

LaTeX Basics

for scientists

Chapter 0: What is LaTeX and why do you want to use it?

What is TeX?

- computer program for typesetting documents
- input document is written in plain text in a text editor
- formatting language: formatting is marked by commands
- built-in macro language; definition of own macros
- conversion of text files into binary file of a page description language (DVI, PDF, PS)
- no word processing program; cannot be compared to word processing programs (different purposes)
- Open Source and operating system independent
- no WYSIWYG ("What You See Is What You Get")
- very well documented

What is special about TeX?

- special strength: mathematical notation and typeface
- layout is considered to be very clean
- rule set for setting distances in mathematical formulas
- text is set paragraph by paragraph (total fit instead of first fit):
 - hyphenation algorithm
 - line break optimization: all line breaks (including word separations) in the paragraph are determined simultaneously
 - page break optimization

What is LaTeX?

- software package simplifying the use of the TeX typesetting system using macros
- collection of TeX macros, which simplify and expand the use of TeX
- logical markup for the outline
- easiest and best maintained access to TeX
- enables simple
 - automatic numbering of chapters, sections, sentences, equations,...
 - Cross-references

TeX Engines

There are several engines that can process TeX input.
Not all are used to the same extent.

TeX

- Original engine from Knuth
- creates DVI document
- can import EPS and PS

pdfTeX

- fully backwards compatible with Knuth's TeX and thus remains an 8-bit system
- creates PDF document

XeTeX

- Unicode-based (UTF-8), i.e. Unicode as input
- uses system-dependent libraries (built into the binary) to access system fonts
- does not produce direct PDF output, but works via XDV (eXtended DVI) as an intermediate format \Rightarrow converted to PDF as part of the xetex run using xdvpdfmx

LuaTeX

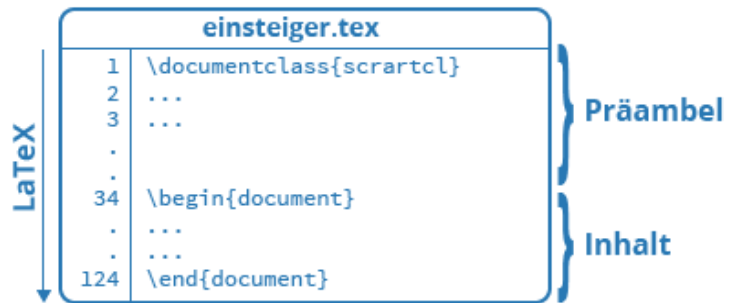
- also Unicode-based (UTF-8), i.e. Unicode as input
- "embedded" Lua interpreter has access to many of the data structures used for text typesetting
- Lua code can be inserted into (La)TeX macros
- Loading of system fonts via Lua code
- ConTeXt is a format for TeX, like LaTeX, but newer and much larger and a pure LuaTeX implementation

Chapter 1: Basics

Syntax

Minimal document setup

```
\documentclass{article}
\begin{document}
\end{document}
```



Reserved characters & spaces

The following characters are part of the LaTeX syntax and are not converted into the document:

character	meaning	example	how to display
\	command	<code>\emph \rightarrow</code> <code>\alpha</code>	<code>\backslash,</code> <code>\textbackslash{}</code>
{}	Grouping of arguments	<code>\emph{italics text}</code> <code>^{superscript textt}</code>	<code>\{</code> <code>\}</code>
#	Arguments in user-defined commands	<code>\newcommand{\mycom}[2]</code> <code>{what to do with my</code> <code>arguments #1 and #2}</code>	<code>\#</code>
\$\$	math mode	<code>\$2^{n \log n}\$</code>	<code>\\$</code>
^	Superscript (in math mode)	<code>\$2^{n \log n}\$</code>	<code>\textasciicircum</code> <code>\^{}</code>
_	Subscript (in math mode)	<code>\$x_i\$</code>	<code>_</code>
&	Cell separation in tables	<code>cell 1 & cell 2 \\\</code>	<code>\&</code>
~	Space without line break	<code>Dr.~Who?</code>	<code>\textasciitilde</code> <code>\~{}</code>
%	comment	<code>% this is a comment</code>	<code>\%</code>

Debugging:

- ! Misplaced alignment tab character &. A & was used unmasked outside a tabular or array environment.

Whitespaces

- One space is the same as one million spaces.
- Tabs are treated like spaces.
- Spaces at the end of a line are ignored.
- Single line breaks are treated as spaces.
- More than one line break marks the beginning of a new paragraph.
- Spaces after commands are ignored ⇒ Use backslash to end command (e.g. `\LaTeX\`).

Commands, groupings & environments

Commands

```
\commandname [<option1>, <option2>, ...] {<argument1>} {<argument2>} ...
```

Define own commands:

```
\newcommand{\mycommand} [<maxNumParam>] [<default>] {what to do with my  
arguments #1 and #2}
```

`maxNumParam` Number of arguments (max 9)

`default` default for the one possible optional parameter (always #1)

Groupings

grouped areas within a LaTeX document that are set within curly brackets

```
{ <content> }
```

Commands within a grouping do not affect the entire document

Groupings can also be created using `\begingroup <content> \endgroup`

Environments

```
\begin{<environment>} ... \end{<environment>}
```

Define own environments

```
\newenvironment{myenvironment} [<maxNumParam>] {<begin>} {<end>}
```

Debugging:

- `! Undefined control sequence ...`
 - Unknown command used: Typing error; required class or package not included (rare if class / package has been updated and the command no longer exists); wrong compiler
- `! LaTeX Error: Environment ... undefined.`
 - Unknown environment
- `! Argument of \@fileswithoptions has an extra }.`
 - Brackets not matching

Creating the document

```

latex <file>      tex compiler with LaTeX initialization files; creates DVI document;
                  imports EPS and PS
pdflatex <file>   pdftex compiler with LaTeX initialization files; creates PDF document;
                  imports JPG, PNG and PDF
xelatex <file>    XeTeX compiler with LaTeX initialisation files; generates PDF
                  Unicode as input; system fonts
lualatex <file>   LuaTeX compiler with LaTeX initialization files; generates PDF
                  Unicode as input; system fonts

```

⇒ compile twice

Auxiliary files

Generated by (La)TeX to store information

```

.aux  stores information from one compilation process to the next (e.g. cross references)
.log  log of the last compilation process
.toc  table of contents
.lof  list of figures
.lot  list of tables
.idx  index register

```

Document structure and preamble

```
\documentclass[<options>]{<class>}
```

```
\begin{document}
```

```
\end{document}
```

Document classes

Document class	Intended use	Default options
article	Documents with small number of pages (scientific articles, ...) no automatic title page no chapters (<code>\chapter</code>)	letterpaper, 10pt, oneside, onecolumn, final
report	smaller theses (project work, bachelor thesis, ...) front page	letterpaper, 10pt, oneside, onecolumn, final, openany

book	larger documents (dissertations, ...)	letterpaper, 10pt, twoside, onecolumn, final, openright
letter	American letter format	letterpaper, 10pt, oneside, onecolumn, final

The `report` class is suitable for quickly converting documents created with `book` into a screen-readable version (e.g. no empty pages)

LaTeX standard classes are based on US typographic conventions and paper formats.

KOMA-Script

- is based on European typographic conventions and DIN paper formats
- complete user interface of the standard classes extended by additional functions

`scrartcl`, `scrbook`, `scrlettr2`

Options:

Font size: 10pt, 11pt, 12pt
Page layout: onecolumn, twocolumn, oneside, twoside
Paper format: a4paper, letterpaper, a5paper, b5paper, executivepaper, legalpaper
Title page: titlepage, notitlepage
Chapters: openany, openright
Graphics: final, draft
Page orientation: portrait, landscape
Space between paragraphs: noparskip, halfparskip
Equations: fleqn, leqno

Debugging:

- ! LaTeX Error: File ``articles.cls'` not found.
→ The class name is wrong or the class is not installed.
- ! LaTeX Error: Missing `\begin{document}`.
→ The `\begin{document}` command is preceded by normal text. Or the command `\begin{document}` is missing
- *
→ `\end{document}` is missing

Additional packages

```
\usepackage[<options>]{<package>}
\usepackage{<package1>,<package2>,<package3>}
```

Debugging:

- ! LaTeX Error: Can be used only in preamble.
→ `\usepackage` command outside the preamble.
- ! LaTeX Error: File ``lmonder.sty'` not found.
→ The package does not exist or the name is misspelled.
- ! LaTeX Error: Unknown option ``...'` for package ``...'`.
→ Unknown option for a package.
- ! LaTeX Error: Option clash for package ``...'`.
→ Different (conflicting) options are set for the same package. The package was loaded twice, possibly by another package.

