

An Introduction to L^AT_EXfor Foundation Year Students

Dr Thomas Carroll

March 3, 2022

Introduction

L^AT_EX is a document preparation system that is fundamentally different to anything that you are likely to have seen before. It's used worldwide by publishers, academics and scientists and it happens to be very good for writing CVs too.

Getting Started

1. Boot in to Linux
2. Log in and start a graphical desktop session
3. Open a terminal
4. If none exists, create the directory `/ex3`
5. Navigate in the terminal to your `ex3` directory
6. Now start up your favourite text editor. This might be *nano*, *nedit*, *gedit*, or some other text editor program.

You are now good to go! You should ensure that you complete all exercises from this lab within the `ex3` directory. You may find it easier to put each exercise in an additional subdirectory within `ex3`, but this is your choice. If you choose to do this, you must not forget to navigate to the appropriate subdirectory in your terminal before running any commands.

1 A Closer Look

In your editor, enter the following text (it's important that you enter it exactly as it is below for now), and then save it as a file called `fire-and-ice.tex` (in case you're wondering, it's the first line from a short poem called 'Fire and Ice' by American poet Robert Frost (R. Frost, 1924); the whole poem is on the Poetry Foundation website if you're interested.)

```
\documentclass[a4paper]{article}
\begin{document}
    Some say the world will end in fire
\end{document}
```

Next, run the command `pdflatex fire-and-ice.tex` (for now it's sufficient to say that `pdflatex` invokes LATEX to create a PDF file as the output; the full story is a bit more complicated.)

LATEX will process your file, and print out a surprisingly large amount of text into the shell window as it does so. If all is well, the last two lines printed out will be something like (**but maybe not identical to**):

Output written on fire-and-ice.pdf (1 page,11853 Transcript written on fire-and-ice.log.

...after which LATEX will return you to the command prompt. If you've made a mistake typing in the file, LATEX will stop part way through processing your work, and display an error (possibly, quite an obscure one) and show a '?' prompt. The best thing to do at this stage is to press [ctrl]d, or x, to tell LATEX to give up trying to process your file. Then fix the mistake, and try again. If you now list the directory contents, you should see that a file `fire-and-ice.pdf` has been created (along with two other files, `fire-and-ice.aux` and `fire-and-ice.log` both of which we will ignore for now). Use one of the PDF readers (such as `evince` - you can try just double clicking the PDF file in a file explorer to open in a PDF Reader) to look at the contents of `fire-and-ice.pdf`. Admittedly, it's not the most exciting result; if everything has worked properly so far, you should see a single-page document with the words 'Some say the world will end in fire,' a little way down from the top of the page. If you have been carried away with the sheer excitement and joy of learning to use LATEX and improvised the text rather than typing what we suggested, then ofcourse you will see something completely different!

But now look more closely. A lot more closely. . . really zoom in. See anything interesting? Even for this very short document, L^AT_EX has taken a fairly sophisticated typographical decision on your behalf and ligated (which is typography-speak for 'joined') the 'f' and the 'i' in the word 'fire'; the spacing between the characters has been subtly altered so that the dot above

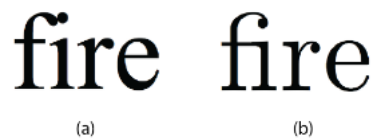


Figure 1: The word ‘fire’ typeset by (a) Microsoft Word 2011 without ligated characters, and (b) by L^AT_EX showing the ligation of the characters ‘f’ and ‘i’.

the ‘i’ and the blob on the end of the curvy bit (the ‘arc of the stem’, if we’re being formal) on the top of the ‘f’ join together to form a single shape called a **ligature**. Why? Because having two blobs side by side simply looks a bit clumsy as you can see in Figure 1. To make this decision, L^AT_EX has had to know a lot about the fonts being used (not all ‘f’s in all fonts have a blobby end; not all ‘i’s in all fonts have a round dot that would merge nicely with the blobby bit on the end of an ‘f’, and so on).

