Resource: 1 LaTeX Tutorial

LaTeX is a document preparation system widely used in science and research-oriented engineering. It allows authors to create high-quality documents with technical content.

Resource 1.1 Overleaf

A great way to collaborate on LaTeX reports is to use Overleaf instead of (or in addition to) installing it locally. This is actually preferred for collaborative projects. A template is available here.

Resource 1.2 The document structure

The following are the highlights of the file structure.

- The `report.tex` file is the one you will mostly be editing.
- The `figures` directory contains a single figure file. This is where you'll put your figure files to be included in your report.
- The `report.bib` file contains the references.
- The `.sty` files load packages and define some macros. You will probably not be editing the `.sty` files. If you would like to add a package, you can do so in the header of the `.tex` file, preferably after the line `\usepackage{mycommands}`.

Resource 1.3 LaTeX basics

My first rule is build often. Error messages are not always helpful, so I recommend building after a complete thought has been expressed. It's a nice cadence, like breathing.

New *paragraphs* can be added with two linebreaks. A single linebreak does not give you a new paragraph.

You can add new sections with the command

```latex
\section{Name of Section}
```

Subsections and subsubsections are created with the command

```latex
\subsection{Name of Subsection}
\subsubsection{Name of Subsubsection}
```

The lines that follow these commands are included in that section/subsection/subsubsection.

You can label a section (and many other objects) with the following syntax.

```
\section{Name of Section} \label{sec:name}
```

The label is `sec:name`, and can later be referred to in the document with the command

```
\ref{sec:name}
```

or

```
\autoref{sec:name}
```

The latter command is usually preferred because it returns the type of object, like "Section 3" or (if the object was a figure) "Figure 2". The **`\ref`** command gives only the number, such as "3" or "2". It is not necessary to label a section with `sec:name`; it could just as well be `name`. However, it is common practice to use prefixes like `sec:` (section), `fig:` (figure), `tab:` (table), and `eq:` (equation) for labels in order to differentiate them.

In the style (`.sty`) files I have included in the template, the `hyperref` package is loaded, so references will be hyperlinked. If you would like to include a url, you can create a hyperlink with the syntax

```
\href{http://ricopic.one}{the coolest website}
```

Resource 1.4 Adding your own figures

The template `report.tex` includes a figure. The figure that is loaded is the external file `figures/data.pdf`, wish is located in the `figures` subdirectory. The syntax we usually use to load figures is

```
\begin{figure}[bt]
  \centering
  \includegraphics[width=1\linewidth]{figures/data.pdf}
  \caption{here's a caption.}
  \label{fig:name}
\end{figure}
```

This creates a `figure` *float* environment, which inserts a figure as close as possible to the location in the document where you've placed it, but at the bottom (`b`) or top (`t`) of a column. Sometimes the placement of a figure is mysterious, and needs to be tweaked at the end by moving it around in your `.tex` file. Don't bother with placement until you've finished the entire document.

You may change the width of the figure by changing the `width` parameter in `\includegraphics[width=1\linewidth]{...}`. It's convenient to leave it in terms of the `\linewidth`, so you may like to use `0.9\linewidth`, for instance. You may also like to use dimensions like `3.2in` or `7cm` or `200pt`, which are all valid.

The file path (`figures/data.pdf` above) can either be relative or absolute. I highly recommend a *relative* path for portability.

The `\caption{}` can always be edited. It should be descriptive, and even redundant with the text. The reason for this is that when many people read technical documents, they read the abstract, look at the figures and captions, and read the conclusions. It should tell a complete story.

The `\label{}` name (`fig:name` above) can be whatever you like (sans spaces or weird characters). I recommend the prefix `fig:` for all figure labels.

Resource 1.5 Adding your own equations

One of the most powerful features of the LaTeX system is the beautiful equations. With a little work, you can master the syntax and write them with ease. There are a few environments for equations.

Resource 1.6 Equation environments

For *inline* equations, include the contents between two `$` signs, like $\cos(x) + 7$. Even when you refer to a variable in your text, you should use an inline environment, like `$x$`. This will ensure the typesetting is consistent for that variable.

For *display* equations, there are a few options. I recommend mostly using

```
\begin{align} \label{eq:name}
  y = x_2
\end{align}
```

Strictly speaking, with one equation, `align` is overkill. Its typical application is when you would like to align two or more equations on consecutive lines. The syntax then is

```
\begin{align}
  y &= x_2  \label{eq:name1}  \\
  z &= x_3  \label{eq:name2}
\end{align}
```

where `&` is the alignment tab that expresses where the equations are to be aligned (in this case at the equal signs). In the given example, both equations will be given a unique equation number. However, this is often undesirable, especially if the second equation is really just a simplification of the first, as in

```
\begin{align}
  y &= 0 - x_2  \\
    &= x_2
\end{align}
```

In this case, it is more common to want to number just one equation, which can be done with the syntax

```
\begin{align}
  y &= 0 - x_2  \label{eq:name}  \\
    &= x_2 \nonumber
\end{align}
```

Note that there is no need to label every equation, even if it is numbered. You'll only want to label equations to which you'll want to refer later.

