

# Meeting One

Aaron Ecaj

February 2, 2011

## 1 Introduction to a $\LaTeX$ file

Welcome to LaTeX. Let's start learning how to use the software by going over this document piece by piece. We'll read the output and the source together, to understand how the latter generates the former. Before we begin, you'll need to be familiar with one command: `\verb`.

This macro takes one argument, delimited by an arbitrary character (I'll be using `+`). It then outputs this argument verbatim (hence the name), without doing any interpretation of special command sequences contained therein. (It also uses a special font.) This is like quotation marks for LaTeX – I'll be using this macro a lot to talk about the commands we will be using.

### 1.1 Preamble

All LaTeX files begin with a preamble. Let's go through the elements of this file's header one by one:

- **Comment.** Comments in LaTeX begin with a percent character: `%`. They last for the rest of the line, and are a good way to include info for yourself in the document. If you want to use a percent character in your document and have it print, you need to precede it with a backslash: `%`. Other characters that have a “special” meaning and can only be typed normally with a backslash include: `$ # { }`. The best way to get a backslash in your output is to use the command `\textbackslash`: `\`.
- **Documentclass.** Every LaTeX document must have a documentclass. There are classes for writing resumes, books, and even for making PowerPoint-style slide presentations. “Article” is the default class – it doesn't do anything funny, so you should use it for everything unless you have a good reason not to.
- **Packages.** These are add-on files that can be loaded to provide new commands, or change the behavior of existing ones. Time for a little excursus on LaTeX command begins with a backslash character. The name of the command follows, and is case-sensitive. Then come the command's arguments. The convention is that optional arguments are in square braces `[ ]` while required arguments are in curly brackets `{ }`. (This generalization does not extend well to some of the more complicated packages, but it is a good rule of thumb.) So above, we are issuing the command `usepackage`.

It takes one required argument, the name of the package being loaded. It can also take an optional argument, which consists of options for that package. Each package defines its own options, and they are specified in that package's documentation. I am loading the package `fullpage`, which makes the margins smaller. The `empty` option suppresses page numbering. The document class can also accept options; I've told this document to be in 12 point font. 10 is the default, 11 is also available.

- `Synctex`. This activates a new feature of LaTeX, which allows you to find the spot in the output corresponding to a specific location in the source, and vice versa.
- Title, author, date. Quite self-explanatory. If you don't set the date, the date on which the document is compiled will be used.
- Final notes. It is tedious to retype all this information each time you create a new document. I recommend that you create a template with all your favorite packages pre-loaded, that you can copy into each of your new documents. The preamble is also where you should put any new commands that you define, and you'll probably accumulate a collection of these as time goes by as well.

## 1.2 Document

Now we have reached the line that reads `\begin{document}`. As one might expect, this is the beginning of the document; anything after this line will be part of the output.

### 1.2.1 Formatting

There are a few basic commands for formatting text:

- **Bold face text**
- *Italic text* – this is actually emphasized text, an example of semantic formatting.
- `SMALL CAPITALS`
- Sans serif
- Typewriter (`monospace`)

You should almost never use these formatting macros in an actual document (except for `\emph`). This is to give you flexibility in making formatting changes. (Here an example will be given.)

### 1.2.2 Spacing

Spacing in LaTeX is a tricky beast. In general, LaTeX tries to make the right margin of the text align, as well as the left. It also puts extra space after periods, unless you give the option `frenchspacing` to the document class. Wrinkle the first: this means extra space after every period, even where you don't want it. So if you naively write Dr. Chomsky, then

you will get the extra space and it will look kind of funny. There are two ways around this. The first is to put a backslash before the space, forcing it to be regular-sized: Dr. Chomsky. However, if it happens that this appears near the end of a line, then the title and the name will be split across the line, which also looks odd: (this is just to fill up some space) Dr. Chomsky. The solution is to use a tilde character: Dr. Chomsky. This guarantees that the two words on either side of the tilde will never be split across lines.

Here is an example of both styles, side by side:

```
adsfjasdfadfadsfadsfasdfadsfadsfads Dr. Chomsky adsfadsfasdfadsfadsfadsfadsfadsfads
adsfjasdfadfadsfadsfasdfadsfadsfads Dr. Chomsky adsfadsfasdfadsfadsfadsfadsfadsfads
```

Furthermore, LaTeX doesn't care how many spaces you put between words: it collapses them all into single spaces. If you want to leave large blank spaces, you can use the `\hspace` command, which takes as an argument a length (in inches, centimeters, or any number of other more-or-less obscure units). Here's a 2.3 centimeter space: `\hspace{2.3cm}`. If you want to space something so it is right-aligned, use the `\hfill` command `\hfill` like this.

Now, let's talk about vertical spacing. If only one carriage return separates two lines, they will be combined into one paragraph. All these words will be on the same line, for example.

Two carriage returns make a new paragraph. By default, paragraphs are indented except when they immediately follow a section heading. If you'd like to suppress the indentation of a paragraph, place the command `\noindent` at the beginning. Finally, if you'd like to break lines without starting a new paragraph, use two backslashes, `\\`, like this. Finally, you can add vertical space just like you can horizontal space, with the `\vspace` command:

Hello. We have moved down 1.4 inches.

