

ATK tähtitieteessä

Osa 6 - \LaTeX

Sébastien Comerón – based on the notes by Jarkko Laine

8. toukokuuta 2017

What is L^AT_EX?

- ▶ Most of the mainstream text editors follow a “What You See Is What You Get” (WYSIWYG) philosophy. You type and edit the text as it is meant to look in the paper or pdf output.
- ▶ L^AT_EX is a typesetting system where how the document looks is taken care of by a compiler. It behaves like a programming language aimed at producing documents. The author writes a document in plain text following certain rules and the compiler interprets it.
- ▶ For example, in a WYSIWYG word processor the author manually chooses where to put an illustration. In L^AT_EX the compiler decides where to place the images. So in L^AT_EX the author is freed from the burden of preparing the text layout.

The advantages of \LaTeX

- ▶ You do not have to take care of the document layout. This might be a very complicated issue, for example, in a WYSIWYG document with a lot of images.
- ▶ Typing mathematical formulæ is relatively easy in \LaTeX .
- ▶ It is easy to add references, footnotes. . .
- ▶ \LaTeX compilers are free and can be found for wide array of platforms
 - ▶ Linux: \TeX Live is provided by default in many Linux distributions.
 - ▶ Windows: Protext
 - ▶ Mac: Mactext

The disadvantages of \LaTeX

- ▶ We need to learn how to “program” in \LaTeX .
- ▶ Installing new packages is not always straightforward.

A plain text editor

- ▶ We need a text editor to write the plain text file that will later be compiled.
- ▶ Some text editors are especially designed for typing \LaTeX documents.
 - ▶ In Linux we can for example use Kile (the one I suggest using today because it is the one I know the best) or Texmaker. Both of them includes features that are useful at writing \LaTeX documents. Alternatively, we can use our favourite text editor such as emacs or vi.
 - ▶ In Windows we can use TexStudio.
 - ▶ In Mac we can use TexShop.

The .tex file

- ▶ \LaTeX documents are written in plain text. The files must have a `.tex` extension.
- ▶ A document is structured as follows:
 - ▶ A preamble that sets the document type and calls the packages that will be used.
 - ▶ The text itself.
 - ▶ The document is ended.

Setting the document style

- ▶ The first line of the document must set the document class

```
\documentclass[options]{document style}
```

- ▶ There are several different document styles:

- ▶ `article`
- ▶ `report`
- ▶ `book`
- ▶ `letter`
- ▶ ...

- ▶ The options can set a lot of different parameters such as

- ▶ The font size, e.g., `10pt`, `12pt`
- ▶ The paper size, e.g., `a4paper`, `a5paper`
- ▶ The number of columns, e.g., `onecolumn`, `twocolumn`
- ▶ ...

- ▶ So, if we want to write an article with a 12 point font in an A4 paper we would write

```
\documentclass[12pt, a4paper]{article}
```

Calling packages

- ▶ We can call packages that activate features that are not available in the basic \LaTeX . The call sequence is

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

- ▶ Some useful packages to be called:

- ▶ To allow the use of “strange” characters

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

Thanks to this command we can write things like

Lähtöjärjestys

instead of

```
L\"aht\"oj\"arjestys
```

- ▶ To allow the document to be hyphenated according to Finnish rules (the default is English)

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


Calling packages

- ▶ More useful packages

- ▶ To allow the use of graphics such as pictures

```
\usepackage{graphicx}
```

- ▶ To add mathematical functionalities

```
\usepackage{amsmath}
```

- ▶ To add fonts and symbols used in mathematical formulæ

```
\usepackage{amsfonts}  
\usepackage{amssymb}
```

- ▶ To be able to change the colour of the text

```
\usepackage{color}
```

- ▶ Many more packages that allow to make fancy tables, drawings, add fonts. . . can be found online

An example of a preamble

- ▶ A preamble for an article in Finnish with images and formulae should look like that

```
\documentclass[12pt, a4paper]{article}
\usepackage[utf8]{inputenc}
\usepackage[finnish]{babel}
\usepackage{graphicx}
\usepackage{amsmath}
\usepackage{amssymb}
\usepackage{amssymb}
\usepackage{color}
```

Beginning and ending a document

- ▶ Once the preamble is written, we can start the document itself by writing

```
\begin{document}
```

Write here whatever you want.

```
\end{document}
```

- ▶ This way of proceeding to begin an environment with the command `\begin{environment}` and ending it with `\end{environment}` is used for many environments in \LaTeX like for making tables, pictures, and equations as we will see later.

The title section

- ▶ We could now start writing text. However, it would be good to add a title to the text. This is done by defining the variables for the title section or page. Common title page variables to be defined are `\title{Title}`, `\author{Author}`, `\date{Date}`...
- ▶ The title page is compiled by including the command `\maketitle`.

