COMPASS Workshop on LaTeX and Overleaf

Erik Wessel

August 22, 2024

1 Introduction

1.1 Why use code to make documents?

Everyone at this workshop has plenty of experience with word processors. All of you have certainly spent hours writing in Word, Google Docs, or Pages. These editors are simple to use: you just type your text, instantly see what the final document will look like, and if you need to change the formatting you click to select text and there are menus and buttons to change the fonts, formatting, etc.

However, many technical documents aren't made this way. The reason is subtle: we expect the documents we read to be formatted in specific ways. For example, if you are reading a textbook, you expect the figures to be numbered, and referred to by number. The section headers should all have the same font, and they should be numbered as well, and those numbers should be in the table of contents. The references should be cited in a standard way, and if there are equations, they should be printed in a very clear, uniform way, rather than hand-drawn.

The common element is that there are many rules that our documents have to follow exactly. But in word processors like Microsoft Word, Google Docs, or Pages, these rules are hard to see, and even harder to change. In particular, typing equations as complicated as the ones you see in textbooks is very hard to do with buttons and menus.

This is the motivation for a different way of writing documents, one that separates the content from the formatting, and allows you to see and control the formatting rules explicitly, like writing computer code. This is why, especially for technical documents featuring equations, we still use LaTeX in many technical fields, even though it is much older than Word, Google Docs, and Pages, and a little slower to learn.

1.2 What is LaTeX?

LaTeX (pronounced *LAH-TEK*) is a software system for typsetting text. It is built on TeX (pronounced *TEK*), which is a typesetting language invented by Stanford Computer Science Professor Donald Knuth in the late 70's. LaTeX defines a large number of macros (special commands that are short-hand ways of invoking other commands), that add new features to TeX and make it easier to use.

With LaTeX, it is possible to write files (traditionally with a .tex extension, although this is optional) that describe the structure of a document, and specify what the formatting rules will be. Typically, you write the content, and specify the document class (with the command `\documentclass{}`) that contains all the formatting details. But you can also create your own document class files, so everything is completely customizable! When you want to see what your document looks like, you use a program that *compiles* it into another format, usually PDF.

1.3 What is Overleaf?

Overleaf is essentially Google Docs for LaTeX documents. This makes it easy to share and collaborate on documents. In addition, since Overleaf lets you create and compile documents all in your browser, it is the easiest way to get started with LaTeX. It also provides lots of friendly documentation and templates, and useful editing tools.

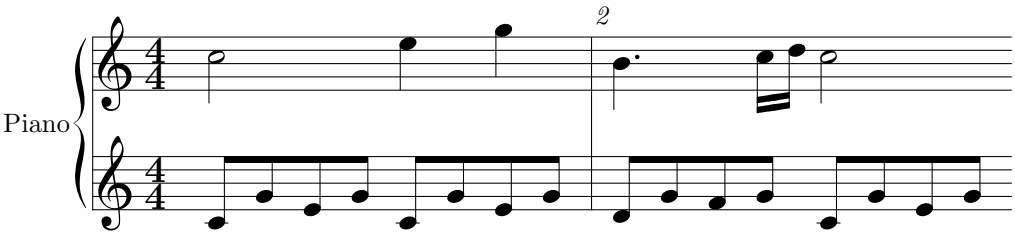
1.4 What is possible?

The programmatic approach to typesetting seems to imply robotic inflexibility–but nothing could be further from the truth! LaTeX is code, and code is $hac{k}a^{bl}e$.

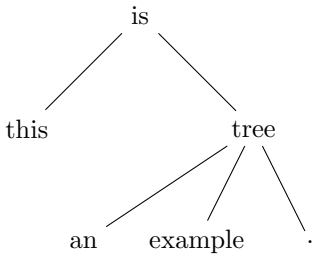
It is possible to do $\mathbf{c_{r}a_{a_{z}}a_{y}}$ things with relative ease!

For example, you can dramatically illustrate that the digits of π go on forever: $3.14159265358979323846264338327950288...$

You can write sheet music:

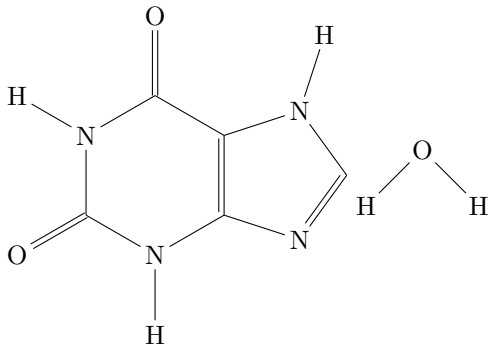


Diagrams:



This is an example tree.

