

ACADEMIC SKILLS IN COMPUTER SCIENCE

Lecture 9: Typesetting in LaTeX

Sebastian Rudolph

Computational Logic

Slides by Markus Krötzsch

TU Dresden, 15th June 2020

Goals for today

Learning goals of this lecture:

- (1) Learn about typesetting reports in LaTeX
- (2) Get to know some of the most important LaTeX commands

Typesetting with LaTeX

LaTeX is the most widely used software for creating research reports in computer science and various other sciences (especially those that require mathematics).

- Often stylised as \LaTeX
- First version by Leslie Lamport (1980s), on top of TeX (Donald Knuth, 1978–1986)

Advantages of LaTeX

- Relatively simple approach of producing relatively high-quality documents
- Excellent support for presenting mathematical expressions
- Free software that runs on every major operating system
- LaTeX sources can be managed like program code, e.g., in a git repository
~> excellent support for versioning, change tracking, and collaboration

Basic Approach

Basic idea: authors write source text in a mark-up/programming language, which is then compiled into PDF documents

Several software tools can be used for the compilation

- Most widespread today is the implementation pdfTeX

Running the command `pdflatex somefile.tex` produces a PDF document from the file `somefile.tex`.

- Another modern implementation is luaTeX
- Several graphical editors are available (calling suitable LaTeX compilers internally)
- ShareLaTeX provides an online service for working on LaTeX sources

Basic Approach

Basic idea: authors write source text in a mark-up/programming language, which is then compiled into PDF documents

Several software tools can be used for the compilation

- Most widespread today is the implementation pdfTeX

Running the command `pdflatex somefile.tex` produces a PDF document from the file `somefile.tex`.

- Another modern implementation is luaTeX
- Several graphical editors are available (calling suitable LaTeX compilers internally)
- ShareLaTeX provides an online service for working on LaTeX sources

Besides a compiler, there are many LaTeX macro packages for extra functionality.

The best way to get everything you need is to install a **LaTeX distribution**:

- **TeX Live** is the leading distribution supporting all major operating systems
Linux distributions provide TeX Live packages and automatic updates
- **MiKTeX** is a popular distribution used on Microsoft Windows
Includes tools to make up for Windows's lack of software package management

A basic LaTeX document

```
\documentclass[a4paper,11pt]{article}

\usepackage[UKenglish]{babel} % set document language
\usepackage[utf8]{inputenc} % use UTF8 character encoding for files
\usepackage[T1]{fontenc} % better hyphenation & PDF quality

\begin{document}
This is where you write the text.
These words will appear on the page in the PDF.
\end{document}
```

The part before the actual text is called the **preamble**. It includes:

- A **document class** (article here; others include, e.g., book)
 - It defines many things (layout, fonts, commands)
 - Options can customise details (e.g., paper format and base font size)
- **Package inclusions** that load extra functionality or define global settings
- **Macro definitions** and other settings as needed

Warning

As a programming language, (La)TeX is not modern!

- Most syntactic sanity is based on conventions, not design
- Symbols (macros, “variables”, etc.) are global without much protection
- Sources are not “parsed” but “interpreted”
- (Misguided) commands can process the whole document
- Syntax errors are often detected only when interpreting the code, and error reporting is often not helpful
- Macro packages can influence each other in surprising ways

Do not despair. Most errors are easy to track down using a web search.

Writing text in LaTeX

The document body of a tex file is mostly text with interspersed markup commands

LaTeX mark-up is used for formatting (e.g., in `\textbf{bold}` or `\textit{italics}`), and for special symbols (e.g., `\{e}` or `\&`).

Some commands need several parameters, e.g., to

```
\href{https://tu-dresden.de/}{link to TU Dresden}.
```

Comments start with percent `%` and extend to the end of line

Writing text in LaTeX

The document body of a tex file is mostly text with interspersed markup commands

LaTeX mark-up is used for formatting (e.g., in `\textbf{bold}` or `\textit{italics}`), and for special symbols (e.g., `\'e` or `\&`). Some commands need several parameters, e.g., to `\href{https://tu-dresden.de/}{link to TU Dresden}`. Comments start with percent `%` and extend to the end of line

produces:

