LaTeX From The Ground Up

Tim Schulte

Albert-Ludwigs-Universität Freiburg
Grundlagen der Künstlichen Intelligenz

9. November 2017

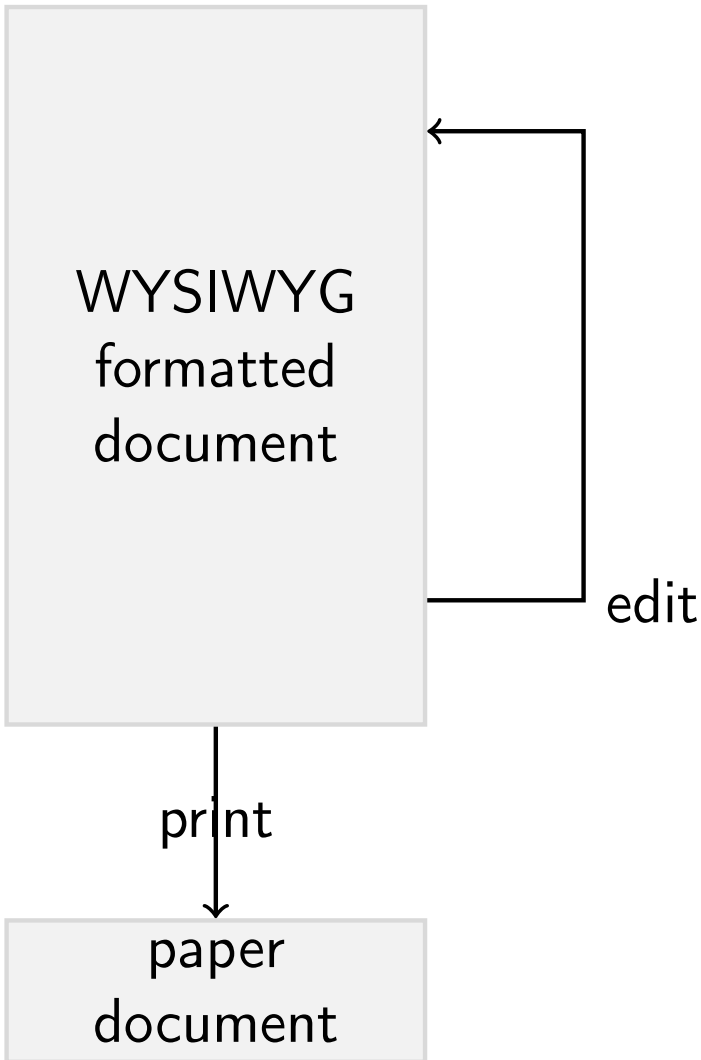
Table Of Contents

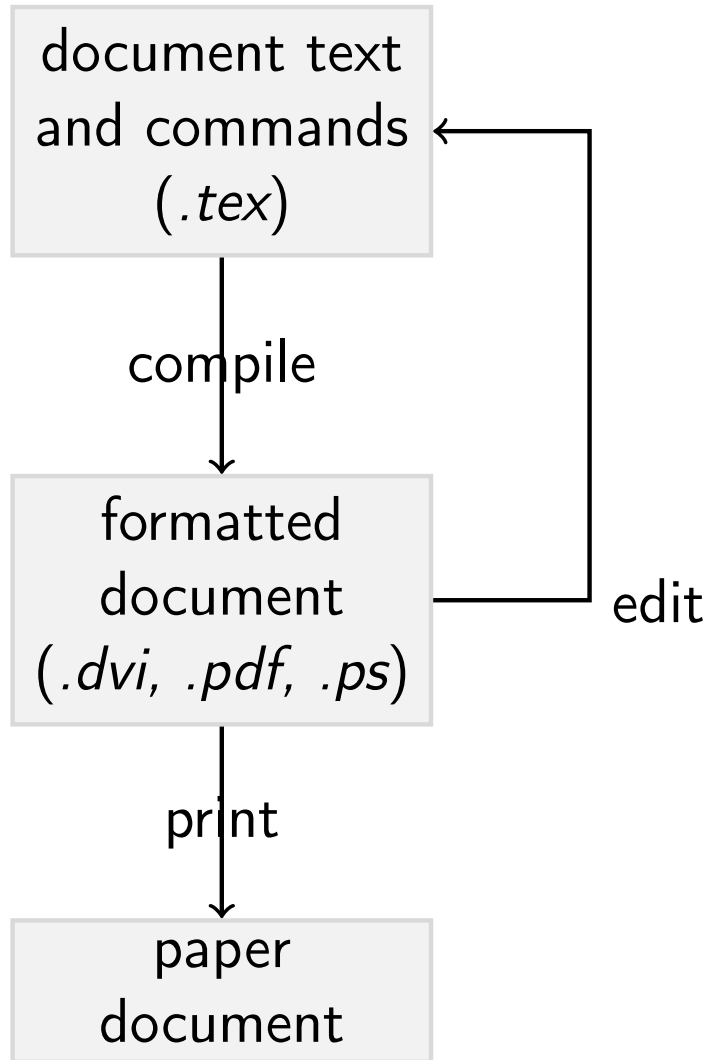
Why LaTeX?

