\
    \hline % chktex 44
  \end{tabular}
\end{table}
```

8.15 Tables with tabularray

The tabularray package separates the layout of a table from its content. The layout is given once as key-value list in front of the rows: the columns (`colspec`), the style of whole rows (`row{...}`), and cells spanning multiple rows or columns (`cell{row}{column} = {r=..., c=...}`). The rows then contain only the content, without `\multicolumn` or `\multirow`. `X` columns share the remaining width as in tabularx, `Q` columns are as wide as their content. The caption is above the table, as for all tables.

Table 8.7: Runtime of four sorting algorithms, set with tabularray

Algorithm	Runtime (s)		Memory (MB)
	1 000 items	1 000 000 items	
Quicksort	0.12	1.80	12
Mergesort	0.15	2.10	24
Heapsort	0.20	2.40	12
Timsort	0.10	1.50	18

Corresponding LaTeX code of `80 - latex hints.tex`

```latex
\begin{table}
  \caption{Runtime of four sorting algorithms, set with tabularray}
  \label{tab:tabularray}
  \centering
  \begin{tblr}{
      colspec = {X[l] Q[r] Q[r] Q[r]},
      row{1,2} = {font=\bfseries},
      cell{1}{1} = {r=2}{l,m},  % "Algorithm" spans the two header rows
      cell{1}{2} = {c=2}{c},    % "Runtime" spans two columns
      cell{1}{4} = {r=2}{r,m},
      row{4,6} = {bg=black!6},  % every second row on a light gray
       ↪ background
    }
    \toprule
    Algorithm & Runtime (s)  &                   & Memory (MB) \\
    \cmidrule{2-3}
              & 1\,000 items & 1\,000\,000 items &             \\
    \midrule
    Quicksort & 0.12         & 1.80              & 12          \\
    Mergesort & 0.15         & 2.10              & 24          \\
    Heapsort  & 0.20         & 2.40              & 12          \\
    Timsort   & 0.10         & 1.50              & 18          \\
    \bottomrule
  \end{tblr}
\end{table}
```

For a table spanning more than one page, `longtblr` takes the same keys and repeats the header row (`rowhead = 1`) on every page.

8.16 Tables spanning multiple pages

The longtable package typesets tables that automatically break across page boundaries. The header (`\endhead`) and footer (`\endfoot`) rows are repeated on every page; once enough rows are added, the table spans more than one page on its own.

Table 8.8: A sample long table.

First column	Second column	Third column
A	BC	D
A	BC	D
A	BC	D
A	BC	D
A	BC	D
A	BC	D
A	BC	D
A	BC	D
A	BC	D
A	BC	D
A	BC	D
A	BC	D

Corresponding LaTeX code of `80 - latex hints.tex`

```
\begin{longtable}{lll}
  \caption{A sample long table.}\label{tab:long} \\
  \toprule
  \textbf{First column} & \textbf{Second column} & \textbf{Third
  ↪ column} \\
  \midrule
  \endfirsthead

  \multicolumn{3}{c}{{\bfseries \tablename\ \thetable{} -- continued
  ↪ from previous page}} \\
  \toprule
  \textbf{First column} & \textbf{Second column} & \textbf{Third
  ↪ column} \\
  \midrule
  \endhead

  \midrule
  \multicolumn{3}{r}{{Continued on next page}} \\
  \endfoot

  \bottomrule
  \endlastfoot

  A & BC & D \\
  A & BC & D \\
  A & BC & D \\
  A & BC & D \\
  A & BC & D \\
  A & BC & D \\
  A & BC & D \\
  A & BC & D \\
  A & BC & D \\
  A & BC & D \\
  A & BC & D \\
  A & BC & D \\
\end{longtable}
```

8.17 Source Code

The minted package highlights listings properly. It uses the pygments library, which in turn requires Python.

Listing 1 shows source code written in XML. Line 2 contains a comment.

