LaTeX for Psychological Researchers

Lecture 2: Basics of the LaTeX language

Sacha Epskamp

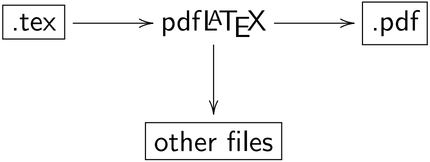
University of Amsterdam
Department of Psychological Methods

10/04/2013

Outline

- Last Week
- The LaTeX Language
  - Commands
- Document Structure
  - Documentclass
  - Preamble
- Body of a document
  - Sectioning
  - Environments
  - Figures and Tables
  - floats
- Mathematics

The LaTeX process



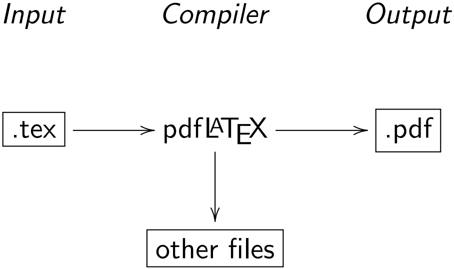
.tex

pdfLaTeX

.pdf

other files

The LaTeX process



Input

Compiler

Output

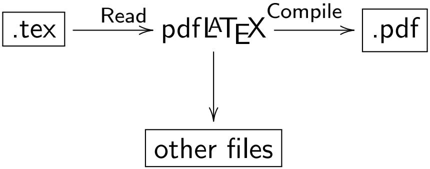
.tex

pdfLaTeX

.pdf

other files

The LaTeX process



.tex

Read

pdfLaTeX

Compile

.pdf

other files

The LaTeX process

LaTeX refers to the *programming language* used to write the input file and the *program* used to interpret this file and compile the output file. It does **not** refer to an editor in which you write the input file.

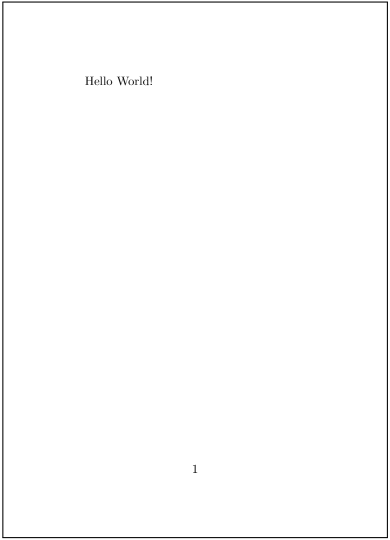
For clarity, in this course LaTeX refers only to the language, the compiler is referred to as "the compiler" generally or pdfLaTeX specifically.

Hello World Example

.tex file:

```latex
\documentclass{article}
\begin{document}

Hello World!

\end{document}
```

.pdf file:



Hello World!

1

Outline

Last Week

The LaTeX Language

Commands

Document Structure

Documentclass

Preamble

Body of a document

Sectioning

Environments

Figures and Tables

floats

Mathematics

Today

- Today we will be looking at the basics of the LaTeX language
- This is possibly a very boring lecture with many commands and terms
- This material is covered in many different resources online. For example, the getting started part of the wikibook:

  `http://en.wikibooks.org/wiki/LaTeX/`

- Try not to memorize everything, but try to understand what is happening!

The LaTeX language

- LaTeX is the markup/programming language that the compiler can understand and process
- LaTeX consists only of ASCII characters
- Just like a natural language LaTeX it uses *syntax* and *semantics*
- Also just like a natural language LaTeX has exceptions
- These must be correct. Unlike humans the compiler cannot understand the language with spelling errors

LaTeX commands

A LaTeX document is mainly defined through **commands**. These generally take one of two formats:

- A backslash \ and then a name consisting of letters only. Command names are terminated by a space, a number or any other “non-letter”
- A backslash \ and exactly one non-letter

For example:

```
\int \alpha \beta \$ \& \[ \]
```

LaTeX commands

Some commands need an argument, which has to be given between curly braces { } after the command name (with some exceptions). Some commands support optional parameters, which are added after the command name in square brackets [ ]. The general syntax is:

```
\commandname[option1,option2,...]{argument1}{argument2}...
```

LaTeX Arguments

```
\commandname[option1, option2, ...]{argument1}{argument2}...
```

