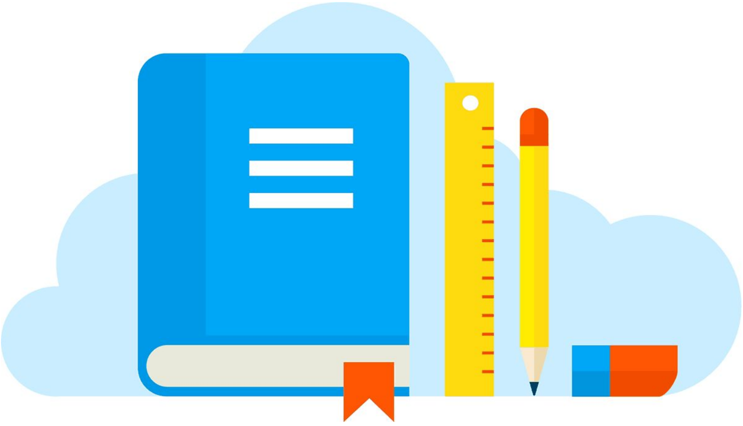

Lecture Notes 0

Date: 04/01/2017

What is LaTeX?

LaTeX is a document preparation system for high quality typesetting. It is used for almost all type of scientific publications (from a scientific papers, reports, etc to textbooks, Physical Geodesy is written by LaTeX). LaTeX is *not* a word processor. You all know Word, and you probably have used it several times. One of the main issues of using Word--or any other word processing applications--is that you need to worry about a few things. Consider this very basic document.

Cartesian closed categories and the price of eggs
Jane Doe
September 1994

Hello world!

The previous document consists of a few things. The title of the text, the author name, the date, and the actual content of the document. And it is quite clear that you need to do some styling for them e.g., the font size of the title should not be the same as the actual text. There are also headings and other stuffs that need to have their own font size and other styling (the heading usually has a big font, and it is on the left side of the page--assuming that we are using a left-to-right language). In a word processor e.g., MS Word you need to specify all of that yourself. The task becomes even harder as you use advance things like figures, equations, etc when you need to track of them. You also need to correctly put the caption with their corresponding item (figure, table). It becomes even harder when you need to cross-reference these tables or figures (I will from here and on call them floats), in your table of contents. In LaTeX, things are very different. LaTeX is based on the idea that writers should only focus on their writing, styling is left for their experts. Let us get back to our previous document.

```
\documentclass{article}
\title{Cartesian closed categories and the price of eggs}
\author{Jane Doe}
\date{September 1994}
\begin{document}
   \maketitle
   Hello world!
\end{document}
```

Figure 1. A very simple LaTeX document. Note that you do not need to specify the title font size, or it's location on the paper or any of these. The same holds true for the other part of the document.

A very simple LaTeX documents (as shown in Fig.1) consists of the following

- It begins with the type of the document, in this case we use the *article* type. There are other types e.g., book, or thesis
- `\title{<some_title_goes_here>}`, `\author{<author_name_goes_here>}`, and `\date{<date_is_written_here>}.` It is important to know that *all* LaTeX commands begins with the backward slash "\"followed by the command name. It is a good analogy to think of LaTeX commands as mathematical *functions*. Square root is a function that takes as an input a number, and it returns its square root. You have used it a lot in your calculator, haven't you? Consider the `\title` command in our LaTeX case, it is a function as we have already agreed, and it has argument(s). LaTeX parses its arguments in a different way. If the argument is only one letter you don't need to use curly brackets "`{}`", however, if the argument is more than a letter you need to group them in a curly brackets. Hence the use of brackets in Fig.1 e.g., `\title{This is a cool document}.` In mathematical notations, symbols are often one letter, but that not always holds true. We will see that later.
- The document begins with the command `\begin{document}` and ends with `\end{document}.` Between those commands is the actual text of your

document. Note that \maketitle does not take any argument. It is just a command that is used to display the title, author name, and the date in your document.

In LaTeX, the document consists of several sections, each section has its subsection(s). The section can be nested for several level using the prefix sub. The section in LaTeX has the command \section as you would expect.

\section{}

\subsection{}

\Subsubsection{}

Figure 2: The sections structure in LaTeX. You can go wild about nesting these sections, but in practice you only use 2 levels at most. Note that these category is for article type only, in the book or thesis class there is a high level which is the Chapter.

It begins by the document type, or class, in this case we specify it to as an article. We have also specified the title, author, and the date. The document begins with the keyword \begin{document}, and it ends with \end{document}. Between those commands you write the actual contents of your document.

