LaTeX for Luddites
(An Ancient Philosopher's Guide)

Willie Costello

beta version 1.0.1 || last updated August 30, 2012

Abstract

This document aims to introduce LaTeX to people (and specifically, ancient philosophers) who have never used it before, indeed, who have never done any sort of coding whatsoever, ever. As such, it will not teach you everything (or even most things) about how LaTeX works. Rather, it is designed to teach you *how to write in LaTeX*, without getting hung up on all the complicated computer wizardry behind it.

This guide is designed to work as both a general introduction to LaTeX and as a go-to resource for while you're writing papers. It takes you from the absolute basics of writing in LaTeX to the utilization of some of its more advanced features. You may not wish to read through this entire guide before getting to work on your paper. Generally, you learn LaTeX best when you are working on your own documents, figuring out to do whatever it is you want to do on the spot, so I actually recommend just reading this guide for as long as it takes you to figure out the basics of writing in LaTeX, and then going off and doing your own thing, referring back to this guide as necessary.

In order to get the full potential out of this document, you should be constantly switching back and forth between the compiled PDF and the original source code. (If you did not compile this PDF from the original source, go to my LaTeX website to download the full package.) The PDF provides an ordinary language explanation of how a feature works, while the source shows how this is actually accomplished *in situ*.

A small note, to avoid any possible confusion: In order to print out sample code on the PDF version of this document (the stuff that looks like `this`), I had to use a special LaTeX

command (`\verb`) in the source code. Because of this, you will not always be able to copy this document's source code into a new document and have it work correctly. Generally, if code appears like `this` in the PDF, directly copy it from there. Refer to the source code only when the guide tells you, or to get a general handle on how things are working; just be wary of anywhere you see a `\verb` command.

Finally, there will surely be many issues you come across while writing papers that are not covered in this guide. Fortunately, nearly any question you'll ever have about LaTeX can be answered by Google these days, simply by typing in your issue into the search bar. (Hint: it helps to include the word "LaTeX" in whatever search you're doing.) Do feel free to drop me a line about questions you have, as they will only help make future versions of this guide even better. You can email me at williecostello@gmail.com.

So, without further ado, let's get to work with LaTeX!

Contents

1 Getting started **5**

2 Writing in sections **5**
2.1 Subsections . . . 6
2.1.1 Subsubsections . . . 6

3 Essential writing tips **6**
3.1 Quotation marks: The easiest mistake to make . . . 6
3.2 Indenting: Don't do it . . . 7
3.3 Hyphens, en dashes, and em dashes . . . 7
3.4 *Italics* . . . 8
3.5 **Boldface** . . . 8
3.6 Small caps . . . 8
3.7 Periods . . . 8
3.8 Ellipses . . . 9
3.9 Comments . . . 9

4 Inserting Greek text **9**

5 Essential structural elements **10**
5.1 Paper title, author, and date . . . 10
5.2 Headers . . . 10
5.3 Footnotes . . . 11
5.4 Environments . . . 11
5.4.1 Block quotes . . . 11
5.4.2 Itemized lists . . . 12
5.4.3 Enumerated lists . . . 13
5.4.4 Single spacing . . . 13
5.5 Page breaks . . . 13
5.6 Table of contents . . . 13

6 Citations and bibliography **14**
6.1 Maintaining a BibDesk bibliography . . . 14
6.2 Adding a bibliography . . . 15
6.3 How LaTeX bibliographies work . . . 15
6.4 Compiling your bibliography . . . 16
6.5 Citations with page numbers . . . 17
6.6 Including uncited works in your bibliography . . . 17

7 Global changes **17**
7.1 Font size . . . 17
7.2 Margins . . . 18
7.3 Line spacing . . . 18

7.4 Font face 18
7.5 Document metadata 19

8 Defining new commands **19**
8.1 New command abbreviations for Greek words 20
8.2 New command abbreviations for the works of Plato and Aristotle 20

9 Cross-references and hyperlinks **20**
9.1 Section cross-references 21
9.2 Page cross-references 22
9.3 Footnote cross-references 22
9.4 Hyperlinking 22
9.4.1 Hyperlinking to the internet 22

10 Continuing with LaTeX **23**

1 Getting started

First, let's see how a simple paragraph works. Basically, once you have your preamble, document title, and header specified, you can just start typing. Now let's say you want to start a new paragraph. This is not accomplished by pressing enter once and starting a new line. Rather, you need to press enter twice, leaving a blank line in between paragraphs.