- Many commands require a single argument, and some commands require even multiple arguments
- Generally an argument is interpreted in one of three ways:
    - The text a command is applied to
        - `\textit{text to make italic}`
    - The mode or specification of a command
        - `\documentclass{classname}`
    - A name or title to be used for something
        - `\usepackage{packagename}`

LaTeX Options

```
\commandname[option 1, option 2, ...]{argument1}{argument 2}...
```

- Some commands can have several options
- Often optional
- Each option is a verbatim term of one of several mutually exclusive options
  - `\documentclass[a4paper,11pt]{article}`
  - `\usepackage[dutch]{babel}`
  - `\sqrt[3]{x}`

Environments

- Two special commands are `\begin{environment}` and `\end{environment}`
- These initiate and exit an environment
- The type of environment is applied to everything between the begin and end commands
- These must be properly nested
- As we will see, the entire LaTeX document is one big environment

Environments

LaTeX code:

```latex
\begin{itemize}
\item Two special commands are \verb|\begin{environment}| and \verb|\end{environment}|
\item These initiate and exit an environment
\item The type of environment is applied to everything between the begin and end commands
\item These must be properly nested
\item As we will see, the entire \LaTeX\ document is one big environment
\end{itemize}
```

Output:

- Two special commands are `\begin{environment}` and `\end{environment}`
- These initiate and exit an environment
- The type of environment is applied to everything between the begin and end commands
- These must be properly nested
- As we will see, the entire LaTeX document is one big environment

Environments

LaTeX code:

```latex
Apply Knuth's advice to \LaTeX:
\begin{quote}
The best way to learn how to use \LaTeX\ is to use it. Thus it's high time for you to sit down at a computer terminal and interact with the \LaTeX\ system, trying things out to see what happens.
\end{quote}
```

Output:

Apply Knuth's advice to LaTeX:

> *The best way to learn how to use LaTeX is to use it. Thus it's high time for you to sit down at a computer terminal and interact with the LaTeX system, trying things out to see what happens.*

Outline

- Last Week
- The LaTeX Language
  - Commands
- Document Structure
  - Documentclass
  - Preamble
- Body of a document
  - Sectioning
  - Environments
  - Figures and Tables
  - floats
- Mathematics

Document Structure

Every LaTeX document has the following form:

```
\documentclass{ CLASS }

    PREAMBLE

\begin{document}

    BODY

\end{document}
```

Document Structure

- The first line is always the `\documentclass` command, specifying the *class of the document*
  - This specifies what *sort* of document you intend to write
- After the document-class comes the *preamble*
  - Load needed packages
  - Define things that affect the whole document
  - Define variables, custom commands or other things needed in the document
- After the preamble comes the *body*
  - Starts with `\begin{document}` and ends with `\end{document}`
  - This is the main document
  - Contains all text, figures, tables, etcetera
- Finally, anything added after `\end{document}` is not compiled. This room can be used for comments

Document Structure

Every LaTeX document has the following form:

```
\documentclass{ CLASS }

    PREAMBLE

\begin{document}

    BODY

\end{document}
```

The Document Class

The first line of any LaTeX document is always to define the document class, written as `\documentclass{CLASS}`. You can define the type of the document that you will write by changing `CLASS` in the class you want.

```
\documentclass[option1,option2,...]{CLASS}
```

The Document Class Arguments

A few possible documentclasses:

Document class	Output
`article`	For writing articles, but this is also your general all-purposes class.
`report`	Can contain chapters. For larger articles, thesis and small books.
`book`	For larger books.
`letter`	For letters.
`apa`	For APA style articles, comes with the apa package and includes apacite.
`beamer`	For making presentations. comes with the beamer package.

