KU LEUVEN

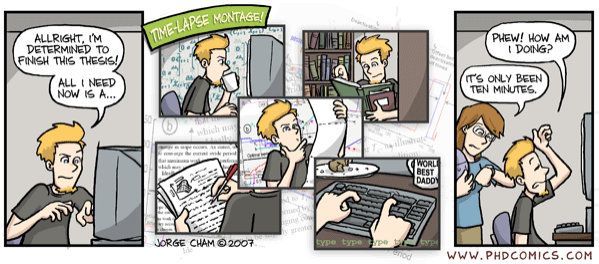
Introduction to LaTeX

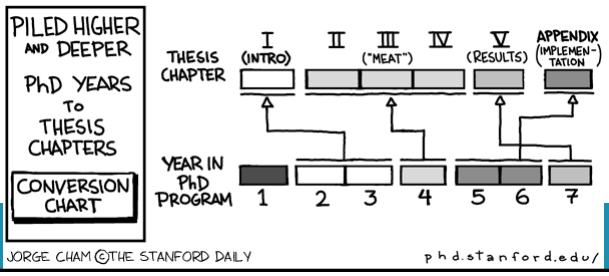
Large documents

1

Contents

- Large documents
- Referencing
- Index
- Glossaries
- Hyperlinks
- In the margin

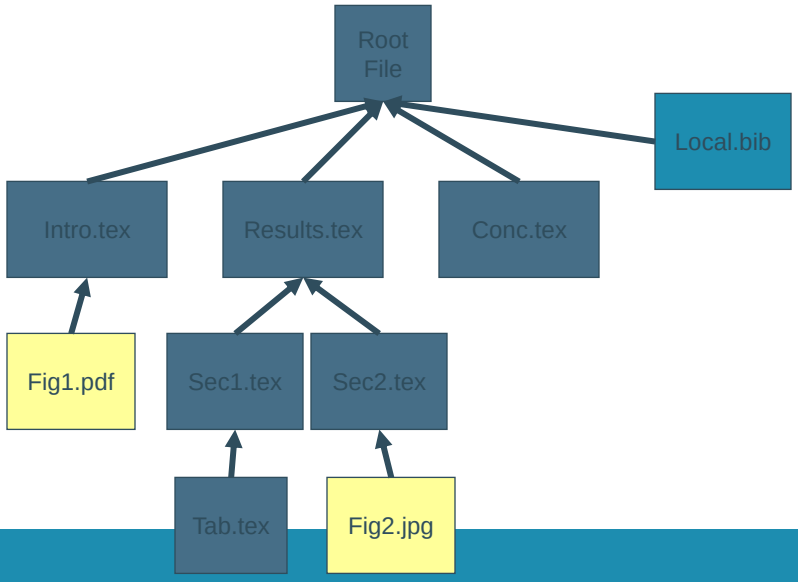




KU LEUVEN

2

Large documents: structure



KU LEUVEN

Large documents

- 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

KU LEUVEN

Project structure

- Get some folder structure
  - create a directory for the project.
  - create two folders:
    - /tex: for the LaTeX files
    - /img: for the images
- create the root document in the root folder
  - ./MyRootDoc.tex
  - ./tex/
  - ./img/
- https://en.wikibooks.org/wiki/LaTeX/Modular_Documents

