

A brief introduction to L^AT_EX

Silke Hofstra
Scintilla's Course Committee

May 30, 2016

Introduction

The goal of this document is not to provide an extensive guide to all that is \LaTeX . For that, books have been written. This document is a very simple introduction to the most important parts of \LaTeX , and isn't as comprehensive as most other documentation out there. Because of this I'd like to link to some more complete, but maybe slightly outdated, resources:

- [The \$\text{\LaTeX}\$ Wikibook¹](#) has a lot of information. This document tries to loosely follow the structure of Wikibooks.
- [The \$\text{\TeX}\$ Stack Exchange²](#) is a high quality Q&A site for \TeX related questions. If you have a question chances are you'll find your answer on there as well.
- For books and other websites [this question³](#) on the \TeX Stack Exchange is a good starting point.

For this course, you can download everything:

- [The presentation⁴](#)
- [The manual & assignments⁵](#)
- [Everything in a zip-file⁶](#)

¹<http://en.wikibooks.org/wiki/LaTeX>

²<http://tex.stackexchange.com/>

³<http://tex.stackexchange.com/q/11/17484>

⁴<https://www.scintilla.utwente.nl/docs/cursus/latex/presentation/presentation.pdf>

⁵<https://www.scintilla.utwente.nl/docs/cursus/latex/manual/manual.pdf>

⁶<https://git.scintilla.utwente.nl/cursus/latex/repository/archive.zip?ref=master>

Contents

Introduction	1
1 The Basics	4
1.1 What is \LaTeX	4
1.2 Installation	5
1.2.1 Linux	5
1.2.2 Mac OS X	5
1.2.3 Windows	5
1.3 Commands, environments and comments	6
1.4 The Preamble	7
1.5 A basic document	8
1.6 Document structure and sections	9
1.7 Title	10
1.8 Packages	11
1.8.1 geometry	11
1.8.2 babel	11
1.8.3 hyperref	11
1.9 Text	13
1.10 Lists	15
1.11 Images	16
1.12 Tables	17
1.13 Labels and references	18
1.14 Footnotes and Margin Notes	19
1.15 Mathematics	20
1.16 Source Code Listings	21
1.17 Getting content from different files	22
1.18 Common problems	23
2 Beyond	24
2.1 Bibliography management	24

2.1.1	Bibliography database	24
2.1.2	Typesetting with a bibliography	25
2.1.3	Bibliography package	25
2.2	Fancy headers	27
2.3	L ^A T _E X engines	28
2.3.1	latexmk	28
2.3.2	X _Y L ^A T _E X & LuaL ^A T _E X	28
2.4	Fonts	29
2.4.1	L ^A T _E X _{2ϵ} Packages	29
2.4.2	L ^A T _E X ₃ : Fontspec	29
2.5	Variables & Commands	30
2.6	Editor tricks	31
3	Assignments	32
3.1	Assignment 1	32
3.2	Assignment 2	33
A	Sample data for the assignments	34
A.1	Samples for assignment 1	34
A.2	Sample data for assignment 2	36

Chapter 1

The Basics

1.1 What is L^AT_EX

L^AT_EX is a language that allows you to create documents in almost any text editor by focussing primarily on *content* and *structure*, not *how it looks*.

Wikipedia describes it as:

LaTeX ([/ˈleɪtɛk/ LAY-tek](#) or [/ˈlɑːtɛk/ LAH-tek^{\[1\]}](#)) is a document preparation system and document markup language. [...] LaTeX is not the name of a particular editing program, but refers to the encoding or tagging conventions that are used in LaTeX documents. Almost any editing program or word-processor may be used to write LaTeX documents [...]

— Wikipedia

1.2 Installation

While `.tex` files are plain-text in and of itself, you need a typesetting engine and the base \LaTeX packages to be able to turn it into a pdf. In this course a working install is provided, but you probably want a working installation at home as well.

1.2.1 Linux

On Linux, there is one large \TeX distribution: [TeX Live](https://www.tug.org/texlive/)¹. Most Linux distributions have \TeX Live in their package manager. This is usually reasonably up to date.

If it's not up to date enough, the [TeX Live installer](https://www.tug.org/texlive/acquire-netinstall.html)² can be used. There are also instructions on the [TeX StackExchange](http://tex.stackexchange.com/q/1092)³ for Debian and Ubuntu.