If you want equations to be numbered as *subequations*, like "(2a)", "(2b)", etc., then use the syntax

```
\begin{subequations}
  \begin{align}
    y &= x_2  \label{eq:name1}  \\
    z &= x_3  \label{eq:name2}
  \end{align}
\end{subequations}
```

Finally, if you want no numbers, use the syntax

```
\begin{align*}
  y &= x_2  \\
  z &= x_3
\end{align*}
```

Resource 1.7 Math syntax

LaTeX math syntax must always be included in a math environment (see above) and is pretty straight-forward. This reference page, beginning with math symbols is very useful. The usual math operators like $+$ and $-$ are the obvious syntax, but others are more subtle. If you'd like an explicit "times" operator or cross product, you'd use the syntax `x \times y`, for instance.

Vectors are best expressed with a boldface font. The best way to do this is to use the include `bm` package with the command `\bm{x}`. One advantage of this is that it works also for Greek symbols like `\alpha`, `\beta`, etc.

Powers and indices use the common syntax `x^2` and `x_2`. If more than one character is in a numerator or denominator, braces `{}` must be used, like `x^{2y}`.

Fractions are usually expressed as either `numerator/denominator` (for inline fractions) or `\frac{numerator}{denominator}` (for display fractions).

Sums and integrals can have the syntax `\sum_{i=1}^{n}` and `\int_0^t`.

Matrices can be included with the syntax (for a two-by-two)

```
\begin{bmatrix}
  m_{11}  &  m_{12}  \\
  m_{21}  &  m_{22}
\end{bmatrix}
```

The "Detexify" service can be used to find other symbols you may need. Simply sketch the the symbol you're looking for and AI magic[5] takes care of the rest.

[5] Not actually magic.

Resource 1.8 Adding your own references

Resource 1.8.1 Setting up references database

Another powerful feature of LaTeX is the inclusion of references via BibTeX. The references you would like to include in your document should be added to the `.bib` file via a reference manager such as JabRef (Windows) or BibDesk (OS X). You can open your `.bib` reference database in one of these apps and add them manually or import them from the internet. Google Books includes downloadable BibTeX references for the books it has in its database. Also, if you're referencing an academic paper, usually the journal's website allows you to download the BibTeX reference file.

On university computers you will be unable to install JabRef. However, it can be run from a flash/thumb drive. Download the `.jar` installation file here. Place the file on your flash drive. Double-click the file (now on the flash drive) and navigate to `Options > Preferences`, then, under the `General` section, check the box `Load and save references from/to jabref.xml on startup (memory-stick mode).` Now you can run JabRef to manage your references on most computers—from your flash drive!

Both JabRef and BibDesk allow you to copy to the clipboard a text reference, like

```
@article{Fedrizzi2015,
Author = {Marcus Fedrizzi and Julio Soria},
Journal = {Measurement Science and Technology},
Number = {9},
Pages = {095302},
Title = {Application of a single-board computer
as a low-cost pulse generator},
Url = {http://stacks.iop.org/0957-0233/26/i=9/a=095302},
Volume = {26},
Year = {2015}}
```

and paste it into the main window. It should create a new entry for you.

Alternatively, you can simply directly edit the `report.bib` file. You can paste in a text reference, like the one shown above, at the end of the file. If you aren't editing it directly in Overleaf, make sure you use a plain text editor like `Notepad` on Windows or `TextEdit` on a Mac. (Don't use `WordPad`, `Word`, or any other "rich text" editor.) This is the quick and dirty method.

Once you have your reference in the database, you must make sure it has a BibTeX *key*. I usually make my keys the author's last name and the year of the publication, like `Picone2015` or `Mathers2000`. Each key must be unique, so sometimes you have to include a letter postfix like `Picone2015a` or `Picone2015b`. These keys are how we will refer to the reference in the text.

Resource 1.9 Including the references database in your document

In the template `report.tex` file I've pointed to the `report.bib` file in the lines

```
\bibliographystyle{plainnat}
\bibliography{report}
```

This bibliography style `plainnat` is a good one, so there's usually no need to tweak it. The location of the **`\bibliography{}`** command is where the bibliography section of the document will be included. It is usually after the main contents in the report, and before appendices.