KU LEUVEN

5

Large documents

```
\documentclass[a4paper]{book}
\title{A Thesis}
\author{MY Self}
\begin{document}
  \frontmatter
    \maketitle
    \tableofcontents
    \listoffigures
    \listoftables
  \mainmatter
    \include{introduction}
    \include{background}
    \include{methodology}
    \include{implementation}
    \include{analysis}
    \include{discussion}
    \include{conclusion}
  \appendix
\include{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}`
- Both commands allow to include 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
  - If you want to focus only on some parts of the text and you delete or put in comment the *inputs* you don't want, 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 or sections.
- Cannot be nested.
- Can only appear in the *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}`
  - Specify the folder and the file. `\import{ }{ }`. The first parameter inside braces is the directory where the file is located, it can be relative to the current working directory or absolute. The second parameter is the name of the file to be imported
  - `\import{sections/}{section1-1.tex}`
  - Just replace `\input{dir/file}` with `\subimport*{dir}{file}` and all LaTeX code will handle relative paths properly.

Modularity

- 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

Referencing

- Reference almost anything that is numbered (sections, figures, formulas)
- LaTeX will take care of numbering, updating it whenever necessary.
- `\label{marker}` give the object a *marker*
- `\ref{marker}` reference the object you have *marked*
- `\pageref{marker}` It will print the number of the page where the object is.

Labels

- Common practice to structure the labels
  - **chap:** chapter
  - **sec:** section
  - **fig:** figure
  - **tab:** table
  - **eq:** equation

  `\label{fig:my_figure}`

  File: demo_label

KU LEUVEN

Labels and references

- References to equations and sections.
- Label the item
  - `\section{...} \label{sec:sec-name}`
  - `\begin{equation} ... \label{eq:eq-name} \end{equation}`
- Having created the labels, refer to the objects using `\ref{label-name}`
- Works also for chapters, subsections, subsubsections, tables, figures, and enumerated lists.
- Run the compilation several times
- *File: demo_referencing_01*

KU LEUVEN

Label and floats

- Use the `\label` command to cross-reference:

```
\begin{figure}
\includegraphics{. . . }
\caption{This is Donald}
\label{Donald}
\end{figure}
```

- *Warning:* If you want to label a figure so that you can reference it later, you have to add the label **after the caption** but **inside the floating environment**. If it is declared outside, it will give the section number.

Hyperlinks

- `\usepackage{hyperref}`
- Importing the `hyperref` package all cross-referenced elements become hyperlinked.
- Be careful when importing `hyperref`. Usually it has to be the last package to be imported.
- https://www.overleaf.com/learn/latex/Hyperlinks

Hyperref: setting parameters

- Use hypersetup (in preamble) to set some parameters
- Every parameter must be comma-separated and the syntax must be in the format parameter=value.
- `\hypersetup` : specify parameters
  - `colorlinks=true`
    - Links will be colored (default red).
  - `linkcolor=blue`
    - Internal links, those generated by cross-referenced elements, are displayed in blue.
  - `filecolor=magenta`
    - Links to local files will be shown in magenta color
  - `urlcolor=cyan`
    - Links to web sites are set to cyan

Hyperlink (web)files

- Links to a web address can be added using:
  - `\url` command to display the actual link
  - `\href` to use a hidden link and show a word/sentence instead.
- `\href{https://www.kuleuven.be}{Some Link}`
- `\url{https://www.kuleuven.be}`
  - This will show the url passed as parameter and make it into a link, useful if you will print the document
- The commands `\href` and `\url` can also be used to open local files

Inserting links manually

- Cross-referenced elements become links once `hyperref` is imported,
  - References to `\label` used in the document will create links
- Use `hypertarget` and `hyperlink` to create links manually
  - `\hypertarget` specifies the target (anchor) with an identifier
  - `\hyperlink` refers to the specifier
- *File: demo_hyperref_1*
- *File: demo_hyperref_2*
- *File: demo_hyperref_3*

KU LEUVEN

Index

- An index is an alphabetical list of words and expressions with the pages of the document upon which they are to be found.
- Index creation involves some work, but LaTeX still makes it easier than doing it by hand.
- Use `makeindex`.
  - Tag keywords in the LaTeX source as index entries.
  - These tags cause LaTeX to record index information in a special output file.
  - Run `makeindex` to process this file to create an index that can be typeset by LaTeX.
- https://en.wikibooks.org/wiki/LaTeX/Indexing
- https://www.overleaf.com/learn/latex/Indices

KU LEUVEN

Index

- Load makeidx package
  ```
  \usepackage{makeidx}
  ```
- Mark words to be indexed by enclosing them in a `\index` command, the text has to be repeated.
  ```
  Superconductors\index{superconductor} conduct
   electricity with zero resistance\index{resistance}.
  ```
- Start the indexing: Place a `\makeindex` command under the `\usepackage{makeidx}` command
- `\printindex` will print the index

Makeindex procedure

- TeXstudio: build + Tools>Index + build
- if you are using a command prompt, you will need to do:
  - pdflatex filename.tex
  - makeindex filename.idx
  - pdflatex filename.tex
- If you are also using BibTeX, you will need to do:
  - pdflatex filename.tex
  - bibtex filename
  - makeindex filename.idx
  - pdflatex filename.tex
  - pdflatex filename.tex

Sub-entries

- You can make an index with sub-entries, and sub-sub entries.
- The general form of \index is:

```
\index{main_entry !sub_entry !sub_sub_entry }
```

- For example, an index entry of the form:

```
\index{provinces!Ontario}
\index{provinces!Saskatchewan}
\index{provinces!British Columbia}
```

- *File: MyDocIndex*

KU LEUVEN

Glossaries and Acronyms

- **Glossary**: an alphabetical list of terms in a particular domain of knowledge with the definition / explanation for those terms
- **Acronym**: an abbreviation formed from the initial letters of other words and pronounced as a word (*ASCII*, *NASA*)
- A possible improvement would be references to the locations in the text where those terms are used.
- https://en.wikibooks.org/wiki/LaTeX/Glossary
- https://www.overleaf.com/learn/latex/Glossaries

KU LEUVEN

Glossaries

- Use package glossaries
  - `\usepackage{glossaries}`
- Create an entry with the command `\newglossaryentry`
- The command `\gls` produces the name of the term given the label
- `\printglossary` where you want your list of entries to appear
  - Alternatively, to display all glossaries use the iterative command: `\printglossaries`
- *File: demo_glossaries_simple.tex*
- *File: demo_glossaries.tex*

Glossaries: steps

- Overview of the steps to generate a glossary
  - Prepare the tex file
    - Load the package
    - Tell LaTeX to generate a glossary `\makeglossaries`
    - Define the terms and their definitions
    - Use the terms
    - Print the glossary `\printglossaries`
  - Compile the document
  - Generate the index file
  - Compile the document again
- http://texblog.org/2014/01/15/glossary-and-list-of-acronyms-with-latex/

Glossaries: tweaking the setup

- Need some setup tweaking in your editor!
  - http://brianhoffmann.de/journal/thesis/2011-08-01/latex-glossaries-with-texniccenter
- Note: if you want to use both glossaries and hyperref, you must load `hyperref` before `glossaries`. This is an exception to the usual advice of loading `hyperref` last.
  http://www.dickimaw-books.com/latex/thesis/html/makeglossaries.html
- Check documentation for beginners
  http://tug.ctan.org/macros/latex2e/contrib/glossaries/glossariesbegin.pdf

Acronyms

- The glossary and the list of acronyms can be displayed separately in different places:
  `\usepackage[acronym]{glossaries}`
- `\newacronym` to create an acronym
- `\printglossary[type=\acronymtype]` to print the list of acronyms
- *File: demo_acronyms_glossaries.tex*

Proclamations

- LaTeX does not by default provide an environment for theorems.
- 2 step procedure:
  - In the preamble, use a `\newtheorem` command to define the proclamation.

    ```
    \newtheorem{thrm}{Theorem}
    ```
  - 2 parameters, the first one is the name of the environment that is defined (thrm), the second one is the word that will be printed (Theorem)
  - In the document body
    - Use: `\begin{thrm} ...\end{thrm}.`
    - The second argument (Theorem) is used to label the statement (title).
- https://www.overleaf.com/learn/latex/Theorems_and_proofs

Proclamations

- LaTeX provides an environment to create theorem-like environments (can be extended by the amsthm).
  - `\usepackage{amsthm}`
  - `\newtheorem{theorem}{Theorem}`
  - `\newtheorem{corollary}[theorem]{Corollary}`
  - `\newtheorem{lemma}[theorem]{Lemma}`
  - `\newtheorem{definition}[theorem]{Definition}`
- use the following environments as appropriate:
  - `\begin{theorem}...\end{theorem}`
  - `\begin{corollary}...\end{corollary}`
  - `\begin{lemma}...\end{lemma}`
  - `\begin{definition}...\end{definition}`
  - `\begin{proof}...\end{proof} % from amsthm`
- http://www.maths.adelaide.edu.au/anthony.roberts/LaTeX/ltxenviron.php

Proclamations

- LaTeX automatically numbers theorems consecutively (independent of chapters, etc.)
- Use * to suppress
  - `\newtheorem*{thm2}{Theorem}`
- *File: demo_theorem_1.tex*
- In general, every `\newtheorem` has its own counter.
- The automatic numbering can be linked to other counters.
- `\newtheorem{cor}[theorem]{Corollary}`
- *File: demo_theorem_2.tex*

KU LEUVEN

Proclamations

- Theorem numbers can be linked with sections, subsections, chapters…