If you look at the source code, you will see a blank line between this paragraph and the previous one, which gets renders as a new paragraph in your compiled document. If you're writing a short, simple essay, you can use this format exclusively. However, there's plenty more that LaTeX can do for you.

First, let's say you want to structure your document in sections. This is very easy with LaTeX; all you do is use the `\section` command. One such command was added at the beginning of this section; that's why it says "Getting started" centred above this paragraph in bold (centred and bolded is the style specified in the preamble for section titles). To use this command, simply start a new line and write:

```
\section{}
```

with your desired section name inside the curly brackets. Thus, to create a new section after this one, all we have to do is start a new line with this command, and…

2 Writing in sections

A new section! If you look at the source code, you'll see that by inserting the code line:

```
\section{Writing in sections}
```

LaTeX creates a new section when compiling the document, and automatically numbers it appropriately. LaTeX also easily handles subsections, which are accomplished, naturally enough, with the `\subsection` command, like so:

2.1 Subsections

Additionally, subsections can be further divided into subsubsections; simply use the intuitively (or rather, unimaginatively) named `\subsubsection` command:

2.1.1 Subsubsections

Don't try to go further than this, though; subsubsubsections would just be being greedy.

Unnumbered Sections

You can also create unnumbered sections by inserting an asterisk (`*`) between `\section` and `{}`, like so: `\section*{}`. Unnumbered sections will not be included in the table of contents, but they will be typeset with the normal section style.

Now let's get to more of the nuts and bolts of writing in LaTeX.

3 Essential writing tips

If you start trying to write a paper in LaTeX, you'll quickly come across things that are easy to do in a WYSIWYG (What You See Is What You Get) word processor like Word, but not as intuitive in LaTeX. Here is a quick rundown of these essential items.

3.1 Quotation marks: The easiest mistake to make

Probably the most notable difference between typing in LaTeX and typing in a WYSIWYG editor is how LaTeX handles quotation marks. In any other text editor, you're probably used to using the single (`'`) or double quotation mark (`"`) for both the start *and* end of quotations. LaTeX works differently: At the start of quotations, you need to use the grave accent mark (`` ` ``): the key under escape and to the left of 1 on standard keyboards. At the end of quotations, you use the apostrophe or standard single quotation mark (`'`). If you want double quotation marks, you use two grave accents at the beginning (`` `` ``) and two apostrophes at the end (`''`).

Doing this incorrectly will not stop your document from compiling; however, ”it will make it look bad” (like that) and thus show everyone that you don't know what you're doing. Proper quotations, on the other hand, “will look like this”, as they should. Check out the source code of this paragraph to see more clearly how this is done.

3.2 Indenting: Don't do it

Another easy mistake to make in LaTeX is to indent or tab in paragraphs, as you might be used to doing with your previous word processor. In LaTeX, there's no need to do this; LaTeX automatically handles indentation for you, knowing when and when not to indent. For instance, this paragraph, since it's the first of a new section, is not indented.

In contrast, this paragraph, being the second paragraph in this section, is indented, and as you'll see, you did not have to tell LaTeX anything for it to do this.

And in the rare case in which LaTeX is indenting a paragraph that you don't want to be indented, you can use the `\noindent` command on the line before the paragraph starts, as done with this paragraph.

Long story short: Never hit tab expecting it to indent your paragraph.

3.3 Hyphens, en dashes, and em dashes

Once of the best parts of LaTeX is how it handles and differentiates between hyphens, en dashes, and em dashes. You may not even be aware of the difference between these, so here's a brief run through:

A hyphen is what's used to separate elements in hyphenated words; it is inserted by typing a single hyphen (-), the key you're used to pressing for hyphens, like so: T-rex.

An en dash is slightly longer than a hyphen, and is used primarily to represent a range of values, such as page numbers. It is inserted into a LaTeX document by typing two hyphens (--), like so: 101–105.

An em dash is an even longer dash, used primarily to introduce parenthetical comments or

an abrupt transition to a new clause. It is inserted into a LaTeX document by typing three hyphens (`---`), like so: The dinosaur—later identified as a diplodocus—stomped through the forest.