The Document Class Options

The options differ per documentclass and each document class has different defaults (for example, `article` and `report` use `oneside` by default, but `book` uses `twoside`). A few common options are:

Option	Specifies
`10pt`, `11pt`, `12pt`	Main font size
`a4paper`, `letterpaper`, `...`	Sets the paper size
`onecolumn`, `twocolumn`	Number of columns for main text
`oneside`, `twoside`	Is the document meant for two sided printing?
`landscape`	Flip the document
`draft`	Enables you to easily spot typesetting errors

Document Structure

Every LaTeX document has the following form:

```
\documentclass{ CLASS }

    PREAMBLE

\begin{document}

    BODY

\end{document}
```

The Preamble

- The preamble is used to define several things that influence the whole document
- Packages are loaded in the preamble
- Define variables, custom commands or other things needed in the document
- Depending on the documentclass several things must be defined in the preamble. For example:
  - The theme of a presentation in `beamer`
  - Authorname, affiliation, title, abstract and more in `apa`

Loading Packages

- LaTeX can be extended through many packages
- Packages are stored online at CTAN. Google “CTAN packagename” for documentation on a certain package
- To use a package, it must first be installed on your computer
  - If you did a full install, this is probably already the case (do not forget to update your distribution each year or so)
  - Otherwise the package need to be installed. MiKTeX can do this automatically (although download might be slow) and for the other distributions you can look at the installation guide
- Secondly, the package must be loaded in the LaTeX document
- Loading a package can enable new commands, change commands and even fundamentally change the way a document looks

Loading Packages

Packages can be with the `\usepackage` command:

```
\usepackage[option1,option2,...]{ PACKAGE }
```

- Most packages do not have options
- Packages can conflict with other packages
- In general it is best to not load packages you don't need
- Some packages however are so commonly used that they should always be loaded. In this course it is assumed these are loaded

Common Packages

Package	Description	
amsmath	Advanced math extensions	*
amssymb	Mathematical symbols	*
graphicx	Needed to include pictures	*
babel	Sets the language of the document through an option	*
geometry	Manages the document margins and papersize through option	*
apa	Comes with the apa documentclass	
apacite	Citing using bibTeX in APA style	
beamer	Creates presentations	

* Recommended to load in every document

Outline

Last Week

The LaTeX Language

- Commands

Document Structure

- Documentclass
- Preamble

Body of a document

- Sectioning
- Environments
- Figures and Tables
- floats

Mathematics

Document Structure

Every LaTeX document has the following form:

```
\documentclass{ CLASS }

    PREAMBLE

\begin{document}

    BODY

\end{document}
```

The Body of a $\LaTeX$ Document

$\LaTeX$ code:

```
this is an example of the \emph{body} of a \LaTeX\ document.

Here we can write all sorts of funny things (e.g.\ Stories, fairy tales, general Microsoft Word bashing). But also equations:

\[
F(k) = \int_{-\infty}^{\infty} e^{-2 \pi i k x} \, \mathrm{d}x
\]
```

Output:

this is an example of the *body* of a $\LaTeX$ document.

Here we can write all sorts of funny things (e.g. Stories, fairy tales, general Microsoft Word bashing). But also equations:

$$F(k) = \int_{-\infty}^{\infty} e^{-2\pi ikx} \, \mathrm{d}x$$

The Body of a LaTeX Document

LaTeX code:

```latex
In a \LaTeX\ body there are 3 different modes in which you can write:

\begin{itemize}
  \item Text
  \item Math
  \item Verbatim
\end{itemize}

Where text mode is the default and the other two are obtained in environments. We will first focus on Text mode.
```

Output:

In a LaTeX body there are 3 different modes in which you can write:

- Text
- Math
- Verbatim

Where text mode is the default and the other two are obtained in environments. We will first focus on Text mode.

Letters and Spacing

LaTeX code:

```
All  letters  that we type are the same in the output.

White space is     interpreted    as      a  single              space between
words.  A newline  is
ignored  but

two newlines are  interpreted  as a change in paragraphs.

After a period the spacing  is slightly   larger.  This might not be what you want (e.g. right  here).  Force a normal space with a backslash (e.g.\ as here).
```

Output:

All letters that we type are the same in the output.

White space is interpreted as a single space between words. A newline is ignored but

two newlines are interpreted as a change in paragraphs.

After a period the spacing is slightly larger. This might not be what you want (e.g. right here). Force a normal space with a backslash (e.g. as here).

Line breaking

LATEX code:

```
A new line can be forced with \
↳textbackslash\textbackslash\
↳which breaks the \\ paragraph
↳and \textbackslash\
↳textbackslash∗ \\∗ to not break
↳ the paragraph.

 Alternatively  \newline \verb|\
↳newline| can be used.

A new page can be made with \
↳verb|\newpage| or \verb|\
↳clearpage|.
```

Output:

```
A new line can be forced with \\ which
breaks the
paragraph and \\*
to not break the paragraph.
Alternatively
\newline can be used.
A  new  page  can  be  made  with
\newpage or \clearpage.
```

Accents

LaTeX code:

```
Because the input is ASCII many special characters can not be written directly. Such a letter will not be compiled.

Accents can be added with escaped characters. For example, \verb|\'{o}| returns \'{o}.
```

Output:

Because the input is ASCII many special characters can not be written directly. Such a letter will not be compiled.

