Introduction to LaTeX: Basic typesetting

Joey Stanley

January 31, 2020

Contents

Part I
Getting Started

1 Introduction

Today's workshop will introduce the basics of using 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. This handout contains more detail than what we will have to cover in the workshop itself, so be sure to look through it for additional comments and examples. 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.), which allow you some additional flexibility and the ability to work wherever you are. Or you can do it entirely online using Overleaf.com, which is probably easier for beginners. 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.

LaTeX has a notoriously steep learning curve. This document will not be able to cover everything. Fortunately there is lots of help online. The two help pages I use the most are Overleaf's documentation, and Wikibooks. I will link to these sites throughout the handout so you can read in greater detail and learn more.

2 Basic document structure

Let's get started. The most basic code you'll need in a LaTeX file is this:

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

Everything between `\documentclass{article}` and the `\begin{document}` line is called the *preamble*, and it's where you'll put information to customize your entire document. Everything after `\begin{document}` and before `\end{document}` is the body of your text, which is where you'll do most of the typing.

In this handout, Part II will mostly cover the body of your document and Part III will cover things you can add to your preamble. Let's get started then.

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 consider.

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:

```
Paragraph one.
Paragraph two.

Paragraph three.
```

Paragraph one. Paragraph two.
Paragraph three.

This template of code + PDF will be used for examples throughout this handout.

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.

Note that the first paragraph after a header will not be indented, following typicaly typographic conventions.

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 scripting language itself. For most of these symbols, you can just precede it with a backslash (\). In the next example, notice that the text stops at the % sign after the +75+.

```
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!

The backslash itself is a little tricker, and you'll need to use the command `\textbackslash` for it to appear.

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 no 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.

For additional help on special characters in LaTeX, see this Wikibooks page. As far as I know, any character can be typed in LaTeX: you may have to install additional packages to get it, but do not be discouraged if you cannot easily get a symbol to appear. Someone else has likely needed that symbol before and has solved your problem.

3.2 Non-English characters

For accented, non-Roman, 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* directly into your editor and they will appear as you expect them to.

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 best one to check is the `babel` package, which has support for over 200 languages, including Arabic, Cherokee, Chinese, Devanagari, Georgian, Greek, Hebrew, Per-

sian, Faroese, Japanese, Korean, Russian, Sanskrit, Vietnamese, and a whole bunch more. Chances are, if you need to type a passage (or an entire document) in a language other than English, `babel` can take care of it.

For additional help on internationalization in LaTeX, see this Wikibooks page and these Overleaf help pages on international language support, `babel`, and other related packages.

3.3 Commenting your code

Before we get into the thick of things, I should mention the importance of commenting your code. Like virtually all scripting and programming languages, LaTeX offers a way for you as the user to insert comments alongside your code so that you can remind youself what, why, and how you did things.

The way to comment your code is with the `%` character. The way it works is that LaTeX ignores anything that follows the percent symbol on a line. This is why in the example above the line seemingly stopped after the `75`. Some people like to put comments for a particular line of code one line above. Others like to put it at the end of the line.

```
% To get the fancy LaTex, use
the \LaTeX command.
\LaTeX is 50\% patience, 50\%
skill, and 50\% art.

We don't stop at 100\%! % Type
\% for a percent sign
```

LaTeXis 50% patience, 50% skill, and 50% art.
We don't stop at 100%!

It may not seem important now, but start getting into the habit of commenting your code. A few weeks from now, you're going to forget why you did what you did. For me, most of the comments are in the preable rather than the body, but if there's something tricky I had to do in the body, I put comments there too. I like to not just explain *what* code does, but *why* I had to do it that way. I also like to put links to online help pages where I learned a particular trick. You can have as many comments as you want, and LaTeX won't care a bit. Your future self will thank you.

For additional reading on commenting your code, see this Overleaf page and this section in Wikibooks.

4 Changing the text (locally)

Now that we've seen how to do the basics of typing text in LaTeX, let's see how we can make local changes to that text. But local, I mean changes that apply to one or maybe a few words at a time.

4.1 Text formatting

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}}`.

Be aware that if you copy and paste text over from Word into LaTeX, it will not preserve this formatting. You will have to manually add these codes in yourself.

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 to uppercase (like turning æ to Æ), you might want to try `\MakeUppercase{}` and `\MakeLowercase` instead.

4.3 Font families

It may also be important to change the font family. Body text is usually written as it is in this handout, in a serif font. *Serifs* are those little tails and horizontal embellishments on the letters. To force text to be in a serif (or *roman*) font, you can use `\textrm{}`. Headers are often typed in a sans serif font, or one without those little tails. 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.

For information on how to change the font in your entire document, see section 10.

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 set 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!
```

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.