A first package: `blindtext`

`\blindtext[<x>]` creates a paragraph sample text with `<x>` repetitions
`\Blindtext[<y>][<x>]` creates `<y>` paragraphs of sample text with `<x>` repetitions

Page layout

Display page layout

Packages for visualizing the page layout:

`\usepackage{showframe}` display page margins

`\usepackage{layout}` output of the current page layout, incl. parameters & values
`\layout`

`\usepackage{layouts}` further possible layout outputs

Adjust page layout

Changing layout variables:

`\setlength{\<lengthcommand>}{}`

Length units:

pt 1/72 inch
mm
cm
in inch
ex font dependent; about the height of an "x"
em Font dependent; about the width of an "M"

Length commands can also be used as length units:

<code>\parindent</code>	length of indent of paragraphs
<code>\baselineskip</code>	vertical space between lines
<code>\parskip</code>	additional distance between paragraphs
<code>\textwidth</code>	line width in the current environment
<code>\textheight</code>	height of the text on the current page

The geometry package

- changes many parameters within the document
- should be included as the last package if possible

```
\usepackage[<options>]{geometry}
```

or

```
\usepackage{geometry}
```

```
\geometry{<options>}
```

Options:

- Options of the document classes are automatically transferred or can be overwritten
- The values of the length variables `\topmargin`, `\oddsidemargin` and `\evensidemargin` are added to the margin values set by the `geometry` package

Margins: `top=xx, bottom=xx, left=xx, right=xx`

Page size: `width=xx, height=xx`

Text block: `textwidth=xx, textheight=xx`

Footer: `footsep=xx, footskip=xx`

...

Headers and footers

```
\pagestyle{<option>}      entire document
```

```
\thispagestyle{<option>} single page
```

Optionen:

`plain` page number in footer (default for `article`)

`empty` no page numbering

`headings` running header on each page; document class dependent what is displayed in the header; default for `book`

`book, report:`

one-sided: — chapter

two-sided: chapter section


```

article
    one-sided:  —          section
    two-sided: section    subsection
myheadings own header design using
    \markboth{<left header>}{<right header>} or
    \markright{<right header>}
  
```

Column titles

Automatically display chapter, section or subsection in header or footer.

`\automark[<right side>]{<left side>}` ⇒ Specify the column title.

Optional argument only necessary for double-page documents

The fancyhdr package:

Customize headers/footers

```

\usepackage{fancyhdr}
\pagestyle{fancy}
  
```

Delete the defaults or previous settings:

```

\fancyhead{}
\fancyfoot{}
  
```

```

\lhead[<lh-even>]{<lh-odd>}          \lfoot[<lf-even>]{<lf-odd>}
\chead[<ch-even>]{<ch-odd>}         \cfoot[<cf-even>]{<cf-odd>}
\rhead[<rh-even>]{<rh-odd>}         \rfoot[<rf-even>]{<rf-odd>}
  
```

Access column titles via `\leftmark` and `\rightmark`

The scrlayer-scrpage package

KOMA extension ⇒ only for KOMA-Script classes

```

\usepackage{scrlayer-scrpage}
  
```

Two new styles:

<code>scrplain</code>	configuration for front matter pages, e.g. table of contents
	Single-page documents: page number in footer (centered)
	Two-page documents: page number in footer (outside)
<code>scrheadings</code>	configuration for the main part of the document
	Single-page documents: column title in header (centered), page number in footer (centered)
	Two-page documents: column title in header (outside) page number in footer (outside)

Styles are automatically activated on the respective pages.

Delete defaults or previous settings: `\clearpairstyles`

<code>\ihead[<definescrplain>]{<definescrheadings>}</code>	inside
<code>\chead[<definescrplain>]{<definescrheadings>}</code>	center
<code>\ohead[<definescrplain>]{<definescrheadings>}</code>	outside

same for `\ifoot`, `\cfoot` `\ofoot`

`\headmark` Output of the current column title according to the respective page

Adjustment of the column title:

<code>\automark[chapter]{section}</code>	default setting
<code>\automark{<links>}{<rechts>}</code>	adjustment

`automark` can also be specified as package option

Package options to create separator lines between header and text:

<code>headtopline</code>	display line above header
<code>headsepline</code>	display line below header
<code>footsepline</code>	display line above footer
<code>footbotline</code>	display line below footer

Change font

`\setkomafont{pageheadfoot}{<define font>}`
`\setkomafont{pagenumber}{<define font>}`

Page numbers

`\pagenumbering{gobble}`

<code>gobble</code>	no page numbers
<code>arabic</code>	arabic numbers Zahlen
<code>roman</code>	lower case Roman numbers
<code>Roman</code>	upper case Roman numbers
<code>alph</code>	lower case letters
<code>Alph</code>	upper case letters

Print current page number: `\thepage` or `\pagemark`

Set page number: `\setcounter{page}{<number>}`

Multi-column documents

```
\documentclass[twocolumn]{article}
```

The `multicol` package

```
\usepackage{multicol}  
\begin{multicols}{2} ... \end{multicols}
```

- up to ten columns
- allows you to switch number of columns within a document
- can be used within other environments (e.g. `figure`)
- balanced column length on the last page
- vertical lines between the columns can be adjusted
- difficulties with floats (use `*` environments, e.g. `\begin{figure*}`, to span full `textwidth`)

```
\setlength{\columnseprule}{<Breite>}    vertical line between the columns  
\setlength{\columnsep}{<Breite>}        horizontal space between the columns
```

Document structure

Title(page)

```
\title{...}, \author{...}, \date{...}  
today's date: \date{\today}
```

```
\begin{titlepage}... \end{titlepage}    generates title page  
\maketitle                               generates titles
```

Abstract

```
\begin{abstract} ... \end{abstract}  
\renewcommand{\abstractname}{<new name>}
```

Chapters/Sections

```
-1  \part{<text>}  
0   \chapter{<text>}  
1   \section{<text>}  
2   \subsection{<text>}  
3   \subsubsection{<text>}  
4   \paragraph{<text>}  
5   \subparagraph{<text>}  
  
    \minisec{<text>}    only in KOMA-Script classes; no numbering
```

```
\section[<short title>]{<long title>}
short title appears in the table of contents, long title in the document
\section*{<Section without numbering>}
* no numbering
```

```
\setcounter{secnumdepth}{1} Numbering depth
\setcounter{tocdepth}{3}    Numbering depth in table of contents
\setcounter{section}{4}    Set value for the numbering of the following sections
```

Table of contents

```
\tableofcontents
\setcounter{tocdepth}{<depth>}    Numbering depth in table of contents (3 is default)
\listoffigures    List of figures
\listoftables    List of tables
```

Commands (Sections, Figures, Tables, ...) are not listed in the table of contents

Insert an unnumbered section into the table of contents:

```
\section*{Introduction}
\addcontentsline{toc}{section}{Introduction}
```

or generally:

```
\addcontentsline{<list>}{<type>}{<entry>}
```

list	toc, lof, lot
type	type of entry chapter, section, subsection, ... for table of contents (toc) figure, table for list of figures (lof) or tables (lot)
entry	Text; commands must be protected with <code>\protect</code>

Indent the entry in the directory:

```
\addcontentsline{lof}{figure}{\protect\numberline{\thefigure}<Bildunterschrift>}
```

Entry of additional information or e.g. extra distance:

```
\addtocontents{<list>}{<entry>}
```

Adjusting headings

The `titlesec` package

```
\titleformat{<level>}[<shape>]{<format>}{<label>}{<sep>}{<beforecode>}
[<aftercode>]
```

level	chapter, section, ...
shape	shape of the heading: hang, block, display, runin, leftmargin, rightmargin, drop, wrap, frame

<code>format</code>	Heading formatting (text)
<code>label</code>	labeling/numbering
<code>sep</code>	horizontal distance between numbering and title
<code>beforecode</code>	code before the title (may also be empty)
<code>aftercode</code>	code after the title (optional)

```
\titlespacing{\level>}{\left>}{\fore-sep>}{\after-sep>}
```

<code>left</code>	left margin
<code>before-sep</code>	vertical space before the title
<code>after-sep</code>	space between title and text

For KOMA script classes:

No use of the titlesec package

```
\addtokomafont{<level>}{<fontformat>}
\renewcommand*{<level>format}{<labelformat>} uses counter \the<level>.
\RedeclareSectionCommand[<options>]{<level>}
```

options: `beforeskip=xx`, `afterskip=xx`

Cross-references

```
\label{<key>}
\ref{<key>}
```

- `key` can contain upper and lower case letters, numbers and punctuation marks (but no spaces)
- `\ref{key}` can be used before definition of `\label{key}`

Usage:

- Sections, chapters, etc.
- Entries in numbered lists
- Floats (`figure`, `table`)
- Theorems etc.
- Mathematic environments
- single lines in certain environments (`align`, ...)