Accents can be added with escaped characters. For example, `\'{o}` returns ó.

Hyphenation

LaTeX code:

```latex
\LaTeX\ automatically hyphenates verylongwords

\verb|\mbox{}| can be used to force \mbox{verylongwords} to not be hyphenated.

\verb|\mbox{}| can be used to force verylongwords to not be hyphenated.

\verb|\-| can be used to make sure very\-longwords hyphenate at a certain point

\verb|\-| can be used to make sure verylong\-words hyphenate at a certain point.
```

Output:

```
LaTeX automatically hyphenates very-
longwords
\mbox{} can be used to force
verylongwords to not be hyphenated.
\mbox{} can be used to force verylong-
words to not be hyphenated.
\- can be used to make sure very-
longwords hyphenate at a certain point
\- can be used to make sure verylong-
words hyphenate at a certain point.
```

Emphasis

LaTeX code:

```
We can \emph{emphasize}
↳important parts of the text with
↳\verb|\emph{}|.

This works nested, in this way
↳\emph{we can \emph{
↳emphasize} while we are
↳emphasizing}.

In \LaTeX\ this is the preferred
↳ method for emphasizing over
↳manually setting text \textit{
↳ italic } or \textbf{bold}.
```

Output:

We can *emphasize* important parts of the text with `\emph{}`.
This works nested, in this way *we can* emphasize *while we are emphasizing*.
In LaTeX this is the preferred method for emphasizing over manually setting text *italic* or **bold**.

Quote marks

LaTeX code:

```latex
\LaTeX\ treats left and right quotes as different entities.

The left quote mark is the backtick (usually under tilde) (`)

The right quote is the normal tick mark (')

A double quote mark is obtained though two ticks at both sides. For ``example'' like so.
```

Output:

LaTeX treats left and right quotes as different entities.
The left quote mark is the backtick (usually under tilde) (‘)
The right quote is the normal tick mark (’)
A double quote mark is obtained though two ticks at both sides. For “example” like so.

Dashes and Footnotes

LATEX code:

```
An n-dash can be included with two dashes--like this--and an m-dash can be included with three dashes---like this---.

A footnote can be added with the \verb|\footnote| command\footnote{Although this does not work in presentations.}.
```

Output:

An n-dash can be included with two dashes–like this–and an m-dash can be included with three dashes—like this—
.

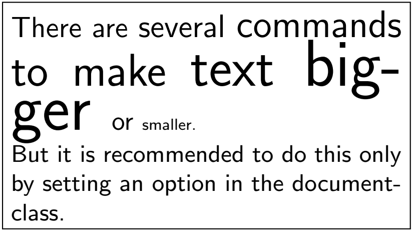
A footnote can be added with the `\footnote` command.

Font size

LaTeX code:

```
\normalsize{There are} \large{several} \Large{commands} \LARGE{to make} \huge{text} \Huge{bigger} \small{or} \tiny{smaller}.

\footnotesize{But it is recommended to do this only by setting an option in the documentclass.}
```

Output:



There are several commands to make text bigger or smaller.

But it is recommended to do this only by setting an option in the documentclass.

Special Characters

LaTeX code:

```
Several characters are used as special commands in \LaTeX\ and can not be entered normally. % For example, anything following a % is a comment and is not compiled.

Generally, you can get most characters by ``escaping'' them (i.e.\ adding a backslash before it): \$ \% \&.
```

Output:

Several characters are used as special commands in LaTeX and can not be entered normally.
Generally, you can get most characters by “escaping” them (i.e. adding a backslash before it): $ % &.

