

# A bits-pieces introduction to $\text{\LaTeX}$

Siva Athreya

Indian Statistical Institute, Bangalore, India

October 19, 2016

- T<sub>E</sub>X is computer program. Donald E. Knuth invented this in 1982. T<sub>E</sub>X as we use it today was released in 1982, with some slight enhancements added in 1989 to better support 8-bit characters and multiple languages.
- T<sub>E</sub>X is pronounced “Tech,” with a “ch” as in the German word “Ach”
- T<sub>E</sub>X is renowned for being extremely stable, for running on many different kinds of computers, and for being virtually bug free.

- L<sup>A</sup>T<sub>E</sub>X is “only” a program and therefore needs more guidance. The author has to provide additional information to describe the logical structure of his work. This information is written into the text as “L<sup>A</sup>T<sub>E</sub>X commands.”
- This is quite different from the WYSIWYG (What you see is what you get.) approach that most modern word processors, such as *MS Word* or *Corel WordPerfect*, take.
- When using L<sup>A</sup>T<sub>E</sub>X it is not normally possible to see the final output while typing the text, but the final output can be previewed on the screen after processing the file with L<sup>A</sup>T<sub>E</sub>X.

# Advantages/Disadvantages

- ▶ Professionally crafted layouts are available, which make a document really look as if “printed.”
- ▶ The typesetting of mathematical formulae is supported in a convenient way.
- ▶ Users only need to learn a few easy-to-understand commands that specify the logical structure of a document. They almost never need to tinker with the actual layout of the document.
- ▶ Even complex structures such as footnotes, references, table of contents, and bibliographies can be generated easily.

# Advantages/Disadvantages

- ▶ Free add-on packages exist for many typographical tasks not directly supported by basic  $\text{\LaTeX}$ . For example, packages are available to include graphics or to typeset bibliographies conforming to exact standards. Many of these add-on packages we will discuss.
- ▶  $\text{\LaTeX}$  encourages authors to write well-structured texts, because this is how  $\text{\LaTeX}$  works—by specifying structure.
- ▶  $\text{\TeX}$ , the formatting engine of  $\text{\LaTeX} 2_{\epsilon}$ , is highly portable and **free**. Therefore the system runs on almost any hardware platform available.

- The input for L<sup>A</sup>T<sub>E</sub>X is a plain ASCII text file. You can create it with any text editor. It contains the text of the document, as well as the commands that tell L<sup>A</sup>T<sub>E</sub>X how to typeset the text.

- “Whitespace” characters, such as blank or tab, are treated uniformly as “space” by L<sup>A</sup>T<sub>E</sub>X. *Several consecutive* “whitespace” characters are treated as *one* “space.”
- An empty line between two lines of text defines the end of a paragraph. *Several* empty lines are treated the same as *one* empty line.

# L<sup>A</sup>T<sub>E</sub>X Special Characters

The following symbols are “reserved characters” that either have a special meaning under L<sup>A</sup>T<sub>E</sub>X or are not available in all the fonts. If you enter them directly in your text, they will normally not print, but rather coerce L<sup>A</sup>T<sub>E</sub>X to do things you did not intend.

# % ^ & \ \_ { } ~ \$

As you will see, these characters can be used in your documents all the same by adding a prefix backslash:

The backslash character `\` can *not* be entered by adding another backslash in front of it `\\` this sequence is used for line breaking. Try the `$ \` backslash `$` command instead. It produces a `'\'`.



L<sup>A</sup>T<sub>E</sub>X “commands” are case sensitive, and take one of the following two formats:

- ▶ They start with a “backslash” `\` and then have a name consisting of letters only. Command names are terminated by a space, a number or any other ‘non-letter.’
- ▶ They consist of a backslash and exactly one non-letter.

When L<sup>A</sup>T<sub>E</sub>X encounters a % character while processing an input file, it ignores the rest of the present line, the line break, and all whitespace at the beginning of the next line.

This can be used to write notes into the input file, which will not show up in the printed version.

# L<sup>A</sup>T<sub>E</sub>X Simplest File Structure

```
\documentclass{...}
```

This specifies what sort of document you intend to write. Then start the body of the text with the command

```
\begin{document}
```

Now you enter the text mixed with some useful L<sup>A</sup>T<sub>E</sub>X commands. At the end of the document you add the

```
\end{document}
```

command, which tells L<sup>A</sup>T<sub>E</sub>X to call it a day. Anything that follows this command will be ignored by L<sup>A</sup>T<sub>E</sub>X.

```
\documentclass{article}  
\begin{document}  
Small is beautiful.  
\end{document}
```

You can include commands that influence the style of the whole document, or you can load “package”s that add new features to the L<sup>A</sup>T<sub>E</sub>X system. To load such a package you use the command

```
\usepackage{...}
```

You can also title the article with author ship

```
\author{H. ~Partl}
```

```
\title{Minimalism}
```

These are are included in the space between

```
\documentclass{article}
```

```
.  
. .  
. .
```

```
\begin{document}
```

```
\documentclass[a4paper,11pt]{article}
% define the title
\author{H.~Partl}
\title{Minimalism}
\begin{document}
% generates the title
\maketitle
% insert the table of contents
\tableofcontents
\section{Some Interesting Words}
Well, and here begins my lovely article.
\section{Good Bye World}
\ldots{} and here it ends.
\end{document}
```

## L<sup>A</sup>T<sub>E</sub>X A Typical Command Line Session

- Edit/Create your L<sup>A</sup>T<sub>E</sub>X input file. This file must be plain ASCII text. On Unix all the editors will create just that. On Windows you might want to make sure that you save the file in ASCII or *Plain Text* format. When picking a name for your file, make sure it bears the extension `.tex`.
- Run L<sup>A</sup>T<sub>E</sub>X on your input file. If successful you will end up with a `.dvi` file. It may be necessary to run L<sup>A</sup>T<sub>E</sub>X several times to get the table of contents and all internal references right. When your input file has a bug L<sup>A</sup>T<sub>E</sub>X will tell you about it and stop processing your input file. Type `ctrl-D` to get back to the command line.

```
latex firstfile.tex or pdflatex firstfile.tex
```