3.4 Italics

You'll often want to italicize certain words or phrases in your document; this is accomplished with the `\emph{}` command. Simply place the text you want italicized inside the curly brackets after `\emph`. Everything inside the brackets *will be italicized, like this*.

(Also note that this italics command is more properly understood as an *emphasis* command, because it does not always italicize the given text, but rather emphasize it in contrast to the emphasis of its surrounding text. For example, the title of this section is actually emphasized. It appears non-italicized because section titles are by default italicized, so to emphasize this section title in contrast to its standard style, the text gets unitalicized. If that didn't make sense, don't worry about it.)

3.5 ***Boldface***

Boldface is less often used than italics, but in case you ever want to do it, it's accomplished with the `\textbf{}` command. Simply place the text you want bolded inside the curly brackets after `\textbf`, just like with the `\emph` command.

3.6 *Small caps*

Even less often, you may want to type something in small caps, like the above section title. This is accomplished with the `\textsc{}`" command, which works just like the previous two.

3.7 *Periods*

This does not come up too often, but sometimes you'll want to type a period that does not mark the end of a sentence, for instance, with abbreviations. LaTeX, however, assumes that every time you insert the period character (`.`), it marks the end of a sentence. Thus, if you just type "`Mrs. Smith`", the rendered space between "Mrs." and "Smith" will look a *little* too big, like this: Mrs. Smith.

Rather, whenever you want to insert a non-sentence-ending period, you should type ". ~" with no space after it instead, as in `Mrs.~Smith`. This, as you can see, comes out looking right: Mrs. Smith.

3.8 Ellipses

On a related note, if you ever want to write ellipses (that is, "…"), do *not* write ". . . ". Instead, use the special LaTeX command `\dots\`, simply by typing `\dots\` in the middle of your sentence … wherever you want the ellipses to be.

3.9 Comments

One of the neat things about writing in a programming language like LaTeX is that you can easily incorporate comments into your paper, which will be visible when viewing the source code but hidden when the document is compiled. These can be little reminders such as "Expand in footnote" or "Compare with *De Anima*" or whatever you so wish.

Comments are introduced in LaTeX by the percent sign (%). Everything following the percent sign will be automatically commented out for the rest of the line (that is, until the next line break). So, if you wish to insert a comment in the middle of a paragraph, simply place the comment where you want it, and then insert *one* line break after it (if you insert two line breaks, that will start a new paragraph). Check out the source code of this section; there's a super secret comment hidden right in the middle of this paragraph!

4 Inserting Greek text

One of LaTeX's major advantages for ancient philosophers is how easily and beautifully it incorporates Greek text into your document. Simply use the `{\gr }` command, placing the desired Greek text inside the curly brackets, leaving a space between `\gr` and the start of the text. Your Greek text can be entered manually or copied and pasted from the Thesaurus Linguae Graecae or some such similar database. Take a look at the source code to see how this text was rendered: {\gr Μῆνιν ἄειδε θεὰ Πηληϊάδεω Ἀχιλῆος}.

5 Essential structural elements

The preceding commands assisted you with various additions *within* paragraphs. However, there are also several things you need to do outside individual paragraphs.

5.1 Paper title, author, and date

Specifying the title of your paper and your name is usually a good idea, and LaTeX makes this super easy to do. Indeed, these are essential structural elements of a LaTeX document.

After the `\begin{document}` command (that is, after your preamble but before your paper text starts), you should include a title section, like so:

```
\title{}
\author{}
\date{}
```

These are about as straightforward as they look; simply insert your paper title inside the curly brackets next to `\title` and your name inside the curly brackets next to `\author`.

If you have a multi-line title and want to break it at a particular point, insert two backslashes (\\) inside your paper title, like this: `\title{First Line\\Second Line}`

Finally, whether you wish to include a date on your paper or not (I often don't), LaTeX requires you to include a date command of some sort, even if only to tell it not to include any date. If you wish to suppress the date, this is easily accomplished by writing the following: `\date{}`. If you wish to specify the date, you can either enter it manually inside the date command's curly brackets, or use LaTeX's handy-dandy today's date command, like so: `\date{\today}`; this will automatically print the date on which the document is compiled.