Sectioning

- There are several commands to properly section your document
- This is useful for adding titles, making a table of contents and cross referencing
- A section is defined by a single command
- A type of section may not be available in certain documentclasses
- Sectioning automatically numbers the sections

Sectioning

Section commands:

Command	Level	Comment
`\part{ name }`	-1	Not in letters
`\chapter{ name }`	0	Only in Books and Reports
`\section{ name }`	1	Not in letters
`\subsection{ name }`	2	Not in letters
`\subsubsection{ name }`	3	Not in letters
`\paragraph{ name }`	4	Not in letters
`\subparagraph{ name }`	5	Not in letters

Sectioning

A single experiment APA research paper:

Command	Examples
`\section{ name }`	Methods, Results, Discussion
`\paragraph{ name }`	Participants, Materials, Procedure

Note: Not the title

Sectioning

An APA paper with multiple experiments

Command	Examples
`\section{ name }`	Experiment 1, . . . , General discussion
`\subsection{ name }`	Methods, Results, Discussion
`\paragraph{ name }`	Participants, Materials, Procedure

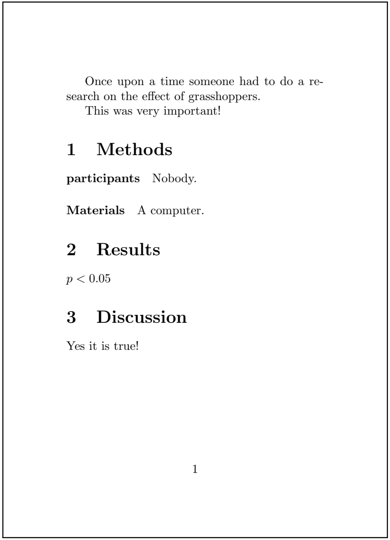
Note: Not the title

Sectioning Example

.tex file:

```latex
\documentclass{article}
\begin{document}
Once upon a time someone had to do a research on the effect of grasshoppers.
This was very important!

\section{Methods}

\paragraph{participants} Nobody.

\paragraph{Materials} A computer.

\section{Results}

$p < 0.05$

\section{Discussion}

Yes it is true!

\end{document}
```

.pdf file:



Once upon a time someone had to do a research on the effect of grasshoppers.

This was very important!

1 Methods

participants Nobody.

Materials A computer.

2 Results

$p < 0.05$

3 Discussion

Yes it is true!

1

Sectioning

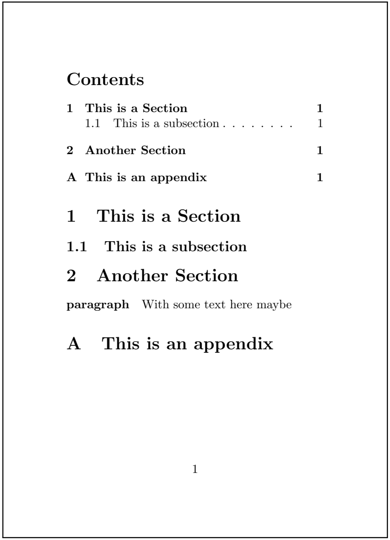
- To omit numbering you can add an asterisk to the command. E.g. `\section*{ name }`
- A table of contents can be added with the `\tableofcontents` command
  - This might require two runs of pdfLaTeX to work
- To start the appendices you can use the `\appendix` command. Sectioning will continue afterwards in letters

Sectioning Example

.tex file:

```latex
\documentclass{article}
\begin{document}
\tableofcontents
\section{This is a Section}
\subsection{This is a subsection}
\paragraph{paragraph} With some text here maybe

\section{Another Section}

\appendix

\section{This is an appendix}
\end{document}
```

.pdf file:



Contents

1 This is a Section

1.1 This is a subsection

2 Another Section

paragraph With some text here maybe

A This is an appendix

1

Environments

- The `\begin` and `\end` commands can be used to create an environment
- This is used for many different things
- An environment breaks the paragraph, but in some cases we can also use inline environments

Environments

LaTeX code:

```latex
We can \emph{itemize} with the \verb|itemize| environment:
\begin{itemize}
\item This creates a bulleted list
\item The symbols depend on your documentclass
\item Can also be nested
\begin{itemize}
\item By adding an environment in an environment
\item Note that these must be properly nested (FILO)
\end{itemize}
\item After such a nested itemization we can continue with the first one
\end{itemize}
```

Output:

We can *itemize* with the `itemize` environment:

- This creates a bulleted list
- The symbols depend on your documentclass
- Can also be nested
  - By adding an environment in an environment
  - Note that these must be properly nested (FILO)
- After such a nested itemization we can continue with the first one

