Introduction to LaTeX

Philippe Cara
pcara@vub.ac.be

June 17, 2019

1 Introduction

The *programming language* TeX was inventend by Donald E. Knuth in 1979[2]. In 1985 Leslie Lamport extended TeX with several useful macros. This extended version of TeX became known as LaTeX[1]. Nowadays most people use LaTeX2$_\varepsilon$ which is a LaTeX version 'in between' the second and the third (which might never appear). This LaTeX runs on top of TeX version 3.1415926 which is believed to be bugfree and appeared in 2008.

In several technical fields, in particular, computer science, mathematics, engineering, and physics, LaTeX has become a de facto standard. Many thousands of books have been published using LaTeX, including books published by Addison-Wesley, Cambridge University Press, Elsevier, Oxford University Press and Springer. Numerous journals in these fields are produced using TeX or LaTeX, allowing authors to submit their raw manuscript written in LaTeX.

LaTeX is similar to `HTML` (the language used for writing WEBpages). The philosophy is that the author should focus on the *content* of his document and **not** on the *lay-out*. The latter is taken care of by 'style definitions' (CSS files for HTML). LaTeX programs are *compiled* and the result of a compilation is a PDF (or other printable/viewable format) document. See Figure 1.

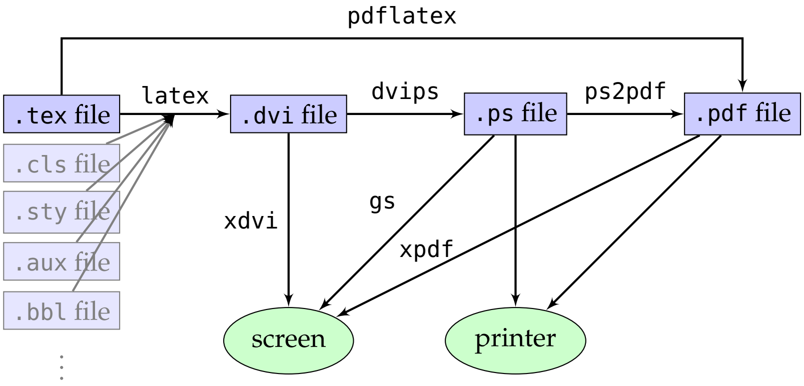

Figure 1: LaTeX workflow

2 Software

There are many implementations of TeX and LaTeX. Most of them are free and the system is available on virtually any computer system know to the human species. General information can be found on `http://www.latex-project.org/`.

In this workshop we use LaTeX software provided by the TeXLive package (`http://www.tug.org/texlive/`), together with the TeXmaker *programming envionment* (`http://www.xm1math.net/texmaker/`). In case you don't want to install LaTeX on your computer (immediately), you can use a cloud solution like `http://overleaf.com`.

Other popular implementations are MikTeX, TeXShop,. . . and as editor you could also use TeXnicCenter, `vim`, `emacs` (with `aucTeX` extension), Notepad, . . .

3 First example

Let's try to compile our first program!

Open TeXmaker and create a new document. Type the four lines of code and save with a sensible name.

```
\documentclass{article}
\begin{document}
  Hello World!
\end{document}
```

Clicking on the arrow in front of "Quick Build" will automatically compile your first LaTeX program and display the result.

This was a minimal document in LaTeX. As you can see, every LaTeX document starts with a `\documentclass` command, taking an argument between curly brackets. This takes care of the layout of your document. Everything you want to appear in your text will be typed between `\begin{document}` and `\end{document}`. This region of your program is an example of an *environment*. There are many more environments in LaTeX which are all started with `\begin{<environment-name>}` and ended with `\end{<environment-name>}`. All LaTeX commands start with a *backslash* sign '`\`'.

You can now add some content and structure to your document.

```
\documentclass{article}
\title{Introduction to \LaTeX}
\author{Your Name}

\begin{document}
\maketitle

\begin{abstract}
  Notes from the first \LaTeX session
\end{abstract}

\section{Introduction}
Hello world!
\subsection{Subsection Heading Here}
Write your subsection text here.
\end{document}
```

You see that sections are numbered automatically and typeset in a uniform way. You can also generate a table of contents with `\tableofcontents`. The number of spaces or blank lines is not important. Feel free to add whitespace to make your LaTeX code more readable! One blank line (or more) initiates a new paragraph.

3.1 Numberless sections

Try `\section*{Numberless section}`.

3.2 Optional arguments

Many LaTeX commands can take *optional arguments* that change their behaviour. These are put between square brackets. Some examples:

- `\documentclass{book}` which provides a `\chapter{}` command
- `\documentclass[12pt]{article}`
- `\documentclass[twocolumn]{article}`
- `\documentclass[12pt,a4paper]{book}`

3.3 Reserved characters

The characters `$`, `&`, `%`, `#`, `_`, `{`, `}`, `~`, `^` and `\` cannot be used in regular text because they have a special meaning in LaTeX. Most of the time you can make them appear by preceding them with a backslash, like in `\$`, `\&`,... To obtain a backslash, do `\textbackslash`. Try to find out what `\\` does.

4 Maths

Try the following code

```
$\sum_{i=1}^n \alpha_i\vec{v}_i = \vec{0}$
```

Try to typeset your favourite formula!