LaTeX is a document preparation system and markup language.

- LaTeX takes care of artistic details and layout design
- Mathematical formulas are exceptionally well supported
- Complex structures can be generated easily (footnotes, table of contents, list of figures, bibliography, references, etc.)
- Encourages writing well-structured texts, because LaTeX itself works by specifying structure
- Highly portable and free

How LaTeX Works





Setting Up LaTeX

Requirements

- TeX distribution (MikTeX, TeXLive, MacTeX, ...)
- Editor (TeXStudio, TeXMaker, any plain text editor)

Examples

- **Windows**: TeXLive + TeXStudio
- **Linux**: TeXLive + Editor (e.g. Gedit, Vim, ...)
- **Mac**: MacTeX
- **Browser**: ShareLaTeX

The First Document

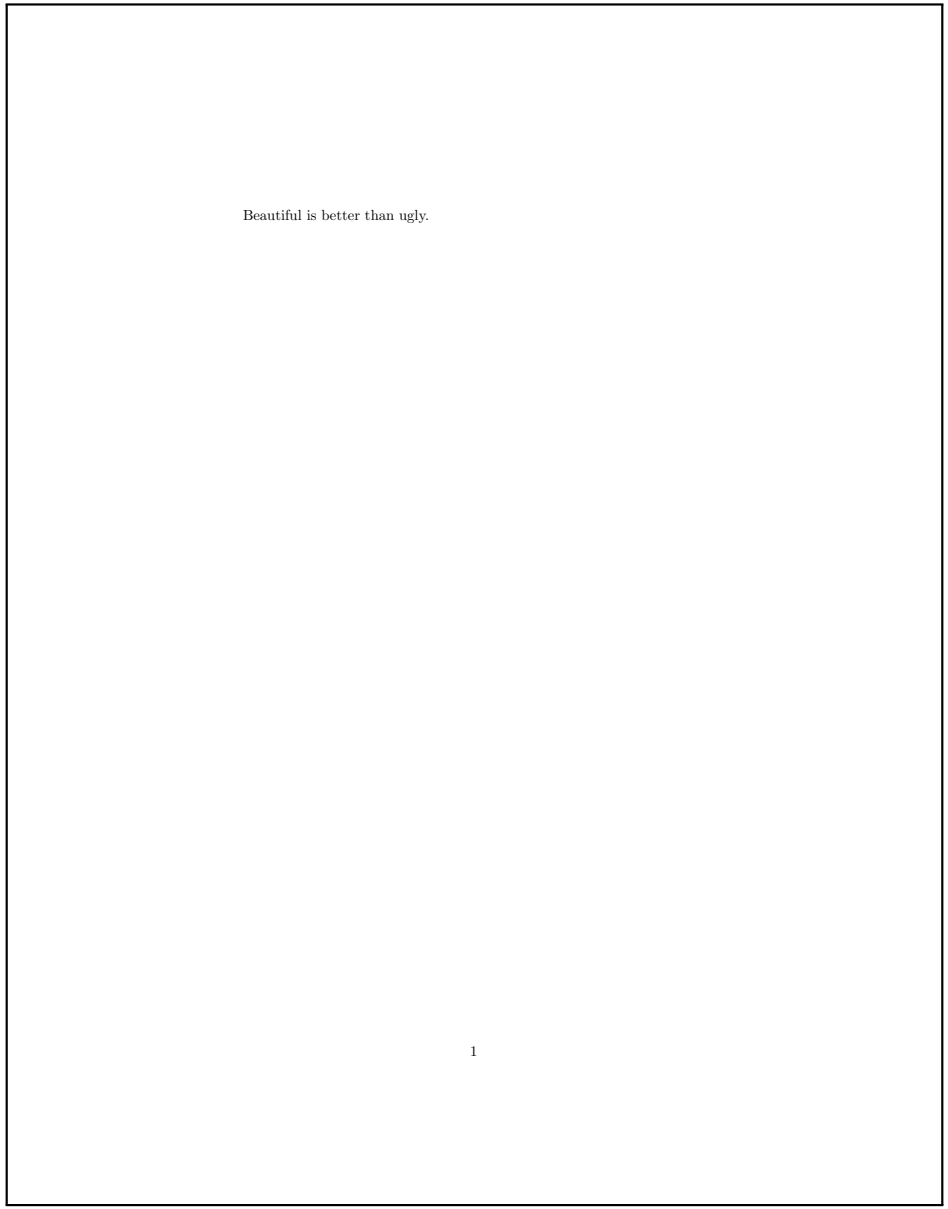
first.tex

```
\documentclass{article}

\begin{document}
Beautiful is better than ugly.
\end{document}
```