You also need a \LaTeX editor, a list can be found on the [TeX StackExchange](http://tex.stackexchange.com/q/339)⁴.

1.2.2 Mac OS X

Simply download and install [MacTeX](https://www.tug.org/mactex/)⁵. This is actually \TeX Live prepackaged with some applications specific for OSX.

1.2.3 Windows

For Windows there are two distributions available:

- [MiKTeX](http://miktex.org/)⁶, installed on Winstin and usually the best choice. One of the features of MiKTeX is that it downloads certain packages when they are used.
- [TeX Live](https://www.tug.org/texlive/)⁷. Because TeX Live is made to be cross platform, it's not really optimized for Windows.

¹<https://www.tug.org/texlive/>

²<https://www.tug.org/texlive/acquire-netinstall.html>

³<http://tex.stackexchange.com/q/1092>

⁴<http://tex.stackexchange.com/q/339>

⁵<https://www.tug.org/mactex/>

⁶<http://miktex.org/>

⁷<https://www.tug.org/texlive/>

1.3 Commands, environments and comments

\LaTeX is mostly written using two ‘structures’: commands and environments. Commands start as follows and can have multiple *arguments*, some of which can be *optional*. The order and amount of arguments varies per command.

```
\command{argument}[optional argument]
```

In \LaTeX curly brackets (`{}`) denote mandatory arguments and square brackets (`[]`) denote optional arguments.

Environments work by using the *begin* and *end* commands.

```
\begin{environment}[optional argument]
  environment content
\end{environment}
```

The most important thing is to make sure every bracket is closed and every environment is ended. If you don't you'll get errors like:

```
Runaway argument?
[...]
! File ended while scanning use of \emph .
```

or

```
! LaTeX Error: \begin{figure} on input line 73 ended by \end{document}.
```

You can comment a line in your document by starting it with a percent sign (%):

```
% This is a comment and will not be shown in your pdf.
```

1.4 The Preamble

A \LaTeX document is usually structured as follows:

```
\documentclass{...}  
% This is the preamble  
\begin{document}  
% This is your content  
\end{document}
```

Everything before `\begin{document}` is called the *preamble*. In the preamble you usually set up your packages and document settings. More generally: in your preamble you determine the look and feel of your document.

The first line, `\documentclass{...}`, determines your *document class*, where `...` is replaced by the name of your document class. The most popular document classes are:

- **article**: for writing simple articles, without chapters.
- **report**: for writing reports, divided in chapters.
- **letter**: specifically for writings letters.
- **book**: for properly writing books.

You can load the document class with options. This document has the following:

```
\documentclass[a4paper,12pt,english,titlepage]{report}
```

Some common options are:

<code>a4paper, letterpaper</code>	Paper size
<code>11pt, 12pt, 13pt</code>	Default font size
<code>oneside, twoside</code>	One or two sided paper. This makes the outer and inner margins flip every page.
<code>onecolumn, twocolumn</code>	Change the default number of columns on every page.
<code>landscape</code>	Paper orientation
<code>titlepage</code>	Let the title take up an entire page
<code>english, dutch</code>	Pick the main language.

1.5 A basic document

Below is a basic document. This one compiles but contains almost no \LaTeX . You can use the template below and extend it to try out content from the next few sections.

```
\documentclass{article}

\begin{document}

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras
sollicitudin sem vel consectetur feugiat. Pellentesque orci tortor,
tincidunt eu ante id, consectetur sagittis mi. Nunc nec ultrices velit.
Donec sit amet mi id ipsum fringilla malesuada et in urna. Etiam
sollicitudin gravida quam vitae egestas. Duis pulvinar ut mi at
tincidunt. Nunc vestibulum hendrerit efficitur. Nulla vitae lorem
tempor, volutpat sapien convallis, blandit mauris. Phasellus nec libero
feugiat, tempor ante eget, hendrerit turpis. Donec rhoncus auctor
auctor. Proin nec dui luctus, consectetur dolor ut, facilisis lectus. Ut
vitae massa pharetra, varius augue eu, congue purus. Proin id ex a elit
malesuada porttitor.

\end{document}
```

1.6 Document structure and sections

A document is (depending on the class) divided in *parts*, *chapters*, *sections*, *subsections*, etc. These number automatically (left column). You can also add a heading without a number or it adding to the count for that heading type (right column).