```xml
<listing name="example">
  <!-- comment -->
  <content>not interesting</content>
</listing>
```

List. 1: Example XML listing using minted

Corresponding LaTeX code of `80 - latex hints - minted.tex`

```latex
% After a period, LaTeX sets the wider sentence space by default. The exception: a period after a capital letter
% is taken for an abbreviation or initial ("T. Scientist", "NASA") and gets the narrow inter-word space.
% "XML." ends a sentence, so \@ before the period restores the sentence space.
% (The other way round, "e.g.\ details" forces the narrow space after a lowercase abbreviation; ~ is a
% non-breaking inter-word space, as in "Prof.~Dr.~Eike Kraft".)
% An abbreviation set with \gls{xml} needs no \@: glossaries ends it with a non-letter, so the period after it
% gets the sentence space. The plain "XML" here is for documents without the glossaries package.
\Zcref{lst:XML} shows source code written in XML\@.
\Zcref{line:comment} contains a comment.

\begin{listing}[htbp]
  \begin{minted}[linenos=true,escapeinside=||]{xml}
<listing name="example">
  <!-- comment --> |\label{line:comment}|
  <content>not interesting</content>
</listing>
\end{minted}
  \caption{Example XML listing using minted}
  \label{lst:XML}
\end{listing}
```

One can also typeset JSON as shown in Listing 2.

```
{
  key: "value"
}
```

List. 2: Example JSON listing using minted

Corresponding LaTeX code of `80 - latex hints - minted.tex`

```
\begin{listing}[htbp]
  \begin{minted}[linenos=true,escapeinside=||]{json}
{
  key: "value"
}
\end{minted}
  \caption{Example JSON listing using minted}
  \label{lst:flJSON}
\end{listing}
```

Java is also possible as shown in Listing 3.

```
public class Hello {
    public static void main (String[] args) {
        System.out.println("Hello World!");
    }
}
```

List. 3: Java code rendered using minted

Corresponding LaTeX code of `80 - latex hints - minted.tex`

```latex
\begin{listing}[htbp]
  \begin{minted}[linenos=true,escapeinside=||]{java}
public class Hello {
    public static void main (String[] args) {
        System.out.println("Hello World!");
    }
}
\end{minted}
  \caption{Java code rendered using minted}
  \label{lst:flJava}
\end{listing}
```

Listings longer than the remaining space continue on the next page if they are *not* floating. A floating listing is always placed on a single page. Thus, use `\captionof{listing}` instead of the `listing` environment. Listing 4 spans multiple pages.

```java
public class Squares {

    /** Returns the square of 1. */
    public int square1() {
        return 1 * 1;
    }

    /** Returns the square of 2. */
    public int square2() {
        return 2 * 2;
    }

    /** Returns the square of 3. */
    public int square3() {
        return 3 * 3;
    }

    /** Returns the square of 4. */
    public int square4() {
        return 4 * 4;
```

```
    }

    /** Returns the square of 5. */
    public int square5() {
        return 5 * 5;
    }

    /** Returns the square of 6. */
    public int square6() {
        return 6 * 6;
    }

    /** Returns the square of 7. */
    public int square7() {
        return 7 * 7;
    }

    /** Returns the square of 8. */
    public int square8() {
        return 8 * 8;
    }

    /** Returns the square of 9. */
    public int square9() {
        return 9 * 9;
    }

    /** Returns the square of 10. */
    public int square10() {
        return 10 * 10;
    }

    /** Returns the square of 11. */
    public int square11() {
        return 11 * 11;
    }

    /** Returns the square of 12. */
```

```
59      public int square12() {
60          return 12 * 12;
61      }
62
63      public static void main(String[] args) {
64          System.out.println(new Squares().square12());
65      }
66  }
```

List. 4: Long Java listing spanning multiple pages

8.18 Itemization

One can list items as follows:

- Item One
- Item Two

Corresponding LaTeX code of `80 - latex hints.tex`

```
588  \begin{itemize}
589    \item Item One
590    \item Item Two
591  \end{itemize}
```

With the package paralist, one can create itemizations with lesser spacing:

- Item One
- Item Two

Corresponding LaTeX code of `80 - latex hints.tex`