5.2 Headers

Having a document header usually looks good, and is also easy to do in LaTeX. After the title, you can include these commands to specify your left and right header.

```
\lhead{}
\rhead{}
```

Simply write in your left header text inside the `\lhead` curly brackets and your right header text inside the `\rhead` curly brackets. (I like to include the paper title in the left header and my name in the right header.)

5.3 Footnotes

Footnotes are an essential element to academic papers, and easily handled by LaTeX, which includes a handy `\footnote{}` command. I've shortened this to `\fn{}` to make things even easier. Simply type `\fn{}` at the point in the text where you want the footnote number to go, and then type your footnote inside the curly brackets.[1]

5.4 Environments

Many other structural elements are handled in LaTeX by special *environments*, which are basically just stretches of text with special formatting guidelines. All environments are introduced in the same way: start a new line, type `\begin{}` with the environment name inside the curly brackets, start a new line and type your text, and once all your text is written out, start a new line and type `\end{}` with the same environment name inside the curly brackets. Inside the `\begin` and `\end` commands, all text will receive the environment's special formatting; after the environment is closed, typed text returns back to normal. And handily, most of the environments you'll need are already defined for you in LaTeX and ready to be put to work. Let's review some of the most common.

5.4.1 Block quotes

Perhaps the most important environments for writing academic papers is block quotes. Fortunately, this is very simple with LaTeX, which includes a quote environment that is intuitively introduced with `\begin{quote}` and closed with `\end{quote}`.

[1] Like this (see source code).

However, if you're like me, you probably dislike typing any more characters than necessary, so what I've done is shortened these commands to the simpler `\bq` and `\eq`, respectively. (Defining new commands will be discussed more below in §8.) Thus, to introduce a block quote, simply start a new line, type `\bq`, press enter once, and then start typing your block quote, like so:

> And now we're in a quote environment. To end your quote, finish typing it, press enter once, type `\eq`, and then press enter once more and you'll be back in a normal paragraph.

Like this. And, if you wish to start a new paragraph after a block quote, simply press enter twice, leaving a blank line after the `\eq` command, before starting your next paragraph.

5.4.2 Itemized lists

Sometimes you'll want to insert an itemized (that is, a bulleted, non-numbered) list. This is, yet again, very easily accomplished with LaTeX, which includes an `itemize` environment, introduced and closed with the `\begin{itemize}` and `\end{itemize}` commands. As before, I've created abbreviated new commands, `\bi` and `\ei`, to make things quicker. Inside this environment, you must introduce each new list item with the `\item` command, followed by a space. Here's an example (look at the source code to get a better idea of how this works):

- First item
- Second item
    - You can also create nested itemized lists, by simply starting a new `itemize` environment inside the first one
    - (Just don't forget to close this environment, too)
- Third item

And that's all there is to it.

5.4.3 Enumerated lists

Enumerated (that is, numbered) lists work in much the same way, except the environment is `enumerate` rather than `itemize`, which I've abbreviated to `\be` and `\ee`.

1. First item
2. Second item
    (a) Nesting also works with enumerated lists
    (b) Second nested item
3. Third item

5.4.4 Single spacing

Occasionally, you may want a section or a paragraph to be single spaced, in contrast to the rest of your document which is double spaced (if you want your *entire* document to be single spaced, skip below to §7.3). This is easily accomplished with the `singlespace` environment, which I've abbreviated to `\bs` and `\es`. Introduce this just like the other environments…

And voilà—single spaced text! Actually, it doesn't look single spaced until I've typed a little more, so I guess I'll type a little more to show you what I mean.

Once you leave the environment, your next paragraph will be returned to double spacing, just as before.

5.5 Page breaks

Most of the time you won't have to worry about making page breaks when working with LaTeX, but occasionally you will want to manually insert one. This is done by simply entering the command `\pagebreak` on a new line.

5.6 Table of contents

If you want to insert a table of contents into your document, simply add the line:

```
\tableofcontents
```

where you want your table of contents to appear. All chapters, sections, and subsections will automatically be listed in the table of contents, alongside their respective page numbers. Also, I find it helpful to specify that the table of contents be single spaced and on its own page, which can be easily accomplished with these five lines of code:

```
\pagebreak
\bs
\tableofcontents
\es
\pagebreak
```

6 Citations and bibliography

Bibliographical citations are yet another major part of any academic paper, and thankfully one that LaTeX makes much easier, once you've got the basic commands down. For the purposes of this guide, I will be assuming that you have already successfully installed BibDesk. This section will now show you how to maintain a BibDesk bibliography and then incorporate citations and a references list into your document.

