Using LaTeX for scientific writing

(part 1)

www.dcs.bbk.ac.uk/~roman/LaTeX

Roman Kontchakov

roman@dcs.bbk.ac.uk

TEX and LATEX

TEX is a computer program created by Donald E. Knuth.
It is aimed at typesetting text and mathematical formulas.

TEX is pronounced "Tech" with a "ch" as in the Scottish "Loch."

LATEX is a macro package which enables authors to typeset and print their work at the highest typographical quality, using a predefined, professional layout. LATEX was originally written by Leslie Lamport.
It uses the TEX formatter as its typesetting engine.

LATEX is pronounced "Lay-tech" or "Lah-tech."

No WYSIWYG

What You See Is What You Get

Advantages

- Several professional styles are available that make documents look "like printed". Changing style requires changing one single line in the document, with consistency ensured.
- High-quality maths typesetting
- Only a few commands to define the structure of text,
  no knowledge of typography or book design required.
- Complex scientific documents can be created automatically:
  - bibliography
  - index
  - cross-references
  - table of contents, lists of figures, tables etc.
  - ...
- Operating-system independent
- Long-term storage of documents: plain-text (ASCII) rather than binary
- Free software with source code available: errors are corrected rapidly

Disadvantages

- Learning curve
- Major changes in layout may require rewriting the style file

  (blessing in disguise)
- One gets an eye for all the bad documents out there

Plan

- Document structure: sections, cross-references, lists, figures, tables
- Mathematics: symbols, formulas, theorems
- Bibliographic references (BibTeX)
- MakeIndex
- Incorporating graphics
- Creating graphics and diagrams in LaTeX
- Thesis, report, article and customised styles
- Presentations
- Anything else? e-mail me: **roman@dcs.bbk.ac.uk**

Resources

Software CTAN `www.tug.org`,

win: MiKTeX 2.9 `www.miktex.org`

WinEdt 10.3 `www.winedt.com`

mac: MacTeX 2019 `www.tug.org/mactex`

win/mac: TeXstudio 2.12 `texstudio.org`

web: Overleaf / ShareLaTeX `https://www.overleaf.com/www.sharelatex.com`

Introduction `http://mirrors.ctan.org/info/lshort/english/lshort.pdf`

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

Literature The LaTeX-companion (MITTELBACH *et al*, 2004)

Symbols Comprehensive symbol list from CTAN

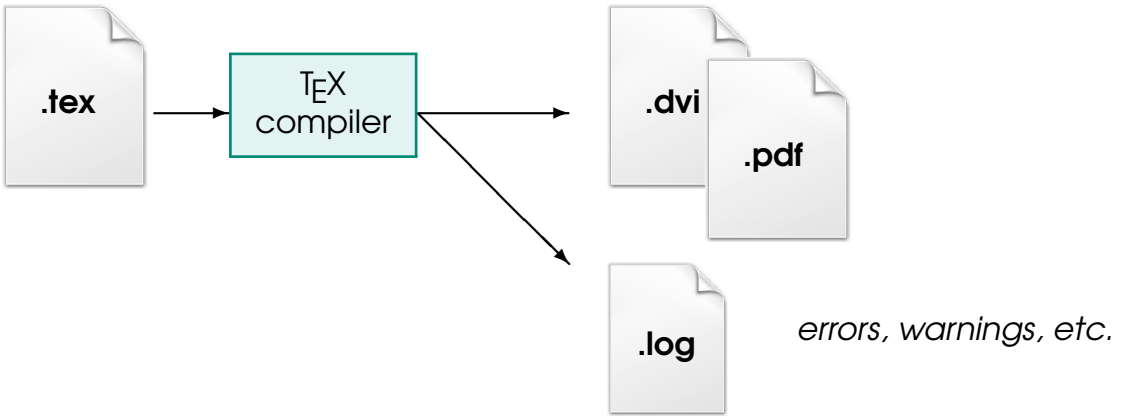
`http://tug.ctan.org/info/symbols/comprehensive/symbols-a4.pdf`

deTeXify: `http://detexify.kirelabs.org/classify.html`

Your first LaTeX document

```
\documentclass{article}
\begin{document}
My first \LaTeX{} document.
\end{document}
```

How does LaTeX work?



Spaces

"Whitespace" characters such as blank or tab are treated uniformly as "space". Several consecutive whitespace characters are treated as one space. Whitespace at the start of a line is generally ignored; a single linebreak is treated as whitespace.

