

A Beginner's Guide to the Documentation of L^AT_EX

C. G. P. J.

November 18, 2022

For Nicole, that we may work together.

Contents

Contents	3
1 Introduction	4
2 Some Ideas Surrounding L^AT_EX	5
2.1 The Structure of L ^A T _E X	5
3 Documentation	11
3.1 Official Documentation, or Just Offal?	11
3.2 A couple of Favourite Documentations	12
3.3 Summary So Far	14
3.4 Other Documentations	14
4 Official Sites and Resources	17
4.1 The Three Main Sites	17
4.2 Bringing It Together	18
5 Online Help	19

Chapter 1

Introduction

I decided to compile a guide to the documentation of \LaTeX . Such a guide is needed, because there are many good works that can help a person to become erudite in \LaTeX , but they are not always obvious; and in every place where a good work is not found, a less satisfactory explanation stands.

This guide is *not* an introduction to LaTeX; such things have already been written. This booklet instead attempts to help people by directing them to documentations and resources that I wish I had used much earlier in my journey. What more I have added is for the purpose of internal consistency.

The guide is accompanied by (and is a part of) a repository, which is available on github. The name of the repository is “A Beginner Guide To LaTeX Resources”.

<https://github.com/chrisgpj/A-Beginner-Guide-To-LaTeX-Resources>

A note on the text

In many places throughout this guide, I have used “LaTeX”, rather than “ \LaTeX ”. While “ \LaTeX ” is the official rendering, “LaTeX” is official within plain-text environments, and I prefer this in prose.

Chapter 2

Some Ideas Surrounding L^AT_EX

Without good documentation, there is no such thing as a good system, or program, or whatever the *thing* is. It is my view that good documentation cannot save a bad project, but bad documentation can hamstring a good one.

This guide discusses documentation and other resources for L^AT_EX. However, it is necessary to speak briefly on L^AT_EX first, or confusion may abound. The following synopsis is minimal, and only to provide context for *this* guide and the accompanying repository materials.

2.1 The Structure of L^AT_EX

The place to begin is with T_EX.

T_EX was made by Donald Knuth to improve the typesetting-quality of academic articles and books, which he noticed to be in decline as then-new digital technologies were introduced into publishing workflows. An axiom of T_EX is that it is a *typesetting* program.

The manual typesetters of old had to forge a path from handwritten manuscripts to formatted print. They had to make decisions about what fonts to use, what types of words to make italic, how to format their chapter-breaks, and

so on. The manuscript would be ‘marked-up’ to communicate specific instances of typesetting action; an underlined word usually meant ‘make this italic’, for example. However, even with these steps completed, there was still work to be done in figuring out exactly how to execute those instructions — when to break to a new line, when to hyphenate a word, and such. Typesetters effectively calculated an optimum within the instructions they had for the text¹

The way TeX works is sometimes likened to the processes of typesetting. The overall instructions and the markup for the text are written using the TeX language; the TeX program takes those instructions and markup, transforming them into machine-readable instructions, and the engine runs calculations to optimise how the document is rendered. One of the usual outputs produced is a PDF (portable document format) file.

You will notice people referring to the different TeX components: the TeX markup/programming language, the TeX program, the TeX typesetting engine. As a whole, TeX is a system; these component are all part of a synchronous workflow. However, there is often call to consider these elements of the workflow separately. I think that the staged workflow has definitely helped TeX to function well, but beyond this opinion I really don’t have the knowledge to comment further on the components of TeX; I think that one of the links below in this chapter will lead you to better information.

TeX uses markup. The use of markup in digital files really exploded in the early 80’s, although it had been around since 1969 with the invention of GML (Generalised Markup Language). The development of markup languages has been important for file encoding. For instance, the ‘x’ in the ‘.docx’ file extension stands for ‘XML’ (Extensible Markup Language)². But for us, the focus is on typography. The idea behind markup in typography is so simple it’s hard to explain. The hand-drawn underline indicating emphasis of a word adds meaning without transforming the word itself. In the same way, digital markup adds meaning without actually changing the original information.

¹I believe that, to this day, we have not surpassed the technical excellence of the typesetters of the 19th century.

²XML is a descendant of GML.

In most common word processors, italicising a word means it is also displayed in that form: the word as we see it is altered. TeX, however, uses plain-text and markup, so it does not conflate the original meaning and presentation. TeX keeps the presentation instructions separate and is a ‘What You *Mean* is What You Get’ system, as opposed to a ‘What You *See* is What You Get’.

This is a sentence.

This is a sentence in italics, which is the usual way of emphasising something in print.

The TeX input looks like this: `\emph{This is a sentence in italics, the usual way of emphasising something in print.}`. If you sent the file to someone whose font, for some reason, did not have italics, they would still know the sentence is emphasised.

