Introduction to LaTeX: Basic typesetting

Joey Stanley

January 29, 2020

Contents

Part I

Getting Started

1 Introduction

Today's workshop will introduce a lot of the basics of LaTeX. The goal by the end of the hour is for you to be able to typset a basic document. Some of the more sophisticated features—including those that you'll need for a full dissertation—will be covered in later workshops. Of course, there are countless tutorials and helps available to you online.

It is assumed that you already have some way to use LaTeX. You can do this on your own computer with a variety of software packages (Atom, TeXStudio, TeXShop, etc.) or you can do it entirely online using Overleaf.com. The concepts that will be covered today will apply regardless of which software you use. If you need help, please get my attention as soon as possible.

2 Basic document structure

To get started, the most basic structure you'll need in a LaTeX file is this:

```
\documentclass{article}
\begin{document}
  ...
\end{document}
```

Everything before the `\begin{document}` line is called the *preamble*, and it's where you'll put information to customize your document. Everything after that (but before `\end{document}`) is the body of your text, and that's where you'll do most of the typing. For now, we'll leave the preamble alone and we'll just start typing stuff in the body.

Part II

Basic Typesetting

3 Basic text

Once you're in the body of your document, you can type almost whatever you want. If you're considering switching over from Word to LaTeX, for example, the vast majority of what you copy and paste over will transfer just fine. However, there are a few things to consdier.

First, to make a new line in LaTeX, you need to hit enter twice, so that in your text editor there is a blank line between paragraphs. If you only hit enter once, LaTeX won't know that you want to do a new line and it'll actually continue from the previous paragraph. Here's an example with what your code might look like on the left, and what the PDF might look like on the right (this template of code + PDF will be used throughout this document):

```
Paragraph one.
Paragraph two.

Paragraph three.
```

> Paragraph one. Paragraph two. Paragraph three.

By default, LaTeX will do some indentation for you at the start of each paragraph. You can suppress indentation for a paragraph by putting `\noindent`. just before it:

```
This is a very short paragraph
but it is long enough to show
that it's indented.

\noindent
This paragraph is slightly
shorter and it's not indented.
```

> This is a very short paragraph but it is long enough to show that it's indented.
> This paragraph is slightly shorter and it's not indented.

3.1 Special characters

There are a few characters that will cause you some trouble in LaTeX. These are characters like \, %, &, and a few others. They won't display properly because they are reserved for specific purposes within the LaTeX language itself. For most of these, you can just precede the symbol with a backslash (\). For the backslash itself, you'll need to use `\textbackslash`.

```
Made up numbers are used in
\latex documents 75% and 90% of
the time!

But, almost 100\% of the time it
doesn't matter!
```

> Made up numbers are used in LATEX documents 75
> But, almost 100% of the time it doesn't matter!

One thing that may not be intuitive is how quotation marks work. Because English text typically uses the curly quotes ("...") instead of straight quotes ("..."), we have to type opening and closing quotation marks differently. For opening quotes, use the tick ' character (which is found to the left of the 1 key on my keyboard). For closing quotation marks, use just the regular apostrophe. For double quotes, just two two ticks or two apostophes. There's not need for the actual double quotation character. For apostrophes, just use the apostrophe.

```
I overheard her say, ''And he's
like, 'I don't believe you!'.''
```

> I overheard her say, "And he's like, 'I don't believe you!'."

Finally, for en-dashes, which are for ranges of numbers and a few other special cases, type two hyphens in a row. For em-dashes, type three hyphens in a row. For a regular dash or hyphen, just one will do fine, as expected.

```
Kelly single-handedly made 4--5
dozen New York--style bagels and
they were---and I cannot stress
this enough---heaven-sent.
```

> Kelly single-handedly made 4–5 dozen New York–style bagels and they were—and I cannot stress this enough—heaven-sent.

3.2   Non-English characters

For accented, non-English, or other less-common characters, you may have to play around with them. Most accented characters will render just fine in LATEX, which means you can feel free to type naïve, résumé, japapeño, vis-à-vis, tête-à-tête, façade, Māori, and háček. For other characters that are more specialized or are non-Latin based, you may have to resort to add-on packages to get them to render properly. The first one to check is the babel package, which has support for over 200 languages, including Arabic, Cherokee, Chinese, Devanagari, Georgian, Greek, Hebrew, Persian, Faroese, Japanese, Korean, Russian, Sanskrit, Vietnamese, and a whole bunch more.