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.

```
It does not matter whether you enter
one or several   spaces after a word.

An empty line starts a new paragraph.
```

It does not matter whether you enter one or several spaces after a word.

An empty line starts a new paragraph.

Special characters

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

These characters can be used in your documents by adding a prefix backslash:

```
\$ \& \% \# \_ \{ \}
```

$ & % # _ { }

LaTeX commands

LaTeX commands are case sensitive and take one of the following two formats:

- start with a backslash `\` and then have a name consisting of letters only (e.g., `\LaTeX`)
  Command names are terminated by a space, a number or any other 'non-letter'
- or consist of a backslash and exactly one non-letter character (e.g., `\$`)

LaTeX ignores whitespace after commands:

```
\LaTeX document
```

LaTeXdocument

Some commands need a **parameter**
which has to be given between curly brackets `{ }` after the command name

Some commands support **optional parameters**
which are added after the command name in square brackets `[ ]`

```
You can \textsl{lean} on me!
```

You can *lean* on me!

Comments

When LaTeX encounters a % character while processing an input file, it ignores the rest of the present line, the linebreak, and all whitespace at the beginning of the next line

```
This is an % stupid
% Better: instructive <----
example: Supercal%
        ifragilist%
   icexpialidocious
```

This is an example: Supercalifragilisticexpialidocious

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

The % character can also be used to split long input lines
where no whitespace or linebreaks are allowed

Document structure

```
\documentclass[options]{class}
\usepackage{...}      (preamble)
\begin{document}
...
\end{document}
```

Document classes

- `article` for articles in scientific journals, presentations, short reports, program documentation, invitations, etc.
- `report` for longer reports containing several chapters, small books, PhD theses, etc.
- `book` for real books
- `...`

Document class options

- `10pt, 11pt, 12pt` sets the size of the main font in the document (`10pt` by default)
- `a4paper, letterpaper, ...` defines the paper size
- `twocolumn` typesets the document in two columns
- `twoside, oneside` specifies whether double or single sided output should be generated (The article and report classes are single sided and the book class is double sided by default.)
- `...`

Page styles

`\pagestyle{`*style*`}`

`plain` prints the page numbers on the bottom of the page, in the middle of the footer (this is the default page style)

`headings` prints the current chapter heading and the page number in the header on each page, while the footer remains empty

`empty` sets both the header and the footer to be empty

It is possible to change the page style of the current page with the command

`\thispagestyle{`*style*`}`

NB. You can create your own headers and footers (more in week 3).

Big documents

The following command is useful, in particular, when working with other people and version control systems because it reduces possible change conflicts in different parts of the document

`\input{`*filename*`}`

simply includes the contents of the file specified (does not start a new page)

The following two commands are useful when working on documents with several chapters (each chapter in a separate file, for example):

`\include{`*filename*`}`

inserts the contents of another file named *filename.tex*

(LaTeX will start a new page before processing the material input from *filename.tex*)

`\includeonly{`*filename*`,`*filename*`,...}`

after this command is executed in the preamble of the document, only `\include` commands for the filenames which are listed in the argument of the `\includeonly` command will be executed

Paragraphs

The most important text unit in LaTeX (and in typography) is the **paragraph**. We call it "text unit" because a paragraph is the typographical form which should reflect one coherent thought, or one idea.

Often books are typeset with each line having the same length (**justified paragraphs**). LaTeX inserts the necessary linebreaks and spaces between words by optimising the contents of a whole paragraph. If necessary, it also **hyphenates** words that would not fit comfortably on a line. How the paragraphs are typeset depends on the document class. Normally the first line of a paragraph is **indented**, and there is no additional space between two paragraphs.

LaTeX always tries to produce the best linebreaks possible. If it cannot find a way to break the lines in a manner which meets its high standards, it lets one line stick out on the right of the paragraph. LaTeX then complains ("**overfull hbox**") while processing the input file. This happens most often when LaTeX cannot find a suitable place to hyphenate a word. You can instruct LaTeX to lower its standards a little by giving the `\sloppy` command. It prevents such over-long lines by increasing the inter-word spacing even if the final output is not optimal. In this case a warning ("**underfull hbox**") is given to the user. In most such cases the result doesn't look very good. The command `\fussy` brings LaTeX back to its default behaviour.