1. Create a new folder and inside it a new text file called *first.tex*
2. Add the above lines to the file using a plain text editor
3. Compile the document using *pdflatex*
4. View document using a PDF viewer

The First Document



General Document Structure

```
\documentclass{article}
...                          % preamble

\begin{document}
...                          % document body
\end{document}
```

Preamble

- commands affecting the whole document
- document class, language, ...
- used packages

Document body

- document content
- logical structure
- formatting instructions

Spaces

```
It does not matter
whether you enter one
or several     spaces
after
a    word.

An empty line starts
a new paragraph
```

It does not matter whether you enter one or several spaces after a word.
An empty line starts a new paragraph

Special Characters

In LaTeX some characters are reserved

```
# % ^ & _ { } ~ $ \
```

Insert a backslash before these characters to get the desired result

```
\# \% \^{} \& \_ \{ \} \~{} \$ \textbackslash
```

Output: # % ^ & _ { } ~ $ \

Commands

LaTeX commands are of the form

```
\commandname[opt1,opt2,...]{arg1}{arg2}...
```

Some commands need an arguments given in curly braces `{ }` while optional parameters are given in square brackets `[ ]`.

Examples:

```
\documentclass{article}, {\huge profit}
```

Environments

Environments specify areas in the document where certain typesetting rules apply.

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

Comments

```
This is an % stupid
% Better: instructive <--
ex%
    amp%
    le.
```

This is an example.

Document Class

The *document class* specifies the overall layout of your document

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

Use `article` for your paper, `beamer` for your presentation.

Classes	
`article`	for articles in scientific journals, presentations, short reports, program documentation
`proc`	a class for proceedings based on the article class
`report`	for longer reports containing several chapters, small books, Master's and PhD theses
`book`	for real books
`letter`	letters
`beamer`	for presentations

Document Class Options

Options	
`10pt, 11pt, 12pt`	Size of the main font in the document
`a4paper, a5paper, letterpaper, ...`	The paper size
`titlepage, notitlepage`	Start a new page after the titlepage?
`onecolumn, twocolumn`	Typeset the documents in one ... ... or two columns
`landscape`	Change layout to landscape mode
`openright, openany`	...
`draft, final`	...

Packages

Packages allow further customization. They are included in the preamble of your document

```
\usepackage[options]{package}
```

Most packages you'll need are already included in the template.

Useful packages

graphicx, epsfig, geometry, tikz, fancyhdr, setspace, amsmath, listings, xcolor, url, inputenc, babel, ...