Molecular structures:



Code snippets:

```
def fibonacci_of(n):
...     if n in {0, 1}:  # Base case
...         return n
...     return fibonacci_of(n - 1) + fibonacci_of(n - 2)  # Recursive case
```

Fancy equations:

$$ {}_{s_1}Y_{\ell_1 m_1}\ {}_{s_2}Y_{\ell_2 m_2} = \sum_{S,L,M} \sqrt{\frac{(2\ell_1+1)(2\ell_2+1)}{4\pi(2L+1)}} \langle \ell_1 s_1, \ell_2 s_2 | LS\rangle \langle \ell_1 m_1, \ell_2 m_2 | LM\rangle\, {}_S Y_{LM}. \tag{1} $$

The equations can even be embedded in-line: $\vec{F} = m\vec{a}$.

And you can also do Chemical equations: $\mathrm{Na_2SO_4} \xrightarrow{\mathrm{H_2O}} \mathrm{Na^+ + {SO_4}^{2-}}$

These examples are just the tip of the iceberg!

2 Getting Started

The best way to learn LaTeX is to start using it. First we will create a document in Overleaf. Go to https://www.overleaf.com. If you've never been there before, you will see the login page. Either log-in with your account on another service, or create a new account with overleaf.

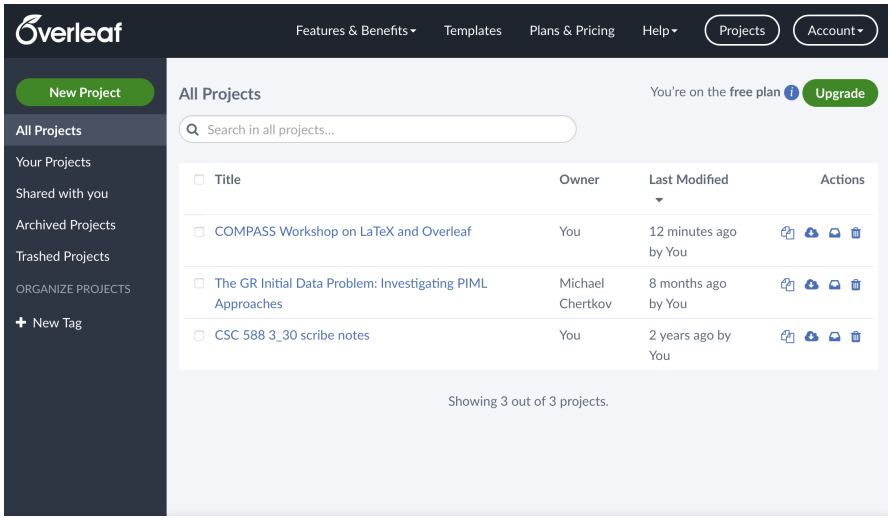


Figure 1: The home project directory of an Overleaf account.

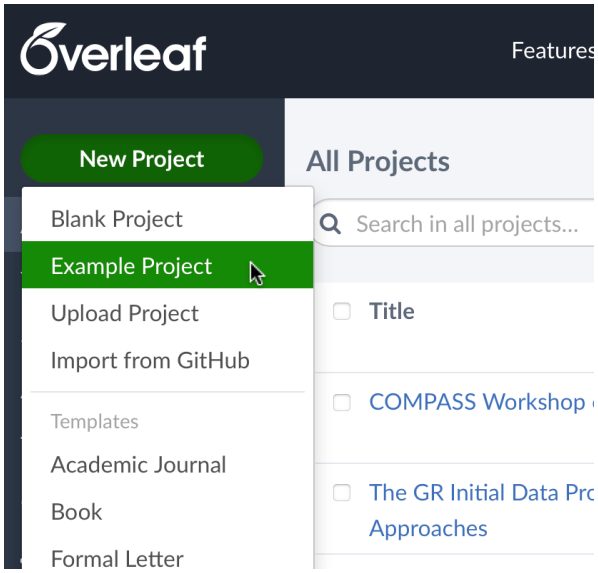


Figure 2: Select the "Example Project" template.

Once you have created your account, you should see your main overleaf project directory, which will look something like the screenshot (Figure 1).