3.3   Commenting your code

Under constrution. . .

4 Changing the text (locally)

4.1 Fonts

In addition to being able to type whatever you want, it's important to also be able to format that text. You can use **bold** with \textbf{} and *italics* with \textit{}. There is also a *slanted text* with \textsl{} in case you need that.

```
This is \textbf{bold}.
This is \textit{italicized}.
This is \textsl{slanted}.
```

> This is **bold**.
> This is *italicized.*
> This is *slanted.*

Also, note that these functions can be embedded within one another so that to do ***bold italics*** you can type \textbf{\textit{bold italics}}.

4.2 Captitalization

You can also adjust the capitalization of your text. Note that these commands ignore whatever case your original text is in and displays them correctly.

```
This is \uppercase{upPerCase}.
This is \lowercase{loWerCaSe}.
This is \textsc{Small Caps}.
```

> This is UPPERCASE.
> This is lowercase.
> This is SMALL CAPS.

Note that if you have accented characters that need to change case, try \MakeUppercase{} and \MakeLowercase instead.

4.3 Typefaces

It may also be important to change font families. Body text is usually written in a serif font, which is one that has the little tails and horizontal embellishments on the letters (called *serifs*). To force text to be in a serif (or *roman*) font, you can use \textrm{}. Headers are often typed in a sans serif font. Computer code is often typed in a typewriter or monospaced font:

```
A \textrm{serif} font.
A \textsf{sans serif} font.
A \texttt{typewriter} font.
```

> A serif font.
> A sans serif font.
> A typewriter font.

4.4 Size

Finally, it may be important to change the font size. While this is not done very often within the body of a document, it's good to be aware of the options.

First off the default size of the text is 10-point font. But you can change it to something different by adding some additional information in the very first line of your document: \documentclass[11pt]{article} will change it to 11-point

font, for example. As with most other things in the preamble, this change will modify your entire document.

The easiest way to *change* sizes is by using one of several built-in commands for changing the size relative to the normal size. The benefit of this is that if you ever need to change your font size, these changes will scale up or down proportionally.

```
\Huge Typesetting!
\huge Typsetting!
\LARGE Typsetting!
\Large Typsetting
\large Typsetting!
\normalsize Typsetting!
\small Typsetting!
\footnotesize Typsetting!
\scriptsize Typsetting!
\tiny Typsetting!
```

Typesetting!
Typsetting!
Typsetting!
Typsetting!
Typsetting!
Typsetting!
Typsetting!
Typsetting!
Typsetting!
Typsetting!

For these, you can put them at the beginnning of a paragraph to modify the entire paragraph (and any subsequent paragraphs until another size change), or you can put them in brackets to modify just a specific word or two:

```
\large
This whole paragraph is large.

Oops. This one is too.
```

This whole paragraph is large.
Oops. This one is too.
Just this text is large.

```
\small
Just {\large this text} is large.
```

Of course you can manually set the font size to a specific size if you'd like. This is accomplished with the template `\fontsize{}{}\selectfont`. Within the first pair of curly brackets, you put the size and the unit (points = `pt`, millimeters = `mm`, centimeters = `cm`, inches = `in`, whatever). Within the second pair, you need to specify the *leading*, which the typographical term for the line spacing, because when you change font size you're probably going to want to change the spacing. Typically, you give it a little extra spacing so that it doesn't run into the line above it.

```
This is 10-point font, which is
the default, but you can change
it to something like
\fontsize{15pt}{19pt}\selectfont
15-point font with 19pt leading.

\normalsize
Just be sure to change
it back when you're done!
```

> This is 10-point font, which is the default, but you can change it to something like 15-point font with 19pt leading. Just be sure to change it back when you're done!

5 Document structure

When moving beyond the paragraph level, it's important to know how to add some elements to your document to give it structure and organization. In this section, we look at sections, lists, and footnotes.