Language Specific Packages

German documents require *umlauts* (äöü)

```
\usepackage[utf8]{inputenc}
```

correct hyphenation

```
\usepackage[german]{babel}
```

and (sometimes) special fonts

```
\usepackage[T1]{fontenc}
```

Other languages are supported likewise.

Font shapes

LaTeX provides commands to change the

- font family
  `\texttt` (typewriter) `\textrm` (roman)
  `\textsf` (sans serif)
- font series
  `\textbf` (**boldface**) `\textmd` (medium)
- font shape
  `\textup` (upright) `\textit` (*italic*) `\textsl` (*slanted*)
  `\textsc` (SMALL CAPS)

You can highlight text using `\emph`.

Sizing text

LaTeX provides commands to change the font size

`\tiny`	tiny font
`\scriptsize`	very very small font
`\footnotesize`	very small font
`\small`	small font
`\normalsize`	normal font
`\large`	slightly larger font
`\Large`	very large font
`\LARGE`	even larger font
`\huge`	huge font
`\Huge`	very huge font

Ex.: `\tiny tiny tiny ... \normalsize normal ...`

Ex.: `{\LARGE large large ...} normal ...`

Line- and page-breaks

- Paragraphs are separated by a full blank line
- `\\` ends a line explicitly without ending the whole paragraph
- `\newpage` (or `\clearpage`) ends a page explicitly

Alignment

Center text using the `center` environment.

Left/right-align text using the `flushleft/flushright` environment.

```
\begin{center|flushright|flushleft}
...
\end{center|flushright|flushleft}
```

Title Page

titlepage.tex

```
\documentclass{article}

\title{My first document}
\author{Emilio Estevez}
\date{\today}

\begin{document}
\maketitle
Beautiful is better than ugly.
\end{document}
```

Abstract

Most research papers have an abstract.

abstract.tex

```
\documentclass{article}

\begin{document}
\begin{abstract}
Your abstract goes here...
\end{abstract}
...
\end{document}
```

The `abstract` evironment is available for `articles` and `reports`.

Sectioning Commands

There are 7 levels of depth for defining sections.

Sectioning commands and levels		
`\part`	-1	not in letters
`\chapter`	0	books and reports only
`\section`	1	not in letters
`\subsection`	2	not in letters
`\subsubsection`	3	not in letters
`\paragraph`	4	not in letters
`\subparagraph`	5	not in letters

Sectioning Commands

sectioning.tex

```
\documentclass{article}
\begin{document}
\section{Section}
Hello World!
\subsection{Subsection}
Structuring a document is easy!
\subsubsection{Subsubsection}
More text.
\paragraph{Paragraph}
Some more text.
\subparagraph{Subparagraph}
Even more text.
\section{Another section}
\end{document}
```

1 Section

Hello World!

1.1 Subsection

Structuring a document is easy!

1.1.1 Subsubsection

More text.

Paragraph Some more text.

Subparagraph Even more text.

2 Another section

Referencing sections

Add labels to sections to reference them in the text.

```
\section{Results}\label{res}
...
As seen in Section \ref{res} ...
```

LaTeX keeps track of section numbers for you.

Table Of Contents

A table of contents can be generated with

```
\tableofcontents
```

Titles of sections are added automatically to the table of contents.

You can modify the text displayed in the ToC

```
Ex.: \section[Introduction]{Rapid Introduction To...}
```

Note

ToC entries are recorded when the document is processed. They are reproduced the next time the document is processed.
$\Rightarrow$ **Run *pdflatex* twice** to ensure that all ToC pagenumber references are correctly calculated.

Modular documents

When writing a book, it makes sense to split the document into multiple `.tex` files.

Getting LaTeX to process multiple files is easy. Just use

```
\input{filename}
```

somewhere in your document, to put the contents from `filename.tex` in place.

Done.

BibTeX file format

mybib.bib

```
@article{maxmustermann,
    author = {Mustermann, Max},
    title = {Mustermann on topics of interest},
    journal = {Journal of Mustermann},
    volume = 46,
    year = 1993,
    number = 2,
    pages = {35--53}
}
...
```

For an indepth description of the database format see

https://en.wikibooks.org/wiki/LaTeX/Bibliography_Management#BibTeX

The BibLaTeX package

```
% in the preamble
\usepackage[bibencoding=utf8,
    backend=biber, style=numeric]{biblatex}
\addbibresource{mybib.bib} % or

% where the bibliography will be printed
\printbibliography
```

There are other styles, like `alphabetic`, `authoryear`, ...