The first thing we will do is use the new project button to create a new document. In the drop-down menu select the "example" option from the available templates (Figure 2)

The document will be created and opened in Overleaf's editor. This will be a split-panel view with the document file on the left, the source code of the currently selected .tex file in the center, and the rendered document on the right. Next, we'll read through this document to understand it.

2.1 Basics of TEX

Lets look at the beginning of the example document. The first few lines look like this:

```
1 \documentclass{article}
2
3 % Language setting
4 % Replace 'english' with e.g. 'spanish' to change the document language
5 \usepackage[english]{babel}
6
```

```
% Set page size and margins
% Replace 'letterpaper' with 'a4paper' for UK/EU standard size
\usepackage[letterpaper,top=2cm,bottom=2cm,left=3cm,right=3cm,marginparwidth=1.75cm]{
    geometry}

% Useful packages
\usepackage{amsmath}
\usepackage{graphicx}
\usepackage[colorlinks=true, allcolors=blue]{hyperref}

\title{Your Paper}
\author{You}

\begin{document}
\maketitle

\begin{abstract}
Your abstract.
\end{abstract}

\section{Introduction}

Your introduction goes here! Simply start writing your document and
```

As we can see, the TEX language is very simple to understand. Inside the document, plain text is the content of the document body. Back-slashes (\) denote macros and commands. The required arguments for a command are usually surrounded by curly braces (`{}`), while optional arguments use square braces (`[]`). The `%` symbol marks comments, which are notes to the writer that are ignored by the compiling program.

We can see that the very first command, `\documentclass{article}`, sets the style of this document. This determines many of the formatting rules, fonts, page sizes, etc. You can create your own document classes, but it is often wise to use an existing one at first.

After that, there are `\usepackage{}` commands. These are what unlock a lot of the magic of LATEX. Packages are collections of macros (LATEX commands) that allow you to do complex things simply. For example, here there are packages that allow language-aware formatting, adjust the paper size, enable math typsetting, and allow graphics to be embedded.

Next, the `\title{}` and `\author{}` commands set the title and author. It is important to realize that these commands do not draw the title! Instead, the commands just supply information about the title and author. The title is drawn once the document is made, by the `\maketitle` command. Comment this command out and re-compile: the title and author will be gone. Likewise, if you attempt to call `\maketitle` without the `\title{}` and `\author{}` set, the compilation will result in errors.

Right before the `\maketitle` command, something very important happens: we begin the document, using the `\begin{document}` command. The `\begin{}` command is used in LATEX to signal that the block of text below it follows special rules. In this case, it follows the rules determined by the document class. Next, the abstract begins with `\begin{abstract}`. This is necessary because the abstract is formatted a little differently than the rest of the document, and has special rules that need to be applied. Later on, we'll see the `\begin{}` command used to start all sorts of special sections, including figures, lists, tables, equations, and code. When the abstract is concluded, this is signaled to LATEX with the corresponding `\end{abstract}` command. Not closing every `\begin{}` with a corresponding `\end{}` will cause an error when compiling.

Finally, we have the `\section{}` command, which takes the section title as its required argument. This command starts out a section, with the text below belonging to the section until the next `\section{}` command is reached. Sections can also include subsections, via the `\subsection{}` command, and so on.

If we scroll all the way to the bottom of the example document, we see:

```
If you have an \href{https://www.overleaf.com/user/subscription/plans}{upgraded
    account}, you can also import your Mendeley or Zotero library directly as a \verb|.
    bib| file, via the upload menu in the file-tree.

\subsection{Good luck!}

We hope you find Overleaf useful, and do take a look at our \href{https://www.overleaf
    .com/learn}{help library} for more tutorials and user guides! Please also let us
```

```
    know if you have any feedback using the Contact Us link at the bottom of the
    Overleaf menu --- or use the contact form at \url{https://www.overleaf.com/contact
    }.
115
116 \bibliographystyle{alpha}
117 \bibliography{sample}
118
119 \end{document}
```

It's worth noting a few things. First, unlike the `\begin{}` command, the `\section{}` commands don't need a corresponding end: LaTeX can figure out where the sections end, because sections only end when another section at the same level starts.

Next, we see two commands that set up the bibliography. The first, `\bibliographystyle{alpha}`, chooses the style that will be used for cross-references and bibliography entries, while the second `\bibliography{sample}` tells LaTeX that the file named `sample.bib` contains all the information for the bibliography. `.bib` files are databases that collect all the citation information you need and keep it out of site. While they can be read and written by hand, it is often better to use a special tool to generate and update them, and we will be introduced to some later.