Environments

LaTeX code:

```latex
We can also itemize with numbers with the \verb|enumerate| environment:
\begin{enumerate}
\item This creates an enumerated list
\item Can also be nested
\begin{enumerate}
\item Like this
\end{enumerate}
\item After nesting we can continue with the original numbering
\end{enumerate}
```

Output:

We can also itemize with numbers with the `enumerate` environment:

1. This creates an enumerated list
2. Can also be nested

   2.1 Like this

3. After nesting we can continue with the original numbering

Environments

LaTeX code:

```latex
With the \verb|description| environment we can itemize with descriptions. So to summarize
\begin{description}
\item[Itemize] Creates bulleted list
\item[enumerate] Creates enumerated list
\item[description] Creates a list with titles such as this one
\end{description}
```

Output:

With the `description` environment we can itemize with descriptions. So to summarize

Itemize	Creates bulleted list
enumerate	Creates enumerated list
description	Creates a list with titles such as this one

Verbatim

LaTeX code:

```latex
To be able to include code as it
↳ is the \verb|verbatim|
↳environment can be used:
\begin{verbatim}
$ Whatever % we write here
 it does not break @
\end{verbatim}

Can also be used inline with the
↳ \verb|\verb| command, but
↳this requires $\mid$ symbols
↳instead of curly brackets.
```

Output:

To be able to include code as it is the `verbatim` environment can be used:

```
$ Whatever % we write here
it does not break @
```

Can also be used inline with the `\verb` command, but this requires | symbols instead of curly brackets.

Verbatim

LaTeX code:

```latex
\begin{center}
The \verb|center| environment can be used to center text and even figures and tables.
\end{center}
```

Output:

The `center` environment can be used to center text and even figures and tables.

Including figures

- With the `graphicx` package we can use the command `\includegraphics` to import figures
- This can be a number of different types of files
  - `jpg`
  - `png`
  - `pdf`
  - `eps`
- Options can be used to specify the size of the image

```
\includegraphics[option1=value,option2=value,...]{filename}
```

Including figures

Options:

`width=xx`	Set preferred width to xx (default in inches)
`height=xx`	Set preferred height to xx
`keepaspectratio=true`	Keeps aspect ratio, can also be false
`scale=xx`	Scaling factor
`angle=xx`	Rotate by xx degrees
`page=xx`	Select what page to include from multipage pdf

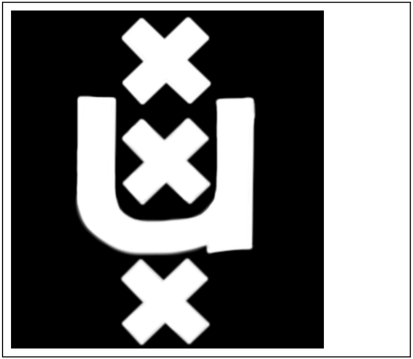
Often you want to make width and height proportional to text width and height. This can be done with e.g. `0.8\textwidth`

Including Figures

LaTeX code:

```
\includegraphics{uva.png}
```

Output:



Including Figures

LaTeX code:

Output:

```
\begin{center}
\ includegraphics {uva.png}
\end{center}
```

Including Figures

LaTeX code:

```
\begin{center}
\ includegraphics [width=0.3\textwidth]{uva.png}
\end{center}
```

Output:

Tables

- Tables can be created with the `tabular` environment
- This requires one argument that specifies each column, indicated with a `l` for left aligned, `c` for right aligned, `r` for right aligned or `p{width}` for a column of fixed width
- Vertical lines are created in this argument with a | symbol
- Horizontal lines are created with the `\hline` command
- Cells are separated with the `&` command
- rows are separated with the `\\` command

In the fourth lecture we will see a way to automate this from `R`!