There’s a reasonable chance that you’ve never heard of ligated characters before— it is, after all, a fairly specialist thing. And there are hundreds of other obscure but important ‘rules’ of typography that go to make professionally typeset documents look good. Look up ‘kerning’, ‘combining characters’ and ‘serif’ on Wikipedia for starters. And if you’re really keen, Donald Knuth’s fascinating and compendious book ‘Digital Typography’ (D. Knuth, 1999) has an entire chapter dedicated to the joys of the letter ‘S’. Probably best not to bring this subject up at this time.

These things might not seem obvious or hugely important, but collectively and subliminally they make the difference between something that looks just-about-acceptable (like most things written using wordprocessors) and things that stand out as looking really professional (you might want to remember this when it comes to putting your CV together.)

Anyway, it really is important that you use exactly the text we suggested. Really, it is. So please change it now if needed.

That’s better.

Thanks.

But that’s enough about the typography of the word ‘fire’ for now. Let’s put together something more substantial.

2 A larger example

Create yourself a new file, called `sections.tex` by copying `fire-and-ice.tex`. In `sections.tex`, replace the line of poetry by copying and pasting the following text from Figure (which contains the first three paragraphs of Dickens’

novel, 'Bleak House' (C. Dickens, 1852)):

LONDON. Michaelmas Term lately over, and the Lord Chancellor sitting in Lincoln's Inn Hall. Implacable November weather. As much mud in the streets as if the waters had but newly retired from the face of the earth, and it would not be wonderful to meet a Megalosaurus, forty feet long or so, waddling like an elephantine lizard up Holborn Hill. Smoke lowering down from chimney-pots, making a soft black drizzle, with flakes of soot in it as big as full-grown snow-flakes — gone into mourning, one might imagine, for the death of the sun. Dogs, undistinguishable in mire. Horses, scarcely better; splashed to their very blinkers. Foot passengers, jostling one another's umbrellas in a general infection of ill-temper, and losing their foot-hold at streetcorners, where tens of thousands of other foot passengers have been slipping and sliding since the day broke (if the day ever broke), adding new deposits to the crust upon crust of mud, sticking at those points tenaciously to the pavement, and accumulating at compound interest.

Fog everywhere. Fog up the river, where it flows among green aits and meadows; fog down the river, where it rolls defiled among the tiers of shipping and the waterside pollutions of a great (and dirty) city. Fog on the Essex marshes, fog on the Kentish heights. Fog creeping into the cabooses of collier-brigs; fog lying out on the yards, and hovering in the rigging of great ships; fog drooping on the gunwales of barges and small boats. Fog in the eyes and throats of ancient Greenwich pensioners, wheezing by the firesides of their wards; fog in the stem and bowl of the afternoon pipe of the wrathful skipper, down in his close cabin; fog cruelly pinching the toes and fingers of his shivering little 'prentice boy on deck. Chance people on the bridges peeping over the parapets into a nether sky of fog, with fog all round them, as if they were up in a balloon, and hanging in the misty clouds.

Gas looming through the fog in divers places in the streets, much as the sun may, from the spongey fields, be seen to loom by husbandman and ploughboy. Most of the shops lighted two hours before their time — as the gas seems to know, for it has a haggard and unwilling look.

Edit the text to make sure that there is a blank line between each of the three paragraphs (it's not enough that each paragraph starts on a new

line of its own, there has to be an empty line in between them). That's how paragraphs are distinguished in LATEX (in fact, it doesn't matter how many blank lines you put as long as there is at least one. . . for the most part LATEX ignores spare whitespace).

After the 'Bleak House' text, type a paragraph of gibberish by randomly hitting keys on the keyboard, and putting spaces in to make 'words' in the text file. You should end up with something like the following, but about three times as long (don't cut and paste this from here. . . it's important that you actually create some unique gibberish yourself!):

Oisjdf oqweqwe oi soijs hbweo kbsd oisjdf oijqwknpiouh ius-
bdfspb sifuhgywqeb usgweijf blimqwoq oieuerwefwiu aokqjw uioshi-
ufds qiqks odsubfi psdiweneq.

Be careful at this stage to only include letters (uppercase and lowercase), commas, full-stops and perhaps numbers in your gibberish text: if you use any other symbols, you may upset LATEX.

Next go to the website *www.lipsum.com* and follow the instructions to generate yourself three paragraphs of what's called Lorem Ipsum text. Copy and paste that text into your LATEX document too (don't worry about what it means at this stage), again making sure that you have a blank line between the paragraphs.