Finally, the document is ended with the `\end{document}` that closes the block started by `\begin{document}` near the beginning.

2.2 More complex commands

In the remainder of the example document, you will see examples of the commands that make,

- Lists
- Tables
- Figures
- Equations
- Citations
- ... etc.

The best way to learn is to read these examples, and try making changes. The following exercises are some suggestions to try.

2.2.1 Exercises

1. In the left panel, scroll to the "How to write Mathematics" section.
   - Notice how mathematics must be written in *mathmode*,
     activated by `$$` inline or a `\begin{equation} - \end{equation}` block.
   - Add a new *in-line expression* to this section: $e^{i\pi} = -1$
   - Add your own *equation* to this section: $\sqrt{1+x} = 1 + \frac{1}{2}x - \frac{1}{8}x^2 \cdots$
   - Change the fractions in the existing equation so that both sides are over $2n^2$ instead of n.
2. Make a section of text *italic* (Hint: use the buttons above the editor window)
3. Make a section of text *bold*
4. Change one of the section or subsection titles.
5. Scroll to the section "How to include figures"
   - Change the Figure caption.
   - Upload your own image and replace the existing one.
   - Make a second Figure.

6. Scroll to the "How to add lists" section
    - Add an item to the list
    - Add an in-line equation to the list
    - Look up how to make the list numbered instead of bullet-points
7. Scroll to the "How to add tables" section
    - Change the numbers in the table
    - Add a new row to the table
    - Create a second table, by copying and modifying the first one.
    - Put an in-line equation of your choice into the table.

If you don't know how to do something, look for an example in the document. If you can't find one, search the internet. Searching will bring up many helpful forum posts and tutorials on LaTeX. There's a good chance you will find yourself back at Overleaf, because in addition to providing a document editing and collaboration system, Overleaf also provides some of the best LaTeX documentation around!

This document lists many useful mathematical symbols and how to write them in LaTeX: http://tug.ctan.org/info/undergradmath/undergradmath.pdf. If you can't find what you're looking for there, do a web search or try this handy website: https://detexify.kirelabs.org

Finally, because of the large number of LaTeX examples and tutorials on the web, ChatGPT (and other LLMs) are pretty good at writing LaTeX code as well! If you have a very specific thing you want, such as a table structured in a specific way, and aren't sure how to begin, describe what you want to a chatbot in words. It should produce well-commented code which has a good chance of compiling and doing roughly what you want. You can read it, understand how it works, and modify it as you see fit!

2.3 References

One of the most useful features of LaTeX is the ability to refer to elements by numbered references, without having to manually assign the numbers. When writing the document you assign a named label to the element you want to references using the `\label{}` command, then you can refer to that label again with the `\ref{}` command. For example, I can refer to subsection 2.2 by placing this label in its title:

```
148 \subsection{More complex commands\label{section:complex}}
```

Then calling `\ref{section:complex}` when I need the section number. Or I can write an equation,

$$E^2 = p^2c^2 + m^2c^4. \tag{2}$$

And then I can reference the equation above as Equation 2 by doing this:

```
172 \begin{equation}\label{eq:rel_energy_momentum}
173 E^2 = p^2 c^2 + m^2 c^4.
174 \end{equation}
175 And then I can reference the equation above as Equation \ref{eq:rel_energy_momentum}
        by doing this:...
```

Note that starting the reference labels with "`eq:`" and "`section:`" is not necessary, it's just a nice convention to follow to keep the types of references straight.

Exercise: Try it in your own document!

2.4 Citations and BibTex files

We saw from the example document that citations in LaTeX require a bibliography file with a .bib extension–called a BibTex file. Here, we'll delve into what those files do, and how to produce them.

2.4.1 Reading BibTex files

The first thing to know about BibTex files is that they are human readable. Go ahead and open the one from the sample document, you will see this:

```
@article{greenwade93,
    author  = "George D. Greenwade",
    title   = "The {C}omprehensive {T}ex {A}rchive {N}etwork ({CTAN})",
    year    = "1993",
    journal = "TUGBoat",
    volume  = "14",
    number  = "3",
    pages   = "342--351"
}
```

As indicated by the different extension, this file is not in the TEX language, although it is closely related. The `@article{...}` block defines a database entry describing the metadata for an article. Within the entry, we can see that properties needed for the bibliography, such as the author, title, year, etc. are all provided.

