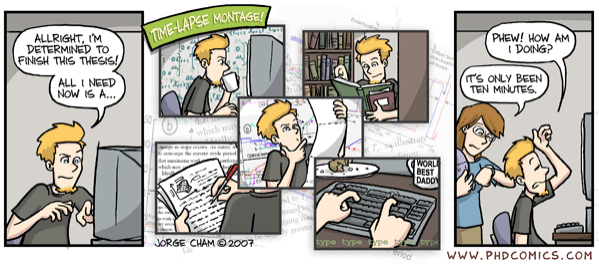
KU LEUVEN

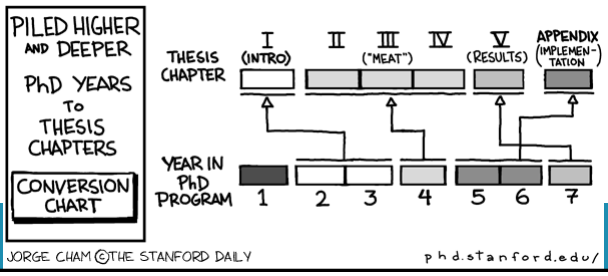
Introduction to LaTeX

Modular documents

Contents

- Large documents

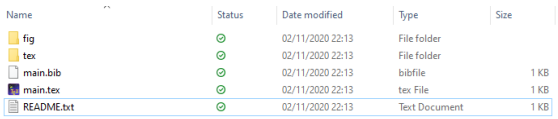




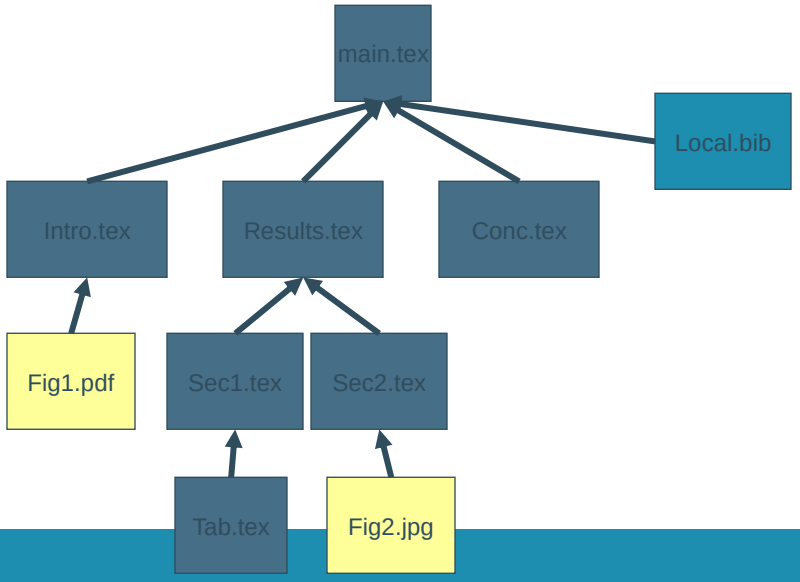
KU LEUVEN

Modular LaTeX

- Create a clear structure of the project:
  - a root directory for the project.
  - directories inside the root folder:
    - for LaTeX documents
    - for images
    - Tip: choose short names. (ex. tex and fig)
- Check:
  - https://eitanlees.github.io/latex/2019/05/17/modular-latex.html
  - https://github.com/nelkinda-templates/template-latex-book
  - https://en.wikibooks.org/wiki/LaTeX/Modular_Documents
  - https://www.overleaf.com/learn/latex/Management_in_a_large_project



Modular LaTeX



Modular LaTeX

- Large document: keeping all the source text in one file becomes unmanageable.
- Advantages to break a document into separate files:
  - Imposes a structure on the document as a whole.
  - Allows you to focus on each part separately.
  - Maintenance of the document becomes easier,
  - (Pre)view only part of the document.
- https://tex.stackexchange.com/questions/22431/everyday-latex-and-workflow/22433#22433

Modular LaTeX

```
\documentclass[a4paper]{book}
\title{A Thesis}
\author{MY Self}
\begin{document}
  \frontmatter
    \maketitle
    \tableofcontents
    \listoffigures
    \listoftables
  \mainmatter
    \input{introduction}
    \input{background}
    \input{methodology}
    \input{implementation}
    \input{analysis}
    \input{discussion}
    \input{conclusion}
  \appendix
    \input{sourcecode}
  \backmatter
    \bibliography{bibthings}
\end{document}
```

- A typical root document

Large documents

- LaTeX supports splitting a document in several files. Two commands will make it easy:
  - `\input{file.tex}`
  - `\include{file.tex}`
- Absolute and relative paths can be used.
- Both commands allow to insert content from external files inside another LaTeX document. The idea is that you have some top level document file and a number of files that get included in this file automatically when you run LaTeX.