Referencing pages

```
\pageref{<key>}
```

The `varioref` package

```
\usepackage{varioref}
```

```
\vref{<key>}
```

depending on the side of the labelled object and the reference:

- Object and reference on the same page: `\vref` generates standard command `\ref` without `\pageref`
- Object and reference on different pages:
 - difference > 1: `\vref` generates `\ref` with `\pageref`
 - difference = 1: `\vref` generates `\ref` and “on the preceding page” or “on the following page” or similar phrases

Chapter 2: Text Formatting

Encoding

pdfTeX-Engine

Input encoding

```
\usepackage[<option>]{inputenc}
```

- Support of extended input character sets with their different encodings
- Conversion of the input character encoding into the internal LaTeX standard language
- `option` -- use character encoding of the editor
- during multiperson processing, problems may occur with different input character encoding
- `standard`: `utf8`

Output encoding

```
\usepackage[<option>]{fontenc}
```

- `standard` font of LaTeX is the Computer Modern (CM) font family
- does not offer all 256 characters of the European character set in T1 encoding by default
- Problem: special characters are reassembled from the limited number of available characters
- `fontenc`-package forces output fonts in appropriate encoding (option `T1` for Western Europe)
- text characters for the corresponding languages as single characters
- more than 100 free fonts are available for LaTeX (LaTeX Font Catalogue)

Language

```
\usepackage[<language1>,<language2>,...]{babel}
```

- adaptation for LaTeX to many languages
- contains hyphenation, special characters, language-specific typographic rules
- translation of the language-specific commands in the document
- several languages can be loaded; last loaded language is active

```
\selectlanguage{<languageA>}
```

```
\foreignlanguage{<languageB>}{<Text in another language>}
```

```
\begin{otherlanguage}{<languageB>} <Text in another language>
```

```
\end{otherlanguage}
```

```
ngerman, english
```

Umlauts / Accents

- If the correct input and output encoding is set, umlauts and accents can be used without complications.
- Without the use of additional packages, umlauts, accents and special characters must be generated with the corresponding LaTeX commands

XeTeX or LuaTeX engine

Input encoding via the engine (Unicode)

⇒ Editor should be set to appropriate character encoding

Output encoding

```
\usepackage{fontspec}
```

System fonts can be loaded

```
\setmainfont{<systemfont>}
```

```
\setsansfont{<systemfont>}
```

```
\setmonofont{<systemfont>}
```

Language

```
\usepackage{polyglossia}
```

```
\setdefaultlanguage[<options>]{<language1>}
```

```
\setotherlanguage[<options>]{<language2>}
```

Options:

variant

spelling

script

...

Commands:

```
\selectlanguage[<options>]{<language>}
```

```
\textlang[<options>]{<language>}{<text>}
```

```
\begin{lang}[<options>]{<language>} ... \end{lang}
```

Hyphenation

define your own rules for hyphenation:

possible hyphenation points for the entire document:

```
\hyphenation{zwei-fels-oh-ne son-nen-ba-den}
```

possible hyphenation points for a single occurrence of a word:

```
zwei\ -fels\ -oh\ -ne
```


Suppress hyphenation:

`\mbox{<text>}`

Command	Function	Example	Possible hyphenation points
-	Hyphen that suppresses other possible breaking points	Mess-Ergebnis	Mess-Ergebnis
"=	Hyphen which allows further breaking points	Mess"=Ergebnis	Mess-Er-geb-nis
"~	Hyphen that is not allowed as breaking point	Mess"~, Schätz"~ und andere Ergebnisse	Mess-, Schätz-, ...*
\-	Breaking point that suppresses other breaking points	Ur\ -instinkt	Ur- instinkt
"-	Additional breaking point	Am"-nestie	Am-nes-tie
""	Breaking point that does not require a hyphen	und/""oder	und/-oder

* only hyphen, no breaking point

Global settings

`\hyphenpenalty=<xx>`

hyphenation penalty

`\tolerance=<xx>`

Tolerance for uneven spacing between words

Typesetting

Optimise typesetting

`\widowpenalty=<xx>` penalty widows

`\clubpenalty=<xx>` penalty orphans

Underfull/overfull warnings

Underfull Box: too much white space in a line → large spaces between words

Overfull Box: there is not enough space to place the text within the margins
→ words exceed the right margin (use the `[draft]` option of the `\documentclass` command to highlight overfull lines in the document)

The `microtype` package

- greatly improves the overall appearance of the text by using various techniques
- allows more words to be placed in each line of text
- overall word spacing is reduced and at the same time more evenly distributed
- reduces the number of "under-" and "overflow box" warnings

Horizontal vs vertical vs mathematical mode

Horizontal mode (paragraph mode)

- Processing of text
- Wrap text in lines and the lines in pages
- LR mode (restricted horizontal mode): `\mbox`; no automatic line breaks
- `\parbox` and `minipage`: Paragraph mode without page breaks

Math mode

- Inline math mode
- Display math mode

Vertical mode

- Creating the output page (list of lines, floats, ...)
- Mode LaTeX is in when starting a document

Commands for manual distance

Commands for vertical distance	
<code>\vspace{<skip>}</code>	vertical distance of size <skip>
<code>\smallskip, \medskip, \bigskip</code>	Distance that is specified by the respective class; special cases of <code>\vspace{}</code>
<code>\smallbreak, \medbreak, \bigbreak</code>	Similar; removes preceding distance marks good point for page break
<code>\addvspace{<skip>}</code>	is merged with other distances to maximum
<code>\vfill, \vspace{\fill}</code>	vertical filling with whitespace
<code>\vspace*{<skip>}</code>	is also inserted after a page break
<code>\vglue{<skip>}</code>	distance at the beginning of a page
Commands for horizontal distance	
<code>\enskip, \quad, \qqquad</code>	horizontal distance of length 0.5em, em, 2em
<code>\hspace{<skip>}</code>	horizontal distance of length <skip>
<code>\hspace*{<skip>}</code>	Analog; is also inserted after a line break (<code>\newline</code>)
<code>\hfill, \hspace{\fill}</code>	horizontal filling with whitespace

Negative distances are also possible

Blank spaces

`\,` small distance (e.g. between number and unit)

`~` normal space without line break: W.~R.~Hamilton

Space after end of sentence (dot after lowercase letter) larger than between words

⇒ Suppress with backslash: Mr.\ Smith, etc.\

⇒ enforce with `\@:` Vitamin A\@.