At the beginning of the entry there is a piece of information with no associated field: `greenwade93`. This defines the label that will be used to cite the article in the text. This label is never seen by the end reader, so it doesn't need to be pretty. But it does need to be memorable for you, the writer, because this is how you will cite that paper. In the main .tex file, you will see the citation is created by the command `\cite{greenwade93}`. This creates a mark in the text, but this mark isn't the label we see in the .tex file, instead, it is determined by the citation style setting `\bibliographystyle{alpha}`. Try changing this to `\bibliographystyle{unsrt}` and recompiling. Or, say you want something more like APA: in that case, try `\bibliographystyle{apalike}`.

2.4.2 Writing BibTex files

The second thing you should know about .bib files is that they are human writable. Given the example above, go ahead and add a second reference into the bibliography–just make something up. Once the reference exists, you can cite it in your text. If you don't cite it, not only will a marker for it not appear in the body of the text, but it also won't appear in the bibliography: *the bibliography only shows entries from the .bib file that have been cited.*

2.4.3 Managing BibTex files

The third thing you should know about .bib files, is that humans shouldn't read or write them! They are database files, the domain of machines. It is far easier, and better practice, to use special software tools to create the .bib files for your research.

One stand-alone, cross-platform tool that can do this is JabRef (https://www.jabref.org). This program creates and views .bib files in a graphical interface. Most usefully, JabRef is very good at pulling all the needed information for a .bib entry off of the internet! Just pasting a DOI link or other reference typically is enough to automatically generate the whole entry.

A more involved option is to use Zotero (https://www.zotero.org). Zotero is an extremely useful application for managing your research sources. You can create many collections of references, store your annotated files, notes, and biblographic metadata all in one place. Zotero comes with a browser extension that allows you to create a library entry from whatever page you are looking at in your browser: automatically retrieving the publication information and downloading the text if it can find it. But what makes this work really well with LATEX is a Zotero extension called Better BibTex (https://retorque.re/zotero-better-bibtex/). Better BibTex allows you to export .bib files from Zotero that are based on the metadata Zotero has collected. This allows you to instantly make your Zotero library citable from any LATEX project, and you can even set up a .bib file to be continuously synced whenever you make changes to the associated Zotero library.

2.5 LATEX Challenges

Test your ability to figure out how to do stuff in LATEX with these ***challenge questions***!

- Figure out how to make a multi-line equation and reference the individual lines.

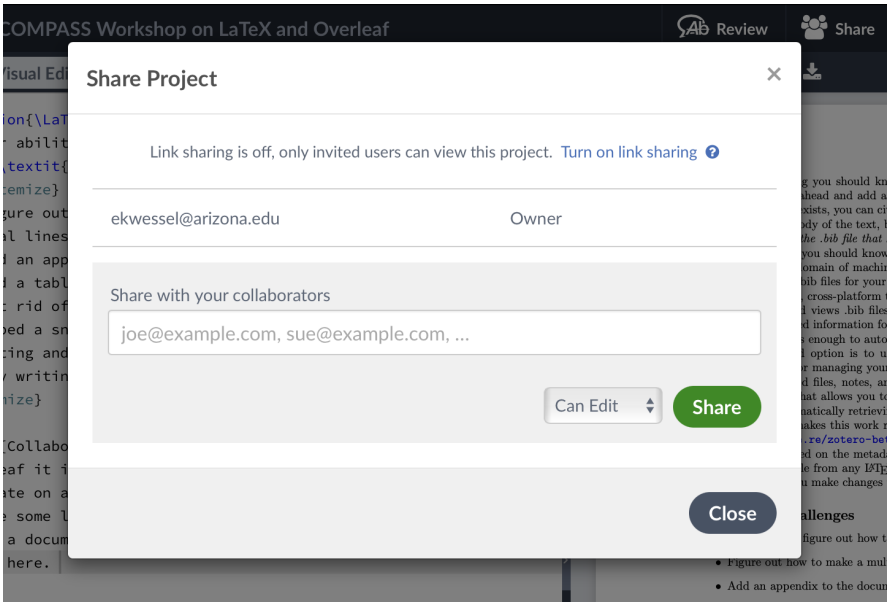


Figure 3: The Overleaf project sharing panel.