Linebreaking and pagebreaking

`\\` or `\newline`

starts a new line without starting a new paragraph

`\\*`

additionally prohibits a pagebreak after the forced linebreak

`\newpage`

starts a new page

`\linebreak[`n`]`, `\nolinebreak[`n`]`, `\pagebreak[`n`]` and `\nopagebreak[`n`]`

do what their names say. They enable the author to influence their actions with the optional argument n. It can be set to a number between 0 to 4. By setting n to a value below 4 you leave LaTeX the option of ignoring your command if the result would look very bad.

More special characters and symbols

quotation marks

```
``Please press the `x' key.''
```

“Please press the ‘x’ key.”

hyphen

```
daughter-in-law
```

daughter-in-law

en-dash

```
pages 13--67
```

pages 13–67

em-dash

```
yes---or no?
```

yes—or no?

minus sign

```
$1$ and $-1$
```

1 and -1

ellipsis

```
Not like this ...
```

Not like this ...

```
but like this: London, Rome, \ldots
```

but like this: London, Rome, . . .

Space between words

LaTeX inserts slightly more space at the end of a sentence, as this makes the text more readable. It assumes that sentences end with full stops, question marks or exclamation marks. If a full stop follows an uppercase letter, this is not taken as a sentence ending (full stops after uppercase letters normally occur in abbreviations).

`\␣` produces a space which will not be enlarged

`~` produces a space which will not be enlarged and additionally prohibits a linebreak

```
(e.g.~live football matches)
```

e.g. live football matches

`\@` in front of a full stop says that this full stop terminates a sentence (even if it follows an uppercase letter)

```
\ldots\ on the NHS\@. Most\ldots
```

... on the NHS. Most...

NB. The additional space after full stops can be disabled with the command

```
\frenchspacing
```

Sectioning

```
\section{Insect evolution and biology}
\subsection{Five factors in a winning formula}
\subsubsection{Flight}
\paragraph{The origin of wings}
\subparagraph{...}
```

1. Insect evolution and biology

1.1 Five factors in a winning formula

1.1.1. Flight

The origin of wings

The following can be used in the `report` and `book` classes:

```
\part{...}
\chapter{...}
```

Document title

The **title** of the whole document is generated by issuing

```
\maketitle
```

The contents of the title have to be defined by the commands

`\title{...}`, `\author{...}` and optionally `\date{...}`

before calling `\maketitle`

In the argument of `\author`, you can supply several names separated by `\and`

Cross references

`\label{`*marker*`}` creates a label with the name *marker*

`\ref{`*marker*`}` prints the number of the section, subsection, figure, table, etc. after which the corresponding `\label` command was issued

`\pageref{`*marker*`}` prints the page number of the page where the `\label` command occurred

↓ 'label'

```
\section{Introduction}\label{sec:intro}
In this paper, we study the problem of\dots

\section{Preliminaries}\label{sec:prelims}
We consider structures of the form\dots

\section{Complexity of the Problem}
\label{sec:cmplx}
The structures introduces in
Section~\ref{sec:prelims} on
p.~\pageref{sec:prelims} enjoy\dots
```

1. Introduction

In this paper, we study the problem of . . .

2. Preliminaries

We consider structures of the form. . .

3. Complexity of the Problem

The structures introduces in Section 2 on p. 4 enjoy. . .

Footnotes

```
Footnotes\footnote{This is a footnote.}
are often used by \LaTeX{}  people.
```

Footnotes[1] are often used by LaTeX people.

[1] This is a footnote.

```
Footnotes\footnote{\label{fn:latex}This is a footnote.}
are often used by \LaTeX{}  people
(one may also refer to footnote numbers: e.g.,~\ref{fn:latex}).
```

Footnotes[1] are often used by LaTeX people (one may also refer to footnote numbers: e.g., 1).

[1] This is a footnote.

NB: `\label` is placed inside the `\footnote` command
(but it can be placed either after `\section`,... or inside those commands)

Lists

```
An example:
\begin{enumerate}
\item You can mix the list
environments to your taste:
\begin{itemize}
\item But it might start to
look silly.
\item[-] With a dash.
\end{itemize}
\item Therefore remember:
\begin{description}
\item[Stupid] things will not
become smart because they are
in a list.
\item[Smart] things, though, can be
presented beautifully in a list.
\end{description}
\end{enumerate}
```

An example:

1. You can mix the list environments to your taste:
   - But it might start to look silly.
   - With a dash.
2. Therefore remember:

   Stupid things will not become smart because they are in a list.

   Smart things, though, can be presented beautifully in a list.