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

This whole paragraph is large.
Oops. This one is too.
Just 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). For example, to change it to one centimenter-high text, you'd use `1cm`. Within the second pair of brackets, 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 make the spacing a little larger than the font size so that the letters don'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!

For more information on font families and sizes, see this Overleaf page or this Wikibooks page. For more information on changing the line spacing in your entire document, see section 9.1.

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 to give it structure and organization. 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.

By default, it'll add the numbers at the start of each line. 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 illustion 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

It's quite common in a dissertation-sized document that you'll want to refer to some other section ("As mentioned in section 2.1..." or "As will be explained in section 5.5..."). In most typsetting programs, you would have to type that number in yourself. But as soon as you start to reorganize your document, such that section 2.1 is now section 3.2, you'll have to go back and change all those references. This is tedious and error-prone. Isn't there a better way? Yes!

One of the selling points in LaTeX is its abilty to handle internal references. They way it works is with a pair of commands: the section that you want to make reference to, you mark with `\label{...}` and the in-text reference to that section is with `\ref{...}`. The best part: LaTeX will make a hyperlink directly to that section for you so that readers can easily navigate to that page.

Let's say I have some section header called "Background". I can add a small bit of code underneath, giving it the label `background`:

```
\subsection{Background}
\label{background}
```

2.1 Background

Then, later in your document, you insert the code instead of the number:

```
As seen in section
\ref{background}...
```

As seen in section 2.1...

Now, if I move the background section around somewhere, that 2.1 will change automatically.

In this document, internal references are indicated using the default, a red box around the text. This is useful for people viewing the PDF, and it doesn't print. You may like the look of a light blue text instead, as if it were a webpage (though be aware that it will show up as light gray when printed). For details on customizing how hyperlinks appear see this Wikibooks page on the `hyperref` package.

When I typset my dissertation, I put labels underneath every section right away. That way, they all had a reference ready for me to use in case I needed

it. In fact, knowing that it was just that easy to make these kinds of internal references made include many more of them that I would have otherwise.

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

Footnotes in LaTeX are quite straightforwardly implemented with the `\footnote{}` function. Wherever you put that code, that's where the little superscript number will appear in your text. Whatever you type between brackets will be the footnote itself. LaTeX will take care of lettering/numbering them automatically:

```
Here's some text!\footnote{And
here's a footnote!}
```

Here's some text![a]

[a] And here's a footnote!

See this Overleaf page and this Wikibooks page for more information on unusual footnote situations, like multiple marks for the same footnote and custom markers on a footnote.

5.5 Long quotations

In dissertations, it's relatively common to quote a longer passage from some previous source. For these cases, it's nice to have the quotation indented a little bit to make it clear that it's a quote and not your words. To accomplish this, you simply use the `quotation` environment, with the usual `\begin{}` and `\end{}` tags.

```
Some famous author said this:
\begin{quotation}
  \noindent
  ``This is a super awesome,
  totally incredible, long
  passage that the person said!''
\end{quotation}
I wish I could be as smart as
them!
```

Some famous author said this:

> "This is a super awesome, totally incredible, long passage that the person said!"

I wish I could be as smart as them!

By default, the first line of the quoted paragraph is indented, but I find it looks a little nicer if it starts without the extra indentation.

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 to make your document look a little nicer.[1]

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 `\vspace{}` and `\hspace{}`. As their names imply, 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 (necessarily) 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 10 points wide. If you want to insert a one-line break between elements on the page, `\vspace{em}` is a really good way to do that.

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{-1em}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.

I find myself using `\vspace{1em}` quite a bit if some element on the page, like a table, figure, or something else, doesn't have enough space before or after it. In fact, in this handout, the gaps between the sample code snippets and the paragraph following them are created in this way.

For more information about spaces, including how to account for optional spaces at the ends of lines or pages, see this page in Wikibooks. Also, sometimes

[1] Be aware though that `ragged2e` is no longer being maintained.

you'll find that commands like `\LaTeX` sort of "eat" up the following space. You can insert sort of an optional space after commands with the `\xspace` command in the `xspace` package.

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. This document more or less uses the default LaTeX settings I won't be able to easily demonstrate what these changes look like in a document.

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 packages

If you're using Overleaf, this section is not super relevant because Overleaf installs and handles packages for you automatically. But for those of you that are working offline, it's important to know how to work with add-on packages.

LaTeXcomes installed with several dozen packages as part of its standard distribution. But occasionally you will need to use some more specialized package for whatever reason.

The easiest and most superficial way to download and use a package is to download it into the directory where your LaTeX files are.