Note the use of `\emph`, meaning ‘emphasis’. This is a descriptive form of markup, because it says ‘emphasise this word’ — the decision about *how* it is emphasised can be another thing. Contrast this approach of describing the features of the text to a more procedural markup, where one might say ‘italicise the text’ with the command `\textit`. LaTeX can feel a bit procedural, because it is for a purpose, but often you get on better, and write more elegant code, taking a descriptive approach to markup.

Outside typographic professionals and programming professionals, most people use L^AT_EX, not T_EX. Up until this point, I have only mentioned TeX, but the workflow that I have described is drawn more from my experience with LaTeX.

LaTeX is a set of macros written for TeX by Leslie Lamport in the early 80’s (TeX had its origins in the 70’s!). In LaTeX, a few simple lines of code sets you up with a document that would be more complicated in TeX. You actually are using TeX in the background, and the typesetting engine is still a TeX engine; you’re just using a few LaTeX commands to represent many more TeX commands. If little customisation is introduced to a LaTeX document, it will naturally resemble one of the base templates (that is: one of the default sets of macros), but it is quite easy to customise the document

from this baseline, and in many cases you will interweave a few commands in base TeX language. Because LaTeX can streamline many parts of the process, we can focus effort where it belongs: on the text itself.

People use LaTeX today because:

- The typography is considered excellent. There are many reasons behind this, but one not mentioned yet is that the algorithms for calculating things like kerning — the spacing between letters and words — are good.
- You can use markup and commands to indicate formatting, meaning that the content (that is: your writing) is separate from the presentation. Being able to 'say what you mean', rather than going through a 'what you see is what you get' system, creates clarity and precision.
- The input file is plain-text, which comes with a number of advantages, including stability of the file.
- It is easy to produce things like tables of contents, bibliographies, and cross-references. Because of these features, LaTeX is considered ideal for thesis-writing.
- The structure of documents can be clearly specified, which can be helpful for denser and/or longer documents.
- The typeset is produced, by the engine, from the plain-text input, so there is less tendency to fiddle with the end-format; interventions in the presentation have to be specified in the input file. The documents are less sensitivity to small shifts of things like pictures or tables.
- Many packages have been written for TeX and LaTeX over time, extending the functionality.
- Tex and LaTeX are a part of the open-source sphere. The systems are maintained and extended with the help of a vast, highly-skilled community³.

³See XeTeX, for example.

An Addendum

Finally, L^AT_EX is a powerful and effective typesetter for maths equations and other technical typography such as tables. The presentation of maths was one of the key things that Knuth wanted to solve with his original TeX system. In this guide I have been biased towards L^AT_EX as a general type-setting system, but that is only because I have used it so much for that purpose.

An introduction to the history of TeX is given here:

<https://tug.org/whatis.html>

You will see in a later chapter that this website (the website of the TeX User Group, or ‘TUG’) is the official homepage of TeX.

Why Not To Use L^AT_EX

The L^AT_EX community is — as one would expect — very pro-L^AT_EX. I am, too, partly because of the specific applications I was able to put it towards as a quality officer. However, I am not blind to possible counter-arguments to L^AT_EX.

Some people promote L^AT_EX on the basis that it encourages one to write documents using logical structure. The header for this section is marked up as a ‘subsection’, for instance, and the formatting for that is either automatic, or whatever I specify for that header level. This is quite different to using different formatting, often made up on the spot, to indicate the document organisation. However, people do not always navigate documents by their logical structure. If you are working with a document identical to someone else’s, you will sometimes communicate by citing page numbers. If you make a note in a physical copy of a novel, you are making a note on a page. I think there is a strong case for replicating the navigation of pages, and I have not come across that possibility for the L^AT_EX input file, even though rapid-viewing of the output, and many other aids, are available.

This discussion gets down to the ‘text is an ordered hierarchy of content objects’ model. Through researching for this guide, I have found some interesting counter-points to that model, which I once accepted unequivocally. To

phrase my own thinking, I would say ‘text is a representation of language characterised by the use spatial dimensions in place of time’.

Anyway, my point is this: While lots of people admire and recommend L^A-T_EX, at the end of the day, it’s just a useful tool, not a solution to the chaos of the world.

Chapter 3

Documentation

3.1 Official Documentation, or Just Offal?

Knuth wrote a manual for TeX, just as he has written many books on computer programming. Some people have commented that this book is not exactly for a general audience. It teaches technical authors how to typeset, and I speculate that one hurdle there is that it may already assume a good deal of technical literacy. However, we need not worry about these things, because we are using LaTeX.