5.1 Headers and subheaders

Unless you're writing a novel, it's probably important to have some headers to break your document up into sections. Fortunately, it's straightforward with the `\section{}` command. For deeper levels, you can do `\subsection{}` and even `\subsubsection{}`.

```
\section{Main level}

\subsection{One level deep}

\subsubsection{Two levels deep}
```

> **1 Main level**
>
> **1.1 One level deep**
>
> **1.1.1 Two levels deep**

It's generally unwise to go a fourth level deep because readers often have a hard time of keeping that many levels straight, so it's not as straightforward to do in LaTeX.

If you don't like the numbers, you can suppress them by adding an asterisk after the name of the command but before the curly brackets:

```
\section*{Main level}

\subsection*{One level deep}

\subsubsection*{Two levels deep}
```

> **Main level**
>
> **One level deep**
>
> **Two levels deep**

Technically, it's possible to just include bigger and bolder text to create the illusion of sections, but that's not recommended. The reason is because when you type things as sections, you can generate a table of contents. You can do

so by putting `\tableofcontents` somewhere near the top of your document (perhaps just under the title). That's exactly what I've done in this document.

5.2 Internal references

Under constrution. . .

5.3 Lists

It doesn't happen very often, but sometimes it's important to make lists of some sort. To make a numbered or a bulleted list, we'll have to introduce the idea of an *environment*. LaTeX environments can be thought of a more complicated function that applies to a larger passage of text. They begin with a `\begin{}` tag and end with a `\end{}` tag.

For *unordered* lists, which typically have a bullet point at the start of each line, you use the `itemize` environment. Each item within that lists starts with the `\item` tag:

```
\begin{itemize}
  \item First bullet
  \item Second bullet
\end{itemize}
```

- First bullet
- Second bullet

For *ordered* lists, which have numbers at the start of each line, use the `enumerate` environment, but the syntax is otherwise identical:

```
\begin{enumerate}
  \item First bullet
  \item Second bullet
\end{itemize}
```

1. First number
2. Second number

You can even nest these lists, and mix and match how you please.

For more information, including how to customize the symbols, change the numeric system (particularly with embedded lists), and start numbered list using a different number, see this tutorial by Overleaf.

5.4 footnotes

Under constrution. . .

5.5 Long quotations

Under constrution. . .

6 Alignment and spacing

6.1 Text alignment and justification

The main body text in LaTeX will be justified, meaning text will be flush against both the left and the right margins. To override this behavior, you can use the `\raggedright`, `raggedleft`, or `\centering` functions before the first paragraph in your document:

This is an example of fully-justified text, which is the default in LaTeX. You can see that the text is flush against the left and right margins. Personally, I think this looks best so long as the body of the text is wide enough.	This is an example of left-justified text. Somewhat unintuitively, I had to use the `\raggedright` command to pull this off, but if you pay attention to the *right* margin, you can see that the text is all ragged and not lined up.
This is an example of right-justified text. Somewhat unintuitively, I had to use the `\raggedleft` command to pull this off, but if you pay attention to the *left* margin, you can see that the text is all ragged and not lined up.	This is an example of centered text. I used the `\centering` command to pull this off. This is typically not recommended unless it's for a one- or two-line passage like a title or something. For long paragraphs, it's hard to read centered text.

For more information on how to apply these changes to your entire document, see this page in Overleaf. You may also want to check out the `ragged2e` package, which implements a more sophisticated method for hyphenation and justification to make your document look a little nicer.

6.2 Adding and removing white space

Sometimes, LaTeX just doesn't quite behave the way you want it to. It is quite common among beginners (and long-time users) to have something that isn't positioned quite right. One little trick for making small adjustments to where things are positioned on the page is with the `\vspace{}` and `\hspace{}` commands. As their name implies, they simply add some *vertical* space or some *horizontal* space. Within the brackets, you put how much space should be added, using the same syntax that we saw above with the font sizes (a number plus a unit of measurement).

```
This gap [\hspace{1in}] is one
inch wide.

\vspace{1cm}
This paragraph is one
centimeter down.
```

> This gap [   ] is one inch wide.
>
>
> This paragraph is one centimeter down.

