Introduction to using LaTeX

Jesper Eriksen[*]

November 18, 2017

Contents

[*] Contact: jeri@business.aau.dk

1 introduction

This document is a (very) short introduction to some of the fundamentals in writing in LaTeX.[1] Of course, there are already a ton of resources that you can look through to get started with LATEX, including The Not so Short Introduction to LaTeX2e by Tobias Oetiker and his co-authors, or Lars Madsen's Introduktion Til LATEX.

I try to provide some of the key elements that you most likely will need to get started writing your semester projects in LaTeX. These include what you need to create your first document, how to insert pictures, tables, and equations, how to make footnotes, creating citations and a reference list, as well as headers for your document.

Most students starting their university journey have been used to writing in word processors like Word, Google Docs, Pages, or the like. In these types of word proecessors, What You See Is What You Get (WYSIWWYG). LaTeX is slightly different, it is a word processing *language*. Instead of writing your text directly into a document in which you pray that you formatting will stay as you want it and pictures doesn't start jumping pages, you write down commands for how you want LaTeX to write the document, and letting LaTeX take care of making your documents look beautiful. This might seem a bit intimidating, but do not worry, tons of other people have already been in your position and asked the same questions as you are doing now - and many of them have done so online on web pages like stackexchange where TeX connoisseurs revel in helping out the less experienced. This, of course, means that if you search for whatever your problem is you most likely will find a good answer - fast.

One of the easiest ways to get started with LaTeX is to use online editors like Sharelatex. Sharelatex allow you to work without having to install a lot of things on your computer, and also to collaborate on several authors on the same document at the same time (which is particularly handy for group projects).[2]Alternately, if you want to be able to write LaTeX documents locally on your computer, you can find several guides online for installing the LaTeX system together with an editor (such as TeXMaker), and relevant packages online.[3] In this document, I will assume that you use ShareLatex, which means that you will have to make workarounds for some things if you do not.

Time to get started!

[1] Currently, it probably also contains quite a lot of flaws. I will appreciate any suggestions for changes.

[2] As a student at AAU you will have an Institution access that allow you to collaborate with your peers in the same file without overwriting each others work - somewhat similar to Google Docs.

[3] The web page of *The LaTeX Project* gives you access to the most recent of the distributions (like MikTex and MacTex) that installs all the things you need to get started.

2 Setting Up A Basic Document

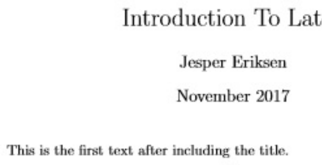
When you log on to Sharelatex, you will notice the option to start a new project. If you click on that option, you will go to a page with three seperate sections. On the righthand side is a pdf document and a large blue bottom saying “Recompile” on top. In the center is a section that contains some code, and to the left a register (like a folder on your computer). LaTeX is not a regular word processor like word, where *what you type is what you get.* Instead, LaTeX is a language, from which pdf’s are compiled. The language relies on you telling the software what should be in the document, and then the software will produce (compile) a document with those specifications. The specifications include what class of document you are writing (e.g. article, report, etc.), the encoding of the input (utf8 is preferable here if you write in Danish), a title, author, and date. When you have these elements, you can tell the software to begin the document, insert the text you want, and then end the document. Written in LaTeX, these basic elements become

Code 1: Example 1

```
\documentclass{article}
\usepackage[utf8]{inputenc}

\title{Introduction To Latex}
\author{Jesper Eriksen}
\date{November 2017}

\begin{document}

\maketitle

This is the first text after including a title.

\end{document}
```

Which, when you compile the code, produces the following text

Figure 1: Example 1 - document



Introduction To Latex

Jesper Eriksen

November 2017

This is the first text after including the title.

The most important thing to notice is that any text you write that should go into the document goes between the `\begin{document}` and `\end{document}`. Let us call this the main body of the code. Anything else, like the packages you use, graphical settings, the title-, author, and date-setting, etc. goes above these into the *preamble*.

Next, notice the `\maketitle` in the body part of the code. Whenever you want to do something beyond simple text, you will often be starting the code with a \. As you might guess, the `\maketitle` inserts a title together with the name of the author(s) and the date. Other useful statements like this is `\tableofcontents`, which produces a table of content, and `\bibliography{file}` that can create a bibliography if you have references in your document, as well as a .bibtex file with your references (more on this later). These all go into the main body of the code.