```
\newtheorem{sectheorem}{Theorem}[section]
```

- Named theorem: pass the name as a parameter

```
\begin{theorem}[The first one]
This is an important theorem.
\end{theorem}
```

- *File: demo_theorem_3.tex*

KU LEUVEN

Proclamations

- Generate a list of theorems
- Package: `thmtools`
- `\listoftheorems`
- *File: demo_theorem_4.tex*

Proclamations

- proof environment
  - Used for proofs.
  - Typesetting somewhat different from theorem
  - Ends with □

```
\begin{proof}
This is obvious.
\end{proof}
```

- *File: demo_proof.tex*

footnote

- The \footnote command places the numbered footnote text at the bottom of the current page.
- `\footnote{footnote text}`
- Referencing is possible
  - Place label inside the note
- *File: demo_footnote.tex*

- Numbering tweaks – see also `package{chngcntr}`
  - Article: 1, 2, …
  - Book, report: no reset per chapter
- *File: demo_footnote_number.tex*

endnotes

- No footnotes at the bottom of the page, but at the end of the document.
- `% In the preamble`
- `\usepackage{endnotes}`
- `\let\footnote=\endnote`
- `% In the document where you want the notes to be printed`
- `\newpage`
- `\theendnotes`
- *File: demo_endnotes.tex*
- https://www.sixhat.net/latex-continuous-footnote-numbers-and-footnote-to-endnote-conversion.html

Margin note

- Create notes in the margin is a really nice/cool feature in LaTeX.
- Edward Tufte: *it lets you keep your notes near your content, which is a good thing*.
- Only short text!
- `\marginpar`
- *File: demo_marginpar.tex*
- `marginnote` package can be used for more flexibility.
- `\marginnote{This note will appear in the margin.}`
- *File: demo_marginnote.tex*

Numbering

- Some document elements (e.g., figures in the book class) are numbered per chapter (figure 1.1, 1.2, 2.1, ...).
  How to achieve continuous numbering (figure 1, 2, 3, ...)?
- Some document elements (e.g., figures in the article class) are numbered continuously.
  How to achieve per-section numbering?
- Use package chngcntr
- https://tex.stackexchange.com/questions/28333/continuous-v-per-chapter-section-numbering-of-figures-tables-and-other-docume
- *File: MyLargeDoc-numbering*
- *File: MyArticle-numbering*

Extra commands

- `\frontmatter` turns off chapter numbering and uses roman numerals for page numbers;
- `\mainmatter` turns on chapter numbering, resets page numbering and uses arabic numerals for page numbers;
- `\appendix` resets chapter numbering, uses letters for chapter numbers and doesn't fiddle with page numbering;
- `\backmatter` turns off chapter numbering and doesn't fiddle with page numbering.
- The hard-and-fast rule:
  - Don't use \appendix after \backmatter, because chapter numbering has already been turned off by \backmatter.
- *File: MyLargeBook-input.tex*
- Taken from: http://tex.stackexchange.com/questions/20538/what-is-the-right-order-when-using-frontmatter-tableofcontents-mainmatter