The last spacing gotcha in LaTeX is spacing after commands that have no arguments. No-argument commands eat any spaces that come after them, like what happens in `LATEX`here. To keep this from happening, we can put an empty pair of curly braces right after the command, but before the space: `LATEX` now does what we want.

### 1.2.3 Punctuation

LaTeX was invented at a time when computers weren't really good at processing symbols not found on the keyboard. Computers have gotten much better at this; LaTeX not so much. We will be using a version of LaTeX, called XeLaTeX, which has really good support for all Unicode characters, as well as fancy fonts in the output (more on this in a later lesson). However, some old habits die hard.

- Quotation marks. Open curly quotes can be typed with two backticks (above the tab

key); close quotes are two apostrophes: “like this.” If you use quotation marks, you will always get close quotes, which looks “ugly.” If you want single quotes, you can use just one backtick or apostrophe, ‘like so.’

- Dashes. There are several of these. Using a single dash character from the keyboard, you get a hyphen, which is OK for built-up words, like “built-up.” For dashes in sentences, you need to use either an en-dash or an em-dash. En-dashes – these are used with spaces before and after, and can be typed using two dash characters from the keyboard. Em-dashes—they are typed with three dashes, and should not be spaced.
- Dashes and numbers. With numbers, special dashes get used (though they are not typed in a special way in LaTeX. For ranges, use two dashes (i.e. an en-dash) but no spaces between the numbers: 123–456. For negative numbers, use this special incantation:  $\$-\$$ : –15 degrees. This is subtly different than –15 with an en-dash: note the spacing. (The dollar signs are used to enter a special mode for typesetting math, something we will save for a future lesson.)
- Ellipsis. The right way to get an ellipsis in LaTeX is with the command `\ldots`, which produces ... dots. These are better looking than the kind that you get by typing three periods, which look like this ...

#### 1.2.4 Environments

In addition to commands, LaTeX uses environments for arranging text. These are ways of wrapping commands both before and after text. The command `begin` begins an environment; it takes one argument which is the name of the environment. It must be matched by a corresponding `end` with the same argument. If you nest environments, you must close the inner one before the outer one. Just like commands, you can define your own. Some of the more useful ones are:

The `center` environment, which centers text.

```
quote, which is used for typesetting extended quotations (>4 lines) ajksdfk ajksdfkj
asdlfj aksdjf kasd f;asd f;lks djf; asdlkfjask;k fasd fa sdf asd fasd f;las dfkajs d;fj as;dfj
;adksl jfj;asd fasd ;fa sd;kfasd ads;
```

Hello from Lilliput

- `itemize`, which produces bulleted lists
  - like this one
    - nested bullets
    - are handled automatically
1. `enumerate`, for numbered lists
    - (a) which is also capable
    - (b) of nested lists
  2. see?

### 1.2.5 Footnotes

Footnotes are very easy to create in LaTeX. Simply use the command `\footnote` with the footnote text as the argument.<sup>1</sup>

### 1.2.6 Sections

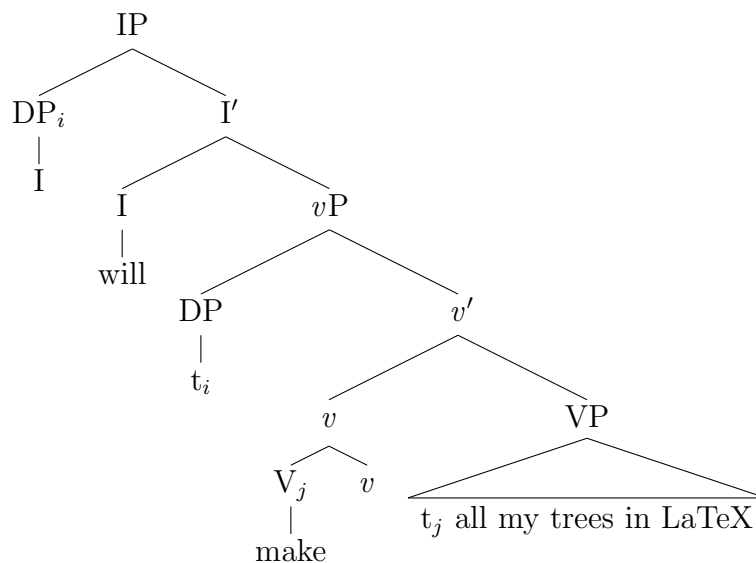
You can divide your document into sections using the `\section` commands. In addition to sections, there are subsections and subsubsections. If you use these commands, you can get an automatically generated table of contents, like so:

## Contents

<b>1</b>	<b>Introduction to a <math>\LaTeX</math> file</b>	<b>1</b>
1.1	Preamble . . . . .	1
1.2	Document . . . . .	2
1.2.1	Formatting . . . . .	2
1.2.2	Spacing . . . . .	2
1.2.3	Punctuation . . . . .	3
1.2.4	Environments . . . . .	4
1.2.5	Footnotes . . . . .	5
1.2.6	Sections . . . . .	5
<b>2</b>	<b>Trees</b>	<b>5</b>

You can also use commands to automatically insert section numbers and the pages they begin on (we will cover this in another session).

## 2 Trees



---

<sup>1</sup>LaTeX will then handle the numbering and positioning for you.