Run `pdflatex` and look at the resulting `sections.pdf`. Again, at first glance, there's nothing hugely exciting going on here: the text from your `.tex` file has been assembled into a (probably) two page PDF document. But in fact quite a few things have happened:

- The blank lines that separate the paragraphs in your source file are gone, and instead LATEX has indented the first line of each paragraph. There's nothing profound here about the decision to indent paragraphs rather than (say) to leave a gap between them (so called block paragraphs); it's just LATEX's default paragraph style, and can be changed easily enough.³
- There are page numbers at the bottom of each page (except page 1). This is a very sensible default, allowing you in meetings to ask people to 'turn to page 4' and so on (it's a bit odd, and a source of some annoyance, that most wordprocessors don't do this by default too). But you can easily switch page numbers off, change their style or move them elsewhere if you need to.
- The text is fully justified so that it's flush on both the left and right hand edges.

- Certain words that fall at the end of lines have been hyphenated, even if they weren't originally split by hyphens in the source text.

These last two points about justification and hyphenation might seem utterly trivial, but getting these factors right turns out to be an important part of making readable documents; readability is not just about which font you choose, but about how the characters and spaces are distributed on each line. It's something that you will have seen happen so often in professionally typeset materials that you'll have taken it for granted, and you may not have spotted that it doesn't happen in quite the same way when you use a wordprocessor. The principle behind justifying text seems easy enough at first glance: you simply have to take any 'left over' space that would be at the end of a line, and redistribute it between the words in some way to 'space them out' a bit more so that they take up all the available horizontal room. But it turns out to be much harder than it seems, and all the simple ways you might think of for doing this on a per-line basis give results that are visually quite unsatisfactory, with nasty patterns made up of neighbouring whitespace appearing in the layout and ruining everything. The added complication here is that to do the optimum job of shuffling the words round on their lines, you really need the flexibility of being able to hyphenate words to give you extra space on lines that would otherwise be tightly packed with characters. But if you hyphenate words badly, by breaking them in inappropriate places, or do it too often then that makes the text hard to read too. It's all deeply inconvenient and inter-related. The best solution to this aesthetic conundrum so far was developed by Knuth and Plass, 1981 and later improved by Frank Liang during his PhD research (F. M. Liang, 1983), and it's a technique that tries to optimise the layout of whole paragraphs rather than individual lines. The algorithm takes into account all manner of different things: language-specific character patterns, the number of consecutive lines that end with hyphens, the word 'tightness' on each line. . . and many others. The details don't really matter here, but the really important insight is that you get better results by working on a bigger chunk of the document (in this case a paragraph at a time) rather than trying to independently optimise lots of small bits (say, lines) and hoping that the result of joining them all together turns out nicely. This idea of optimising things globally versus locally is something you'll see many times throughout your Computer Science degree; watch out for it in all the algorithms courses.

This highlights a fundamental difference between the WYSIWYG paradigm and the way that documents are produced using LATEX. As we've seen, small changes down at the level of individual characters can have fairly large knock-on consequences for the optimal layout of a document: changing one

letter for another affects the length of a line, which in turn affects the hyphenation options, which then affects the length of a paragraph, which means that the best place for a figure that was on page 99 might now be on page 100, but that means that all references to ‘page 99’ now need a bit more room to account for the extra digit which means that the some line lengths have changed which means. . . well, you get the idea. It’s more complicated than it first seems. LATEX gets to see the whole of your document at once, and can therefore make decisions about how best to hyphenate lines, position figures or whatever else it likes before it shows you the results; so it can make solutions that are optimal for the whole document, as well as looking at details such as ligated characters. Wordprocessors can’t afford to do this since it would be deeply distracting if every time you typed a character, the whole layout of the document flapped around in front of your eyes. Wordprocessors can only sensibly make little optimisations at a local (probably per-line level). It’s not that wordprocessors are rubbish, or that the people that programmed them were lazy: it’s a fundamentally different way of working, and knowing the pros and cons of the different approaches will help you pick the right tool for the job.

