A quick start to LaTeX

Claudio Vellage

26. November 2013

Inhaltsverzeichnis

1 How to use LaTeX

This is a quick introduction to the most common features of LaTeX. For more features, check the other lessons on `http://www.latex-tutorial.com` and the articles on `http://blog.latex-tutorial.com`.

1.1 Document structure

The documents in LaTeX are structured slightly different from Word. The document consists of two parts, the preamble and the main document. If you're familiar with programming, then you might want to compare the structure with that of a C/C++ file.

1.1.1 Preamble

The purpose of the preamble is to tell LaTeX what kind of document you will set up and what packages you are going to need. A package is a set of additional functions such as *amsmath* for additional math formatting. For this document, the preamble looks like this:

Listing 1: Preamble of this document

```
% Preamble
% ----
\documentclass{article}

% Packages
% ----
\usepackage{amsmath} % Advanced math typesetting
\usepackage[utf8]{inputenc} % Unicode support (Umlauts
    etc.)
\usepackage[ngerman]{babel} % Change hyphenation rules
\usepackage{hyperref} % Add a link to your document
\usepackage{graphicx} % Add pictures to your document
\usepackage{listings} % Source code formatting and
    highlighting
```

You can set the class of the documentclass with the *documentclass* command and add packages with the *usepackage* command. Only the *documentclass* command is mandatory, you can compile a document also without packages, yet some functions may be missing in this case. The *usepackage* command *must not* be used in the main document.

1.1.2 Main document

The main document is contained within the *document* environment like this:

Listing 2: Main part of a LaTeX document.

```
\begin{document}
% ...
% ... Text goes here
% ...
\end{document}
```

Within those two statements, we can add the content of our document. But just adding the text is probably not enough, since we also have to apply formatting to it.

1.2 Formatting in LaTeX

Formatting in LaTeX can be applied by the use of commands and environment. The topmost environment is the document environment as described in Listing 4. So there are obviously more environments, but how to find them? Well the easiest way is to download a LaTeX cheat sheet which provides a list of the most useful commands and environments. For most packages there is also a manual available, which can be found on Google.

1.2.1 Math typesetting

To introduce you to math typesetting and environments, I will demonstrate you how to format some simple equations:

$$f(x) = x^2 \tag{1}$$

$$f'(x) = 2x \tag{2}$$

$$F(x) = \int f(x)dx \tag{3}$$

$$F(x) = \frac{1}{3}x^3 \tag{4}$$

This can be done with the following code:

Listing 3: Typesetting equations in LaTeX.

```
\begin{align}
f(x) &= x^2\\
f'(x) &= 2x\\
F(x) &= \int f(x)dx\\
F(x) &= \frac{1}{3}x^3
\end{align}
```

As you can see, we again have a *begin* and *end* statement, this it's applying the *align* environment. This will align the equations at the ampersand (&) sign. Naturally, those will be placed in front of the equality sign. If you watched carefully, you will see that LaTeX magically added sequential numbers to all equations. LaTeX does this for many other elements too. More about that in the next chapter.

1.2.2 Document layout

Usually a document does not only consist of a bunch of equations, but needs some kind of structure too. We are usually going to need at least:

- Title/Titlepage

- Table of contents

- Headlines/Sections

- Bibliography

LaTeX provides all the commands we need. The following commands will help us:

Abbildung 1: This figure shows the logo of my website.

Listing 4: Useful commands to structure a document.

```
\section{Text goes here} % On top of document hierarchy;
    automatically numbered
\subsection{}
\subsubsection{}
\paragraph{} % Paragraphs have no numbering
\subparagraph{}

\author{Claudio Vellage} % The authors name
\title{A quick start to \LaTeX{}} % The title of the
    document
\date{\today{}} % Sets date you can remove \today{} and
    type a date manually
\maketitle{} % Generates title
\tableofcontents{} % Generates table of contents from
    sections and subsections

\\ % Linebreak
\newpage{} % Pagebreak
```

There are commands to create sections, the sections are numbered automatically and the *tableofcontents* command will use them to generate the table of contents. You don't have to do it yourself, ever. LaTeX also provides commands to generate the title using the *maketitle* command. This needs the *author*, *title* and *date* command to be set. If you place the *maketitle* or *tableofcontents* command in your document, the Commands will be added at that exact place, so you probably want them in the very beginning of your document. If you want the title to appear on a single page, simply use the *newline* command.

1.2.3 Adding a picture

Most documents will also need some kind of picture. Like this:
Adding them is fairly easy:

Listing 5: Adding pictures in LaTeX.

```
\begin{figure}
\includegraphics[width=\textwidth]{picture.png}
\caption{This figure shows the logo of my website.}
\end{figure}
```

You have to embed the picture within the *figure* environment, then use the *includegraphics* command to select the image. Note that the picture file has to be in the same directory as your .tex file or you specify the path like this:

Listing 6: How to specify a path.

```
% ...
\includegraphics[width=\textwidth]{FOLDERNAME/picture.png
    }
% ...
```

It makes sense to manage all your pictures in subfolders if you have many of them.

1.3 Sourcecode formatting

Throughout the book I used the *listings* package to format the LaTeX code snippets. You can specify the language for each *lstlistings* block, in this case i used LaTeX of course and added a caption as well as enabled linebreaking:

Listing 7: How to use the listings package."

```
\begin{lstlisting}[language={[LaTeX]TeX},caption=,
    breaklines=true,frame=single]
% Source code goes here
\end{lstlisting}
```

The language is specified as *[DIALECT]LANGUAGE*, here LaTeX is dialect of TeX. A list of all programming languages can be obtained from the *listings* package manual, but you can also just try out if it works.

1.3.1 References and Bibliography

You can specify labels for all the things that are automatically numbered. If you want to refer to a section of your document, you'd simply use the *label* and *ref* (reference) command. Where the label indicates what you want to refer to and the reference will print the actual number of the element in your document. This will also work interactively in your PDF reader You can try this feature in section 1.3.1.

Listing 8: Labels and references in LaTeX

```
\section{}\label{sec:YOURLABEL}
% ...
I've written text in section \ref{sec:YOURLABEL}.
```

Papers usually include a lot of references to the great works of other people. In order to properly cite them, we'd want to use the biblatex package. For this purpose we'd simply add the following code to our preamble:

Listing 9: Preamble code to use biblatex.

```
\usepackage[backend=bibtex,style=verbose-trad2]{biblatex}
    % Use biblatex package
\bibliography{FILENAME} % The name of the .bib file (name
    without .bib)
```

All bibliographic information will be stored in the bibliography (.bib) and *must not* be inside of the .tex file An example could look like this:

Listing 10: Example bibliography file.

```
@ARTICLE=
{
VELLAGE:1,
AUTHOR="Claudio Vellage",
TITLE="A quick start to \LaTeX{}",
YEAR="2013",
PUBLISHER="",
}
```

Now i could add a self reference using the *cite* command:

Listing 11: The cite command.

```
This feature works as I described in \cite{VELLAGE:1}.
```

The *biblatex* is very smart and wil print autogenerate the bibliography if we want to. We'd usually do this in the end of the document. Simply add

Listing 12: The cite command.

```
\printbibliography
```

to our document. More examples can be found on the website.