You can think of \begin and \end as a commands (or functions) that takes in other LaTeX commands (or functions) to perform certain operation. In the previous example \begin and \end took document as their argument, and they construct the document for you. Other type of \begin and \end groups are also available. Common types of group commands in LaTeX are

- Itemize. To construct unordered list
    - \begin{itemize}
    - \item 1 ...
    - \end{itemize}
- Enumerate. It constructs ordered list
    - \begin{enumerate}
    - \item 1 ...
    - \end{enumerate}

- `Table`. The construction of a table is quite difficult than other commands. We will discuss it with detailed examples later.

What you see is what you get (WYSIWYG)

LaTeX is not WYSIWYG. That might be difficult for the first time, but after a while you will see that it is not that important. Wysiwyg is actually helpful at some cases. When you are working on some document you need to see the exact output as you are working on it. A lot of new users spend a lot of time compiling their LaTeX files just to see the small changes that they have made, even though these changes are a few words! In Word, you see changes as you type, in LaTeX however, you need to compile your document first before you see the output. In the first section we will have a live session of LaTeX, so most of these concepts will become very easy there.

LaTeX

LaTeX has many advantages over word processing systems. In LaTeX everything is automated. You do not need to worry about the place of your title, or to keep track of the numbers of the equations you used or any of that. Generation of bibliography is also easy. You can do most of that in Word, but in LaTeX yo get to do them much faster, by many order of magnitudes in fact. Word was not designed for publication, or scientific publication in particular. It was meant to be used by journalists or people like them to publish their contents as fast as possible. While learning in LaTeX is quite steep, once you get it it becomes the most powerful tool you have ever used. Another important reason why using LaTeX is that it is much faster than a word processor. In a word processor software you need to use your mouse a lot, which takes a time. In LaTeX however, everything is under your keyboard, you do not need to use a mouse at all! If you want to insert an image you just add its path to your document (using some commands that we will describe later). If, for some reason, you decide to change that image it is much easier than in Word, you just need to only update the new path.

A very simple LaTeX document

Let us try to write a very simple LaTeX document. We have already explored the basics of LaTeX, so some of these commands should become familiar to you. To make things quite interesting, we will use your assignments template as an example.

To start, kindly open assignment_template.tex file. Because of the extension it might not be recognized by your system. You can open it with notepad (in Windows systems), or you can open it by any text editor in Linux systems (gedit can be fine). In both cases I highly recommend to use a text editor more advanced than notepad. You can use sublime text[1]. It is very powerful, free, and it will give you a nice syntax highlighting for *tex* files.

Your file starts with a few lines with leading “%”, they are comments and they are generally used for describing the file. *Each* LaTeX document starts with \documentclass command which specify the type of the document that you are working on. In your assignment templates we used scrartcl[2] document type. It is similar to the article type, but more appropriate for assignment templates. When we pass the name of the document class to \documentclass command it searches for this package in your packages that came preinstalled with miktex[3]. If it is not available you will get an error. More on the errors will be discussed later. You may have noticed the brackets in the previous command. Those brackets are used to pass the arguments for your command. In this case, scrartcl comes with defaults values, if you want to override them you have to pass the new values in brackets. In this case, for scrartcl package (or command to be consistent), the default value for the fontsize is 12 (just assuming), we want to make it to 11. We pass the new argument in those brackets. You may also need to use those brackets to ensure that you get the behaviour you want i.e., we wanted to ensure that the fontsize is 11, and the paper is A4.

```
\documentclass[paper=a4, fontsize=11pt]{scrartcl}
```

There are several lines start with \usepackage{some_package}, as you may inferred, we use \usepackage command to include a specific package to our document. There are tons of packages in LaTeX, it’s really good to only include the one that you are actually using in your work. In our template we used some packages for styling e.g., secsty, we also used fancyhdr package for styling the headers. The most important packages are amsmath packages that allow us to use advanced equations. You can find more informations about the packages we used in your assignment template file.

[1] "Sublime Text - Download." https://www.sublimetext.com/3. Accessed 3 Jan. 2017.
[2] "CTAN: Package scrartcl." https://www.ctan.org/pkg/scrartcl?lang=en. Accessed 3 Jan. 2017.
[3] (2016, December 30). MiKTeX. Retrieved January 3, 2017, from https://miktex.org/

If you jump to `\title` command you will find that there are weird things e.g., \normalfont \horrule, etc. These are basically used for styling `\normalfont` normalizes the font, `\horrule` draws the horizontal lines on your document title and so on. You do not need to worry about them, neither should you modify any of them. You *only* need to specify your institution name, and your name (and index).

In the date command, you can either specify the date of writing this article, or even better use `\today` command that automatically generate the date of writing this document. LaTeX is pretty cool, isn't it?