```
597  \begin{compactitem}
598    \item Item One
599    \item Item Two
600  \end{compactitem}
```

One can enumerate items as follows:

1. Item One

2. Item Two

Corresponding LaTeX code of `80 - latex hints.tex`

```
606 \begin{enumerate}
607   \item Item One
608   \item Item Two
609 \end{enumerate}
```

With the package paralist, one can create enumerations with lesser spacing:

1. Item One
2. Item Two

Corresponding LaTeX code of `80 - latex hints.tex`

```
615 \begin{compactenum}
616   \item Item One
617   \item Item Two
618 \end{compactenum}
```

With paralist, one can even have all items typeset after each other and have them clean in the TeX document:
1. The first item, 2. the second item, and 3. the third item appear in one line.

Corresponding LaTeX code of `80 - latex hints.tex`

```
624 \begin{inparaenum}
625   \item The first item,
626   \item the second item, and
627   \item the third item appear in one line.
628 \end{inparaenum}
```

8.19 Writing in Markdown

If you are new to LaTeX, you can start by writing your text in Markdown using the markdown package. LaTeX then takes care of the typesetting. Markdown and LaTeX can be mixed: Table 8.9 is written in Markdown and referenced from LaTeX. A whole chapter can be written in a separate Markdown file included by

Table 8.9: A table written in Markdown

Right	Left	Centered
12	12	12
123	123	123

`\markdownInput`, as Chapter 7 shows. The markdown-latex-quickstart shows a complete document written in Markdown.
Text can be *emphasized* or **bold**, and `code` is set in a monospaced font. A citation looks like this [Sch17], and here is a footnote.[8] Referenced footnotes work as well[9].

- first point
- second point
  - sub-point

1. firstly
2. secondly

Table 8.9 shows a table with caption. Its identifier `{#tbl:markdown}` is referenced by `<#tbl:markdown>`.

[8]This is an inline footnote.
[9]This is a footnote defined at a different place in the text.

Corresponding LaTeX code of `80 - latex hints.tex`

```
  \begin{markdown}
Text can be *emphasized* or **bold**, and `code` is set in a monospaced
↪ font.
A citation looks like this [@mwe], and here is a footnote.^[This is an
↪ inline footnote.]
Referenced footnotes work as well[^detail].

[^detail]: This is a footnote defined at a different place in the text.

- first point
- second point
  - sub-point

1) firstly
2) secondly

| Right | Left | Centered |
|------:|:-----|:--------:|
|    12 | 12   |    12    |
|   123 | 123  |   123    |

: A table written in Markdown {#tbl:markdown}

<#tbl:markdown> shows a table with caption.
Its identifier `{#tbl:markdown}` is referenced by `<#tbl:markdown>`.
\end{markdown}
```

8.20 Abbreviations

With `\gls{...}`, you can enter abbreviations: the first time you call it, the long form is used. When reusing `\gls{..}` the short form is automatically displayed. The abbreviation is also automatically inserted in the abbreviation list. With `\glspl{...}` the plural form is used. If you want the short form to appear directly at the first use, you can use `\glsunset{...}` to mark an abbreviation as already used. The opposite is achieved with `\glsreset{...}`. Both only change the “already used” flag: the effect shows at the next `\gls{...}` of that abbreviation, so they have to come before it in the text. `\glsresetall` (e.g.,

after the abstract) lets every abbreviation appear in its long form again.
Abbreviations are defined in `abbreviations.tex` by means of `\newabbreviation{key}{short}{long}` (or `\newacronym` for an acronym that is read as a word).
More information at: `https://ctan.org/pkg/glossaries-extra`.
At the first pass, the error rate (ER) was 5. At the second pass was ER 3. The plural form can be seen here: ERs. To demonstrate what the list of abbreviations looks like for longer description texts, Relational Database Management Systems (RDBMS) must be mentioned here.
Deutschsprachige Anwendervereinigung TEX e.V. (DANTE e.V.) is a local TEX user group. The German-speaking local TEX user group is DANTE e.V.. A General Practitioner (G.P.) is a medical doctor. I went to my surgery to see the G.P..

Corresponding LaTeX code of `80 - latex hints.tex`

