This is My First LaTeX Document

Pizofreude*

January 2020

*Self-funded by [Pizofreude](https://ko-fi.com/pizofreude).

Contents

1 Quick Introduction

First document. This is a simple example, with no extra parameters or packages included.

This is second paragraph. A bit short but no body is perfect and I am nobody.

This is the third paragraph that I wrote wholeheartedly without much thought or in fact any thought at all.

There are currently Title, Author, and Date in this first LaTeXdocument!

Not a comment anymore, and now we know % indicates comment in LaTeX.

Just as I thought, new line or paragraph will end the comment command. Great and very intuitive!

This is how to **bold**, *italic*, and underline text in LaTeX. We can combine it to our heart contents.

For example this is how to ***bold, italic, and underline*** text all at once!

Emphasis is another great text format. If used in *normal text*, it will *italicized* it. If used in **bold text**, it will ***bold and italicized it***. If used in *italic text*, it will present it as normal text.

Note: some packages, such as Beamer, change the behaviour of the \emph command. We can use \ to avoid escape character. E.g. &, { and } as well as at the end of \ to introduce space afterwards.

2 Adding Image

You need to upload images to your Overleaf project first before embedding it. \usepackage{graphicx}: LaTeXpackage to import graphics in preamble. \graphicspath{{images/}}: In preamble, it configures the graphicx package filepath. It is recommended to create a directory name images in the working directory. \includegraphi- cs {imageName}: In body. it loads the imageName. Although the full file name, including its extension, is allowed in the \includegraphics command, it's considered best practice to omit the file extension because it will prompt LATEX to search for all the supported formats. Here's an example an image by uploading it from local machine to Overleaf:

Generally, the graphic's file name should not contain white spaces or multiple dots; it is also recommended to use lowercase letters for the file

Figure 1: Hex logo.

Figure 2: PulseX logo.

extension when uploading image files to Overleaf. Usage of camelCase for imageName is also recommended. Here's another example of image by URL:

Another method of adding image amongst other options is inserting an image with cross-reference in the paragraph completed with page number where it is positioned. As you can see in Figure 3 and this image can be found on page 5.

There are several noteworthy commands in the example:

\includegraphics

$$width = 0.5ntextwidth$$

{PulseChain}: This form of \includegraphics instructs LaTeXto set the figure's width to 50% of the text width—whose value is stored in the \textwidth command.

Figure 3: Pulsechain logo.

\caption{Pulsechain logo.}: As its name suggests, this command sets the figure caption which can be placed above or below the figure. If you create a list of figures this caption will be used in that list.

\label{fig:Pulsechain logo.}: To reference this image within your document you give it a label using the \label command. The label is used to generate a number for the image and, combined with the next command, will allow you to reference it.

\ref{fig:Pulsechain logo.}: This code will be substituted by the number corresponding to the referenced figure.

Images incorporated in a LaTeXdocument should be placed inside a figure environment, or similar, so that LaTeXcan automatically position the image at a suitable location in your document.

Further guidance is contained in the following Overleaf help articles Positioning of Figures and Inserting Images.

3 Creating lists in LaTeX

You can create different types of list using *environments*, which are used to encapsulate the LaTeXcode required to implement a specific typesetting feature. An *environments* starts with \beginenvironment-name and ends with

\endenvironment-name where *environment-name* might be figure, tabular or one of the list types: **itemize** for unordered lists or **enumerate** for ordered lists.

3.1 Unordered lists

```
 % The verbatim environment is used to write LaTeX syntax without
processing it.
\begin{itemize}
  \item The individual entries are indicated with a black dot, a
  so-called bullet.
  \item The text in the entries may be of any length.
\end{itemize}
```

The output will be as follows:

- The individual entries are indicated with a black dot, a so-called bullet.
- The text in the entries may be of any length.

3.2 Ordered lists

Ordered lists use the same syntax as unordered lists but are created using the enumerate environment:

```
\begin{enumerate}
  \item This is the first entry in our list.
  \item The list numbers increase with each entry we add.
\end{enumerate}
```

The output will be:

1. This is the first entry in our list.
2. The list numbers increase with each entry we add.

As with unordered lists, each entry must be preceded by the \item command which, here, automatically generates the numeric ordered-list label value, starting at 1. For further information, refer to:

- Demonstrating various types of LaTeX list
- Comprehensive List Styles in LaTeX

4 Writing Code Syntax in LaTeX

Inline code for LaTeXsyntax with **WYSIWYG** method can be achieved by using \textbackslash command. For *code block*, we can use the *verbatim environment* to write LaTeXsyntax as a *code block* in LaTeXand ignore the LaTeXcommands. The *verbatim environment* is used to display text exactly as it is written, without interpreting any LaTeXcommands.