For more information on installing packages, see this Wikibooks page and this documentation from CTAN.

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.

8.3 Page numbering

9 Paragraph and line properties

9.1 Line spacing

Line spacing (single-spaced, double-spaced, etc.) is an important part of the typography of your document. There is no perfect number because it depends on a variety of factors like what font you're using, the size, how wide your page is, etc.

The easiest way to set the spacing is be redefining a function called `\baselinestretch`, which takes the default spaacing and stretches it by some amount. You can redefine like this: `\renewcommand{\baselinestretch}{1.6}`. The value there is not terribly intuitive, but by setting it `1.6` it'll give you double spacing. For one and a half spacing, set it to `1.3`. I'd recommend something a little smaller like `1.1`, but do what you think looks best.[2]

9.2 Line breaks

It's important to at least give some thought to hyphenation and justification in your document. Because text must flow from one line to the next, LaTeX has to determine where the line should end. When a word spills over into the margin, should it go on the next line, should the previous words squeeze together a little bit to make room, or should it be hyphenated? LaTeX has to make these decisions with every single line in your document. And it usually does a pretty good job. But sometimes it doesn't.

The simplest way to control for where line breaks are is by including specific codes in your document to help LaTeX out a bit. For example, if there's a particularly long word that LaTeX doens't want to split up, you can insert a "soft hyphen", meaning LaTeX will use that spot for hyphenation if it needs to, but if it doesn't, the hypen isn't there. The code for this is just `-`.

In this example, LaTeX but the hyphen between the *l* and the *i*, but I think it makes more sense to put it between the *i* and the *g*, so I forced it there with a soft hyphen. Note that there are other soft hypens in the word, but they're not displayed since they're not at the end of the line.

[2] In my opinion double-spacing is far too much. It's reminiscent of typewriter days, it wastes paper, and the fact that you're using LaTeX means that we certainly don't need to emulate what a typewriter does. At the same time though, single-spaced is a little too narrow. I appreciate that sometimes people like to strike a balance and go for maybe 1.5 spacing, but even that is a little too much.

```
Mary Poppins sings supercalifrag
ilisticexpialidocious!

Mary Poppins sings super\-cali\-
fragilistic\-expiali\-docious!
```

Mary Poppins sings supercalifragilisticexpialidocious!
Mary Poppins sings supercali-fragilisticexpialidocious!

Alternatively, if you want a break to happen at a space, try putting `\linebreak` there instead of the space.

Kind of the opposite problem is that sometimes LaTeX will put a line break when you don't want it to. In this example, there's a break between *Mrs.* and *Smith* and I don't like it. I can use a *non-breaking space* with a tilde (`~`), instead of a regular space, and it'll tell LaTeX to never put a line break there.

```
I had a teacher named Mrs. Smith
who loved Halloween.

I had a teacher named Mrs.~Smith
who loved Halloween.
```

I had a teacher named Mrs. Smith who loved Halloween.
I had a teacher named Mrs. Smith who loved Halloween.

It'll move things around to comply with your imposed rule. Of course, now there's the separate problem of the spaces being too wide between words.

Adding soft hypens and non-breaking spaces is handy when you need to force a very small change. But it can get annoying because as soon as you change some text, it'll move the word around and you may have a whole different set of problems. But you can tell LaTeX to avoid some of these problems altogether by implementing a more sophisticated algorithm for hyphenation (see the next section).

In some cases, you may want to disable hyphenation entirely. This is usually not recommended, but it's useful to see how. You can do this by putting the command `\hyphenpenalty=100000` somewhere near the top of your body text (not the preamble). The details of this aren't super important, but LaTeX assigns penalties to certain things if a conflict arises, and this sets it to some huge number so that any alternative is better than hyphenation. Sometimes you may not want to remove them entirely, but you can adjust the degree to which hyphenation occurs by setting that penalty lower (to perhaps `1000`) while also adding `\tolerance=1000` (and adjusting it from there). It may take some toying around to get the results you want.

9.3 Handling overfull lines

Once you've written several paragraphs in your document, you may find that there are lots of warnings when you compile, saying there are a bunch of "overfull" lines. What this basically means is that there's text that is spilling into the margins and LaTeX couldn't fix it for you. It gives you a warning message so that you can take care of it. You can try adding soft hyphens, non-breaking spaces,

or rewriting the text a little bit to resolove the problem, but sometimes that just creates a new set of overfull lines.

A bandaid over this problem is to add `\sloppy` somewhere near the top of your document. It will force LaTeX to prevent these overfull lines. The results may be good enough, but you may find occasional lines that just didn't end up looking right (the spaces between words/letters is too wide or too narrow). But if you're just sick of those warning errors, this is a way to shut most of them up.