Tables

LaTeX code:

```latex
\begin{tabular}{ ll }
First  Column & Second Column \\
\hline
1 & 2 \\
3 & 4  \\
5 & 6
\end{tabular}
```

Output:

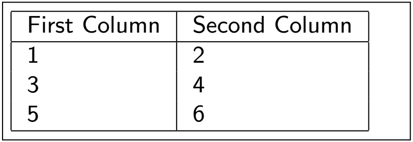
First Column	Second Column
1	2
3	4
5	6

Tables

LaTeX code:

```
\begin{tabular}{|l|l|}
\hline
First  Column & Second Column \\
\hline
1 & 2 \\
3 & 4  \\
5 & 6 \\
\hline
\end{tabular}
```

Output:



First Column	Second Column
1	2
3	4
5	6

Tables

LaTeX code:

```latex
\begin{tabular}{|p{0.2\textwidth}|p{0.6\textwidth}|}
\hline
First  Column & Second Column \\
\hline
1 & 2 \\
3 & 4  \\
5 & 6 \\
\hline
\end{tabular}
```

Output:

First Column	Second Column
1	2
3	4
5	6

Floats

- Including a figure and tables at exactly a certain place in the text can make it look unprofessional
- Furthermore we want to add captions to figures and tables
- A float environment is a special kind of environment that can hold a figure or table and sort of floats through your document to find a good place to end up
- This is usually near the place you specified the float, but at the top or bottom of a page or in another way your documentclass specifies it
- The float environment for figures is `figure` and for tables is `table`

Floats

In general:

```
\begin{ FLOAT }[placement options]
\centering
% Code for figure or table
\caption{ }
\label{ }
\end{ FLOAT }
```

Floats

For figures:

```
\begin{figure}[placement options]
\centering
\includegraphics{ }
\caption{ }
\label{ }
\end{figure}
```

Floats

For tables:

```
\begin{table}[placement options]
\centering
% Code for tabular
\caption{ }
\label{ }
\end{table}
```

Floats

Placement options:

Specifier	Permission
`h`	Try to place float approximately at this point in text
`t`	Float can be placed at top of pages
`b`	Float can be placed at bottom of page
`H`	From `float` package, places float exactly here

Only do this when your text is complete

Floats

- The `\caption` command is used to add a caption to the float
- the `\label` command is used in referencing
    - `\label{key}` stores the float under some key, this can then be referenced in the text with `\ref{key}`
    - To get this right, run pdfLaTeX twice
    - Note that `\label` should come after `\caption`

Floats Example

.tex file:

```latex
\documentclass{article}
\usepackage{graphicx}
\begin{document}
Figure \ref{fig:uva} shows the UvA logo.
\begin{figure}
\centering
\includegraphics[width=0.5\textwidth]{uva.png}
\caption{The UvA logo}
\label{fig:uva}
\end{figure}
\end{document}
```

.pdf file:

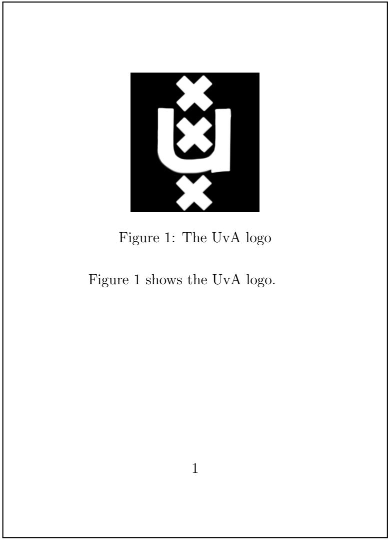


Figure 1: The UvA logo

Figure 1 shows the UvA logo.

1

Outline

- Last Week
- The LaTeX Language
  - Commands
- Document Structure
  - Documentclass
  - Preamble
- Body of a document
  - Sectioning
  - Environments
  - Figures and Tables
  - floats
- Mathematics

- LaTeX is often used for its strong capabilities of writing mathematical text
- This is done in "math mode", which can be enabled in several ways
- In the next few slides is a brief overview of the basics. For more information on this topic see one of many resources online

Math mode

LaTeX code:

```
We can use the \$ or  sign  to enable  and  disable  math mode in text.  Within math mode spaces are ignored and  letters re  written  as  variables :  $1 apple + 2 apples = 3 apples$

To obtain  roman  letters  in  math  mode we can use the \verb|\text| and \verb|\mathrm| commands: $1\text{ apple} + 2\text{ apples} = 3\text{ apples}$.

 Alternatively  \textbackslash( and \textbackslash) can be used
```

Output:

We can use the $ or sign to enable and disable math mode in text. Within math mode spaces are ignored and letters re written as variables: $1 apple + 2 apples = 3 apples$
To obtain roman letters in math mode we can use the `\text` and `\mathrm` commands: $1\text{ apple} + 2\text{ apples} = 3\text{ apples}$.
Alternatively \(and \) can be used