Here's an example of how to use the verbatim environment:

```
\begin{verbatim}
This is an example of the verbatim environment.
It can be used to display LaTeX code without interpreting any
commands.
For example, \textbf{this} will not be bold.
\\end {verbatim}
```

Alternatively, you can also use the *lstlisting environment* provided by the *listings package*, which is a powerful package for typesetting source code in LaTeXdocuments. It has many options for customization, including syntax highlighting. User can import \usepackage{listings} in the **preamble**.

```
\begin{lstlisting}[language=TeX]
This is an example of the lstlisting environment.
It can also be used to display LaTeX code without interpreting
any commands.
For example, \textbf{this} will not be bold.
\end{lstlisting}
```

The output will be:

```
This is an example of the lstlisting environment.
It can also be used to display LaTeX code without inter-
preting any commands.
For example, \textbf{this} will not be bold.
```

4.1 Code Block for Other Programming & Tech Languages

User can include many other programming languages as well as tech languages such as Markdown and etc pp. We can use bash language as alternative to *verbatim environment* and *lstlisting environment* which looks more professional as well. Refer to these lists for further information:

1. Code Listing in LaTeX
2. Code Highlighting with *minted environment*

5 Adding math to LaTeX

One of the main advantages of LaTeXis the ease with which mathematical expressions can be written. LaTeXprovides two writing modes for typesetting mathematics:

inline math mode used for writing formulas that are part of a paragraph *display math* mode used to write expressions that are not part of a text or paragraph and are typeset on separate lines.

5.1 Inline Math Mode

Here's an example of how to write inline math:

```
\documentclass[12pt, letterpaper]{article}
\begin{document}
In physics, the mass-energy equivalence is stated
by the equation $E=mc^2$, discovered in 1905 by Albert Einstein.
\end{document}
```

The output will be:

In physics, the mass-energy equivalence is stated by the equation $E = mc^2$, discovered in 1905 by Albert Einstein. $InlineMathFormula$.

To typeset inline-mode math you can use one of these delimiter pairs: \(... \), $ ... $ or \begin{math} ... \end{math}, as demonstrated in the following example:

```
\begin{math}
E=mc^2
\end{math} is typeset in a paragraph using inline math mode---as
is $E=mc^2$, and so too is \(E=mc^2\).
```

and the output is:

$E = mc^2$ is typeset in a paragraph using inline math mode—as is $E = mc^2$, and so too is $E = mc^2$.

5.2 Display math mode

Equations typeset in display mode can be numbered or unnumbered, as in the following example:

```
\documentclass[12pt, letterpaper]{article}
\begin{document}
The mass-energy equivalence is described by the famous equation
\[ E=mc^2 \] discovered in 1905 by Albert Einstein.

In natural units ($c = 1$), the formula expresses the identity
\begin{equation}
E=m
\end{equation}
\end{document}
```

and the output:

The mass-energy equivalence is described by the famous equation

$$E = mc^2$$

discovered in 1905 by Albert Einstein.

In natural units ($c = 1$), the formula expresses the identity

$$E = m \tag{1}$$

To typeset display-mode math you can use one of these delimiter pairs: \[... \], \begin{displaymath} ... \end{displaymath} or \begin{equation} ... \end{equation}. Historically, typesetting display-mode math required use of $$ characters delimiters, as in $$ ... display math here ...$$, but this method is no longer recommended: use LaTeX's delimiters \[... \] instead.

5.3 More Math Examples

This subsection demonstrates more range of mathematical content typeset using LaTeX.

```
\documentclass{article}
\begin{document}
Subscripts in math mode are written as $a_b$ and superscripts
```

```
are written as $a^b$. These can be combined and nested to write
expressions such as

\[ T^{i_1 i_2 \dots i_p}_{j_1 j_2 \dots j_q} = T(x^{i_1},\dots,
x^{i_p},e_{j_1},\dots,e_{j_q}) \]

We write integrals using $\int$ and fractions using $\frac{a}{b}$.
Limits are placed on integrals using superscripts and subscripts:

\[ \int_0^1 \frac{dx}{e^x} =  \frac{e-1}{e} \]

Lower case Greek letters are written as $\omega$ $\delta$ etc.
while upper case Greek letters are written as $\Omega$ $\Delta$.

Mathematical operators are prefixed with a backslash as
$\sin(\beta)$, $\cos(\alpha)$, $\log(x)$ etc.
\end{document}
```