Resource 1.10 Citing sources in the text

Now you're ready to cite sources in the text. The usual way to do this is at the end of a sentence or paragraph with the syntax

```
\citep[p.~77]{Picone2015}
```

The `[p.~77]` is an optional argument, but references a specific page number. Leave it out if you don't want to reference a page number. The key `Picone2015` identifies which source you're citing from your database. This command gives a parenthetical inline citation like (Picone, p. 77).

All the sources that you reference in your text will be listed automatically in the "References" section of your document. If there is a source in your database that you don't reference, it will not be listed.

Overleaf automatically handles updating your references. If you're using TeXShop as your editor with `pdflatexmk` as your build protocol, the references will be updated automatically on each build. If you're using TeXWorks as your editor with `pdflatex` as your build protocol, whenever you add a new reference to the text or change the details on a reference in your reference manager, you will have to build with `pdflatex` once, then build with BibTeX once, then build with pdflatex once or twice more. This will update all the references in your document. If you encounter a reference in your output pdf with a question mark like "(?)", it usually means you need to run BibTeX again.

Resource 1.11 Adding your own tables

LaTeX tables are pretty cumbersome. An example is included in the template file using the tabularx environment. Another useful environment is the standard tabular. I usually have to review this documentation when I make a table.

The following website provides a nice way to enter and edit your table before putting it into LaTeX syntax:

www.tablesgenerator.com.

Resource 1.12 Including full-page pdf documents

If you need to include one or more pages from a pre-existing, multi-page pdf document, you can use the following process. You need the pdfpages package and use the following command to include all pages

```
\includepdf[ % put this where you want the pdf
  pages=-, % all pages
  frame, % frame the pages
  scale=.9, % scale the pages
  pagecommand={\pagestyle{fancy}} % make it sexy
]{
  file.pdf % this is the pdf file name
}
```

If only certain pages are to be included, these can be listed as described in the following example. To include pages 1–3, a blank page, 5, and 9, use the command

```
\includepdf[pages={1-3,{},5,9}]{file.pdf}
```

Box 01.4 Overleaf or local?

If you'd like to build LaTeX documents locally, proceed with the rest of the tutorial. Otherwise, you're done!

Resource 1.13 Installing LaTeX locally

The Cebula Hall 101 computers have LaTeX (TeX Live) installed. The following is for those of you looking to install LaTeX on your personal computer.

For all platforms, I recommend installing the latest version of TeX Live. For each platform, there is a different method to do this.

If you're using a Mac, I recommend installing TeX Live via MacTeX.

If you're using a Windows machine, I recommend installing TeX Live by downloading `install-tl-windows.exe` here and following the directions on the page, which include the quick installation instructions.

If you're running a Unix-based OS, you're on your own, but you're used to it. (And it's actually pretty straightforward.)

Resource 1.14 Building documents

LaTeX provides a markup language that allows you to write a "codey" text document with extension `.tex` that LaTeX *builds* into a `.pdf` (usually). This is a different paradigm from the what-you-see-is-what-you-get (WYSIWYG) word processors (like MS Word) you've used. It requires a little learning curve, but it's well worth it in the end.

Resource 1.15 Editing the template files and your first build

Download the template from Overleaf. Extract the contents of the archive to a convenient directory.

There are multiple options on each platform for editing. On a MS Windows installation, I recommend using the app *TeXWorks*, which will be installed automatically with TeX Live. On OS X, I recommend the app *TeXShop*, which also comes bundled with TeX Live. These apps work in a similar manner that I will now describe.

You will be able to open the provided template file `report.tex` in your editor (TeXWorks or TeXShop). You *should* be able to just "build" immediately: ctrl+T in TeXWorks and cmd+T in TeXShop. You should see in another window a `.pdf` file generated. Congratulations!

Try changing some text from the document and building it again.

Resource 1.16 TeXWorks configurations

Near the green "play" button in the toolbar, select the dropdown menu. You will see several options, like pdflatex. This is the processing tool used (primarily) to build the `.tex` file into a `.pdf`. If you use pdflatex, you should be fine, unless you need to update your bibliography or index. I will address the bibliography question below. To ensure all your changes have been included in the build, it's best to run pdflatex twice (occasionally thrice).

Resource 1.17 TeXShop configurations

Near the Typeset button in the toolbar, select the dropdown menu. You will see several options, like LaTeX and pdflatexmk. These are processing tools used (primarily) to build the `.tex` file into a `.pdf`. I recommend pdflatexmk for its ease of use. If you choose to use LaTeX, you will have to also have to execute the BibTeX processing tool on the file each time you change your references. You will also need to run LaTeX twice at the end.