LaTeX mark-up is used for formatting (e.g., in **bold** or *italics*), and for special symbols (e.g., é or &). Some commands need several parameters, e.g., to link to TU Dresden. Comments start with percent

Writing text in LaTeX

The document body of a tex file is mostly text with interspersed markup commands

LaTeX mark-up is used for formatting (e.g., in `\textbf{bold}` or `\textit{italics}`), and for special symbols (e.g., `\{e}` or `\&`). Some commands need several parameters, e.g., to

```
\href{https://tu-dresden.de/}{link to TU Dresden}.
```

Comments start with percent `%` and extend to the end of line

produces:

LaTeX mark-up is used for formatting (e.g., in **bold** or *italics*), and for special symbols (e.g., é or &). Some commands need several parameters, e.g., to link to TU Dresden. Comments start with percent

- Latin letters, numerals, and many punctuation symbols appear as text in the output
- LaTeX commands have the form `\command{parameter1}{parameter2}...`
- Most non-Latin letters cannot be typed directly (even with UTF file encoding), but are produced by commands

Reserved characters

The symbols # \$ % ^ & _ { } ~ \ have special meaning in LaTeX:

- **Backslash** \ is reserved for initiating commands
- **Braces** { } are conventionally used to group input symbols, especially to delimit parameters¹
- Not so special: **Brackets** [] are conventionally used for optional parameters, but work as plain text if not used after a command
- The **dollar sign** \$ is reserved for delimiting mathematical expressions
- **Ampersand** & separates columns in formatting tables, matrices, etc.
- **Tilde** ~ encodes a non-breaking space
- **Percent** % initiates comments
- **Caret** ^ and **underscore** _ initiate superscripts and subscripts in math, respectively
- **Hash** # refers to parameters in LaTeX command declarations

Commands to print them: \# \\$ \% \^ \& _ \{ \} \~ \textbackslash

¹TeX has no official delimiter; implementations of commands can use other ways to find their input, and leaving away braces does not cause syntax errors (but leads to unexpected behaviour)

Common special characters

Many other characters are not on the keyboard

Unicode input does not work in LaTeX, but there are commands for many symbols

- English quotation marks “ and ” are produced by `` `` and `' '`
- Dashes are produced by `--` (en-dash, `–`) and `---` (em-dash, `—`)
- There are commands for all kinds of accents, e.g., `\" {a}` (`ä`), `\' {a}` (`á`), `\^ {a}` (`â`), ...
- Ellipsis (`...`) can be produced by `\ldots`
- Dedicated packages provide thousands of further symbols

For (much) more, see the [Comprehensive LaTeX Symbol List](#) (link)

Whitespace and linebreaks

Breaking or indenting lines in the source code does often not change the result:

Spaces, tabs, and single line breaks are treated as a single space.

Whitespace and linebreaks

Breaking or indenting lines in the source code does often not change the result:

Spaces, tabs, and single line breaks are treated as a single space.

- An empty line is used to start a new paragraph
- Commands `\\` and `\newline` start a new line (within paragraph)
- `\` and `\~` create a normal space and a non-breaking space (neither collapse)
- Custom spaces can be made, e.g., `\hspace{1cm}` makes a space of 1cm length
- Closing lines with `%` “comments out the linebreak” (not even single space inserted)
- Initial spaces (indentation) is ignored
- Parameter-free commands “eat” spaces that follow

Whitespace and linebreaks

Breaking or indenting lines in the source code does often not change the result:

Spaces, tabs, and single line breaks are treated as a single space.

- An empty line is used to start a new paragraph
- Commands `\` and `\newline` start a new line (within paragraph)
- `\` and `\~` create a normal space and a non-breaking space (neither collapse)
- Custom spaces can be made, e.g., `\hspace{1cm}` makes a space of 1cm length
- Closing lines with `%` “comments out the linebreak” (not even single space inserted)
- Initial spaces (indentation) is ignored
- Parameter-free commands “eat” spaces that follow

This example illustrates how `spaces` `\` `\` `\` can `\hspace{15mm}` be controlled with `\textbackslash` `hspace`.

produces:

This example illustrates how spaces can be controlled with `\hspace`.

“French” spacing

LaTeX defaults to (old-fashioned) extra spacing after full stops.

Add the following to your preamble for modern spacing:

```
\frenchspacing
```

When not using this, extra space is added after every “.” (even for abbreviations within a sentence). To avoid this, write `.\` instead of `.` in such cases.