The title section

- ▶ Here comes a brief example. After the preamble, to get a nice title section we should write something like:

```
\begin{document}

\title{The Hobbit}
\author{J.~R.~R.~Tolkien}
\date{21 September 1937}

\maketitle
```

Note: ~ is used after a dot to indicate that this dot is not the end of a sentence. This changes the spacing between the dot and the next word. In a sentence end this spacing is larger. By using ~ we make it smaller, like the space between two words.

- ▶ If you use `\date{\today}` for the date field, you will automatically obtain today's date as indicated by the computer clock.

Compiling the .tex file

- ▶ Let's compile out first document! In a command line in the same folder as we have the *tex file we can write

```
latex name_of_the_document.tex
```

This will produce a *dvi file that can be read with some linux software like xdvi or okular.

- ▶ If we want to obtain a pdf file we can type

```
pdflatex name_of_the_document.tex
```

- ▶ Those two commands will show warnings if we have made mistakes at coding our *tex file.
- ▶ Kile and Texmaker also have buttons that will produce *dvi and *pdf outputs. However the command line might be necessary to see the error messages to help us in the debugging process. F11 in kile will allow you to see the line numbers, which is very useful for debugging. F10 will divide long lines.

Basic rules of typing

- ▶ Multiple spaces are considered as a single space. Spaces at the beginning of a line are ignored.
- ▶ A per cent symbol % marks a comment. What comes after % in a line will not be compiled and shown in the output.
- ▶ If you want to start a new paragraph you have to leave a blank line.
- ▶ If you want to start a new page use `\newpage`.
- ▶ Several symbols cannot be written straight away (% , \$, & , { , } , - , \) because they are used to mark comments or other features in a *tex document. We should write

```
\%, \$, \&, \{, \}, \-, \textbackslash
```

instead.

- ▶ To change the font size write `{\FontSize Place the text here}`. The available font sizes are `\Huge`, `\huge`, `\LARGE`, `\Large`, `\large`, `\normalsize` (default), `\small`, `\footnotesize`, `\scriptsize`, and `\tiny`.

Basic rules of typing

- ▶ You can have several text modes, for example:
 - ▶ `\emph{text}` – *italics*.
 - ▶ `\texttt{text}` – type-writer font.
 - ▶ `\textbf{text}` – **bold face**.
 - ▶ `\underline{text}` – underlined text .
 - ▶ `{\color{blue}{text}}` – blue text.
- ▶ There is a humongous amount of symbols that can be included in \LaTeX documents. A long list can be found at https://www.rpi.edu/dept/arc/training/latex/LaTeX_symbols.pdf.

Dividing the document in sections, subsections and so on

- ▶ If you want to create a section, just use one of the following commands
 - ▶ `\section{section title}`
 - ▶ `\subsection{subsection title}`
 - ▶ `\subsubsection{subsubsection title}`
 - ▶ `\paragraph{paragraph title}`
 - ▶ `\subparagraph{subparagraph title}`
- ▶ The sections will be automatically numbered. For example, the first section will be section number **1** and if you build a subsection inside it will be called subsection **1.1**.

Creating list

- ▶ A non-numerated list can be created by using the `itemize` environment. Example:

```
\begin{itemize}
  \item 1950: Giuseppe Farina
  \item 1951: Juan Manuel Fangio
\end{itemize}
```

results in

- ▶ 1950: Giuseppe Farina
 - ▶ 1951: Juan Manuel Fangio
- ▶ If instead we want a numbered list (so with ordinal numbers instead of bullet points), we have to use the `enumerate` environment instead of `itemize`.

Creating tables

- ▶ Tables are created within the `tabular` and `table` environments.
- ▶ Let's first see the example of the code for a table:

```
\begin{center}
\begin{table}
\begin{tabular}{l | c | c | r}
Driver & Wins & Podiums & Points \\
\hline
Nico Rosberg & 9 & 16 & 385 \\
Lewis Hamilton & 10 & 17 & 380 \\
\hline
\end{tabular}
\end{table}
\end{center}
```

Driver	Wins	Podiums	Points
Nico Rosberg	9	16	385
Lewis Hamilton	10	17	380

Creating tables

- ▶ In the example the `table` and `tabular` environments have been placed within a `center` environment so the table is found at the centre of the page.

- ▶ The line

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

tells us that the table contains four columns. The first column is aligned to the left (`l`), the second and the third columns are centred (`c`), and the third column is aligned to the right (`r`). The vertical lines also called pipes (`|`) indicate that each of the columns has to be separated by a line.

- ▶ The command `\hline` indicates an horizontal line.
- ▶ In a given row, the values in different columns are separated by the symbol `&` and the changes of row are indicated by `\\`.