You may have noticed that actual text in your document (and in fact, this one) takes up a (perhaps) surprisingly narrow horizontal region of the page. This last ‘feature’ might seem odd at first, but it illustrates another important typographical point that you may never have thought of before: reading long lines of text is hard, because your eyes find it hard to move in a completely straight line as they scan from left to right. So the longer the line of words, the more you have to work to keep your eyes from accidentally drifting up or down onto a neighbouring line (perhaps you’ve experienced the phenomenon of accidentally re-reading the same line over and over as your eyes get tired). It may never have occurred to you, but most books have relatively short lines of text; and those that use large pages often split text into several columns so that individual lines are kept relatively short. The widely trusted book ‘The Elements of Typographic Style’ (R. Bringhurst, 1992) makes the following assertion:

‘Anything from 45 to 75 characters is widely regarded as a satisfactory length of line for a singlecolumn page set in a serified text face in a text size. The 66-character line (counting both letters and spaces) is widely regarded as ideal.’

LATEX knows about this issue, so made a sensible default decision for you. You may have noticed that there is a subtle difference in appearance of this chapter from previous ones: the lines are shorter and paragraphs are

indented. We wanted to use the default LATEX layout for this chapter, but changed it for earlier chapters in an effort to save paper.

There's one last unusual thing to note: your paragraph of gibberish text in between the real words from Bleak House and the Lorem Ipsum almost certainly stands out and looks really quite odd (it might even have confused LATEX's hyphenation rules enough to have ended up with one or more lines sticking out proud of the right hand margin). Let your eyes relax so that your vision blurs for a moment: you can still probably spot that the gibberish paragraph looks out of place. It's quite likely that you even found creating the gibberish unexpectedly hard. The interesting thing here is that as humans, we spend so much time reading and writing 'proper' natural language that we become attuned to the patterns of letters words and the underlying 'rhythm' of our written language that its very hard to artificially reproduce something that looks plausible; the frequency of the letters you chose by randomly hitting keys, and the length and patterns of the words you put in place most likely don't match the natural patterns of real languages, so they just look plain wrong. Curiously the Lorem Ipsum text that you got from the website probably doesn't look so bad, in spite of it not being made up of real words. But that's not surprising since Lorem Ipsum is specifically created to be used as a placeholder for real text when typesetting. While the words are meaningless Latin-like things, their length, and the distribution of characters and word lengths are chosen to mimic patterns in real language.

Next let's give our document some structure by dividing it up into sections. On the line before Bleak House starts, add the command

```
\section{Bleak House}
```

making sure it appears on a line of its own. Then just before your paragraph of gibberish, put a similar instruction (again, on its own line), and then create a final section before the Lorem Ipsum. Rerun pdf_latex and look at the results. You should see the section headings appear in the appropriate places in the PDF, automatically having been given section numbers. Experiment by adding in a couple of sub-sections towards the end of the document using the `\subsection` command, which like the `\section` command above takes as a parameter some text contained in curly brackets. Now, at the top of your document after the `\begin{document}` line and before the command defining the first section, add the following two commands, each on their own line:

```
\tableofcontents  
\newpage
```

Re-run pdf_latex twice, and check that the effect is what you'd expect. You

need to process the source twice because on the first run, LATEX gathers and stores information about what to put in the table of contents, and only creates it on the second run.

2.1 Common Formatting Features

Before we give you some more exercises to do, here is a brief introduction to some commonly used features of LATEX.

Enumerated and bulleted lists Creating bulleted or numbered lists in LATEX is straightforward, and is done like this:

```
\begin{enumerate}
  \item My first item
  \item My second item
  \item The last thing on my list
\end{enumerate}
```

You must make sure that you both begin and end your list, and that each item is terminated by a newline. Changing `enumerate` to `itemize` gives you a bulleted rather than numbered list (note the American spelling of ‘itemize’).

Changing the appearance of text Text can be typeset in bold font using the `\textbf`, in italics using `\textit` and in a fixed-width font using `\texttt` as in the example below:

Some `\texttt{say}` the `\textbf{world}` will end in `\textit{fire}`.
Give it a try, and see what happens!

Cross referencing LATEX allows you to cross-reference almost anything in the document, which includes sections, sub-sections and figures. To use the cross-referencing feature you simply insert the command `\label{mymarker}` at the point in the document you want to refer to, and then use the command `\ref{mymarker}` when you want to use the reference.