A very useful unit of measurement to be aware of is the *em*. Despite what the name implies, it is not the width of the character *M*, but rather it is the *height* of your text. So if you're typing in 10-point font, an em is 12 points. If you want to insert a one-line break between elements on the page, `\vspace{em}` is a really good way to do that.[1]

Also, it's good to know that you can put negative numbers as the size of the gap. This is useful in case you need to squeeze things together or remove an indent or something.

```
\hspace{-2em}This paragraph has
a negative indent and spills
into the edge of the box.

\vspace{-0.5em}
This paragraph is uncomfortably
close.
```

> This paragraph has a negative indent and spills into the edge of the box.
> This paragraph is uncomfortably close.

---

[1] In fact, the gaps between these sample code snippets and the paragraph following them are created in this way.

Part III

Global settings

Most of what we've done so far is how to type basic text and how to make small, isolated changes to how that text is displayed. With this foundation, we can now move on to how to make global changes to your document. By global, I mean they're things that you just do one time (usually in the preamble) and they effect the look of the entire document. The majority of these changes will happen in the preamble of your document, often with the use of add-on packages.

You may notice that all LaTeX documents start by looking the same way. There's a consistent font, the headers are styled a specific way, the text is fully-justified, and the margins are a bit wider than the academic tradition of 1 inch on all sides. The next few sections illustrate how you can change each of these properties, and how to make your document look a little more professional, crisp, and aesthetically pleasing.

7   Installing libraries

Under constrution. . .

8   Page properties

One of the biggest things that will determine the look of your document is how big the paper is and what the margins are. This section shows how to change these using the `geometry` package.

8.1   Page size

By default, LaTeX will use A4 paper, which is 21cm wide and 29.7cm tall. Note that it's close to, but not exactly, 8.5 inches by 11 inches.

To change the paper size of your document, you'll need to load the `geometry` package in the preamble of your document. In square brackets before the name, specify what size paper you want. Here, I'll change it to legal paper size, which is a bit longer.

```
\usepackage[legalpaper]{geometry}
```

`geometry` has a handful of paper sizes built in (like `legalpaper`). To see a full list of paper sizes, look at section 5.1 of the package's documentation.

In the event that you're a normal human and don't know the paper sizes by their official name, you can also specify an explicit size using the `paperwidth` and `paperheight` arguments. This creates a half-sized sheet of paper, which is used for some books.

```
\usepackage[paperwidth=5.5in, paperheight=8.5in]{geometry}
```

Note that UGA dissertations must be exactly 8.5 by 11 for them to be able to be bound at Tate Print and Copy. My recommendation is to specify these dimensions exactly at the top of your document rather than relying on the default A4 size.

8.2 Margins

The margins of your document are also controlled with `geometry`. The easiest way to specify them is by adding the `margin` option and tell how big you want them to be. Here's how you'd get 8.5×11 paper and 1 inch margins on all sides

```
\usepackage[paperwidth=8.5in,
            paperheight=11in,
            margin=1in]{geometry}
```

(Note that I separated out these components onto separate lines simply because there wasn't enough space to do it in one line. In your code, you can do the same thing if you'd like, in case it helps make your code clearer.)

By the way, an alternative syntax for the above block of code is this:

```
\usepackage{geometry}
\geometry{paperwidth=8.5in, paperheight=11in, margin=2in}
```

They are identical under-the-hood and will accomplish the same task for you.

If you would like to specify specific margins, you can do that as well.

```
\usepackage{geometry}
\geometry{paperwidth=8.5in, paperheight=11in,
          left=1in, right=1in, top=1in, bottom=1.25in}
```

Typographically, it's nice to have the bottom margin about a quarter inch bigger than the top margin to achieve the appearance of a more balanced page.

There are many more aspects of the page's layout that you can change with the `geometry` package besides the page size and margins. I encourage you to look at this guide from Overleaf, this guide from WikiBooks, and of course the package's documentation.

9 Fonts

Link to the list of fonts

Line spacing

10 Hyphenation and justification

Look through the typsetting book. What did I do for my dissertation? Also add the code for overfull lines.

11 Widow and Orphans

Also mention ragged bottom in case that's relevant.

Part IV
Final Remarks

12 Where to go for help

You may feel overwhelmed at first. Fortunately LaTeX has been around for a long time, and there's a *lot* of free resources online.

One really good place to start is Overleaf's help pages.

Another good one is WikiBooks.

13 Conclusion

This document has outlined some of the key components of LaTeX. With these skills, you should be able to write the majority of your dissertation. Other topics like mathmatical formulas, images, and tables will be covered at a later date.