5 Extending LaTeX by packages

Although standard LaTeX can already produce very nice documents you might want or need more. You can add macros to LaTeX yourself (see section 12) but if you are lucky someone has already solved your problem and made a *package* available.

You can load packages in the *preamble* of your document. That is the part between `\documentclass` and `\begin{document}`.

```
\usepackage[dutch]{babel}
\usepackage{a4wide}
```

There are thousands of packages available and you can find them on `http://www.ctan.org/pkg/`.

6 Some lay-out

```
\pagestyle{empty} \pagestyle{
   plain} \pagestyle{headings}
\thispagestyle{empty}
\pagenumbering{roman}
\textbf{MOFshape project}
\textsf{MOFshape project}
\textit{MOFshape project}
\texttt{MOFshape project}
\textsl{MOFshape project}
\emph{Emphasized text}
```

MOFshape project
MOFshape project
MOFshape project
`MOFshape project`
MOFshape project
Emphasized text

7 special symbols

```
\usepackage{marvosym}
Do you owe me 50\EUR?
Wash \AtSixty
```

A very long list of available symbols can be found on `http://www.ctan.org/tex-archive/info/symbols/comprehensive/`.

8 Numbered and unnumbered lists

```
\begin{itemize}
\item first
\item second
\begin{enumerate}
\item [first.]subitem
\item subitem 2
\end{enumerate}
\end{itemize}

\begin{itemize}
\item first
\item second
\begin{description}
\item [first.]subitem
\item subitem 2
\end{description}
\end{itemize}
```

9 Tabular material

```
\begin{tabular}{|l|l||c|}
\hline
Name & Age & Gender\\ \hline\hline
Stijn & 24 & M\\
\hline
\end{tabular}
```

Name	Age	Gender
Stijn	24	M

Try to get the following.

Name	Age	Gender
Stijn	24	M

Table 1: properties Stijn

10 Inserting pictures

```
\usepackage{graphicx}

\begin{figure}[hbtp] %here bottom top page
\begin{center}
\includegraphics[width=0.7\textwidth,angle=90]{latexfoto}
\end{center}\caption{first figure}
\end{figure}
```

Figure 2: first figure

11 Crossreferencing

```
\begin{figure}[hbtp] %here bottom top page
\begin{center}
\includegraphics[width=0.7\textwidth,angle=90]{latexfoto}
\end{center}\caption{first figure}\label{firstfig}
\end{figure}

As we can see in figure~\ref{firstfig}
As we can see in section~\ref{sec:1}\\
In section \ref{sec:1} on page\pageref{sec:1}
```

As we can see in figure 2 As we can see in section 5
In section 5 on page 3

12 Extending LaTeX

```
\newcommand{\temperature}[1]{$#1\ ^{\circ}$C}

\temperature{12}
```

Today the temperature is 12 °C.

13 Counters, lengths and redefining commands

One interesting feature of TeX is that the programming language itself can be altered by a program. We won't go very deep but just show how one can redefine existing commands.

As an example we will take the numbering system used in the `enumerate` environment. The first level of enumeration is displayed in the arabic numbering system, followed by a full stop. Suppose we want to change this to roman numbers.

13.1 Counters

In TeX everything that generates a number uses a *counter*. Counter names are one of the few things in TeX that don't start with a backslash. The counter for page numbering is called `page`. You can try

```
\setcounter{page}{99}
```

and see what happens.

The command used to display a counter is always `\the<countername>`. So you can type `\thepage` anywhere in a text to display the current pagenumber.

The counter used for the first level of enumeration is `enumi`. To get roman numbering in enumerations we will redefine the command `\theenumi`.

```
\renewcommand{\theenumi}{\roman{enumi}}
```

13.2 Lengths

Besides counters LaTeX uses many *lengths* which influence the lay-out. One length that we have encountered is `\textwidth` which is the width of a line of text. You can change a length as follows.

```
\setlength{\textwidth}{20cm}
```

LaTeX is very precise with lengths. It understands the following units:

Abbreviation	Definition
`pt`	a point is 1/72.27 inch, that means about 0.0138 inch or 0.3515 mm.
`mm`	millimeter
`cm`	centimeter
`in`	inch
`ex`	roughly the height of an 'x' in the current font
`em`	roughly the width of an 'M' (uppercase) in the current font

There are more measure units but the above are the more important ones.

In this document I used the following settings to get rid of paragraph indentation and separate paragraphs by some blank space.

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1.5ex}
```

14 Further topics

- How to make the document into 2 columns but the abstract is not in column format?
- VUB style. See `http://huisstijl.vub.ac.be/` for some LaTeX templates.
- Changing fonts. Try `\usepackage{times}`
- Bibliography management (other workshop)
- When inserting pictures is it possible to put several side by side and make sure they are scaled in such a way that their height remains the same?
- Making slides for presentation. Use the LaTeX `Beamer` class.
- Making posters. Can also be done with beamer.
- How to add the references, footnotes and how to make citation?
- PDF forms
- More mathematics formulae.
- Modern fonts and encoding schemes: use XeLaTeX.
- . . .

References

[1] Leslie Lamport 'LaTeX: A Document Preparation System', Addison-Wesley (1986). ISBN 0-201-15790-X.

[2] Donald E. Knuth 'The TeXbook' Addison-Wesley (1984).