Whenever you have added something to the code-side of your document, you need to *recompile* the document. In Sharelatex, simply push the blue button. Doing so tells the website that you want a document created based on the code/text that you have added, and a new document should appear in the window to the right of the code. If it does not, you might have a problem in your code (check for example that you have all the { and }'s and the *begin* and *end*'s that you need). Sharelatex will also guide you to where any problems might arise through the colored sections that appear around the code (red means you need to check these lines).

With these things in mind, you should be ready to start compiling you own first document in LaTeX.

3 Starting New Sections

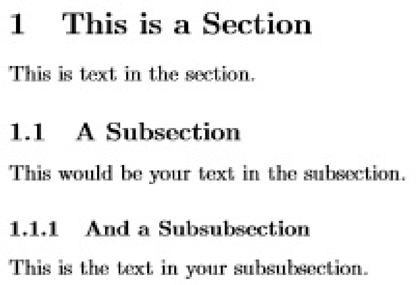
Most documents contains more than one section, and in many cases you also need subsections (for example for theory A, B, and C, as well as a comparison).

You create sections, subsections, and subsubsections in LaTeX using the following commands:

Code 2: Sections, subsection, and subsubsections

```
\section{This is a Section}

   This is text in the section.

\subsection{A Subsection}

   This would be your text in the subsection.

\subsubsection{And a Subsubsection}

   This is the text in your subsubsection.
```

Figure 2: Example 2 - Sections, subsections, and subsubsections



1 This is a Section

This is text in the section.

1.1 A Subsection

This would be your text in the subsection.

1.1.1 And a Subsubsection

This is the text in your subsubsection.

As you can see, each section is numbered. LaTeX *handles* this for you. In addition, by inserting a `\label{textlabel}` in the section command, like this:

```
\section{This is a Section \label{lab_sections}}
```

You can even refer to the section number without having to change it by using the command `\ref{lab_sections}`. For example, this is section 3. If you want to know more about all the fascinating ways you can structure your documents, you can have a look here

4 Figures and Tables

Often we have figures and tables that we want to insert in our documents. In Word you just dump the picture in the text and pray that the formatting will not be messed up completely as you continue working on the document. In LaTeX, you tell the software which picture you want, approximately where it should be placed, and then the software will make sure that the document is put in a suitable place. The same, of course goes for tables.

4.1 Figures

To insert pictures in your text, you will need to add the package "graphicx" to the top of your latex document.[4] You do this by inserting `\usepackage{graphicx}` above the `\begin{document}` command. Next, you need to save the figure, graph, plot or whatever else in a .jpg, .png, or .pdf format and place this file in the same folder as your main .tex document (in ShareLaTeX this is done simply by uploading the file to the main folder you see on the left side panel). You can then include the picture by using the *figure* environment (anything that goes between a `\begin{something}` and `\end{something}` is called an environment). Below is an example of latex code that inserts a picture of Aalborg University which I saved in the same place as the .tex file.

Code 3: Figures

```
\begin{figure}[H]
        \caption{A picture of AAU}
        \label{fig:AAU}

        \centering
        \includegraphics[width=0.85\textwidth]{Aalborg_university.jpg}
\end{figure}
```

Figure 3: A Picture of AAU

In the figure environment you include a picture by using the `\includegraphics[width=0.85\textwidth]{Ex.jpg}` command, where $[width = \dots]$ determines the width of the picture, and *Aalborg_university.jpg* is the file we want to include. What goes in between the [] are called options - beyond determining width, you can also set the height, rotate the picture and other things by putting options in here.

The figure environment also takes options. Here I have inserted *H* to tell the software that I want the picture *H*ere. Alternatives that gives more flexibility are *htbp* and *h*. There are also plenty more options that you can play around with.

To include a caption for the figure, we can use the command `\caption{something}`. LaTeX again takes care of numbering the figures, and you can refer to the specific figure with `\ref{fig:AAU}` like this 3.

[4] There are of course other alternatives, but graphicx appears to be the standard package.

You might also note the `\centering` command. This tells the software that within this enviroment, it should center everything. Leaving it out would render the picture to the left like in figure 4.

Figure 4: A Picture of AAU

There are many more options for figures, including setting pictures up in groups, allowing for wrapping text around the pictures and much more. You should have a look here if you want to know more.

4.2 Tables

Tables, for example from summary statistics, regressions, or perhaps just showing some different concepts in relation to each other, are another often appearing aspect in our writing. Like for figures, there is a specific environment for tables. Actually, tables require two of them (for the most common table you will meet): the outer *table* environment, and the inner *tabular* environment. In addition, tables also have a rather specific syntax to them, which consists of & and \\'s. As a new user, you could probably make your life easier by using a site like tablesgenerator.com to create the at times intricate tables.