<pre>% With numbers \part{Part Name} \chapter{Chapter Name} % Only book/report classes \section{Section} \subsection{Subsection} \subsubsection{Subsubsection} \paragraph{Paragraph} \subparagraph{Subparagraph} % Doesn't appear in some classes</pre>	<pre>% Without numbers \part*{Part Name} \chapter*{Chapter Name} % Only book/report classes \section*{Section} \subsection*{Subsection} \subsubsection*{Subsubsection} \paragraph*{Paragraph} \subparagraph*{Subparagraph} % Doesn't appear in some classes</pre>
---	---

This structure is apparent when you print the table of contents:

```
\tableofcontents
```

(The result is visible on page [2](#))

1.7 Title

By default, the `article`, `report` and `book` document classes come with a way to add a title. For this, you first need to define the author and title:

```
% In your preamble
\author{Me!}
\title{A document}
% \date is optional
% It's the current date (\today) by default
\date{2015-3-21}
```

Afterwards, you can add the following inside your document.

```
\maketitle
```

When you add the `titlepage` option to your document class, the title will have its own page (like this document).

1.8 Packages

By default, \LaTeX is pretty basic: you can't do much. Many of the advanced features of \LaTeX are provided through *packages*.

A package is loaded in your preamble. Packages are loaded in one of the following three ways:

```
\usepackage{packagename}
\usepackage[package-arguments]{packagename}
\usepackage{packageone,packagetwo}
```

Now let's go over some widely used packages:

1.8.1 geometry

The *geometry* package can change margins and other dimensions of your layout. Most of the time one uses it with the `a4paper` option. This can also be done when defining your *document class*.

```
\documentclass[a4paper]{article}
\usepackage{geometry}
```

```
\documentclass{article}
\usepackage[a4paper]{geometry}
```

1.8.2 babel

The *babel* package is a very nifty package that translates and changes hyphenation when a language is specified. Usage is very much like *geometry*:

```
\documentclass[dutch]{article}
\usepackage{babel}
```

```
\documentclass{article}
\usepackage[dutch]{babel}
```

1.8.3 hyperref

The *hyperref* package makes all internal links such as references and the table of contents clickable. It also adds the metadata to your PDF like the table of contents, the title and the author. It also provides clickable links to the internet. This package is a bit more complex. Loading it is simple though:

```
\usepackage[ocgcolorlinks]{hyperref} % the ocgcolorlinks argument tries
to make the link-colors non-printable.
```

You can also customize the appearance of the links:

```
\usepackage[usenames,dvipsnames,table,svgnames]{xcolor}
\hypersetup{
  colorlinks,
  citecolor=Goldenrod!80!Black,
  filecolor=Cerulean,
  linkcolor=BrickRed,
  urlcolor=Blue
}
```

It also adds the `\url` and `\href` commands:

```
\url{www.google.com} \ \      www.google.com
\href{www.google.com}{Google} Google
```

1.9 Text

Text formatting in \LaTeX is more like HTML and Markdown than Microsoft Word. What you see is *not* what you get. This has some advantages and some disadvantages, which will become clear in a moment. Let's start with some simple things:

```
\LaTeX\ ignores pretty
much    all
        whitespace (spaces and enters).
Because of this a new line has
to be done using a command: \\
\LaTeX\ also has a way of
        formatting paragraphs:
two newlines.
```

```
In this paragraph I show that you
can \emph{emphasize} words, but
you can also make them
\textbf{bold}, \textit{italic} or
\textbf{\textit{bold italic}}.
Another thing  $\LaTeX$  can do is
\textsc{small capitals}.
```

\LaTeX ignores pretty much all whitespace (spaces and enters). Because of this a new line has to be done using a command:

\LaTeX also has a way of formatting paragraphs: two newlines.

In this paragraph I show that you can *emphasize* words, but you can also make them **bold**, *italic* or **bold italic**. Another thing \LaTeX can do is SMALL CAPITALS.

By default a new paragraph is indented. This can be changed by using the *parskip* package (for more on packages, see [section 1.8](#)).