Manual breaks

Line breaks	
<code>\newline</code>	Line break
<code>\\[<skip>]</code>	short version for line break defined differently in some environments optional vertical distance <code><skip></code>
<code>*[<skip>]</code>	prevents page break after the line break optional vertical distance <code><skip></code>
<code>\linebreak[<number>]</code>	Line break of priority <code><number></code> from 0 (can be ignored) to 4 (force break) Line is pulled to the margin
<code>\break</code>	Line break without filling
<code>\par</code>	New paragraph
Page breaks	
<code>\newpage</code>	Page break
<code>\pagebreak[<number>]</code>	Page break of priority <code><number></code>
<code>\nopagebreak[<number>]</code>	Prohibit page break with priority <code><number></code>
<code>\clearpage</code>	Page break; all floats are placed here
<code>\cleardoublepage</code>	Page break; all floats are placed here next page begins on the right (inserts blank page if necessary)

Debugging:

- ! LaTeX Error: There's no line here to end.
- An `\\` or `\newline` was used incorrectly (to create blank line or vertical spacing).
Instead: `\vspace`, `\vspace*` or `\bigskip`

Indents

Prevent indent: `\noindent`
 Force indent: `\indent`
 Length of indent: `\setlength{\parindent}{<length>}`

Line spacing

`\renewcommand{\baselinestretch}{2}` double line spacing

The `setspace` package

```
\usepackage{setspace}
\usepackage[<spacingoption>]{setspace}                      global
\begin{<spacingoption>}... \end{<spacingoption>}            environment
```

Options:

Global	environment	
<code>singlespacing</code>	<code>singlespace</code>	normal (single) line spacing
<code>onehalfspacing</code>	<code>onehalfspace</code>	one and a half line spacing
<code>doublespacing</code>	<code>doublespace</code>	double line spacing

```
\begin{spacing}{<line spacing as number>}... \end{spacing}
```

Text Alignment

Alignment	Environment	Command
left-aligned	<code>flushleft</code>	<code>\raggedright{<text>}</code>
right-aligned	<code>flushright</code>	<code>\raggedleft{<text>}</code>
centered	<code>center</code>	<code>\centering{<text>}</code>

Fonts

Font types and sizes

Fonts	Local version	Global version
roman serif	<code>\textrm{<text>}</code>	<code>{\rmfamily <text>}</code>
sans serif	<code>\textsf{<text>}</code>	<code>{\sffamily <text>}</code>
typewriter	<code>\texttt{<text>}</code>	<code>{\ttfamily <text>}</code>
Font width		
fett	<code>\textbf{<text>}</code>	<code>{\bfseries <text>}</code>
medium	<code>\textmd{<text>}</code>	<code>{\mdseries <text>}</code>
Font style		
gerade	<code>\textup{<text>}</code>	<code>{\upshape <text>}</code>
kursiv	<code>\textit{<text>}</code>	<code>{\itshape <text>}</code>

<code>\schräg</code>	<code>\textsl{<text>}</code>	<code>{\slshape <text>}</code>
<code>\Kapitalchen</code>	<code>\textsc{<text>}</code>	<code>{\scshape <text>}</code>
Default font	<code>\textnormal{<text>}</code>	<code>{\normalfont <text>}</code>

`\emph{<text>}` highlight/emphasize text (usually in italics)

Font sizes

`\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`,
`\Large`, `\LARGE`, `\huge`, `\Huge`

Underline:

`\usepackage{ulem}` **Attention: overwrites `\emph{}`**
`\normalem` **reset `\emph{}` to italics**

`\uline{<text>}` underline
`\uwave{<text>}` Wavy line
`\sout{<text>}` cross out

Use system fonts

The `fontspec` package

`\setmainfont{}[]`
`\setsansfont{}[]`
`\setmonofont{}[]`

Symbols and Icons

`fontawesome5` package

Full icon list in the package documentation

Lists

Tabulator Environment

```
\begin{tabbing}
text \= text \>[5pt]
text \> text \>[5pt]
\end{tabbing}
```

`\kill` Ends dummy line that is not printed (defining the tab positions)
`\=` Tab positions can be changed again
`\+` at the end of a line; subsequent lines begin with the first tab
`\-` reset `\+`

Bulleted lists

```
\begin{itemize}
\item ...
\end{itemize}
```

Customize:

```
\renewcommand{\label<itemi>}{<character>}
```

Bullet: `\label<level>`

Level: `itemi, itemii, itemiii, itemiv`

Numbered lists

```
\begin{enumerate}
\item ...
\end{enumerate}
```

Customize:

```
\renewcommand{\the<level>}{\<zaehler>{<level>}}
```

```
\renewcommand{\label<level>}{\the<level>}
```

Level: `enumi, enumii, enumiii, enumiv`

Counter: `\arabic, \alph, \Alph, \roman, \Roman, \fnsymbol`

The `enumerate` package:

for more individualization of numbering

A, a, 1, I and i are placeholders for the respective counters (otherwise use brackets)

```
\begin{enumerate}[i.]
\item ...
\end{enumerate}
```

The `enumitem` package

Various options that can be passed to a list:

`itemsep, leftmargin, labelsep, itemindent, nosep`

Both enumeration and references can be defined here:

`label=<aufzaehlungsformat>`

`ref=<verweisformat>`

Debugging:

- ! LaTeX Error: Too deeply nested.
→ Nesting too deep (>4 levels)
- ! LaTeX Error: Something's wrong--perhaps a missing \item.
→ environment is used without the \item command.

- ! LaTeX Error: `\begin{document}` ended by `\end{itemize}`
 → `\end{itemize}` **without** `\begin{itemize}`
- ! LaTeX Error: `\begin{itemize}` on input line XX ended by `\end{document}`.
 → `\end{itemize}` **missing**
- ! LaTeX Error: Lonely `\item`--perhaps a missing list environment.
 → `\item` **command outside of itemize environment**
- ! Missing `\endcsname` inserted.
 → `\begin{\itemize}`
- ! LaTeX Error: Command `\end{itemize}` invalid in math mode.
 → **Within the** `itemize` **environment, a math environment was started but not closed**

Description lists

```
\begin{description}
\item[label] ...
\end{description}
```

Quotes

Environment	Usage
<code>\begin{quote} ...</code> <code>\end{quote}</code>	<ul style="list-style-type: none"> • for short quotations consisting of a single paragraph • slightly indented from left and right, i.e. line length slightly reduced compared to the rest of the text
<code>\begin{quotation}</code> ... <code>\end{quotation}</code>	<ul style="list-style-type: none"> • for longer quotations consisting of more than one paragraph. • also slightly indented from left and right • first line of a new paragraph slightly indented
<code>\begin{verse} ...</code> <code>\end{verse}</code>	<ul style="list-style-type: none"> • for poems • single lines explicitly terminated by <code>\\</code> • very long lines are indented in the following lines and thus marked as belonging together

Footnotes, marginal notes etc.

Footnotes

`\footnote{<text>}`

- Footnotes in `minipage` environment have their own counter (`mpfootnote`) and are placed at the bottom of the `minipage`
- Footnotes in tables are only set correctly in `tabularx` and `longtable` environments

Customize:

Counter	
<code>\renewcommand{\thefootnote}{\<zaehler>{footnote}}</code>	Change counter. Possible counters, see numbered lists
<code>\setcounter{footnote}{<zahl>}</code>	Set counter to specific value
<code>\@addtoreset{footnote}{section}</code>	Reset counter to 0 with every new section
<code>\footnote[<num>]{<text>}</code>	num replaces the actual counter of the footnote; counter is not incremented
Formatting	
<code>\footnotesize</code>	Font size
<code>\footnotesep</code>	Additional distance above footnote
<code>\setlength{\skip\footins}{<height>}</code>	Space between main text and footnotes
<code>\footnoterule</code>	Line separating the footnotes from the main text

Reference to previous footnote

Works only in the KOMA-Script classes

`\footref{<key>}`

The endnotes package:

`\endnote{text}` instead of `\footnote{text}`

or:

`\renewcommand{\footnote}{\endnote}`

Group notes at the end of a chapter etc. via `\theendnotes\bigskip`

Margin notes

```
\marginpar{text}
```

Customize:

<code>\normalmarginpar</code>	margin notes on the default page margin (outside)
<code>\reversemarginpar</code>	margin notes on the other side margin (inside)
<code>\marginparwidth</code>	Width of marginal notes
<code>\marginparsep</code>	Space between marginal note and main text
<code>\marginparpush</code>	minimum distance between two marginal notes

Colors

```
\usepackage{color}  
\usepackage[usenames,dvipsnames,svgnames]{xcolor}
```

<code>usenames</code>	Names of the 16 basic colours (HTML)
<code>dvipsnames</code>	additional 64 colors
<code>svgnames</code>	additional 150 colors
<code>x11names</code>	additional 300 colors

define your own colors:

```
\definecolor{<mycolor>}{<model>}{<color-spec>}
```

Color models: gray, rgb, RGB, HTML, cmyk

color-spec different depending on the color model

```
\colorlet{<mycolor>}{<color1>!<percentage>!<color2>}  
Mixes percentage% of color1 with (100-percentage)% of color2
```