6.1 Maintaining a BibDesk bibliography

BibDesk works by keeping a database of all the works you ever care to reference. You can add works to this database either manually or from the internet.

To add a new work manually, click on the big green plus sign for "New", which will open a dialog box. Select the relevant kind of work you're entering from the drop-down menu (e.g. "book", "article"), and then insert the relevant data ("Title", "Author", etc.). Lastly, insert a "citation key" for this work. A citation key is the unique string of letters by which you'll refer to the work in the document; it cannot be shared by any other entry. If you leave this field blank, BibDesk with autogenerate a citation key for you.

To add a new work from the internet, if it's a book, go to the Searches menu bar and select Library of Congress and then search for your work there and then click Import. If it's an article,

go to the Bookworks menu bar and select Google Scholar. Change the settings in Google Scholar to show links to import citations into BibTex under "Bibliography manager". Then search for your article in Google Scholar and click "Import into BibTeX". You may want to check the entries they import for the occasional error.

Save your BibTex database. Now whenever you wish to reference a work in a LaTeX document, search your BibDesk database for the work, find its citation key, and then use that citation key as specified below.

6.2 Adding a bibliography

To add a list of cited references to your document, simply insert these lines of code at the desired point (usually at the very end):

```
\bibliography{BibliographyName}{}
\bibliographystyle{BibliographyStyleFileName}
```

The first line tells LaTeX where to locate your bibliography (.bib) file. This is the file you maintain in BibDesk. I have included a sample file in the folder with this document (toybiblio.bib).

The second line tells LaTeX where to find your bibliography style (.sty) file. I have included with this document a style file which I created and really like (biblio.bst). Creating your own style file is possible but hard to explain, so I won't get into it here.

So, don't forget: To get your bibliography to work in LaTeX you will need *two* files, a .bib and a .bst file. (They don't have to placed in the same folder as your document, but this makes them easier for LaTeX to find.)

6.3 How LaTeX bibliographies work

To compile your bibliography, LaTeX searches your document for all the cited works and then creates a bibliography from just those references, automatically. That means you don't have to worry about keeping track of the works you cite, or figuring out how to correctly format their entries: LaTeX does all the work for you.

Making a citation is easy, too, and accomplished with the `\citet{}` command. Simply insert your specially specified "citation key" for the work you wish to reference (the citation key you defined in BibDesk; check your BibDesk library if you forget a work's citation key). For example, if I wanted to cite Plato's *Republic*, and my citation key for the work was "PlatoRepublic", my citation command would look like this:

```
\citet{PlatoRepublic}
```

However, in order for this command to compile properly, we need to learn how to properly compile your bibliography.

6.4 Compiling your bibliography

Very important! To get your bibliography to compile correctly, you cannot just compile your document like usual. In addition, you must run a BibTeX compile. In TeXShop, this is accomplished simply by going to the Typeset menu bar and selecting BibTeX, or by pressing Shift+Command+B.

You should run this BibTeX compile only *after* you have already compiled your document once normally (that is, in LaTeX). After this, run a BibTeX compile, and then *run a normal LaTeX compile again.* At this point, your cited references should now be incorporated into the document, and your bibliography should be printed with all cited works. If they are still not showing up, try running a normal LaTeX compile again. And then maybe one more time. If they're still not showing up, run one more BibTeX compile, and then another LaTeX compile. I know it sounds silly, but the majority of the time this repetitive process will actually fix the issue.

Now we can actually try getting a citation to compile correctly in this document. In the next paragraph I've included the `\citet` command introduced above. If you've compiled this PDF yourself from LaTeX, this citation probably hasn't compiled correctly, as you haven't yet done a BibTex compile. (When citations aren't compiled correctly, LaTeX prints out "**??**" instead of the appropriate reference.) Now try running a BibTex compile, followed by another LaTeX compile (and perhaps another, as the case may be). After this, the following citation should work, and should be noted in the bibliography at the end of this document.

Does this citation work (Y/N): Plato (350 BCE).