```
688 At the first pass, the \gls{er} was 5.
689 At the second pass was \gls{er} 3.
690 The plural form can be seen here: \glspl{er}.
691 To demonstrate what the list of abbreviations looks like for longer
    ↪ description texts, \glspl{rdbms} must be mentioned here.
692
693 \gls{dante} is a local \TeX\ user group.
694 The German-speaking local \TeX\ user group is \gls{dante}.
695 A \gls{gp} is a medical doctor.
696 I went to my surgery to see the \gls{gp}.
```

6 [Todo 6: Include difference between acronym and abbreviation, see https://tex.stackexchange.com/a/154566/9075]

8.21 Other Features

The words “workflow” and “dwarflike” can be copied from the PDF and pasted to a text file.

Corresponding LaTeX code of `80 - latex hints.tex`

```
The words \enquote{workflow} and \enquote{dwarflike} can be copied from
    ↪ the PDF and pasted to a text file.
```

The symbol for powerset is now correct: $\powerset$ and not a Weierstrass p ($\wp$).
$\powerset(1,2,3)$

Corresponding LaTeX code of `80 - latex hints.tex`

```
The symbol for powerset is now correct: $\powerset$ and not
    ↪ a Weierstrass p ($\wp$).

$\powerset({1,2,3})$
```

Brackets work as designed: <test> One can also input backticks in verbatim text: `test`.

Corresponding LaTeX code of `80 - latex hints.tex`

```
Brackets work as designed:
<test>
One can also input backticks in verbatim text: \verb|`test`|.
```

Acronyms can be typeset in running text with `\initialism`, which sets them as small caps that blend into the surrounding lowercase text. It is the lightweight alternative to the full acronym and glossary management (`\gls`, with an automatic list of abbreviations) provided by the thesis variants: `\initialism` only typesets the acronym and does not add it to any list. The macro and the ready-made acronyms (`\OMG`, `\BPEL`, `\BPMN`, `\UML`) are defined in `commands.tex`, where you can add your own.
The UML is maintained by the OMG; related standards include BPEL and BPMN. Any acronym can be set inline with REST or HTTP.

Corresponding LaTeX code of `80 - latex hints.tex`

```
722 The \UML is maintained by the \OMG; related standards include \BPEL and
    ↪ \BPMN.
723 Any acronym can be set inline with \initialism{REST} or
    ↪ \initialism{HTTP}.
```

8.22 Zref-vario examples

Demonstration of zref-vario: zvref[S] at the beginning of a sentence, zvref in all other cases.
Figure 8.1 on page 42 shows a simple fact, although Figure 8.1 on page 42 could also show something else.
Table 8.1 on page 41 shows a simple fact, although Table 8.1 on page 41 could also show something else.
Section 8.7 on page 41 shows a simple fact, although Section 8.7 on page 41 could also show something else.

Corresponding LaTeX code of `80 - latex hints.tex`

```
733 \zvref[S]{fig:ex:cref} shows a simple fact, although
    ↪ \zvref{fig:ex:cref} could also show something else.
734
735 \zvref[S]{tab:ex:cref} shows a simple fact, although
    ↪ \zvref{tab:ex:cref} could also show something else.
736
737 \zvref[S]{sec:ex:cref} shows a simple fact, although
    ↪ \zvref{sec:ex:cref} could also show something else.
```

8.23 Citations

When referencing something from the bibliography file, it will automatically appear in the references section. If a reference is not cited, it is not appearing there.
Standard case: Citing indirectly citing something [Sch17]. In case one wants to name the author: Scharrer [Sch17] shows a minimal LaTeX example.

Corresponding LaTeX code of `80 - latex hints.tex`

