

# Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>What are all these *TeX things?</b>           | <b>1</b> |
| <b>2</b> | <b>Structure of a Latex Document</b>             | <b>1</b> |
| 2.1      | Document Class . . . . .                         | 2        |
| 2.2      | Preamble . . . . .                               | 2        |
| 2.2.1    | Pacakges . . . . .                               | 2        |
| 2.3      | Document . . . . .                               | 2        |
| 2.4      | Top Matter . . . . .                             | 2        |
| <b>3</b> | <b>Files used by L<sup>A</sup>T<sub>E</sub>X</b> | <b>2</b> |
| 3.1      | Inputs . . . . .                                 | 3        |
| 3.2      | Intermediate . . . . .                           | 3        |
| 3.3      | Outputs . . . . .                                | 3        |
| <b>4</b> | <b>Compiling Latex</b>                           | <b>3</b> |
| 4.1      | Adding Bibliography . . . . .                    | 4        |
| 4.2      | Automated Tools . . . . .                        | 5        |

## 1 What are all these \*TeX things?

Up until now I've been talking purely about the T<sub>E</sub>X. What is this L<sup>A</sup>T<sub>E</sub>X thing?  
Some resources 1 2 3 and my summary:

1. T<sub>E</sub>X does the typesetting
2. L<sup>A</sup>T<sub>E</sub>X adds a lot of features (macros) on top of T<sub>E</sub>X that make it easier to use
3. pdfL<sup>A</sup>T<sub>E</sub>X (or XeT<sub>E</sub>X, etc) are the engines that compile the L<sup>A</sup>T<sub>E</sub>X
4. T<sub>E</sub>XLive (or MiK<sub>T</sub><sub>E</sub>X, etc) are distributions that provide the engine and a number of commonly used pacakges

We write in LaTeX because it adds a bunch of useful things to TeX – using LaTeX is much less manual.

## 2 Structure of a Latex Document

The structure of a latex document looks like this:

```
Anything here is an error
\documentclass[options]{some class}
Preamble
\begin{document}
Top matter
```

```
Content
\end{document}
Anything here will not be typeset
```

## 2.1 Document Class

See here for a list of some document classes.

The document class affects which macros are defined and what they do. For example, in articles the main divisions are sections and then subsections. In a book they are chapters, sections, subsections. In a letter none of those are defined!

## 2.2 Preamble

This contains commands that affect the document as a whole. It is a place to define things, import packages and give L<sup>A</sup>T<sub>E</sub>X commands.

### 2.2.1 Packages

LaTeX by default can't do everything. In the same way that programming languages have libraries, L<sup>A</sup>T<sub>E</sub>X has packages. Some packages come with the LaTeX distribution (e.g. I use TexLive), others can be manually downloaded and installed. Including a package with `\usepackage` will make whatever macros are provided in that package available.

Theoretically you can get documentation with `texdoc <package>` but this breaks for me so just go to CTAN.

Some examples:

- Color: `Hi!`
- Hyperlinks: `hyperref` docs

## 2.3 Document

We need this to separate what is preamble and what is content.

## 2.4 Top Matter

You can define things about the article, e.g. who the author is, when it was written. This can easily be typeset – at any location – with `\maketitle`. It is just convention that these commands go at the top.

## 3 Files used by L<sup>A</sup>T<sub>E</sub>X

Until now we have just had single .tex files. However, with packages and compilation there are many file types to know about. See L<sup>A</sup>T<sub>E</sub>X ancillary files for a full list.

### 3.1 Inputs

- .tex: The main file.
- .sty: A package. Load into the main file with `\usepackage`
- .cls: A class. Loaded with `\documentclass` (e.g. `article.cls`)
- .bib: A bibliography file in the bibtex format.

### 3.2 Intermediate

Note that most of these are just text files so you can look in them and see what they contain.

- .log: A detailed account of what happened during the last compiler run
- .toc: A list of section headers (table of contents). See also .lof (list of figures), .lot (tables) etc.
- .aux: Another file that transports info from one compiler run to another [1].
- .out: list of hyperrefs
- .synctex: A file that allows the pdf and text file to be synced – i.e. if you click somewhere in the pdf it moves the cursor to the same place in the text (if the editor supports this).
- .blg: The log file from bibtex
- .bbl: The output of bibtex containing the formatted references

### 3.3 Outputs

- .dvi: An output format (the default for  $\text{\TeX}$ ). See wikipedia.
- .pdf: Thank goodness we finally got here!

## 4 Compiling Latex

A  $\text{\LaTeX}$  document can be very complicated. There may be figures cross referenced in the text, bibliographic references, and a table of contents. Compiling this all into a pdf seems like it might be quite hard. So what happens? Some resources: how does the compile chain work basics.

To start with note “It may be necessary to run  $\text{\LaTeX}$  several times to get the table of contents and all internal references right” [1].

Let’s cleanup all intermediate files for this doc and compile (`pdflatex latex.tex`) and see what happens:

1. We now have the aux, log, out, pdf, toc files. However the place where we expect the *Table of contents* to be is empty and we get a warning in the log and printed as output:

```
Package rerunfilecheck Warning:
File 'latex.out' has changed.
Rerun to get outlines right
```

The package `rerunfilecheck` provides additional rerun warnings if some auxiliary files have changed. So this is telling us that an intermediate file changed so we should rerun to have this info taken into account in the final pdf.

2. No warnings, TOC is inserted, but some page numbers are wrong.
3. pdf now looks good!

Why do we need multiple runs? L<sup>A</sup>T<sub>E</sub>X is a single pass compiler. So it doesn't go back and update things. So, for example if you want a TOC at the start of the document, you can't have it because we don't get to know what the sections are or what page they will be on! You first need to run the compiler to generate the list of sections and their locations, store these in temp files, then run again to insert the TOC. Note that adding the TOC might move things around (so the page numbers might be wrong) so you might need to run it again!

## 4.1 Adding Bibliography

Having a bibliography can complicate this even more. Let's add a simple .bib file referencing the TexBook [2] and the Not so Short Introduction to L<sup>A</sup>T<sub>E</sub>X [1]. To do this we add:

```
\bibliography{latex}      % the file name of the bib
\bibliographystyle{unsrt} % the style we want to use
```

to the end of the document. Now how to compile this?

As before we can run `pdflatex` 3 times to get everything else correct, but the bibliography is still empty and the citation is a question mark. The final error is

```
No file latex.bbl.
LaTeX Warning: There were undefined references.
```

So how do we get a .bbl file?

```
bibtex latex.bib
```

Bibtex reads the aux file which contains the location of the bibliography, the bib style and the list of citations. Note that this means that bibtex needs to be run **after the pdflatex has generated the .aux file**. Bibtex then reads the bib file which contains all possible citations. It pulls out the ones that are

used and formats them according to the given style. These it saves in the .bbl file.

Running pdflatex twice more eventually gets this all included in the pdf, though I'm not exactly sure why it takes two runs. This is why the routine that works in most (though not all) is often

```
pdflatex file && bibtex file && pdflatex file && \
pdflatex file
```

## 4.2 Automated Tools

Manually compiling is a pain. Fortunately there are automated tools that work out exactly what needs to be run how many times. See latexmk.

To compile a single file in one command, run

```
latexmk -pdf file
```

To watch for updates to the files and continually compile

```
latexmk -pdf -pvc file
```

To cleanup all non-input files

```
latexmk -c -bibtex latex
```

## References

- [1] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. *The Not So Short Introduction to LaTeX*. June 2007. Version 4.22.
- [2] Donald E. Knuth. *The TeXbook*. Addison-Wesley Professional, 1986.