Lamport wrote the manual, “LaTeX: A Document Preparation System”. Even though it is from 1985, that book is, for the most part, viable. However, very few people seem to use it. It is barely ever referenced on forums, and I suspect that it has simply not been made accessible enough for it to remain at the top of the recommended resource list. One of my reasons for this view is examples of an alternative. I like the documentation for Python. It is easy to access, and you know that it is a definitive source. With time spent reading, you can make headway. With LaTeX however, there are many disparate resources that refer to the manual, but the manual itself does not seem to be possessed by all LaTeX users. Instead, the community has immediate online access to other resources, some of which are written by people who have made excellent packages for LaTeX and have been deeply involved in its maintenance. I feel that the absence of ‘official documentation’ explains some of the difficulty that people have in learning LaTeX. I hope that this guide saves some time and stress in that regard.

3.2 A couple of Favourite Documentations

So, what has the community produced and made available for free? From what I can see, probably everything you need to get started.

Memoir

One of my favourite contributions is the memoir class documentation. In my own mind, this is the official documentation for LaTeX.

Now, the document class is the *very first thing* you tell LaTeX. It corresponds to the TeX macros that you are working with as your baseline. Memoir is the most popular and well-liked class for writing books and theses. It is the class I use, because it has great specifications for page sizes and margins, and these are things I need to consider when binding. The documentation for memoir is so good that it is well-regarded as a general documentation for LaTeX by the broader community. However, some things in it are specific to the class.

The memoir documentation is in the repository, in the “Documentations” folder.

I like it because it is comprehensive, well-explained, sensitive to broader concepts and theories while still being detailed, does not make too many assumptions about your circumstances, and has excellent cross-referencing to other resources. It is quite long, because of how deep thorough some of the explanations are.

I advise you to look at the memoir documentation, but not necessarily to use it as your first starting point. I know that you are a wonderful, intelligent, trailblazing scientist, but one of the below recommendations will still serve you well as the first step; memoir is something to come back to later, especially if you ever work in a book format.

The Not So Short Introduction

“The Not So Short Introduction to LaTeX” is a well-established reference guide and introduction to LaTeX. I like it because it acts as a reference book, but then does not skim over things too fast. It genuinely introduces

you to LaTeX by carefully thinking about the sorts of things that new LaTeX users would find useful. Consequently, it is indeed not very short — but that is why such things have chapters and a table of contents.

I recommend this guide as one of your possible starting points. It will provide you with a good introduction, good reference material, and it will help you to write concise code from the outset.

The bibliography for the *Note So Short Introduction* is an excellent list of publications and documentations.

Overleaf

Continuing our journey to a good beginning-point: Overleaf. Now, Overleaf is not really a documentation source. It is a website that offers a free, online LaTeX editor, and has some pretty good tutorial/reference-guides available as well. I think that you will get on quite well starting with the overleaf editor if you do not have something like VS Code set up; without such a set-up, LaTeX can get a bit overwhelming¹.

But my purpose here is to examine the documentation side of things. Overleaf has a ‘documentation’ of sorts, the first part of which is a tutorial ambitiously named “learn LaTeX in 30 minutes”. This may be a good place to start if you want to get going. However, as you move forwards, you should transition to also using things like the ‘Not So Short Introduction’, and the memoir class documentation. These documentations are essentially whole books at your disposal. They are more comprehensive than overleaf, and allow you to get around LaTeX in a structured way.

I like overleaf because it is aware of the symbiotic role it plays in the LaTeX community by supplying these beginning points.

https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes

¹See chapter three for installing LaTeX

3.3 Summary So Far

- $\text{T}_{\text{E}}\text{X}$ can mean a typesetting language, program, or engine; or all at once (and by reading elsewhere you will discover much more nuance in the components of the $\text{T}_{\text{E}}\text{X}$).
- $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ is a streamlined way of working with $\text{T}_{\text{E}}\text{X}$
- A well-made online platform, with documentation, like overleaf is not a bad place to start, but you should have more cogent resources on hand as well. Setting up your own workflow is also essential — some discussion on this is provided elsewhere in the repository.
- The ‘Not So Short Introduction’ is excellent as both introduction and reference book.
- The memoir class documentation is excellent, and provides an in-depth walk-through of one of LaTeX ’s most useful classes. Understanding memoir will help you to understand LaTeX at a deeper level.
- $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ is an interesting case of a program where the closest thing to the official documentation is not widely used. This has led to a nebulous certainty about where one should go, and that is why I have written this guide. Some documentations provided in this chapter are suitable for beginners, yet are impressively comprehensive, written at a high level by people who have contributed to LaTeX .

3.4 Other Documentations

XeTeX

LaTeX predates the compatibility of character encoding and fonts that we now take for granted. Character encoding has been particularly important for programs being able to work for multiple languages².

Because the initial TeX system predates things like Unicode, new engines have been built that to achieve compatibility. One advantage of these new engines is that they can use a wider array of fonts. Most LaTeX users are

²Meaning natural languages, such as English, Finnish, etc.

actually using variations, usually XeTeX or LuaTeX. I use XeTeX³, as it seems to be the most comprehensive option, and I have not encountered any case against it.