There are also three font families: roman (serif), sans-serif and typewriter (monospaced). The font family can be changed similarly to the font style:

```
\textrm{This text is serif, as is
all text by default.}
\textsf{This text is sans-serif.}
\texttt{This text is monospaced.}
```

This text is serif, as is all text by default. This text is sans-serif. This text is monospaced.

Sans-serif (or monospaced) can also be made default with the following command:

```
\renewcommand*\familydefault{\sfdefault}
```

Also look out for quotation marks and hyphens:

`This' - or ``This'' —
or——``This".

'This' - or "This" - or—"This".

1.10 Lists

You have two kinds of lists in \LaTeX *enumerated* and *itemized*:

Enumerated:

```
\begin{enumerate}
  \item An item
  \begin{enumerate}
    \item A subitem
    \item A subitem
  \end{enumerate}
\end{enumerate}
\item An item
\end{enumerate}
```

Itemized:

```
\begin{itemize}
  \item An item
  \begin{itemize}
    \item A subitem
    \item A subitem
  \end{itemize}
\end{itemize}
\item An item
\end{itemize}
```

Enumerated:

1. An item
 - (a) A subitem
 - (b) A subitem
2. An item

Itemized:

- An item
 - A subitem
 - A subitem
- An item

1.11 Images

Another thing \LaTeX does well is including images. For this the `graphicx` package is required:

```
\usepackage{graphicx}
```

An image can be added as follows:

```
\begin{figure}[ht]
  \centering
  \includegraphics[width=.8\linewidth]{logo}
  \caption{This is the Scintilla logo}
  \label{fig:logo}
\end{figure}
```



Figure 1.1: This is the Scintilla logo

Let's look at all elements:

```
\begin{figure}[ht]
\end{figure}
```

The figure environment, and where to place it. In this case 'h' (here) 't' (top). This isn't required to include an image, but is required if you want to label it.

```
\centering
```

Center the image and caption.

```
\includegraphics[width=.8\linewidth]
{logo}
```

Add the image at .8 of the text width. The file extension is not required.

```
\caption{This is the Scintilla logo}
```

The caption with text.

```
\label{fig:logo}
```

This makes sure that we can later reference this image.

You can set the default width of images by adding the following to your preamble:

```
\setkeys{Gin}{width=\linewidth}
```

1.12 Tables

One of the drawbacks of L^AT_EX's plain-text format is entering tables. Entering tables is simple enough, but can become convoluted quickly.

```
\begin{table}[ht]
  \centering
  \begin{tabular}{r | c l}
    Left & Center & Right \\ \hline
    1 & 2 & 3 \\
    3 & 6 & 9 \\
  \end{tabular}
  \caption{This is a table}
  \label{tab:table}
\end{table}
```

Left	Center	Right
1	2	3
3	6	9

Table 1.1: This is a table

Let's look at all parts of the table we haven't seen before.

```
\begin{tabular}{r | c l}
\end{tabular}
```

Start a table with three columns: one right aligned, one centered and one left aligned. Add a separator after the first column.

```
&
\\
\hline
```

End a column.

End a line.

Add a horizontal bar after the line.

When having to enter a large table, it might be convenient to use a converter like [this online tables generator](http://www.tablesgenerator.com/)⁸ or [this one](http://truben.no/table/)⁹.

⁸<http://www.tablesgenerator.com/>

⁹<http://truben.no/table/>

1.13 Labels and references

One thing that is very useful when making reports is L^AT_EX's way of handling references. Let's refer to the figure and table:

```
figure~\ref{fig:logo} \\
table on page~\pageref{tab:table} \\
\autoref{fig:logo} \\
\nameref{tab:table}
```

figure 1.1
table on page 17
Figure 1.1
This is a table

It's also possible to link to sections:

```
\label{sec:this one}
\autoref{sec:this one}
on page~\pageref{sec:this one}
```

section 1.13 on page 18

The first lines are L^AT_EX's native way of using references. Both `\autoref` and `\nameref` are provided by the `hyperref` package, which also turns the references into hyperlinks.

There is one caveat: often references will not resolve after one compilation of your document, so you should always run it twice. In chapter 2 the `latexmk` command is discussed, which solves this problem.

1.14 Footnotes and Margin Notes

In \LaTeX there are two main ways of adding notes: at the bottom (foot notes) and in the margin (margin notes).