`\begin{document}..\end{document}` contains all of the commands regarding your document e.g., sections, etc.

```
\begin{document}
\section{Problem 1}
\subsection{A subsection}
\section{Problem 2}
\end{document}
```

We have discucssed the concepts the group commands earlier, they are used for commands that have several lines e.g., itemize to set unordered lists. So far we have learned two important things about using "{}" and \begin, \end commands in LaTeX.

General Examples in LaTeX

In this section we will go through common commands that you are very likely to encounter them in your work. Like HTML, LaTeX also parses new lines as spaces! The lesson to take away here is; when you hit Enter key when you are typing in e.g., Word the cursor would take you to the the beginning of the next line. In LaTeX, however, if you hit Enter that will not make a new line for you. If you want to start a new line you have to type "\\". Let us have an example to demonstrate this.

"This is just a demonstration sentence

And doesn't mean anything at all"

The output would be

"This is just a demonstration sentence And doesn't mean anything at all"

If you, however, want to print a newline, you should do

“This is just a demonstration sentence\\ And doesn’t mean anything at all”

The output would be

“This is just a demonstration sentence

And doesn’t mean anything at all”

- Lists are used widely in writing. There are two type of lists in LaTeX, *unordered lists* like this one, and *ordered lists*.

Ordered lists are often used when the order of the items is important, while unordered lists used when the order is not important.

```
\begin{itemize}
\item Item one
\item two
...
\end{itemize}
```

The output would be something like this

- Item one
- Item two

...

For the ordered lists we use the command \enumerate

```
\begin{enumerate}
\item Item one
\item Item two
...
\end{enumerate}
```

And the output would be something like this

1. Item one
2. Item two

...

You can also nest the items if you like, to do that you have to nest another listing environment under the item you are working on.

```
\item Top Item
\begin{itemize} % you can replace it by enumerate if you want
\item Subitem
\item Another subitem
\end{itemize}
\item Another item
```

The output would be as follows

- Top Item
  - Subitem
  - Another subitem
- Another item

You get the idea I guess. You can nest them however you want, but it is not recommended to use nested lists as that can lead to confuse the reader.

It is often the case that you need to work with tables and figures. The syntax might seem quite difficult, but we will go through it and you will find that it’s very simple. Let us begin by writing a simple table in LaTeX. We will use booktabs[4] package as it provides scientific tables (a.k.a tables you often see in textbooks). It is already included in your assignments, however if it is not, go to beginning of your document and include it using \usepackage{booktabs}.

Generally, a table comes with a few thing, like the caption of the table (what is this is about?), the location of your table within your page i.e., whether it is centered, or left aligned, etc. you *cannot* know exactly where your table will be on the paper, it is just a float on your paper, you only specify at that unknown location how it is aligned. You also need to specify a *label* for your table so that you can reference it from any place in your document.

A typical table in LaTeX starts with the \table environment that can be accessed by (?). The table environment has nothing to do with the content of your table. It only controls the properties of your table e.g., position, label, and the caption. All of these commands are passed to the table command. An example will help you to understand all of the above concept.

```
\begin{table}[] % we do not need to pass any argument for table
command
\centering % you can remove this attribute if you don’t want to
center your table
\caption{This is a nice table, is not it?}
\label{table:my-table} % usually the good way to name your labels is
as follows: first specify the name of the float e.g., table, or
figure, or eqn. Then, use colons to specify the label of that float,
in this case I call it “my-table”. Notice that I did not use “_”
underscore as it is preserved for mathematics symbols, I could use it
but I had to escape it with “\”.
  \begin{tabular}{@{}lll@{}}
  % the l’s indicate the number of columns. In this case we are using
a table with three columns. We only need to specify the numbers of
the columns, while the number of rows can be any number of rows.
 % @{} are used for adding extra spaces to your document. They are
just for styling purposes.
  \toprule % toprule command is used to add a horizontal line to the
top of your table. There are usually three types of horizontal lines
in a typical table. The \toprule, the \midrule, and the \bottomrule.
 Column1 & Column 2 & Column 3 \\
```

[4] "CTAN: Package booktabs." https://www.ctan.org/pkg/booktabs?lang=en. Accessed 3 Jan. 2017.

```
 % we use the ampersand sign “&” for separating the rows items in our
table.
 \midrule
 item 1 & item 2 & item 3 \\
 item 4 & item 5 & item 6 \\
 \bottomrule
  \end{tabular}
\end{table}
```

Figures in LaTeX

Figures are used widely in writing. The most important properties of figures are *caption, label,* and its *position.* Usually you need to include some packages to convert figures to LaTeX. In this case we will use graphix package. You can add using \usepackage{}

Date: 04/01/2016