Citations

Add references to your document with

`\cite`

Example

```
Redundancy \cite{maxmustermann}
...methodology \cite{entry1, entry2, ...}
```

Bibliography summary

1. Create a BibTeX file (the database)
2. Include the `biblatex` package in the preamble and add the database as a `bibresource`
3. Print the bibliography somewhere in the document body

Note

To build the bibliography, first compile the document, then generate the necessary `.bbl` file, and compile the document again.

```
> pdflatex <myfile.tex>
> biber <myfile>
> pdflatex <myfile.tex>
```

If you use an IDE it will probably take care of this for you.

Lists

There are two basic types of lists (which can also be nested).

List environments

```
\begin{itemize}
    \item This list
    \item is
    \item \emph{unordered}
\end{itemize}
```

- This list
- is
- *unordered*

```
\begin{enumerate}
    \item This list
    \item is
    \item \emph{ordered}
\end{enumerate}
```

1. This list
2. is
3. *ordered*

Figures

Use the `graphicx` package + `figure` environment to embed pictures.

```
\usepackage{graphicx}
```

Figure environment

```
\begin{figure}
    %\centering
    \includegraphics[width=.35\linewidth]{gopher}
    \caption{A gopher.}
    \label{fig:gopher1}
\end{figure}
```

Figure Positioning

LaTeX may choose to put the picture on a different location.

Adding [h!] behind the figure environment forces the figure to be shown at the exact location in the document.

```
\begin{figure}[h!]
```

Positioning flags

`h` (here)	same location
`t` (top)	top of page
`b` (bottom)	bottom of page
`p` (page)	on an extra page
`!` (override)	will force the specified location

Tables

Demo

See `https://en.wikibooks.org/wiki/LaTeX/Tables`

Verbatim

To introduce text that won't be interpreted by the compiler, use the `verbatim` environment.

```
\begin{verbatim}
The verbatim environment
  simply reproduces every
 character you input,
including all  s p a c e s!
% & { }
\end{verbatim}
```

```
The verbatim environment
  simply reproduces every
 character you input,
including all  s p a c e s!
% & { }
```

Math Environments

LaTeX needs to know when text is mathematical.

There are two main environments

1. Inline formulas (within text)
   `\(...\)`
2. Displayed equations (separated from text)
   `\[...\]`

Symbols

Mathematics has symbols. Some can be accessed directly

```
+ - = ! / ( ) [ ] < > | ’ :
```

Others require distinct commands. For instance

```
\( \forall x \in X, \quad \exists y \leq \epsilon \)
```

Produces: $\forall x \in X, \quad \exists y \leq \epsilon$

For more advanced mathematic operators see
https://en.wikibooks.org/wiki/LaTeX/Mathematics
http://www.hostmath.com

Common Mistakes

A space right after a period following a lowercase letter by default ends a sentence and LaTeX inserts an extra whitespace. There are several occasions where you do not want to have the default behaviour.

Example

```
Ms. Bean is \ldots\\
Ms.\ Bean is \ldots

I left at 12:00 P.M. In \ldots\\
I left at 12:00 P.M\@. In \ldots

\LaTeX is fun.\\
\LaTeX\ is fun.
```

Common Mistakes

In math mode variables with more than two characters must be wrapped inside `\mbox` or `\mathit` environments.

Example

```
\[
gbfs = 35\\
\mathit{gbfs} = 35
\]
```

Common Mistakes

Using the wrong quotation marks.

`‘American’`	‘American’
`‘‘American’’`	“American”
`,,German‘‘`	„German“
`<<French>>`	«French»

For european quoting style use T1 font encoding:
`\usepackage[T1]{fontenc}`

Common Mistakes

Not using UTF8

Just put

```
\usepackage[utf8]{inputenc}
```

in your preamble and you're done.

Common Mistakes

Not using version management software (git, mercurial, svn)