\input

- `\input`
  - Easy to use: segment the text into chunks, run LaTeX on the top-level file, the contents of each chunk will be read in at the specified points as if its contents have been typed at that point.
- Top-level

```
\documentclass{...}
...
\begin{document}
\input{firstfile}
\input{secondfile}
...
\input{lastfile}
\end{document}
```

\input

- The name of each included file must have the .tex extension
- `\input` can be nested
  firstfile can contain calls to other files to input.
- Each inputted file is not a standalone LaTeX file (no `\documentclass{...}`, `\begin{document}` `\end{document}`).
- calls to input can be mixed with other arbitrary text and LaTeX commands.
- *File: MyLargeBook-input.tex*

\input

- Limitations when using not all the input files:
  - The numbering of sections, page numbers will only rely on the parts that are included.
  - Cross-references will not be resolved.
- Typical use:
  - *Swap out* the preamble
    - put the preamble commands in a separate file and re-use it
  - Keep stuff like tikz figures, complex tables, etc. in separate files

\include

- `\include` works in a similar way as `\input` but there are some differences:
- `\include` implicitly starts new pages. `\include{filename}` behaves like:

```
\clearpage
\input{filename}
\clearpage
```

- Useful for page ranges such as chapters.
- Cannot be nested.
- Can only appear in the *(top)document* body,
- Supports a mechanism of choosing which parts of the document you wish to compile (`\includeonly`).

\include

- Top-level (same as with \input)

```
\documentstyle{...}
...
\begin{document}
\include{firstfile}
\include{secondfile}
...
\include{lastfile}
\end{document}
```

\include

- Each included file gets its own .aux file.
- LaTeX looks at the other aux files, it knows about section and page numbers, cross-references,…
- Each included file will automatically begin on a new page,
- `\includeonly` controls which files will be read by LaTeX
  - multiple files specified in the `\includeonly` line, have to be separated by commas with no intervening spaces.
  - can only appear in the preamble.
- *File: MyLargeBook-include.tex*
- *File: MyLargeBook-includeonly.tex*

\include vs \input

- `\include{blah}` starts a new page and inserts the file blah.tex while `\input{blah}` simply inserts blah.tex.
- use `\include` only for top-level items like chapters where you want to start a new page.
- `\input` simply drops in a block of LaTeX code as-is.
  It can be useful for inserting tables which are machine-generated.
- `\input` can be nested, `\include` not.

	\input	\include
Nesting allowed	X	
Start new page		X
Suited for chapter subfiles		X
Suited for any subfile	X	

Hands-on

- Use handson_large_01 and the subfiles handson-large1, handson-large2, handson-large3
  - Compile the text
  - check the result.
- Split handson-large2 into smaller subfiles and check the result
- Use `\include` instead of \input
- Use `\includeonly` to compile only a part of the text

\import

- In some cases `\input` and `\include` can cause trouble if nested file importing is needed. `\input` needs the full filename starting from the working directory
- https://danielsank.github.io/tex_modularity/
- https://tex.stackexchange.com/questions/58465/how-to-use-the-import-package
- Package import
  - `\usepackage{import}`

subfiles

- A disadvantage of solely using \input and \include is that only the base document can be compiled.
- Working on individual sections of text and editing and compiling those separate from the main file is possible with the packages:
  - `subfiles`
  - `standalone`
- https://en.wikibooks.org/wiki/LaTeX/Modular_Documents
- https://jonasdevlieghere.com/modular-latex-with-subfiles/
- https://texfaq.org/FAQ-multidoc
- https://www.overleaf.com/learn/latex/Multi-file_LaTeX_projects