```
Standard case: Citing indirectly citing something~\cite{mwe}.
In case one wants to name the author: \textcite{mwe} shows a minimal
↪ \LaTeX{} example.
```

Note that `\textcite{mwe}` prints both the author and the reference to the bibliography entry.
A thesis is cited like any other work: a master's thesis [Sha40] or a doctoral thesis [Nas50]. The entry types `@MastersThesis` and `@PhdThesis` are labeled in the document language ("MA thesis" or "Masterarbeit", "PhD thesis" or "Diss."). For a Bachelor's thesis, use `@Thesis` with `type = {bathesis}`; for a Diplomarbeit, `type = {Diplomarbeit}` is printed as is.
Remember that you have to call `biber thesis-example` to generate the bibliography data for `lualatex`. You will need to run `lualatex` twice so that the page numbers are up to date.
In the bibliography, use `\textsuperscript` for "st", "nd", …: E.g., "The 2nd conference on examples". When you use JabRef, you can use the cleanup command to achieve that. See `https://help.jabref.org/en/CleanupEntries` for an overview of the cleanup functionality.

8.24 Miscellaneous Examples

Reference test: Section 8.24, Figure 8.11 and Algorithm 8.1.
Checkmark: ✓. Crossmark: ✗.

Corresponding LaTeX code of `80 - latex hints.tex`