Footnotes are a way to provide additional information if you don't have a wide margin to the side of your page. They are well supported in \LaTeX , and are numbered automatically:

```
Footnote\footnote{Look below!} Footnote10
```

Margin notes are a bit different in the sense that you need a different layout for your document to use them. They are also not numbered, and are represented like normal text:

```
By default:  
\marginpar{This text is placed in the margin.}
```

By default:

This text is
placed in the
margin.

¹⁰Look below!

1.15 Mathematics

Maybe one of the best features \LaTeX has is its extensive support of mathematics. The [\$\text{\LaTeX}\$ Wikibook](#)¹¹ has an extensive page on it, along with all characters.

The \LaTeX notation for mathematics is quite widespread, in the latest versions of Microsoft Word simple \LaTeX mathematic notation can almost be copied verbatim into MS Word equations.

The most important thing about mathematics in \LaTeX is that there are two styles:

- Inline style mathematics are entered between $\$$ and $\$$. The height of all mathematics is reduced so it fits more comfortably in a line of text.
- Display style mathematics are entered between $\backslash[$ and $\backslash]$. The height of the mathematics is unrestricted, which allows for quite convoluted mathematics to be displayed properly.

This equation can be displayed as

```
 $\int_0^\infty \frac{1}{\sqrt{2}} \mathrm{e}^{-x} \mathrm{d}x$ ,  
 $\int_0^\infty \frac{1}{\sqrt{2}} \mathrm{e}^{-x} \mathrm{d}x$ ,
```

or as:

```
 $\int_0^\infty \frac{1}{\sqrt{2}} \mathrm{e}^{-x} \mathrm{d}x$ ,  
 $\int_0^\infty \frac{1}{\sqrt{2}} \mathrm{e}^{-x} \mathrm{d}x$ 
```

This equation can be displayed as $\int_0^\infty \frac{1}{\sqrt{2}} \mathrm{e}^{-x} \mathrm{d}x$, or as:

$$\int_0^\infty \frac{1}{\sqrt{2}} \mathrm{e}^{-x} \mathrm{d}x$$

When working with complex mathematics, the `amsmath` package provides more characters and some environments. The most important environments are `equation` (which allows numbered and labeled equations) and `align` (which allows aligned mathematics):

```

$$v_C = \frac{1}{sC} \cdot i_C$$
  

$$v_C = \frac{1}{sC} \cdot i_C \\ = \frac{1}{sRC} \cdot v_R$$

```

$$v_C = \frac{1}{sC} \cdot i_C \quad (1.1)$$

$$v_C = \frac{1}{sC} \cdot i_C \quad (1.2)$$

$$= \frac{1}{sRC} \cdot v_R \quad (1.3)$$

Numbering can be turned off by using the `equation*` and `align*` environments.

¹¹<http://en.wikibooks.org/wiki/LaTeX/Mathematics>

1.16 Source Code Listings

Sometimes you have to show your code, for example in a document for a \LaTeX course. The `listings` package provides the `lstlisting` environment and the `lstinputlisting` command:

Regular listing:

```
\begin{lstlisting}
[Some source code here]
\end{lstlisting}
Input from (this) file:
\lstinputlisting[firstline=21,
  lastline=23]{manual.tex}
```

Regular listing:

```
[Some source code here]
```

Input from (this) file:

```
-
\let\parindentold\parindent
\usepackage{fontspec,geometry,babel,listings,parskip,
\usepackage[ocgcolorlinks]{hyperref}
```

You can also style your source code listings so they stand out more:

```
\lstset{
  language=[LaTeX]TeX,
  upquote=true,
  basicstyle=\small\ttfamily,
  numberstyle=\tiny\color{Gray}\ttfamily,
  stepnumber=1,
  numbersep=5pt,
  rulecolor=\color{black},
  backgroundcolor=\color{black!10},
  tabsize=4,
  captionpos=b,
  breaklines=true,
  title=\lstname,
  keywordstyle=\color{RoyalBlue},
  commentstyle=\color{ForestGreen},
  stringstyle=\color{WildStrawberry},
}
```

1.17 Getting content from different files

When working on a large document (say, a thesis), it's often a good idea to split your document in multiple parts. When working with multiple people this also allows you to work on different parts at the same time.

The easiest way to get content from a different file is using the `\input` command, this simply uses the contents from a different file as if they were entered in the document itself.