6.5 Citations with page numbers

The `\citet` command normally just takes a citation key, but sometimes you want to specify the page number of the citation as well. This is easily accomplished by added a few extra characters to your command, like so:

```
\citet[Page#]{CitationKey}
```

So, once you've got your document compiling correctly, the following text should appear with a page number: Plato (350 BCE, 511b).

6.6 Including uncited works in your bibliography

Occasionally, you will want to include certain works in your bibliography which are never actually cited. This is easily accomplished by including, somewhere in your document, a command of the form:

```
\nocite{CitationKey}
```

and entering the relevant citation key inside the curly brackets. This work will now be included in your bibliography.

7 Global changes

Another major advantage of LaTeX is how easy it is to make global changes to your document, switching the font size, face, margins, or line spacing with just a single line of code. Here's all you need to do…

7.1 Font size

If you look at this document's preamble, the first thing you'll see listed is a line that reads:

`\documentclass[12pt]{article}`.

If you wish to change the font size of the entire document, simply change the '12pt' to '10pt' or '11pt' (no other options are available).

7.2 Margins

Further down in the preamble you'll see a line that reads:

```
\geometry{verbose,tmargin=1in,bmargin=1in,lmargin=1in,rmargin=1in}
```

If you wish to change the margins of the entire document, simply change these numbers to different values.

7.3 Line spacing

Yet further down in the preamble you'll see a line that reads:

```
\usepackage{setspace}
```

If you wish to change the line spacing of the entire document, on the line below this specify either:

`\singlespacing` (for single spacing)
`\onehalfspacing` (for 1.5 spacing), or
`\doublespacing` (for double spacing)

7.4 Font face

I wouldn't recommend changing the font face of your document, since Linux Libertine is so good, but if you must, find the line in the preamble which reads:

```
\setmainfont[Mapping=tex-text]{Linux Libertine O}
```

and change the text inside the curly brackets to your desired font face.

7.5 Document metadata

In lieu of any better place to put it, I should also mention that LaTeX allows you to automatically edit the "metadata" of your document: special data coded into your PDF document, such as your paper's title, author, subject, and keywords. (This makes your document more easily searchable.) To edit the metadata of your document, scroll down to the bottom of the preamble, right before the `\begin{document}` line and find the section that begins `\hypersetup{`. After this, simply edit the text inside the brackets of `\pdftitle={}`, `\pdfauthor={}`, and so on. LaTeX will take care of the rest.

8 Defining new commands

Defining new commands is a super handy part of working with LaTeX. Let's say you're writing a paper on Neoplatonic metaphysics. You might very well get tired of typing out the phrase "the unhypothetical first principle of all" each time you want to refer to The One. However, with LaTeX you can simply define a new command in the preamble such that all you have to type is, say, `\ufp`. To do this, simply insert a new line in the preamble of the form:

```
\newcommand\CommandName{CommandText}
```

For example, if we wanted to create the new command mentioned above, we would add:

```
\newcommand\ufp{the unhypothetical first principle of all}.
```

Important! If you're going to be defining new commands, be aware that if you want there to be a space after a new command, you must end the command with a backslash. That is, most of the time you'll be writing things like

```
What Plotinus actually means by \ufp\ is unclear
```

If, however, a punctuation mark of some sort follows the new command, do not add a backslash to the end, as in

```
The One, or \ufp, is the ultimate principle of Neoplatonic metaphysics
```

8.1 New command abbreviations for Greek words

A really handy feature of new commands is the ability to easily incorporate Greek words directly into your document. This is especially useful if you are continually referencing a certain Greek word in your paper, for example, *φαντασία*. Normally, to get this to show up you'd write a bit of code like this: `{\gr φαντασία}`. However, if you define a new command like this:

```
\newcommand\phantasia{{\gr φαντασία}}
```

then each time you want to write *φαντασία* in your document, all you'll have to do is type your new command `\phantasia`. Just remember the important proviso above.

8.2 New command abbreviations for the works of Plato and Aristotle

I've also included in this document's preamble a list of handy abbreviations for the titles of all of Plato's dialogues and Aristotle's treatises. This means that you can simply type, for example, `\rep` and will get *Republic*. These work titles are emphasized in the new command, so they will automatically be italicized when they appear in your main text. Consult the end of the preamble for a complete list of abbreviations.