- Add an appendix to the document.
- Add a table of contents to the document.
- Look up how to get rid of the section numbers and try to do that.
- Embed a snipped of code in the document with automatic syntax highlighting and line numbering.
- Try writing these crazy equations from my research!

$$\nabla_\mu \left(\int d\Omega n^\gamma n^\mu I \right) = \int d\Omega n^\gamma (j - \alpha I). \tag{3}$$

$$\nabla_\mu \left(\frac{n^\mu I_\nu}{\nu} \right) + \frac{\partial}{\partial \nu} \left(-n^A n^B \omega^I_{AB} n_I I_\nu \right) \tag{4a}$$

$$+ \frac{1}{\sin(\theta)} \frac{\partial}{\partial \theta} \left(\sin(\theta)(-n^A n^B((\cos\phi \omega^X_{AB} + \sin\phi \omega^Y_{AB})\cos\theta - \sin\theta \omega^Z_{AB})) \frac{I_\nu}{\nu} \right) \tag{4b}$$

$$+ \frac{\partial}{\partial \phi} \left((-n^A n^B (\cos\phi \omega^Y_{AB} - \sin phi \omega^X_{AB}) \frac{1}{\sin\theta}) \frac{I_\nu}{\nu} \right) = \frac{j_\nu - \alpha_\nu I_\nu}{\nu}. \tag{4c}$$

3 Collaboration via Overleaf

On overleaf it is possible for multiple authors to share and collaborate on a document. Simply click the "share" button at the top of the screen. The sharing window will pop up (Figure 3). The options are pretty self-explanatory, and not too different from Google Docs.

Unfortunately, free accounts are limited in the number of collaborators they can have. The only way to have more than one collaborator is to work with someone who has one of the paid plans. The details can be found here https://www.overleaf.com/user/subscription/plans.

4 Advanced Topics

4.1 Using LaTeX without Overleaf

We saw in section 3 that Overleaf charges for using it's collaboration features for more than one person, which is definitely an inconvenience.

However, there is a way around this problem. LaTeX is *free software*, it existed long before Overleaf, and we don't need to be dependent on it to use LaTeX! It is possible to install LaTeX compilers on your own machine. Because LaTeX is essentially *source code* for your documents, you can edit it with any tool you use to edit code: Vim, Emacs, Sublime Text, VSCode, even jupyter if you want to be crazy! Or, you can download a special purpose LaTeX editor, there are several to choose from.

Then, if you want to collaborate with someone else, you can do so the way you collaborate on code by creating a git repository on GitHub or Bitbucket.

The key to doing this is getting LaTeX on your machine. A good LaTeX distribution is TeX Live, which has installation instructions for all major platforms at this link https://www.tug.org/texlive

Once you have installed LaTeX on your own machine, you can call the compiler manually. One was is to use a LaTeX editor, which some versions of TeX Live come with, to build the document. Another way is to call a compiler, such as `pdflatex` in the command line, and pass it your `.tex` file, for example: `pdflatex mydocument.tex`. The details here will depend on exactly what you have installed and what system you are on, but once this is set up, you are free to build LaTeX documents on your own without needing Overleaf.

4.2 Defining your own commands / macros

In LaTeX, most commands that begin with `\` are what are programmers call "macros". Macros are pieces of code that stand in for other pieces of code, so that, when code is run, the macros are replaced with more complicated statements, and the resulting program is run. And LaTeX has a command (itself really a macro) that lets you define your own command macros:

```
\newcommand{\<NAME OF COMMAND>}{<LaTeX CODE THAT DEFINES COMMAND>}
```

For example, lets say I'm writing some Calculus notes and I'd like to save the hassle of writing out partial derivatives. I could define:

```
\newcommand{\ddx}{\frac{\partial}{\partial x}}
```

And then later in an equation do:

```
$\ddx \sin(x) = \cos(x)$
```

The result would be $\frac{\partial}{\partial x}\sin(x) = \cos(x)$.

In fact, this system is so powerful that all of LaTeX is build in it! LaTeX in fact is *not a language at all*: the only language here is TeX, which defines the basic formatting commands. Instead, LaTeX is a huge bundle of macro definitions that enable a large amount of functionality very simply.

For details on how to go further and make more complicated commands, see the wonderful Overleaf article on the topic to get started: https://www.overleaf.com/learn/latex/Commands.

In additon, there is a related article that teaches you how to create your own *environments*: https://www.overleaf.com/learn/latex/Environments.

4.3 Creating your own document class

Finally, it may come as no surprise that you can create your own document class in the form of custom `.cls` files. For a getting started guide on doing this, go here: https://www.overleaf.com/learn/latex/Writing_your_own_class.