As an example table, we can look at the following table

Code 4: 2-by-2 table

```
\begin{table}[H]
        \centering
        \caption{A cell-table}
        \label{tab:tab1}
                \begin{tabular}{lc}

                \\
                \hline
                Cell 1 & Cell 2 \\
                Cell 3 & Cell 4 \\
                \hline
        \end{tabular}
\end{table}
```

Table 1: 2-by-2 table

Cell 1	Cell 2
Cell 3	Cell 4

Firstly, notice the *table* environment. It takes a placement option like the figures in section 4.1, and within it we can also use `\centering` to center the table, `\caption{text}` to add a caption, and `\label{text}` to refer to the table.

To actually create the table, we use the *tabular* environment. The tabular environment takes a special type of options put in {}. In this, we insert letters for each column in the table. The options are *l* for left-adjusted, *c* for centered, and *r* for right-adjusted columns. In the example I have *lc*, which creates two columns. The first left-adjusted, the second centered.

Within the *tabular* environment we put the actual table. \\ moves you to the next row in the table, and & moves you to the next cell in the table. Between these you put the content of each cell. You can use `\hline` to insert a horizontal line across the table (these look particularly good in the top of the table, and not so good in the middle of the table).

Of course, there are many options for developing the table, including merging cells and adding text by using `\multicolumn{#cols spanned}{lcr-adjustment}{Text}`, and adding vertical lines using `\vline`. You can also create tables that span several pages by using the *longtable* environment (make a google search for this, if you need it).

For more information on how to write tables in LaTeX, you could have a look here.

5 Listings

Once in a while we also want bullet point lists, or perhaps numbered sections. The simplest ways to create these are by using the *itemize* and *enumerate* environments.

For example, to create a bullet point list in two layers, we can write

Code 5: Item-list

```
\begin{itemize}
        \item This is the first item
        \item this is the second item
        \begin{itemize}
                \item This is an item within an item
        \end{itemize}
\end{itemize}
```

- This is the first item
- this is the second item
  - This is an item within an item

Note that for each new item in the list, we have to write *item*. We can also embed sub-lists in the list by starting a new itemize environment.

You can make enumerated lists in a very similar way - you simply need to change *itemize* with *enumerate* in the code:

Code 6: Enumerate-list

```
\begin{enumerate}
        \item This is the first item
        \item this is the second item
        \begin{enumerate}
                \item This is an item within an item
        \end{enumerate}
\end{enumerate}
```

1. This is the first item
2. this is the second item

(a) This is an item within an item

The two types of lists can also be used together. For example, an enumerated list within an itemized list would be

Code 7: Item and enumerate-list

```
\begin{itemize}
        \item This is the first item
        \item this is the second item
        \begin{enumerate}
                \item This is an item within an item
        \end{enumerate}
\end{itemize}
```

- This is the first item
- this is the second item

    1. This is an item within an item

There are also many more options in terms of creating lists (including descriptions, and specially numbered enumerations), but this should get you started. You can always search online for any of the many other ways to create lists, but a general reference is this.

6 Equations

Writing equations in LaTeX is a particular joy for many users simply because of how much better an equation written in LaTeX looks that your standard word processor.

Firstly, to write mathematical expressions in your document, you will need to add the following packages to your preamble:

Code 8: Equation

```
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
```

To write an equation on it's own line, you can use the *equation* environment such as

Code 9: Equation

```
\begin{equation}
        y^* = \frac{s}{n + \delta}^{\frac{\alpha}{1-\alpha}}
\end{equation}
```

$$y^* = \frac{s}{n+\delta}^{\frac{\alpha}{1-\alpha}} \tag{1}$$

If you are instead interested in an inline equation, such as $Y = K^{\alpha} L^{1-\alpha}$, you can use the inline math setting $$ to write `$Y = K^{\alpha} L^{1-\alpha}$`.

You will note that many things requires special commands, including making a fraction `\frac{num}{den}`, writing greek letters like `\alpha`, or creating a more-than-one-letter exponent (`L^{1- \alpha}`). Many of the features you might be interested in are well explained at this site.

7 Footnotes

Academics likes to write footnotes.[5][6] You can do so particularly easily by using the command `\footnote{Text in footnote}`, which would produce[7] as your footnote.

8 Citations and a Reference List

The final thing I will introduce in this document, is how to easily make references in your document, as well as generate a literature list.