Use colours:

```
\textcolor{<color>}{<text>}  
\pagecolor{<color>}
```

Boxes

Boxes are treated like characters

Position options

t	top
b	bottom
l	left
r	right
c	center
s	stretch

LR (left-right) Boxes

`\makebox[<width>][<pos>]{<text>}` <width> as linear measure
`\makebox(<width>,<height>)[<pos>]{Text}` <width> and <height> as number
`\mbox{<text>}`

Box with frame

`\framebox[<width>][<pos>]{<text>}`
`\fbox{<text>}`

`\fboxrule` Line width (default 0.4pt)
`\fboxsep` distance between frame and content (default 3pt)
`\raisebox{<lift>}[<height>][<depth>]{<text>}`

Par (paragraphs) Boxes

Content is set in paragraph mode

`\parbox[<pos>]{<width>}{<text>}`

Position:

t top line aligned with baseline
m center aligned to baseline
b bottom line aligned with baseline

`\begin{minipage}[<pos>]{<width>}`
text
`\end{minipage}`

Rules / Bars

`\rule[<lift>]{<width>}{<height>}`

Color Boxes

`\colorbox{<color>}{<text>}`
`\colorbox{<background_color>}{\textcolor{text_color}{<text>}}`
`\fcolorbox{<frame_color>}{<background_color>}{<text>}`

The tcolorbox package

`\begin{tcolorbox}[<options>]`
<content>
`\tcblower`
<content>
`\end{tcolorbox}`

`\tcblower` creates dashed dividing line (optional)

Options:

<code>colback=<color></code>	background color
<code>colframe=<color></code>	frame color
<code>colbacktitle=<color></code>	color of the title
<code>coltext=<color></code>	color of the text
<code>title=<text></code>	title
<code>fontupper=</code>	font main text e.g. <code>\Large\bfseries</code>
<code>arc=<length></code>	arc curvature
<code>boxrule=<line width></code>	line width

Chapter 3: Figures and tables

Importing images

The `graphicx` package

- Supported image formats depend on the driver used by `graphicx` \Rightarrow is automatically selected depending on the compiler
 - `latex` compiler can import EPS and PS
 - `pdflatex` compiler can import JPG, PNG and PDF

```
\includegraphics[<options>]{<image>}
```

```
Global image directory: \graphicspath{{dir1/}{dir2/}}
```

Options:

<code>width=xx</code>	width of the image
<code>height=xx</code>	height of the image
<code>keepaspectratio</code>	image is not distorted during scaling
<code>scale=xx</code>	scaling factor
<code>angle=xx</code>	rotation by <code>xx</code> degrees (counterclockwise)
<code>trim=<l> <r> <t></code>	trimming of the image (<code>trim = 10mm 80mm 20mm 5mm</code>)
<code>clip</code>	necessary for <code>trim</code> to work

```
\reflectbox{} flip image
```

Floats

- Optimized placement of floats for beautiful page layout in continuous text mode
- Avoids cluttered paragraphs and half-empty pages

```
\begin{<float>}[<placement>]
<content>
\end{<float>}
```

Placement options:

<code>h</code>	Float is inserted in the text approximately at this spot
<code>t</code>	Placement at the top of the page
<code>b</code>	Placement at the bottom of the page
<code>p</code>	Placement on special page only for floats
<code>!</code>	Overwriting of the internal LaTeX parameters
<code>H</code>	Placement at precisely this location. Requires the <code>float</code> package, though may cause problems occasionally. This is somewhat equivalent to <code>h!</code>

Sliding objects at the bottom of the page might appear below footnotes

⇒ use the `footmisc` package to avoid: `\usepackage[bottom]{footmisc}`

Debugging:

- ! LaTeX Error: Not in outer par mode.
 float has been used within another environment (e.g. within `minipage`, `parbox`, `tabular`, `framebox`, `table`, `figure`, `math environments`)

Figures

```
\begin{figure} [<placement>]
\includgraphics{<figure>}
\caption{text}
\label{key}
\end{figure}
```

Tables

```
\begin{table}[placement specifier]
\caption{text} \label{key}
\begin{tabular}{...}
... table data ...
\end{tabular}
\end{table}
```

Define own floats

The `float` package

```
\newfloat{<type>}{<placement>}{<ext>}[<within>]
```

type Name of the new floating object command
 placement default placement (*t*, *b*, *p*, *h*) — can be overwritten
 ext File extension for object list (corresponds to *lof*, *lot* for figure and table)
 within Numbering

```
\floatname{<type>}{<floatname>}    Actual name of the new floating object
\floatstyle{<style>}            affects all following newfloat commands
```

Style options:

plain normal
 boxed framed float; caption below the frame
 ruled Caption above float; delimitation by horizontal lines

```
\restylefloat{<type>}            apply the new style also to already defined floats
                                  (e.g. figure and table)
```

Float placement

- for two-column text: `table*` and `figure*` environments span both columns
- `\label{<key>}` after `\caption{}`; can be referenced via `\ref{<key>}` or `\pageref{<key>}`
- without `\caption{}` command an unnumbered figure/table is generated
- several `\caption{}` commands create several figures/tables within a float
- `\caption[<short>]{<long>}` optional argument for short version of caption in `\listoffigures` or `\listoftables`
- counter `figure` and `table` can be changed using `\setcounter{<counter>}{<number>}`
- floats cannot be used within a `parbox` or `minipage`

`\clearpage` inserts all floats and a page break
`\FloatBarrier` inserts all floats without page break (`placeins` package)

The `placeins` package:

`\usepackage[section]{placeins}` `\FloatBarrier` command before each section
`\usepackage[below]{placeins}` similar, but less restrictive

Adjust placement parameters:

`\setcounter{<parameter>}{<number>}`
`topnumber` max number of floats at the top of the page (default 2)
`bottomnumber` max number of floats at the bottom of the page (default 1)
`totalnumber` max number of floats per page (default 3)

Adjust float-text ratio:

`\renewcommand{<command>}{<definition>}`
`\textfraction` Minimum percentage of a text page that must be filled with text (default 0.2)
`\topfraction` Maximum proportion of the upper part of the page that may be filled with floats (default 0.7)
`\bottomfraction` Maximum proportion of the lower part of the page that may be filled with floats (default 0.3)
`\floatpagefraction` Minimum portion of a float page that must be filled with floats (default 0.5)

The wrapfig package

```
\begin{wrapfigure} [<numberoflines>] {<placement>} [<overhang>] {<width>}
...
\end{wrapfigure}
```

There is also a wraptable environment

<numberoflines> Number of lines to be spanned
 <overhang> Overhang beyond the text margin

Placement options: Capital letters ⇒ float; small letters ⇒ exact placement

r R	right
l L	left
i I	inside (for twoside)
o O	outside (for twoside)

Float captions

The subcaption package

```
\begin{subfigure} [<pos>] {<width>} ... \caption{<text>} \end{subfigure}
\begin{subtable} [<pos>] {<width>} ... \caption{<text>} \end{subtable}

\subcaption{<text>}
```

The caption package

Further `\caption*{...}` in addition to normal `\caption{...}`
 (e.g. as table legend)

The sidecap package

Placing captions right/left of the float

```
\begin{SCfigure} [<relativecaptionwidth>] [<placement>] ... \end{SCfigure}
\begin{SCTable} [<relativecaptionwidth>] [<placement>] ... \end{SCTable}
```

`relativecaptionwidth` width of the caption relative to the size of the float

Package options:

<code>rightcaption / leftcaption</code>	Caption right / left of the float
<code>outercaption / innercaption</code>	Caption outside / inside
<code>wide</code>	float may extend beyond the text margin
<code>raggedright / raggedleft / ragged</code>	better text alignment for narrow captions

Tabulars

```
\begin{tabular} [<position>] {<columns>} <rows> \end{tabular}
```

Position parameters:

vertical positioning of the table

t upper edge aligned with baseline
 b lower edge aligned with baseline
 c centered