If you really want to take full control over how LaTeX does its hyphenation and justification, take a look at the `microtype` package. It's far too much to discuss in this introductory workshops, so I'll let you look through the documentation yourself (you may need to learn a little about typography to fully understand the jargon). I will say though that after implementing it in my dissertation, it really looked sharp. The best tutorial I've seen, which also explains the little nuances and details you may not have thought of, is this one by Siarhei Khirevich.

9.4 Widows and orphans

Sometimes you may find that a paragraph starts at the last line of a page, and spills over onto the next page. Some people find this one-line-at-the-end-of-the-page paragraph, which are called *widows*, ugly. Similarly, when the last line of a paragraph is the first line of a new page, that's called an *orphan*.

There are different strategies for preventing widows and orphans. This TeX-FAQ page recommends playing around with the penalty, like you did with hyphenation above, which will force LaTeX to do something lik making the bottom margin bigger to force a widow to start on the next page or something. For a document like this handout that has many small paragraphs and code chunks, this effect is more apparent, but for a dissertation with potentially many longer paragraphs, it's not as noticeable. Alternativly, LaTeX may shrink the space between lines just a tiny bit for just one page to allow room for the orphan to join the rest of its paragraph.

One easy way is to remove widows and orphans entirely is to put `\usepackage[all]{nowidow}` in your preable and LaTeX will do a mxiture of these techniques. I've found though that sometimes it'll add extra space between paragraphs, and it looked like there was some weird double-spacing thing going on between them. So I found that adding `\raggedbottom` helps with that, and forces any extra space to occur at the bottom of the page (in other words, it allows for a flexible bottom margin size).

10 Fonts

Some people like the default font used in LaTeX. Others don't. If you want to change the font, you're in luck because there are many available to you. Unfortunately, changing the font in LaTeX is not as straightforward as it is in other typsetting programs.

The best place I've found for fonts is The LaTeX Font Catalogue. This site contains hundreds of fonts for you to choose from, conveniently organized by their properties (serif vs. sans serif, for example) and whether they support math symbols.

For example, let's say I looked through the Font Catalogue and decide I want to change the font to Garamond. It's a classic and (and in my opinion) sophisticated-looking font.[3]. On the page for Garamond on Font Catalogue, I see under the "Usage" section, the following code:

```
\usepackage[cmintegrals,cmbraces]{newtxmath}
\usepackage{ebgaramond-maths}
\usepackage[T1]{fontenc}
```

All I need to do is copy those three lines of code to somewhere in my preamble and the PDF will use that new font.

Now, be aware that not all fonts are straightforward like that. Each one is a little different and you may be required to do more or less work to get it to render properly. For example, Garamond only works because I've got it installed on my computer already. You may need to consult the font package's documentation (there should be a link at the bottom of its Font Catalogue page) and take a look at the readme to see how you can get it to work. Sometimes, you must actually buy the font in order to use it. Also, the way to change fonts may depend on what LaTeX compiler you're using.

[3] This is the font I used for my dissertation so I'm little biased.

Part IV
Final Remarks

11 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. Here are the main ones that I've linked to a lot throughout this document:

- Overleaf's help pages does a great job at explaining the basics of how to use LaTeX. I've used them many times myself. They include lots of actual LaTeX files that use this code, so you can tinker around with them in sort of a sandbox mode without messing up your file. They purposely don't go into too much detail so that beginners don't get too overwhelmed.
- Another good one is WikiBooks. This is a Wiki-fied version of a free eBook (you can get it as a PDF too). This one goes into more depth than the Overleaf site, on more topics. It will explain how to do things several different ways, weighing the pros and cons of each one. It may be a little intimidating at first, but just as any skill, with some practice you'll be able to use it.
- Every package that I've mentioned, any many many more, can be found on CTAN. Each package also has an in-depth documentation to go along with it that will go into more detail than any tutorial could. It will explain all the options when using that package, and will often show lots of example code and usage. Some documentation is better than others, and again it takes some skill. But if you want to know more about something that I've mentioned, the documentation is where you'll find your answer.

As always, countless people have written blog posts and other help guides, which are available for free online. Google is your friend. Do not be afraid to ask the internet for help because we all do it, even the pros.

12 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. Learning LaTeX is a useful skill and will help you think more about layout and typography. Above all though, most of us who use it find it to be kind of fun. We avoid the headaches of Word and other typesetting programs, and our documents have the potential to look stunning. Above all, the document should be readable, and I think LaTeX is an excellent way to do that for your readers.