To produce a citations and a reference list you will need the following things:

- A bibtex file with all of your references:
  - If you are using Mendeley, you can make Mendeley create a file with all of you references, which you can upload to Sharelatex. Here is how to do it:
    1. In Mendeley, go to Tools/Options/bibtex
    2. Allow “Enable BibTeX syncing”
    3. Choose where the file should be stored.
    4. Locate that file, and upload it to Sharelatex. The file will most likely be names “library.bib”. If not, either rename it, or change the following code.
- In your preamble (the code above the main body), include the following:

  Code 10: References 1

  ```
  \usepackage[authoryear,round]{natbib}
  \usepackage{apalike}
  \bibliographystyle{apalike}
  ```

  Here the first package, natbib, is the one that will allow you to make references. apalike is a citations type package (this one allows you to choose the style your references - this one is the reference style according to the American Psychological Association). Finally, the last part sets the styling of the bibliography, or list of references at the end of the document. This, of course, can also be changed.
- At the place in your main body, where you want your references, include the following:

  Code 11: References 2

  ```
  \bibliography{library}
  ```

  This will gather all of your references from the bibtex file that you collected from Mendeley before.
- Finally, you are now ready to include citations in your text! natbib has several commands that allows for different ways of doing so. Each one relies on the citation key that you specified for the reference in Mendeley when creating the reference. For example, you can include an author (year) reference like Marshall (1920) by writing `\cite{Marshall1920}` if you have a reference with Marshall’s Principles of Economics from 1920 stored in Mendeley with the

[5] Partly because we just cannot contain ourselves and need to write more than we ought,

[6] and partly because some things just really does not fit into the text but are necessary to know.

[7] Text in footnote

key Marshall1920. You can also do complete citations in parentheses (Author, year) using `\citep{Marshall1920}`, showing up as (Marshall, 1920).

- The reference list. Whenever you cite something in your text, this will now show up in the References list at the place where you put your *bibliography* command. Have a look below in the section “References”, where you will find the Marshall citation!

As with all other things in LaTeX, there are more than many online resources for to get you going with producing references. A good introduction to using the natbib package is provided by Sharelatex at Sharelatex’ web site.

9 Commenting Yours and Others Work

This section contains a final feature that can be quite useful for editing your documents when using Sharelatex. In Word, Google Docs, etc. you often send the document (or a link to it) to others to get their comments. Your friends, mother, or if you are particularly in need of help, dog, can then insert comments in the margin.

Sharelatex has a similar feature that allow people with access to the document to make comments in the code-text. Just look in the upper-right part of the code window, and you will find a tiny button that allows people to add comments.

10 List of Resources

All of the resources that I have mentioned in the document above are collected here together with their links. If you have suggestions for something that should be added or find a link that does not work, do send me an email, and I will put it on the list or fix the link.

- Sharelatex: `https://da.sharelatex.com`
  - Their documentation side with plenty of help to get started writing in LaTeX: `https://da.sharelatex.com/learn/Main_Page`
- Tobias Oetiker’s “Not so Short Introduction to LaTeX2e”: `https://tobi.oetiker.ch/lshort/lshort.pdf`
- Lars Madsen’s “Introduktion til LaTeX”: `https://data.math.au.dk/system/latex/bog/version3/beta/ltxb-2011-09-13-20-10.pdf`
- The LaTeX Project website - a resource for getting the latest version of LaTeX, packages, and an editor for you personal computer: `https://www.latex-project.org/get/`
- A reference for handling figures in your document: `https://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions`
- Sections, chapters, paragraphs, and other ways of partitioning your document: `https://en.wikibooks.org/wiki/LaTeX/Document_Structure`
- Tablesgenerator - your website for making any table to input in LaTeX without having to struggle (it even takes excel files!): `https://www.tablesgenerator.com/`

- For general information on how to write tables in LaTeX: `https://en.wikibooks.org/wiki/LaTeX/Tables#.40-expressions`
- Lists and enumerations in LateX? Look here: `https://en.wikibooks.org/wiki/LaTeX/List_Structures`
- Math in Latex: `https://en.wikibooks.org/wiki/LaTeX/Mathematics`
- An introduction to references and bibliographies in LaTeX: `https://da.sharelatex.com/learn/Bibliography_management_with_bibtex`
- Citing in your text using the natbib package: `https://da.sharelatex.com/learn/Natbib_citation_styles`

References

Marshall, A. (1920). *Principles of Economics.* Macmillian and Co., Limited, London, 8 edition.