Columns

Column types: one argument per column

l left-justified column
 r right-justified column
 c centered column
 p{<width>} Paragraph column with fixed width (left justified)
 | vertical line (|| also possible)
 @{<text>} intercolumnar text
 e.g. @{\hspace{width}} or r@{.}l for decimal places
 *{<num>}{<columns>} several identical columns in succession, e.g. *{15}{c}

Column modifiers: effect entire column

>{<command>} Commands that are executed before the column content
 <{<command>} Commands that are executed after the column content

When changing text alignment of the column (e.g. using `\raggedright`), `\arraybackslash` is needed to restore the function of `\` (array package)

The array package

enables paragraph columns

p{<width>} Paragraph column with top aligned text
 m{<width>} Paragraph column with vertically centered text
 b{<width>} Paragraph Column with bottom aligned text

The tabularx package

Define your own column types

```
\newcolumntype{<name>} [<arg>] {<column definition>}
```

tabularx environment

X columns are extended to the width of the table

```
\begin{tabularx}{<width>}{ |X|X|X|X| } ... \end{tabularx}
```

The tabulary package

```
\begin{tabulary}{<width>}{<col>} ... \end{tabulary}
```

L left-aligned, balanced column
C centered, balanced column
R right-aligned, balanced column
J justified, balanced column

The dcolumn package

Decimal places `D{<separator>}{<output separator>}{<decimal places>}`
`D{.}{,}{-1}` use -1 to not limit the number of decimal places

Debugging:

- Extra alignment tab has been changed to `\cr`.
more columns were used than were defined. Possibly a non-masked `&` was used.
- ! LaTeX Error: Missing p-arg in array arg.
width specification is missing for p column
- ! LaTeX Error: Missing @-exp in array arg.
The @ column element is missing the argument
- ! LaTeX Error: Illegal character in array arg.
An invalid letter was used within the column definition of a table

Rows

<code>&</code>	Separator between the columns
<code>\\</code>	new row
<code>\hline</code>	horizontal line
<code>\cline{i-j}</code>	horizontal line from column i to column j
<code>\vline</code>	vertical line of the height of the line

Merge cells

Horizontally merge cells

```
\multicolumn{num_col}{alignment}{content}
```

The multirow package

```
\multirow{<num_rows>}{<width>}{<content>}
```

- adopts natural width
- The following rows must contain empty cells for the respective merged cells

```
\renewcommand{\multirowsetup}{\centering}     default: \raggedright
```

Individualize tabulars

Change table parameters:

`\setlength{<parameter>}{<value>}`
`\tabcolsep` Half the distance between the columns
`\arrayrulewidth` Width of vertical and horizontal lines
`\doublerulesep` Distance between two lines

`\renewcommand{\arraystretch}{<factor>}` Spacing between the rows

The booktabs package: for nicer lines

`\toprule`
`\midrule`
`\bottomrule`

The colortbl package

- For colored tables
- Can also be loaded via the `table` option with the `xcolor` package

`\rowcolors{<starting_row>}{<odd_color>}{<even_color>}`

Define before tabular environment

`\hiderowcolors` deactivates table coloring; at the beginning of the row
`\showrowcolors` activates table coloring; at the end of the row
`\cellcolor{color}`

`\rowcolor{color}` colored row
`>{\columncolor{<color>}}` colored column

Large tables

The longtable package

instead of `\begin{tabular} ... \end{tabular}`

`\begin{longtable}`
`... \endfirsthead ... \endhead ... \endfoot ... \endlastfoot`
`\end{longtable}`

`\endfirsthead` Table header on first page
`\endhead` Table header on the following pages
`\endfoot` Table footer on all pages
`\endlastfoot` Table footer on the last page

The pgfplotstable package

Import tables from .csv files

<https://www.latex-tutorial.com/tutorials/pgfplotstable/>

Spreadsheet

The spreadtab package

```
\usepackage{spreadtab}
```

```
\begin{spreadtab}{<tabletype>}{<parameters>}
```

```
...
```

```
\end{spreadtab}
```

Table cell is identified by the pair <colref><rowref> (absolute) or [<horizontal offset>, <vertical offset>] (relative).

Indicate text field by @

Mixed cells: <text> :={<number>} <text>

Rotation

The rotating package

```
\begin{sideways} ... \end{sideways}      90 degree counterclockwise rotation
\begin{turn}{<angle>} ... \end{turn}      Rotation by <angle> degrees;
                                           inserts necessary spacing
\begin{rotate}{<angle>} ... \end{rotate}  Rotation by <angle> degrees without
                                           additional spacing
\begin{sidewaysfigure} ... \end{sidewaysfigure}
\begin{sidewaystable} ... \end{sidewaystable}
printed on an extra page
```

Package options:

counterclockwise	in one-sided documents
anticlockwise	synonym for counterclockwise
clockwise	in one-sided documents
figuresright	in two-sided documents; all on the right side
figuresleft	in two-sided documents; all on the left side

The rotfloat package

allows to use H as a placement argument

```
\begin{sidewaystable}[H] ... \end{sidewaystable}
```

Chapter 4: Math and algorithms

Packages of the AMS (American Mathematical Society)

`\usepackage{amsmath}` for mathematical formulas

`\usepackage{amssymb}` for mathematical symbols

Math environments

- LaTeX formats mathematical notation differently than normal text
- the entered letters and symbols are interpreted as part of a formula
- In formulas, letters are always interpreted as names of variables and are therefore printed in italics
- Spaces within formulas are ignored and must be set manually using spacing instructions such as `\,` or `\;`
- Blank lines are not allowed
- LaTeX determines spaces within formulas using its own internal rules

mode		environment	LaTeX short version	TeX short version
Text / inline	formulas are displayed inline within the text	<code>\begin{math}</code> <code>\end{math}</code>	<code>\(...\)</code>	<code>\$. . . \$</code>
display	formulas as independent centered paragraph	<code>\begin{displaymath}</code> <code>\end{displaymath}</code>	<code>\[...]</code>	<code>\$\$. . . \$\$</code>
equation	also adds numbering	<code>\begin{equation}</code> <code>\end{equation}</code>		

Commas in math environments are no decimal separators but item separators.

For decimal separators use `{,}`.

Equations

For each equation environment, there is a numbered and an unnumbered (*) version.

Automatically create the math mode

`equation` single formula on a single line with a single number

No line break possible

`gather` several formulas, one formula per line, each line numbered separately

line break using `\\`

`multline` single formula ranging over several lines with a single number

`align` line break using `\\`
 first line aligned to the left; second line aligned to the right
 if there are more than two lines, the middle lines are centered
 single formula ranging over several lines with a single number;
 two column alignment using `&` (if possible after the `=`)
 line break using `\\`

`\nonumber \\` suppresses numbering for this line
`\numberwithin{equation}{section}` sectionwise numbering

All numbered environments for equations can be labelled using `\label{<key>}` and referenced in the text using `\ref{<key>}` or `\eqref{<key>}` (equation number in round brackets).

Debugging:

- `! Missing $ inserted.`
 → e.g. if blank lines have been inserted in a maths environment

Matrices

`\begin{array}{ccc} ... \end{array}`

`\begin{matrix}`
`... & ... & ... \\`
`... & ... & ... \\`
`\end{matrix}`

`vmatrix` `||`
`Vmatrix` `|||`
`pmatrix` `()`
`bmatrix` `[]`
`Bmatrix` `{ }`

`\bordermatrix{}`
`\begin{smallmatrix} ... \end{smallmatrix}` in-text matrices
`\begin{cases} ... \end{cases}` for distinction of cases

Fonts and symbols

Symbols in LaTeX

The most important ones: [LATEX Mathematical Symbols](#)
 all: [The Comprehensive LATEX Symbol List](#)

Greek letters: `\alpha \Alpha ...`

Fonts in math mode

Font formatting in math mode

<code>\mathrm{...}</code>	roman
<code>\mathit{...}</code>	italics
<code>\mathbf{...}</code>	bold
<code>\mathsf{...}</code>	sans serif
<code>\mathtt{...}</code>	typewriter
<code>\mathcal{...}</code>	calligraphy (only capital letters)
<code>\mathbb{...}</code>	"board" letters
<code>\mathfrak{...}</code>	Fraktur
<code>\mathscr{}</code>	requires <code>\usepackage{mathrsfs}</code>

Text in math mode

To print text in math mode use `\mbox{<text>}` or `\text{<text>}`

Formulas

Greek letters

`\alpha \beta \gamma \rho \sigma \delta \epsilon`

Relation operators

`< > \subset \supset \subseteq \supseteq`

Binary operators

`\bigoplus \bigotimes \bigodot \bigcup \bigcap \biguplus \bigsqcup`
`\bigvee \bigwedge`

Further functions

`\sum \prod \coprod`
`\int \oint \iint \iiint \idotsint`
`\arccos \arcsin \arctan \arg \cos \cosh \cot \coth \csc \deg \det \dim`
`\exp \gcd \hom \inf \ker \lg \lim \liminf \limsup \ln \log \max \min`
`\Pr \sec \sin \sinh \sup \tan \tanh \to \infty`
`\substack` allows line breaks via `\\` to write limits (from...to) over several lines

`\frac{<numerator>}{<denominator>}`

`\cfrac{ }{ }` for nested fractions

`\binom{ }{ }`

`\sqrt[<n>]{<expression>}`

`\neq`

`\limits` sets limits in e.g. fractions also above/below the sum,...