You replace the text `mymarker` with **something more meaningful**.

An important tip here is to call the marker something that refers to the content of that part of the document, and to avoid the temptation to use numbers (so don’t call your marker `section1`, instead call it `introduction`; if you reorder your document, it’s still likely to be the introduction, but it may no longer be Section 1).

For example, give this one a go:

```

\section{\label{welcomeweek} Reflection on Welcome Week}
I spent most of welcome week getting to know my way around campus.
I should have made more effort to societies I think.
I must remember to get to the Students Union and take a look around
there.
\subsection{Things I enjoyed}
Everything so far has been wonderful, with the caveat mentioned
in Section \ref{welcomeweek}.
\subsection{Things I didnt enjoy}
Ive discovered that I dont like soggy chips.

```

Figures and pictures Including figures and pictures in your document is easy. You do it like this:

```

\begin{figure}
\centering
\includegraphics[width=10cm]{dutchCheese.png}
\caption{Some men, possibly Dutch, looking at cheeses
(from Wikipedia article on Cheese).}
\label{figure:menandcheeses}
\end{figure}

```



Figure 2: Some men, possibly Dutch, looking at cheeses (from Wikipedia article on Cheese).

which is the code we used to create Figure 2. You can probably see what's happening. The command `\includegraphics{}` gets your image, which can

be PDF, PNG, JPG, GIF or PostScript. The `\begin{figure}...\end{figure}` code ‘wraps’ up whatever picture you’re including, and allows LATEX to treat it as an unbreakable floating thing that it will position for you as best it can in the document, while maintaining an overall nice typographical layout. This floating of figures can sometimes result in the figure ending up in a place you didn’t expect, but in most cases LATEX will make the most sensible choice. It’s possible to employ finer control over figure placement, but that’s beyond the scope of this introduction. The `\includegraphics` command is not built in to core LATEX but is in an additional package, which needs to be explicitly loaded. We load this package by using the command `\usepackage{graphicx}` in the document preamble, i.e. after the `\documentclass` command, but before `\begin{document}`. There are hundreds of such packages that extend LATEX’s functionality.

3 Including Images

Create a document `image.tex`, that contains some text (maybe from ‘Lorem Ipsum’), together with a figure containing an image of your choice (I ahve included the cheese judging image in the blackboard task if you want it) . There should also be a reference to the figure in the text. You should first use the program `convert` (check its man page) to create a PDF version of the image. `conve`

4 A reflection on your studies

Your first proper LATEX document will be a very brief reflection on your studies so far. Write a hand-crafted LATEX document called `courses-reflection.tex`. It will have the following structure.

1. A table of contents.
2. A brief paragraph saying what the document is about.
3. A section, appropriately titled, for one of your course units, containing:
 - a brief paragraph saying what the unit is about.
 - A sub-section, appropriately titled, containing an enumeration of the three things you like the most about the unit.
 - A sub-section, appropriately titled, containing an enumeration of the three things you like the least about the unit.

After completing and successfully compiling and viewing your document, you should spell-check it as follows: `ispell courses-reflection.tex`

5 Next Steps

LATEX is a very powerful tool, and it does take a little getting used to. But it's not as scary as it might initially seem, and you should find that it doesn't take long to learn the common commands and techniques to make your documents well-structured and readable. As you become more comfortable with the basics and understand the principles of typesetting, finding out how to do more complex and sophisticated things is easy—there are hundreds of tutorials on the web. Don't get carried away though, and use the features judiciously: just because you've found a cool new command doesn't mean that you should use it just to show off your LATEX skills: content and structure always trumps typographic bragging.

For example, footnotes¹ are, for some reason, a really tempting feature to use in LATEX. Footnotes were originally a means for mechanical typesetters to insert comments, corrections or additions to existing documents without having to adjust the layout of a whole page, and of course in digital typography this is no longer necessary. They can justifiably be used for citations (though it's not common to do so these days) and URLs if you don't want these intruding into the body of your text. And in some very very rare cases, they can be used as an aside to the main flow of the text (which is what has been done in this document). But this should be used sparingly. If something is important to say, put it in the main body of your text. If it's not, then perhaps you should think about leaving it out completely, or find some other way of including it.

¹like this