Another option is the `\include` command, this works almost the same, but analyses the files separately and adds it after a page break. It also create an `.aux` file. `\include` cannot be used in the preamble, and cannot be used in a file that is used via `\include`. But using `\include` does allow the omission of all other `\include` commands by using `\includeonly`.

```
\documentclass{article}
\begin{document}
% Get the content from `introduction.tex`
\input{introduction}
% Get the content from `conclusion.tex` on a new page.
\include{conclusion}
\end{document}
```

Another option is to use subfiles, this is explained very well in [the L^AT_EX Wikibook](http://en.wikibooks.org/wiki/LaTeX/Modular_Documents#Subfiles)¹².

¹²http://en.wikibooks.org/wiki/LaTeX/Modular_Documents#Subfiles

1.18 Common problems

L^AT_EX is sometimes a little unpredictable. Some common causes of problems are:

- Spaces in filenames.
- Text encoding: always use utf-8. When using `pdflatex` add `\usepackage[utf8]{inputenc}` to your preamble. `xelatex` (see section 2.3.2) solves this problem.

Another thing to look out for is *gobble*: L^AT_EX commands without an argument ‘eat’ the next space. This can be avoided by ending the command with a backslash (`\`) or two brackets (`{}`).

<code>\LaTeX is awesome, \</code>	L ^A T _E X is awesome,
<code>\LaTeX\ is awesome, \</code>	L ^A T _E X is awesome,
<code>\LaTeX{} is awesome, \</code>	L ^A T _E X is awesome,
<code>{\LaTeX} is awesome.</code>	L ^A T _E X is awesome.

What you should also look out for are some reserved characters, some of them are shown in table 1.2.

Symbol	Command	Output
<code>^</code>	<code>\^{} </code>	<code>^</code>
<code>_</code>	<code>_ </code>	<code>_</code>
<code>&</code>	<code>\& </code>	<code>&</code>
<code>\</code>	<code>\textbackslash </code>	<code>\</code>
<code>{</code>	<code>\{ </code>	<code>{</code>
<code>}</code>	<code>\} </code>	<code>}</code>
<code>\$</code>	<code>\\$ </code>	<code>\$</code>
<code>%</code>	<code>\% </code>	<code>%</code>
<code>~</code>	<code>\~{} </code>	<code>~</code>
<code>#</code>	<code>\# </code>	<code>#</code>

Table 1.2: Troublesome symbols

For more common mistakes please look at [this question on stackexchange](http://tex.stackexchange.com/q/139873)¹³.

¹³<http://tex.stackexchange.com/q/139873>

Chapter 2

Beyond

2.1 Bibliography management

When using \LaTeX for a large report or thesis it's often good to use \LaTeX for the bibliography as well. This works by using a bibliography database, a program to parse it, and a package to show your bibliography in your document.

2.1.1 Bibliography database

A bibliography is usually included as a `.bib` database. `.bib` files are plain-text, and quite simple to type yourself.

```
@article{glashow,  
  author      = {Glashow, Sheldon},  
  title       = {Partial Symmetries of Weak Interactions},  
  journaltitle = {Nucl.~Phys.},  
  date        = 1961,  
  volume      = 22,  
  pages       = {579–588},  
}
```

Most important is the *key*, here `glashow`. This key is used to refer to an entry in your database.

There is also a lot of software that can help in managing a bibliography, JabRef for example. Some more software is listed in [the \$\LaTeX\$ Wikibook¹](#).

¹http://en.wikibooks.org/wiki/LaTeX/Bibliography_Management#Helpful_tools

2.1.2 Typesetting with a bibliography

When using a bibliography, you also need to parse your bibliography database. This is done using a parser like `biber`. When you typeset your document, the workflow is now:

- `pdflatex`
- `biber`
- `pdflatex`

This is not very convenient, but there are solutions that do this for you. More on this is explained in section [2.3.1](#).

2.1.3 Bibliography package

To load the bibliography features, use the `biblatex` package.

```
\usepackage[backend=biber,style=ieee]{biblatex}
% Load `biblatex-examples.bib`
\bibliography{biblatex-examples}
```

When loading `biblatex` you can select the bibliography parser (`bibtex`) and the citation style (IEEE). The citation style can be changed to whatever is required for your document. Some well known styles are ACS (as `chem-acs`), AIP (as `phys`), nature, science, `ieee`, Chicago (as `chicago-authordate`), `mla` and `apa`.