`x_{y}, x^{y}` Sub-/superscript
`\overset{}{}` `\underset{}{}`

Greek letters: `\alpha`, `\beta`, `\Gamma`, `\gamma`, ...

Ellipses:

`\ldots` ellipsis aligned with the baseline of the text
`\cdots` ellipsis aligned to the centerline of the text
`\dotsc` ellipsis between commas
`\dotsb` ellipsis for operators
`\dotsm` ellipsis in multiplications
`\dotsi` ellipsis for integrals

Brackets:

`\Biggl(` `\biggl(` `\Bigl(` `\bigl(` `\bigr)` `\Bigr)` `\biggr)` `\Biggr)`
`\Biggl|` `\biggl|` `\Bigl|` `\bigl|` `\bigr|` `\Bigr|` `\biggr|` `\Biggr|`
`\Biggl\{` `\biggl\{` `\Bigl\{` `\bigl\{` `\bigr\}` `\Bigr\}` `\biggr\}` `\Biggr\}`

Bracket size defined by LaTeX:

`\left|` `\right|`
`\left(` `\right)`
`\left\{` `\right\}`

Define your own operators

`\DeclareMathOperator{\<operatorname>}{<operatordefinition>}`

Theorems

- different font to distinguish it from the surrounding text
- name and number for referencing

Define theorem environment: (e.g. for lemmas, tasks, ...)

`\newtheorem{<mytheorem>}[<counter>]{<name>}[<numbering>]`

`<counter>` name of an already defined theorem environment, if the counter should be the same

`<numbering>` optional argument for numbering e.g. section

`\begin{<mytheorem>}[<note>] ... \end{<mytheorem>}`

The `amsthm` package

contains several styles for theorems

```
\theoremstyle{<stylename>}
all theorems defined below use this style
```

Styles:

`plain` common for theorems, lemmas, propositions
`definition` common for definitions, examples
`remark` common for comments

Theorem 1. *Theorem text.*
Definition 2. Definition text.
Remark 3. Remark text.

define your own styles:

all arguments that are left empty take default values

```
\newtheoremstyle
{name}%            name of the new style
{abovespace}%      vertical distance to the previous text
{belowspace}%      vertical distance to the following text
{bodyfont}%        font of the text (e.g. \scshape or \bfseries)
{indent}%           indentation of the header
{headfont}%        font of the header
{headpunct}%       punctuation after the header
{headspace}%       (horizontal) space between header and text
{headspect}%       individual header;
                    contains \thmname{#1}, \thmnumber{ #2} and \thmnote{ #3}
```

Proofs

```
\begin{proof}[<name>] ... \end{proof}
```

Algorithms

Floats for algorithms

The `algorithm` package

```
\usepackage[<numbering>]{algorithm}
\begin{algorithm} ... \end{algorithm}
```

`<numbering>` Numbering of the floats by section, chapter etc.
 Use `default` for continuous numbering throughout the document

```
\listofalgorithms        creates a table of algorithms
```

Pseudocode

The `algpseudocode` package

```
\begin{algorithmic}[<step>] ... \end{algorithmic}
```

<step> line numbering

Basic commands of the pseudocode environment:

```
\State <text>
\If{<condition>} \State{<text>} \Else \State{<text>} \EndIf
\For{<condition>} \State{<text>} \EndFor
\For{<condition> \To <condition> } \State{<text>} \EndFor
\ForAll{<condition>} \State{<text>} \EndFor
\While{<condition>} \State{<text>} \EndWhile
\Repeat \State{<text>} \Until{<condition>}
\Loop \State{<text>} \EndLoop
\Require <text>
\Ensure <text>
\Return <text>
\Print <text>
\Comment{<text>}
\And, \Or, \Xor, \Not, \To, \True, \False
```

Sourcecode

Formatted text

```
\begin{verbatim} ... \end{verbatim}
LaTeX commands are ignored; all spaces are printed
```

The `listings` Package

```
\begin{lstlisting}[<options>]
...
\end{lstlisting}
```

Import source code from file

```
\lstinputlisting[language=<language>, firstline=<start>,
lastline=<stop>]{<source-file>}
```

Customize Code

Possible settings:

```
\lstset{
backgroundcolor=\color{white},
basicstyle=\footnotesize,                  font size
commentstyle=\color{mygreen},
```

<code>escapeinside={\%*}{*},</code>	to add LaTeX code
<code>extendedchars=true,</code>	enables the use of non-ASCII characters (only for 8-bit encodings)
<code>breaklines=true,</code>	automatic linebreaks
<code>keepspaces=true,</code>	keeps code indentation
<code>title=\lstname,</code>	file name
<code>captionpos=b,</code>	
<code>frame=single,</code>	
<code>rulecolor=\color{black},</code>	frame color
<code>language=Octave,</code>	
<code>keywordstyle=\color{blue},</code>	
<code>deletekeywords={...},</code>	delete keyword of the selected language
<code>otherkeywords={*,...},</code>	add additional keywords
<code>stringstyle=\color{mymauve},</code>	
<code>numbers=left,</code>	placing of the line numbers (none, left, right)
<code>numbersep=5pt,</code>	spacing between line numbering and code
<code>numberstyle=\tiny\color{mygray},</code>	
<code>stepnumber=2,</code>	stepsize of line numbering
<code>showspaces=true,</code>	show spaces using underscores
<code>tabsize=2</code>	
<code>}</code>	

The `minted` package

- requires the program Pygments (syntax highlighter) <https://pygments.org/>
- In Overleaf the package works "out of the box"

```
\begin{minted}[<options>]{<language>}
...
\end{minted}
\inputminted[<options>]{<language>}{<filename>}
```

Options

<code>frame=lines</code>	
<code>framesep=2mm</code>	
<code>baselinestretch=1.2</code>	
<code>bgcolor=<color></code>	
<code>fontsize=\<fontsize></code>	
<code>linenos</code>	
<code>mathescape</code>	enabled math mode in comments
<code>rulecolor=<color></code>	
<code>showspaces</code>	
<code>firstline=<linenumber></code>	only for <code>\inputminted</code>
<code>lastline=<linenumber></code>	only for <code>\inputminted</code>

```
\usemintedstyle{<style>}
```

Floats for listings

```
\begin{listing}
```

```
...
```

```
\caption{<caption>}
```

```
\label{<key>}
```

```
\end{listing}
```

```
\renewcommand\listoflistingscaption{List of source codes}
```

```
\listoflistings
```


Book structure

```
\documentclass[<options>]{scrbook}
```

default options: `letterpaper`, `10pt`, `twoside`, `onecolumn`, `final`, `openright`

Structure

```
\begin{document}
\frontmatter                roman page numbers
\maketitle                  title
\chapter{Preface}           unnumbered chapters; do not insert sections
\tableofcontents
%
\mainmatter                 arabic page numbers; starts again with page 1
\chapter{First chapter}    all chapters that contain the actual content
...
%
\appendix                  Chapters and Sections are numbered as appendix (A,...)
\chapter{First Appendix}
%
\backmatter                unnumbered chapters
\chapter{Last note}
\bibliographystyle{abbrv}
\bibliography{book}        bibliography
```