9 Cross-references and hyperlinks

One of the things that LaTeX handles extremely well is intratextual cross-references, that is, references to other parts of your documents. Say I want to reference a previous section of this document, say, the section in which I introduced new commands for Greek words. With LaTeX, instead of entering this section number in manually, I can reference it with a line of code, like so:

```
\ref{New commands Greek}
```

When this is compiled, LaTeX renders this as 8.1, because that is the relevant section number. Most significantly, this means that if I go back and change around the order of my sections, LaTeX will automatically correct these cross-references to the appropriate numbers.

Cross-references actually require two lines of code: first, a line like the one shown above, which tells LaTeX to print out some cross-reference; and second, a label line of code, which tells LaTeX where to look for the cross-reference. Labels look like this:

```
\label{New commands Greek}
```

and are placed wherever you want the cross-reference to be. Thus, if you look at this document's source code and scroll up to the section on creating new commands for Greek words, you will see a line of code like above right below the `subsection` command.

A final note on cross-references, before we see some further examples: Sometimes, LaTeX will print out **??** instead of the appropriate cross-reference number. This means LaTeX can't find the relevant label. Occasionally this will be because you've forgotten to include or mistyped the label, but oftentimes all you need to do to fix this is to compile your document a second time.

9.1 Section cross-references

We already saw how to reference a section in the above example, but here's one further tip: In order to keep everything in your document clean and more precise, it is often helpful to label your labels and cross-references with the *type* of thing you're referencing. Thus, instead of just typing:

```
\label{The awesome section}, and
\ref{The awesome section}
```

you may want to type instead:

```
\label{sec:The awesome section}, and
\ref{sec:The awesome section}
```

This will remind you that what you're referencing is a section, as opposed to some other element.

Note that section cross-references work exactly the same way regardless of whether you are referencing a section, a subsection, or a subsubsection.

9.2 Page cross-references

To reference a page of the document, create a label as before, but instead of using the `\ref` command, use `\pageref` instead. Thus, if we wanted to reference which page the section on new commands in Greek is on, we would type:

```
\pageref{New commands Greek}
```

When compiled, this gets rendered, appropriately, as: 20.

9.3 Footnote cross-references

To reference a footnote number, create a label *within* your footnote (preferably with the `ftn:` abbreviation), like so:

```
\footnote{A footnote.\label{ftn:Footnote}}
```

then simply reference it like any other cross-reference. Thus, if I wanted to refer back to the footnote on page 11, I could simply type:

```
\ref{ftn:Footnote}
```

which LaTeX would render appropriately as: 1

9.4 Hyperlinking

Finally, one of the really handy things LaTeX can do with cross-references is hyperlink them—that is, make it such that you can click on the reference numbers in the PDF and then automatically be brought to the referenced element. This feature has already been included in the document's preamble, and there's nothing more you need to do to make it work. Try it out with some of the previous cross-references!

9.4.1 Hyperlinking to the internet

You can also hyperlink text in your document—namely, email addresses or web links—to the internet, such that a user can click on the text and automatically be brought to the referenced

webpage or email address. This is accomplished with the `\href` commands, which takes two arguments (that is, which has a *pair* of curly-bracketed items after it): first, the internet address you want to link to; and second, the text you want to be clickable in the document. An example should clarify this:

```
\href{http://untilweeatagain.wordpress.com}{this author's food blog}
```

When this is inserted into the document it gets compiled as: this author's food blog, and you can click on that string of words to be brought directly to the relevant website.

To link to an email address, a similar formula is followed, except you add a `mailto:` command at the beginning of the first argument, like so:

```
\href{mailto:williecostello@gmail.com}{this author's email address}
```

And this, when compiled, looks like this: williecostello@gmail.com.

10 Continuing with LaTeX

And so we have reached the end of this guide. Hopefully I have managed to address the majority of issues you will come across with LaTeX, but I am sure I have missed a few, or that you will eventually find the need to do something even I don't know how to do. In cases like these, your best bet is to Google your query; practically every question I've ever had about LaTeX has been answered by this method. However, please do drop me a note at williecostello@gmail.com about your issue as well, as I would like to incorporate any oversights into future versions of this guide.

Thanks for reading!

References

Plato, 350 BCE. *The Republic*. The Academy Press.