Now you can use the citation commands `\textcite` and `\cite` to refer to publications in your database:

```
As \textcite{glashow} noted.
We know that this
works\cite{spiegelberg}.
```

But

```
\textcite[p.~172]{aristotle:rhetoric}
noted this in
untrue\cite[p.~12]{augustine}.
```

```
As Glashow [1] noted. We know that
this works[2]. But Aristotle [3, p. 172]
noted this in untrue[4, p. 12].
```

What remains is printing the bibliography:

```
\printbibliography
```

Bibliography

- [1] S. Glashow, “Partial symmetries of weak interactions”, *Nucl. Phys.*, vol. 22, pp. 579–588, 1961.
- [2] H. Spiegelberg, “‘Intention’ und ‘Intentionalität’ in der scholastik, bei brentano und husserl”, *Studia Philosophica*, vol. 29, pp. 189–216, 1969.
- [3] Aristotle, *The Rhetoric of Aristotle with a commentary by the late Edward Meredith Cope*, E. M. Cope, Ed., with a comment. by E. M. Cope, 3 vols. Cambridge University Press, 1877.
- [4] R. L. Augustine, *Heterogeneous catalysis for the synthetic chemist*. New York: Marcel Dekker, 1995.

2.2 Fancy headers

To get headers on your page (like this one). You can use the `fancyhdr` package:

```
\usepackage{fancyhdr}
\lhead{Author's Name, top left}
\chead{\today}
\rhead{\thepage}
\lfoot{\textit{Author's Name, bottom left}}
\cfoot{\thepage}
\rfoot{\textsc{\today}}
\pagestyle{fancy}
```

There are a lot more options, but for those I'll refer to the [Wikibooks page](#)²

²http://en.wikibooks.org/wiki/LaTeX/Page_Layout#Customizing_with_fancyhdr

2.3 L^AT_EX engines

2.3.1 latexmk

As has been noted before, you need to run `pdflatex` multiple times. Sometimes mixed with other commands. This is inconvenient, but there is a solution for this: `latexmk`.

`latexmk` checks your files to determine what commands to run and how many times. It makes sure to resolve your references, load your bibliography and generate the required files.

If your editor doesn't have a `Latexmk` option, the command line is:

```
latexmk -pdflatex="pdflatex -synctex=1 %0 %S" -pdf #filename#
```

In TeXworks, it can be added (in Settings → Typesetting → Commands) with the following settings:

```
Program: latexmk
Arguments: -pdflatex=pdflatex $synctexoption %0 %S
           -pdf
           $fullname
[X] View PDF after running
```

`Latexmk` can also do a cleanup of all generatable files:

```
latexmk -C #filename#
```

2.3.2 Xe^LA_TE_X & Lua^AT_EX

Two alternatives for `pdflatex` are `xelatex` and `lualatex`. Both of these engines are newer and allow you to make use of features that are coming in L^AT_EX3. This means you can use several packages and features you cannot use with `pdflatex`. In the next few sections examples are included that only work in these two engines.

`Latexmk` can be configured to use this engine as follows:

```
latexmk -pdflatex="xelatex -synctex=1 %0 %S" -pdf #filename#
```

2.4 Fonts

One thing you probably want to change about your document is the font. This is not one of the things $\text{\LaTeX} 2_{\epsilon}$ excels at. \LaTeX was created before fancy things like TrueType and OpenType fonts, so it doesn't have any support for them.

$\text{\LaTeX} 3$, however, aims to fix things like this. When using \XeTeX or \LuaTeX you can make use of this using the `fontspec` package.

2.4.1 $\text{\LaTeX} 2_{\epsilon}$ Packages

Using a font using the old way can be cumbersome. To make this easier many people have created packages with converted font files. In [the \$\text{\LaTeX}\$ Font Catalogue](#)³ many font packages are listed.