XeTeX, and the situation with encodings and fonts that led to its development, are introduced in the XeTeX Companion, which is in the Documentations folder.

Both *Memoir* and the *Not So Short Introduction* suggest XeTeX and themselves provide a gentle initiation to the underlying ideas and application.

BiBTeX

LaTeX may have become a truly powerful typesetting system for academics with the advent of BiBTeX, an in-built program that produces bibliographies. The bibliographies are produced from a library of citations, a citation style, and the corresponding citation keys that are in the document.

There are a few ways to go about citations, as explained in *Memoir* and the *Not So Short Introduction*. However, the most common way of approaching citations and bibliographies appears to be BiBTeX, or one of the direct descendant of BiBTeX that has introduced another facet of compatibility.

My impression is that BiBTeX can seem more complicated than it is, at first, because it involves working with different files. But once the pieces are brought together it is just as concise as any other method.

It is not yet clear to me whether one single descendent of BiBTeX is best option, but I get the impression that they are very similar to BiBTeX. I have provided the documentation to BiBTeX in this repository. While originally written in 1988 and with documentation updated in 2010, BiBTeX is still maintained, and if it ever does become obsolete, it will be easy to use one of the descendants, such as BiBLaTeX.

The *Not So Short Introduction* suggests BiBLaTeX (which is invoked by loading the biblatex package). I found this easy to use. The documentation for biblatex is also provided in the repository.

³Or XeLaTeX.

And Many More

At this point, I feel that the main resources for learning LaTeX have been provided. However, specific requirements will bring up the need for more specific information.

Packages have been written to cater for certain needs, and each one comes with its documentation. I encourage you to at least have these on-hand when you go into some specific area, as it is important to remember that packages are add-ons, and may even clash with some functionality that you are drawing on elsewhere; or there may be some similar nuance to their use.

As mentioned above, the bibliography for the *Not So Short Introduction* is effectively a list of common packages.

- for Tables:
 - tabularx package
 - array package

To be Continued...

Chapter 4

Official Sites and Resources

4.1 The Three Main Sites

The LaTeX Project

This is the official website of LaTeX in that it is the website of the project responsible for maintaining the entity of LaTeX.

<https://www.latex-project.org/>

However, the project funds are actually administered by the Tex Users Group.

Tex User Group

<https://www.tug.org/>

This website is the most official base for LaTeX and TeX that there is on-line.

The Tex Users Group is the administrator of the LaTeX Project.

The Comprehensive Tex Archive Network

<https://www.ctan.org>

Example of things you would go to this site for are font catalogues, docu-

mentation downloads, and mirrors for packages. It functions as a kind of repository for the TeX Users Group.

4.2 Bringing It Together

So far we have been introduced to LaTeX, its documentation, and the TeX User Group. A good next step would be installing TeX and getting started.

The thing you are usually after is *TeXLive*, the distribution of TeX that is kept up-to-date with updated frequently, engine options, fonts¹, and such. It also provides a simple program for actually running LaTeX: for entering input and running it through the engine, producing the output.

TeXLive is distributed by TUG, but if you go to any of the above websites, you will get there.

I will add a resource on setting up workflow, soon, if I come to a good one.

I started off using MacTeX (the version of TeXLive for Macs), which provided me with a program called TeXShop. Then I used overleaf for a short time. Now, I use Visual Studio Code with the LaTeX Workshop extension, and recommend that you use a similar IDE (integrated development environment²).

It does not really matter if you are using TeX or LaTeX. TeXShop, for instance, is fine with both: one just needs to make sure that the program/engine that is selected matches the way the input-file has been written.

¹A favourite of mine is Cormorant Garamond.

²Consider R Studio: I would call that a good IDE for R.

Chapter 5

Online Help

There are a few places you can go for help online, including tutorial websites and forums. There are only a couple of places I recommend. In terms of tutorials, Overleaf (mentioned above) can be a good resource. As for forums, Stack Exchange is the only one that seems to be staying relevant.

<https://tex.stackexchange.com/>

Visit Stack Exchange with caution. Many of the answers are *excellent*, but some are quite mediocre.

I have found the longer documentation for LaTeX to more useful than any of the shorter online resources. Online help has been crucial for me, but I have also been through a phase of bad habits, formed by picking out bits and pieces without proper guidance. This is not a critique of stack exchange, but rather an observation that long-form resources have an immense synergy with shorter resources that help with specific problems.

I have to admit that my main limitation to learning LaTeX is not being comfortable in the digital sphere. Too many things seem arbitrary. It is like a world where acronyms abound, without knowledge of their meaning being essential to aptitude. It could be that I prefer referring to written documents, rather than other media, so that I can process the assumptions behind each piece of information. I have not yet found YouTube to be a great resource on LaTeX, but others may get on well there.

Now my guide is finished.