Adding pictures

- ▶ Pictures can be added with the `figure` environment.
- ▶ Within the `figure` environment a picture can be included with the following command:

```
\includegraphics[options]{image_name}.
```

- ▶ Usually \LaTeX places the images wherever it wants following some somewhat mystic internal rules. We can force \LaTeX to place it in a specific place by adding the preferred location of the image within square brackets after `\begin{figure}` (this does not necessarily work very well):
 - ▶ `h`: here.
 - ▶ `t`: on the top of the page.
 - ▶ `b`: on the bottom of the page.
 - ▶ `p`: on the same page as the surrounding text

Adding pictures

- ▶ There is plenty of nice options for `\includegraphics`. A few examples:
 - ▶ `width=0.48\textwidth`: the picture is scaled so it occupies 0.48 times the width occupied by the text.
 - ▶ `height=0.3\textheight`: the picture is scaled so it occupies 0.3 times the height occupied by the text.
- ▶ Captions can be added with `\caption{Caption text here}`
- ▶ A full example on a centred picture added using the `figure` environment occupying half of the vertical extension of the page and located on the top:

```
\begin{figure}[t]
  \begin{center}
    \includegraphics[height=0.5\textheight]{cmb.pdf}
  \end{center}
  \caption{The Cosmic Microwave Background
as seen by Planck.}
\end{figure}
```

The table of contents

- ▶ \LaTeX can create tables of contents based on the sections defined in the text. To do so, just use the command `\tableofcontents`.

Mathematical formulæ

- ▶ There is two ways to introduce mathematical symbols. One is within the text between two \$ symbols. For example, typing `$E=mc^2$` yields $E = mc^2$.
- ▶ Another possibility is the `equation` environment. For example, the code

```
\begin{equation}
\left(\frac{H}{H_0}\right)^2=
\frac{\Omega_{\rm r,0}}{a^4}+
\frac{\Omega_{\rm m,0}}{a^3}+\Omega_{\Lambda,0}+
\frac{1-\Omega_{\rm r,0}-\Omega_{\rm m,0}-\Omega_{\Lambda,0}}{a^2}
\end{equation}
```

yields the following result

$$\left(\frac{H}{H_0}\right)^2 = \frac{\Omega_{\text{r},0}}{a^4} + \frac{\Omega_{\text{m},0}}{a^3} + \Omega_{\Lambda,0} + \frac{1 - \Omega_{\text{r},0} - \Omega_{\text{m},0} - \Omega_{\Lambda,0}}{a^2}. \quad (1)$$

Mathematical formulæ

► Frequently used features:

- ▶ `\frac{1}{2}` division ($\frac{1}{2}$),
- ▶ `\times` , `\cdot` multiplication (\times , \cdot),
- ▶ `\sqrt{x}` square root (\sqrt{x}),
- ▶ `\ge` greater or equal than (\geq),
- ▶ `\le` smaller or equal than (\leq),
- ▶ `\ne` not equal (\neq),
- ▶ `\in` included in (\in),
- ▶ `\sum` summation (\sum),
- ▶ `\prod` product (\prod),
- ▶ `\int` integral (\int),
- ▶ `\x_{a}` subindex (x_a),
- ▶ `\x^{2}` superindex (x^2),
- ▶ `\alpha` , `\beta`, greek letters (α , β),
- ▶ `\leftarrow` , `\rightarrow` right and left arrows (\leftarrow , \rightarrow).

Mathematical formulæ

- ▶ When we want to place a parenthesis, we may use `(` and `)`. However, those will have a normal size. If we want to enclose something that is larger than normal text (think of a fraction), we need to place `\left(` and `\right)`. The same applies for square brackets and similar things.
- ▶ A typing convention is that subindices that are not running subindices are not in italics. However, in \LaTeX the mathematical mode uses italic characters by default. To shift to non-italic characters we use the `\rm` command.
- ▶ \LaTeX has an infinity of mathematical symbols that we can use. It is however difficult to remember them all. Kile has on its left side a menu (indicated by the α symbol) where you can find many mathematical symbols not covered by this introduction.

Labelling an referencing

- ▶ A label is placed by using command:

`\label{label}`.

within an environment or section. In a table environment it has to be placed within a caption to work well. You can refer to sections, graphics, equations, and tables.

- ▶ You can refer to the label by using `\ref{label}`.

Labelling an referencing

- ▶ An example:

```
\begin{center}
  \begin{table}
    \begin{tabular}{l | c | c | r}
      Driver & Wins & Podiums & Points \\ \hline
      Nico Rosberg & 9 & 16 & 385 \\ \hline
      Lewis Hamilton & 10 & 17 & 380 \\ \hline
    \end{tabular}
    \caption{\{table} Results of the F1 in 2016.}
  \end{table}
\end{center}
```

In `\ref{table}` we see the results of the 2016 F1 World Championship.

Labelling an referencing

- ▶ The result:

Driver	Wins	Podiums	Points
Nico Rosberg	9	16	385
Lewis Hamilton	10	17	380

Taulukko: Results of the F1 in 2016.

In Table 1 we see the results of the 2016 F1 World Championship.