Text formatting

LaTeX has many commands to modify the font of selected text

- `\emph{text}` emphasises some text (usually by showing it in italics)
- `\textbf{text}` sets font weight to bold
- `\textit{text}`, `\textsf{text}`, `\textrm{text}`, `\texttt{text}`, and `\textsc{text}` set font shape to italics, sans serif, roman, typewriter, and small caps
- Sizes are selected relative to base size (not by selecting points)

Size commands (in order of size) include:

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

Size commands do not take parameters, but change the size for all text that follows. To delimit their effect, one needs to enclose them in braces:

```
{\tiny this text is tiny}
```

- The base font size is defined by the document class and its parameters (or otherwise tweaked in the preamble)

Setting document fonts

Fonts are usually selected on the document level in LaTeX

- Most easily, packages are loaded to update the fonts
- It is possible to update fonts for one shape (e.g., sans) and keeping another unchanged (e.g., roman)
- By default, Knuth's custom [Computer Modern](#) font families are used

Setting document fonts

Fonts are usually selected on the document level in LaTeX

- Most easily, packages are loaded to update the fonts
- It is possible to update fonts for one shape (e.g., sans) and keeping another unchanged (e.g., roman)
- By default, Knuth's custom **Computer Modern** font families are used

The following lines in the preamble load the most standard of fonts:

```
\usepackage{newtxtext,newtxmath} % use "Times"  
\DeclareMathAlphabet{\mathcal}{OMS}{cmsy}{m}{n} % optional  
\usepackage[scaled=0.92]{helvet} % use "Helvetica" (scaled!)  
\usepackage{courier} % use "Courier"
```

- Avoid the outdated package `times` (it fails to switch math fonts)
- The optional line resets the `\mathcal` font (looking like: \mathcal{ABC}) to LaTeX's default
- Helvetica is scaled since it is too large for Times at the same pt size
- The PDF will use clones of non-free fonts (e.g., Nimbus Roman instead of Times)

For most uses, you can just copy these lines or stick to the default.

Paragraphs

Paragraphs are blocks of text

- Separated by empty lines (i.e., two consecutive line breaks)
- Justified by default – other modes can be selected:

```
\begin{flushleft}  
left aligned text  
\end{flushleft}
```

```
\begin{flushright}  
right aligned text  
\end{flushright}
```

```
\begin{center}  
centered text  
\end{center}
```

- Automatic hyphenation is used for justified texts
(use proper language setting in preamble to load correct hyphenation patterns)

```
% Some common choices:  
\usepackage[UKenglish]{babel} % English (British)  
\usepackage[USenglish]{babel} % English (American)  
\usepackage[ngerman]{babel} % German (new orthography)
```

Sections

Documents can be structured in sections:

```
\section{The Section Title}
```

Text content ...

- Sections are automatically numbered appropriately
- Section-like headlines without number can be created with the `starred` command:

```
\section*{An Unnumbered Section}
```

- Several levels of headlines are usually available: `\chapter‡`, `\section`, `\subsection`, `\subsubsection`, `\paragraph`, `\subparagraph‡`

[‡] not in all document classes

- All types of sections can be made unnumbered by adding `*`
- **Note:** “subsections” are still “sections” in English; don’t use phrases like “As explained in Subsection 3, ...”

Footnotes

Footnotes are easy to produce:

Here is some normal `\footnote{for an example}` text.
Footnotes that refer to a whole sentence go after
the full stop. `\footnote{as shown here}`

produces:

Here is some normal^a text. Footnotes that refer to a whole sentence go after the
full stop.^b

^afor an example

^bas shown here

- The style of footnotes depends on the document type
- Do not overuse this feature

Lists

Lists and enumerations can be produced as follows:

```
\begin{itemize}
\item First item in list
\item Second item
\end{itemize}
```

```
\begin{enumerate}
\item First enumerated item
\item Second item
\end{enumerate}
```

which produce:

- First item in list
- Second item

1. First enumerated item
2. Second item

- Lists and enumerations can be nested
- Enumeration labels can be customised using one of two incompatible packages:
 1. `\usepackage{enumerate}`
allows simple syntax such as `\begin{enumerate}[(a)]`
 2. `\usepackage[shortlabels]{enumitem}`
same simple syntax; many further options, but more incompatibilities

Labels and references

LaTeX enumerates many things automatically

We can refer to these numbers using `\label` and `\ref`:

```
\section{Preliminaries}\label{sec_prelims}  
...  
\section{Our Contribution}
```

As discussed in Section~\ref{sec_prelims}
on page \pageref{sec_prelims} ...