With such a package, usage is often very easy:

```
\usepackage{tgtermes} % Times clone, roman (rm)
\usepackage{tggheros} % Helvetica clone, sans-serif (sf)
\usepackage{tgcursor} % Courier clone, monospace (tt)
\usepackage[T1]{fontenc}
```

In case of a *sans-serif* font you can make the `sf`-family default.

```
\renewcommand*\familydefault{\sfdefault}
```

2.4.2 $\text{\LaTeX} 3$: Fontspec

Usage of `fontspec` is a lot easier, you can just use locally installed fonts:

```
\usepackage{fontspec}
\setmainfont{Times New Roman} % Roman (rm)
\setsansfont{Arial} % Sans-serif (sf)
\setmonofont{Courier} % Monospace (tt)
```

³<http://www.tug.dk/FontCatalogue/>

2.5 Variables & Commands

In \LaTeX you can make use of three ways to define a command:

```
% TeX
\def\firstmacro#1#2{Macro #1}
% LaTeX2e
\newcommand\secondmacro[1]{Macro #1}
% LaTeX3, from the `xparse` package
\DeclareDocumentCommand\thirdmacro{
  m }{Macro #1}

First: \firstmacro{1} \\
Second: \secondmacro{2} \\
Third: \thirdmacro{3}
```

I do *not* recommend using `\def` for anything except storing simple text, because it's very finicky and will overwrite commands without prompt. `\newcommand` is also very limited, so unless you have no choice: use `\DeclareDocumentCommand`.

The syntax of `\DeclareDocumentCommand` is quite straightforward:

```
\DeclareDocumentCommand\foo{ s o O{bar} m }{%
  % s : star, like \foo vs \foo*
  \IfBooleanTF{#1}{With star}{Without star}
  % o : optional without default
  \IfNoValueTF{#2}{Without argument}{With argument: #2}
  % O{bar} : optional with default value (bar)
  #3
  % m : mandatory argument
  #4
}
% If the command is already defined and you want to replace it:
\RenewDocumentCommand\foo{ m }{Argument: #1}
```

There are a lot more options, but for those you should really look at [the documentation](#)⁴ or [this presentation](#)⁵.

⁴<http://mirrors.ctan.org/macros/latex/contrib/l3packages/xparse.pdf>

⁵<http://uk.tug.org/wp-content/uploads/2009/11/xparse.pdf>

2.6 Editor tricks

Most \LaTeX programs support so-called *directives*. These are comments at the start of the file that instruct your editor to use a certain setting.

- `% !TEX program = xelatex`

The program to run this file with. This is not an actual command line and needs to correspond to a configured command.

- `% !TEX spellcheck = en_GB`

The language to use for the spell-check of your editor. This is usually the language code.

- `% !TEX root = main-document.tex`

If the current file is included, adding this will make sure the editor typesets the main document instead of the current file.

Chapter 3

Assignments

3.1 Assignment 1

Create a document specifically for this course of at least 5 pages with:

- A title page
- At least one section and one subsection
- Some text (go wild!)
- A list of things
- Two images
- Two tables
- References to both images and tables
- At least one footnote somewhere
- A mathematical and verbal description of the abc-formula, with the discriminant analysed separately and aligned properly.
- Split your content over multiple files

Sample data is provided in [Samples for assignment 1](#).

3.2 Assignment 2

Modify the document created in [section 3.1](#). Add the following features:

- Cite something using a bibliography.
- Show the author and page number in the header.
- Use a custom font.

Sample data is provided in [section A.2](#).

Appendix A

Sample data for the assignments

A.1 Samples for assignment 1

Sample images:

- Image 1:http://imgs.xkcd.com/comics/file_extensions.png
- Image 2:<http://imgs.xkcd.com/comics/actually.png>

×	1	2	3
1	1	2	3
2	2	4	6
3	3	6	9

Table A.1: Multiplication table

+	1	2	3
Nancy	Nancy Davolio	Nancy Davolio	Nancy Obesanjo
Justin	Justin Saunders	Justin Timberland	Justin Daviolio

Table A.2: Modified Age Table (names only)

ABC-formula:

$$0 = ax^2 + bx + c \tag{A.1}$$

$$x = \frac{-b \pm \sqrt{d}}{2a} \tag{A.2}$$

$$d = b^2 - 4ac \tag{A.3}$$

Hint:

```
% In preamble
\usepackage{lipsum}
% Elsewhere
\lipsum[1-3]
```

A.2 Sample data for assignment 2

- Bibliography database:

<https://www.scintilla.utwente.nl/docs/cursus/latex/manual/biblatex-examples.bi>