```
Checkmark: \dingcheck.
Crossmark: \dingcross.
```

[Placeholder 2: missing figure]



Figure 8.11: Figure

[Placeholder 3: missing figure]

Figure 8.12: Rotated figure

Manuscript version: #b201c7 - 2026-11-20 14:31:21Z

8.24.1 Algorithms

Algorithm 8.1 *algo*

1: $a \leftarrow 0$
2: State 2

Algorithm 8.2 Algorithm 2

1: $a \leftarrow 0$
2: State 2

Algorithm 8.1 has already shown an algorithm. Test of line referencing: line 2 (Algorithm 8.1) and line 2 (Algorithm 8.2).

8.24.2 Definitions

Definition 8.1 (Title)
Definition Text

Definition 8.1 shows …

8.24.3 Enumerations

(a) a

(b) b

(c) c

(d) d

Equivalent to paralist's inparaenum: (a) a (b) b (c) c (d) d

first First

second Second

third Third

first First

second Second

third Third

first First

second Second

third Third

first label with a long description text breaking over one line. Enabled by the enumitem package First

second Second

third Third

first label with a long description text breaking over one line. Defined in the template

First

second

Second

third

Third

- First
- Second
- Third

The optional parameter changes the marker that is prepended. See `http://www.weinelt.de/latex/item.html`.

A First

B Second

C Third

Incorrect use of the optional parameter as follows:

first First

second Second

third Third

Note that, when `enumitem` is loaded, this behaves differently than with `paralist`.

8.24.4 The fquote environment

> *The program will be finished by next Friday.*
>
> — T. Scientist

> *The program will be finished by next Friday.*
>
> —T. Scientist

Chapter 9

Conclusion and Outlook

Your conclusions are more than a factual summary of your work: they position, interpret, and defend your findings against the state of the art that you discussed in Section 1.2. You specifically outline which concrete findings or methodological contributions advance our knowledge towards the general objective you introduced in Section 1.1. Objectively discuss which parts you solved and in which parts you failed. State who benefits from your results (developers, architects, testers, researchers, …) and how. Name the lessons you learned, e.g., on the evaluation setup or the literature search, that help someone repeating or extending your study.

You should explicitly discuss limitations and shortcomings of your work and detail what kind of future studies are needed to overcome these limitations. Be specific in the sense that your arguments for future work should be based on concrete findings and insights you obtained in your report.

Bibliography

[BCR94] V. R. Basili, G. Caldiera, H. D. Rombach. “The Goal Question Metric Approach”. In: *Encyclopedia of Software Engineering*. John Wiley & Sons, 1994, pp. 528–532 (cit. on p. 29).

[LLDS26] H. Lichter, J. Ludewig, M. Deininger, K. Schneider. *Student Thesis Projects: A Guide to Organizing, Conducting, and Presenting Student Thesis Projects Using the Example of Computer Science*. 7th ed. Zollikon: vdf Hochschulverlag, 2026. ISBN: 978-3-7281-4241-2. DOI: 10.3218/4241-2. URL: https://vdf.ch/product/student-thesis-projects-en.html (visited on 09/23/2026) (cit. on p. 14).

[Nas50] J. F. Nash. “Non-Cooperative Games”. PhD thesis. Princeton University, 1950 (cit. on p. 73).

[RH09] P. Runeson, M. Höst. “Guidelines for conducting and reporting case study research in software engineering”. In: *Empirical Software Engineering* 14.2 (2009), pp. 131–164. DOI: 10.1007/s10664-008-9102-8 (cit. on p. 31).

[Sch17] M. Scharrer. *The* mwe *Package*. 2017. URL: http://texdoc.net/mwe (cit. on pp. 33, 44, 68, 72).

[Sha40] C. E. Shannon. “A Symbolic Analysis of Relay and Switching Circuits”. MA thesis. Massachusetts Institute of Technology, 1940. URL: https://hdl.handle.net/1721.1/11173 (cit. on p. 73).

All links were last followed on September 25, 2027.

Appendix A

My first appendix

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Appendix B

Use of AI Tools

AI Tool	Used for	Used where
ChatGPT5	Generating figures	Figures 8.2 and 8.3
Google AI search	Pseudo code for algorithm XYZ	Algorithm 8.1
DeepL Write	Rephrasing the introduction	Chapter 1

List of Figures

4.1 A scientific figure that has to be explained in the text 25

6.1 Another scientific figure that has to be explained in the text . . 30

8.1 Example figure for zcref demo 42
8.2 Simple Figure 44
8.3 Ulm Minster. Photo: wkrauth, CC0, via Wikimedia Commons. . 45
8.4 Example figure with two sub figures. 46
8.5 A regular grid generated easily with two for loops. 47
8.6 Plot of sin(x) directly inside the figure environment with pgfplots. 48
8.7 Explicit coordinates plotted with pgfplots. 49
8.8 Class diagram drawn with MetaUML 50
8.9 Component diagram drawn with MetaUML 51
8.10 Sequence diagram drawn with pgf-umlsd 52
8.11 Figure 73
8.12 Rotated figure 74

List of Tables

7.1 A table written in Markdown 34

8.1 Example table for zcref demo 41
8.2 Simple Table 53
8.3 Table with colored cells 54
8.4 Table with a footnote 55
8.5 Wide table scaled to the line width 56
8.6 Table with diagonal line 57
8.7 Runtime of four sorting algorithms, set with tabularray 58
8.8 A sample long table. 60
8.9 A table written in Markdown 68

List of Listings

1 Example XML listing using minted 62
2 Example JSON listing using minted 63
3 Java code rendered using minted 63
4 Long Java listing spanning multiple pages 66

List of Definitions

8.1 Title 75

List of Abbreviations

DANTE e.V. Deutschsprachige Anwendervereinigung TEX e.V. 70

ER error rate 70

G.P. General Practitioner 70

RDBMS Relational Database Management System 70

Zebra Notes

Type	Count
todo	6
fixed	1
comment	2
note	1
placeholder	3
fixme	1
source	2
Total	**16**

List of notes

Todo 1 13
The chapter structure of this template is only an example. Agree on the actual structure of your thesis with your supervisor.

Source 1 20
Study on the sleep of giraffes

Todo 2 37

A comment on it.
Comment 1 38
Comment
Todo 3 38
A lot of text still needs to be produced here
Todo 4 38
Add the evaluation
Fixme 1 38
The numbers are outdated
Source 2 38
Survey on the tools used for writing
Todo 5 `@FA` 39
Add a citation
Note 1 39
Shorten this paragraph?
Comment 2 `@FA` 39
Is this claim backed by the evaluation?
Fixed 1 39
Typo corrected
Placeholder 1 39
Table with the results
Todo 6 70
Include difference between acronym and abbreviation, see https://tex.stackexchange.com/a/154566/9075
Placeholder 2 73
missing figure
Placeholder 3 74
missing figure