- We use ~ here to avoid an unintended line break
- Mentioning the page is rarely needed
- The marker `sec_prelims` can be any string of letters and some special characters (using a prefix like `sec` is not required, but avoids confusion)
- `\label` can assign names not only to sections (of any level), but also to items in enumerations, figures, tables, equations, etc.

Compiling multiple times

LaTeX processes documents in a single pass, start to end

- auxiliary files are created to store relevant information, such as labels
- some information may be found too late (example: a label that was ref'ed before occurring in the source)

~> it can be necessary to run LaTeX a second time to get cross-references right

- Unknown references appear as ?? in the document
- The corrected reference may be of different length, so that some text moves when inserting it

~> it can be necessary to run LaTeX a third time to get page references right

Hyperlinks

PDF supports hyperlinks to places inside the document, and to external URLs.

This can be achieved with the `hyperref` package:

- Package is simply loaded in preamble
- Footnote anchors and references link to the right place in the document
- An interactive PDF table of contents is created
- Commands `\url` and `\href` can be used to link to external URLs
- Several other features ...

The `hyperref` package has many options – here is a good way of loading it:

```
\usepackage[hidelinks, % disable coloured boxes around links!
pdfpagelabels, % support named PDF pages
pdftitle={Your report title}, % set document title
pdfauthor={Names of authors}, % set document author(s)
pdfkeywords={keywords, if desired} % set keywords (optional)
]{hyperref}
\urlstyle{rm} % format URLs like normal text
```

Summary: a standard preamble

```
\documentclass[a4paper,11pt]{article}

\usepackage[UKenglish]{babel} % set document language
\usepackage[utf8]{inputenc} % use UTF8 character encoding for files
\usepackage[T1]{fontenc} % better hyphenation & PDF quality
\frenchspacing

\usepackage{newtxtext,newtxmath} % use "Times"
\DeclareMathAlphabet{\mathcal}{OMS}{cmsy}{m}{n} % optional
\usepackage[scaled=0.92]{helvet} % use "Helvetica" (scaled!)
\usepackage{courier} % use "Courier"

\usepackage[hidelinks, pdfpagelabels, pdftitle={Your report title},
pdfauthor={Names of authors}, pdfkeywords={Your keywords}]{hyperref}
\urlstyle{rm}
```

Spaces and lengths in LaTeX

Layout is defined by many “lengths”

- Length of indentation of paragraph
- Distance between two items in an enumeration
- Width of left margin
- ...

In LaTeX, lengths are named and can be set using common units:

```
\setlength{\parindent}{0mm} % no paragraph indent (0mm)  
\setlength{\parskip}{1em} % distance between paragraphs  
% (Note: em = "length of letter m" is a unit depending font size)
```

Other predefined length macros are useful:

```
The next space \hspace{0.5\textwidth} is half as wide as the text.
```

More spaces and lengths

- Spaces can be negative, e.g., for `\hspace{-1em}example` produces “example”
- The newline command `\\` takes an optional space to increase or decrease space between lines, e.g., the next line is added with `\\[-2mm]`, which produces some overlap
- Vertical space (positive or negative) can also be created with the `\vspace` command
- Many lengths in LaTeX are not fixed but allow some stretching or compressing (when this helps to improve the layout)

```
% Let space grow up to 2pt and shrink up to 1pt:  
\setlength{\parskip}{1em plus 3pt minus 1pt}
```

- Flexible vertical spaces are created by `\smallskip`, `\medskip`, and `\bigskip`

Summary

LaTeX is how most computer scientists produce research reports today

For basic usage, LaTeX behaves like a mark-up language where we write plain text mixed with commands that control format

Essential style commands control text format, sections, and listings

Labels, cross-references, and hyperlinks are a strong point of LaTeX

What's next?

- More LaTeX (tables, figures, citations, math)
- Persuasive writing and argumentation
- Oral presentations