Design title pages

Design your own title page (instead of using `\maketitle`)

```
\thispagestyle{empty}
```

... self-designed front page ...

```
\cleardoublepage
```

Page break, next page starts on the right (empty page if necessary)

Hyperlinks and metadata

The `hyperref` package

- links and references within PDF documents
- works only for `pdflatex`
- change settings within the PDF document
- uses already existing references (`\label` and `\ref` system, headings, the bibliography, etc.)
- should be included as last package (but before `geometry` package)

```
\usepackage[<options>]{hyperref}
```

```
\hypersetup{<options>}
```

Options

option	description	possible values	default value
Document metadata:			
pdftitle=<text>	title		
pdfauthor=<text>	author(s)		
pdfkeywords=<text>	keywords		
pdfsubject=<text>	topic/subject		
Link options:			
colorlinks=<boolean>	colored link text instead of frame		false
linkcolor=<color>	links withing document		red
anchorcolor =<color>	color of text links		black
citecolor=<color>	literature references		green
filecolor=<color>	Local files		magenta
pagecolor=<color>	other opages		red
urlcolor=<color>	external URL links		cyan
frenchlinks=<boolean>	small caps, instead of color		false
View options			
bookmarks=<boolean>	creates bookmarks (table of contents)		false
bookmarksopen=<boolean>	open table of contents when opening the file		false
pdfstartview=<value>	Opening view (e.g. adjust to window size)	Fit, FitH, FitV, FitR, FitB, FitBH, FitBV	FitB
pdfpagelayout=<value>	Page layout at opening	SinglePage, OneColumn, TwoColumnLeft, ...	empty

Examples only; further options in package description

Referencing using `hyperref`

`\autoref{<key>}` returns not only numbering but also type of the referenced object (e.g. Figure)

`\nameref{<key>}` does not reflect the numbering but the name of the referenced object
(e.g. the section heading)

Adaptation of the default names for `\autoref`:
`\renewcommand{\figureautorefname}{<text>}`

Bibliography

```
\begin{thebibliography}{<widest-label>}
\bibitem{<key1>} <reference1>
\bibitem{<key2>} <reference2>
\end{thebibliography}
```

`<widest-label>` not the number itself is the parameter, but the number of digits

```
\cite{<key>}
\cite[<text>]{<key>}
\cite{<key1>,<key2>}
```

Literature management using JabRef

automatically generated keys:

<code>[auth]</code>	Last name of the first author
<code>[authorLast]</code>	Last name of the last author
<code>[shorttitle]</code>	the first three words of the title
<code>[veryshorttitle]</code>	the first word of the title
<code>[year]</code>	Year of publication
<code>[shortyear]</code>	the last two digits of the year

<code>:abbr</code>	abbreviation
<code>:lower</code>	lowercase letters
<code>:upper</code>	uppercase letters

Bibliography using BibTeX

```
\bibliographystyle{<citation_style>}
\bibliography{<bibliographyLower case_file>}
several files possible
```

Citation styles:

- Format of the abbreviation in continuous text: e.g. numbers, text
- Format of the entries in the bibliography: e.g. sorted, unsorted, abbreviations, ...

`abbrv, acm, alpha, apalike, ieetr, plain, siam, unsrt, ...`

Format of citation in the text:

```
\bibpunct[<nr1>]{<nr2>}{<nr3>}{<nr4>}{<nr5>}{<nr6>}{<nr7>}
<nr1>      Character before <post> information
<nr2>      open parenthesis
<nr3>      close parenthesis
<nr4>      characters between several citations (comma, semicolon, ...)
<nr5>      reference: n - numeric; s - superscript; a - author-year default
<nr6>      character between author and year
<nr7>      character between several citations from same author
```

The biblatex package

Package options:

```
style=numeric,
citestyle=numeric-comp,
sorting=none,
backend=bibtex,
maxcitenames=1,           number of names in the text
maxbibnames=3,           Number of names in the bibliography
minbibnames=3,           Number of names in the bibliography
natbib=true,             to use natbib citation commands
url=false,
isbn=false,
doi=false,
autocite = superscript,
giveninits=true
```

```
\addbibresource{<bibfile>} File with literature database (can already be found in
                             preamble be called)
\printbibliography         at this point the bibliography is generated
```

Bibliography using biber

- alternative to BibTeX
- Bibliography processor especially for biblatex
- uses BibTeX only for sorting, not for formatting
- does not need a .bst file
- full Unicode support, which is difficult to achieve with BibTeX
- solves all BibTeX problems (correct sorting due to Unicode support, memory requirements, encodings etc.)
- several bibliographies possible
- some LaTeX packages depend explicitly on BibTeX and do not work with biber

- instead of `natbib` the package `biblatex` must be used here
- `biblatex` package has to be loaded with `backend=biber` to generate `foo.bcf` is written
- `.bcf` is short for "biber control file" and contains instructions (which bib file, which sorting etc.)

```
\usepackage[backend=biber]{biblatex}  
\addbibresource{<bibfile>}  
\printbibliography
```

List of abbreviations

The `acronym` package

options:

<code>footnote</code>	Output long version as footnote
<code>nohyperlinks</code>	prevent automatic linking by <code>hyperref</code>
<code>printonlyused</code>	list only abbreviations that are actually used
<code>smaller</code>	reduced displaying of abbreviations in the document
<code>dua</code>	always output the abbreviation in long version
<code>nolist</code>	no list of all abbreviations created

```
\begin{acronym}[<longestabbrevaiation>]  
\acro{<abbreviation>}[<abbreviation>]{<long version>}  
\end{acronym}
```

In the text:

<code>\ac{<abbrev>}</code>	first use is written out, all others are not
<code>\acs{<abbreviation>}</code>	abbreviation only
<code>\acf{<abbreviation>}</code>	long version
<code>\acl{<abbrev>}</code>	long version without abbreviation in brackets
<code>\acp{<abbrev>}</code>	plural (also works for <code>\acsp</code> , <code>\acfp</code> , <code>\aclp</code>).

→ is always completed with `s`; alternatively:

```
\acroplural{<abbrev>}[<short plural>]{<long plural>}
```

Create index

The `makeidx` package

`\makeindex` (must be called in preamble)

Create index entries throughout the text using:

```
\index{<term>}
```

```
\index{<term>!<subterm>}
```

```
\index{<term>!<subterm>!<subsubterm>}
```

creates an entry `<term>` in the index, which refers to the current page

```
\printindex      Index ausgeben
```

Creating the document:

Similar to generating the bibliography:

```
makeindex <mydoc>      Creating the .idx file
```

Chapter 6: Special LaTeX Documents

Letters

The `g-brief` class

```
\documentclass[<options>]{g-brief}
\Name{}
\Strasse{}
...
```

```
\trennlinien
\lochermarke
\faltmarken
\fenstermarken
```

CV

Design your own CV

- CVs can be freely designed in LaTeX
- a suitable class would be for example `scrartcl`
- In particular using the `tabular` environment

CV templates

A large selection of templates for resumes can be found on Overleaf:

<https://www.overleaf.com/latex/templates/tagged/cv>

Scientific publications

- depending on the field and journal it is common or unwanted to submit publications in LaTeX
- a large selection of templates can be found on Overleaf:
<https://www.overleaf.com/latex/templates/tagged/academic-journal>
- Of course, publications can also be freely designed

Presentations with LaTeX

The `beamer` class

```
\documentclass{beamer}  
\usetheme{<beamertheme>}
```

Slide structure

```
\frame{  
\frametitle{<Title of the slide>}  
... }
```

`\section{}` and `\subsection{}` for structuring (outside the slides)

Animation

using `<>` or `\pause`

Blocks

```
\begin{block}{<Name>}  
<content>  
\end{block}
```

`block` for information

`exampleblock` for examples

`alertblock` for particularly noteworthy information

Audio and video material

The `multimedia` package

```
\movie[<attributes>]{<movie-label>}{<moviefile>}  
\sound[<attributes>]{<sound-label>}{<soundfile>}
```

Dual display notes:

<https://www.scivision.dev/beamer-latex-dual-display-pdf-notes/>