Math mode

LaTeX code:

```latex
To write equations, the commands \verb|\[| and \verb|\begin{equation*}| can be used. For an enumerated equation you can use \verb|\begin{equation}|:
\[
1+1=2
\]
\begin{equation*}
1+1=2
\end{equation*}
\begin{equation}
1+1=2
\end{equation}
```

Output:

To write equations, the commands `\[` and `\begin{equation*}` can be used. For an enumerated equation you can use `\begin{equation}`:

$$1+1=2$$

$$1+1=2$$

$$1+1=2 \tag{1}$$

Math mode

LaTeX code:

```
To align several equations you
↳can use the \verb|align*| or \
↳verb|align| environments.
↳These use the \& sign to align
↳ and the \textbackslash\
↳textbackslash sign to go to a
↳new line:
\begin{align*}
x^2 + 100 &= 200 \\
x^2 &= 100 \\
x &= \sqrt{100} \\
&= 10
\end{align*}
```

Output:

To align several equations you can use the `align*` or `align` environments. These use the & sign to align and the \\sign to go to a new line:

$$\begin{aligned} x^2 + 100 &= 200 \\ x^2 &= 100 \\ x &= \sqrt{100} \\ &= 10 \end{aligned}$$

Math mode

LaTeX code:

```
Many mathematical operators work (and are properly spaced) as expected:
\[
f(x) = 5x + 2
\]
For superscripts you can use \verb|^| and for subscripts \verb|_|:
\[
f(x_1,x_2) = x_1^2 + 5x_2 - 1
\]
To group characters together you can use \{ and \}:
\[
f(x) = e^{2x+1}
\]
```

Output:

Many mathematical operators work (and are properly spaced) as expected:

$$f(x) = 5x + 2$$

For superscripts you can use ^ and for subscripts _:

$$f(x_1, x_2) = x_1^2 + 5x_2 - 1$$

To group characters together you can use { and }:

$$f(x) = e^{2x+1}$$

Math mode

LaTeX code:

```
All mathematical operators can be used. These require you to know the commands. Some editors have these commands in the menu.
\[
\int_{-\infty}^{\infty} \prod_{j=1}^m \left( \sum_{i=1}^n \frac{x\sqrt{_i}}{\ln y_j} \right) \, \mathrm{d}x
\]
```

Output:

All mathematical operators can be used. These require you to know the commands. Some editors have these commands in the menu.

$$\int_{-\infty}^{\infty} \prod_{j=1}^{m} \left(\sum_{i=1}^{n} \frac{x\sqrt{_i}}{\ln y_j} \right) \, \mathrm{d}x$$

Math mode

LaTeX code:

```latex
Many Greek letters can be obtained with \verb|\letter| for lower case letters or \verb|\Letter| for upper case letters:
\[
\gamma, \Gamma, \theta, \Theta, \lambda, \Lambda
\]
Note that some Greek letters, such as omicron and capital beta, do not differ from roman letters and are not included.
```

Output:

Many Greek letters can be obtained with `\letter` for lower case letters or `\Letter` for upper case letters:

$$\gamma, \Gamma, \theta, \Theta, \lambda, \Lambda$$

Note that some Greek letters, such as omicron and capital beta, do not differ from roman letters and are not included.

Math mode

LaTeX code:

```
To obtain bold symbols the \verb|\boldsymbol| command can be used:
\[
\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}
\]
```

Output:

To obtain bold symbols the `\boldsymbol` command can be used:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

Math mode

LaTeX code:

```
Matrices can be made by, among others, the \verb|bmatrix| environment, which works somewhat similar to the \verb|tabular| environment:
\[
\boldsymbol{A} =
\begin{bmatrix}
1 & 2 \\
3 & 4 \\
\end{bmatrix}
\]
```

Output:

Matrices can be made by, among others, the `bmatrix` environment, which works somewhat similar to the `tabular` environment:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Very useful equation editor:

`http://www.codecogs.com/latex/eqneditor.php`

Next week: APA style, bibTEX and beamer.

Contact Details

Course website:

- http://sachaepskamp.com/latex-course

E-mail:

- sacha.epskamp@gmail.com