These produce the output:

Subscripts in math mode are written as a_b and superscripts are written as a^b. These can be combined and nested to write expressions such as

$$T^{i_1 i_2 \dots i_p}_{j_1 j_2 \dots j_q} = T(x^{i_1},\dots,x^{i_p},e_{j_1},\dots,e_{j_q})$$

or

$$T^{i_1 i_2 \dots i_p}_{j_1 j_2 \dots j_q} = T(x^{i_1},\dots,x^{i_p},e_{j_1},\dots,e_{j_q}) \tag{2}$$

We write integrals using $\int$ and fractions using $\frac{a}{b}$. Limits are placed on integrals using superscripts and subscripts:

$$\int_0^1 \frac{dx}{e^x} = \frac{e-1}{e}$$

or

$$\int_0^1 \frac{dx}{e^x} = \frac{e-1}{e} \tag{3}$$

Lower case Greek letters are written as ω δ etc. while upper case Greek letters are written as Ω Δ.

Mathematical operators are prefixed with a backslash as $\sin(\beta)$, $\cos(\alpha)$, $\log(x)$ etc.

The next example uses the equation* environment which is provided by the amsmath package, so we need to add the following line to our document preamble:

```
\usepackage{amsmath}% For the equation* environment
```

For further information on using amsmath see the help article.

```
\documentclass{article}
\usepackage{amsmath}% For the equation* environment
\begin{document}
\section{First example}

The well-known Pythagorean theorem \(x^2 + y^2 = z^2\) was proved
to be invalid for other exponents, meaning the next equation has
no integer solutions for \(n>2\):

\[ x^n + y^n = z^n \]

\section{Second example}

This is a simple math expression \(\sqrt{x^2+1}\) inside text.
And this is also the same:
\begin{math}
\sqrt{x^2+1}
\end{math}
but by using another command.

This is a simple math expression without numbering
\[\sqrt{x^2+1}\]
separated from text.

This is also the same:
\begin{displaymath}
\sqrt{x^2+1}
\end{displaymath}
```

```
\ldots and this:
\begin{equation*}
\sqrt{x^2+1}
\end{equation*}
\\end{document}
```

which outputs:

5.3.1 First example

The well-known Pythagorean theorem $x^2 + y^2 = z^2$ was proved to be invalid for other exponents, meaning the next equation has no integer solutions for $n > 2$:

$$x^n + y^n = z^n$$

5.3.2 Second example

This is a simple math expression $\sqrt{x^2+1}$ inside text. And this is also the same: $\sqrt{x^2+1}$ but by using another command.

This is a simple math expression without numbering

$$\sqrt{x^2+1}$$

separated from text.

This is also the same:

$$\sqrt{x^2+1}$$

…and this:

$$\sqrt{x^2+1}$$

The possibilities with math in LaTeXare endless so be sure to visit the help pages for advice and more examples on specific topics:

- Mathematical expressions
- Subscripts and superscripts
- Brackets and Parentheses

- Fractions and Binomials
- Aligning Equations
- Operators
- Spacing in math mode
- Integrals, sums and limits
- Display style in math mode
- List of Greek letters and math symbols
- Mathematical fonts

6 Basic document structure

Next, we explore abstracts and how to partition a LaTeXdocument into different chapters, sections and paragraphs.

6.1 Abstracts

Scientific articles usually provide an abstract which is a brief overview/summary of their core topics, or arguments. The next example demonstrates typesetting an abstract using LaTeX's abstract environment:

```
\documentclass{article}
\begin{document}
\begin{abstract}
This is a simple paragraph at the beginning of the
document. A brief introduction about the main subject.
\end{abstract}
\end{document}
```

This example produces the following output:

Abstract

This is a simple paragraph at the beginning of the document. A brief introduction about the main subject.

6.2 Paragraphs and new lines

With the abstract in place, we can begin writing our first paragraph. The next example demonstrates:

how a new paragraph is created by pressing the ”enter” key twice, ending the current line and inserting a subsequent blank line; how to start a new line without starting a new paragraph by inserting a manual line break using the \\ command, which is a double backslash; alternatively, use the \newline command. The third paragraph in this example demonstrates use of the commands \\ and \newline:

```
\documentclass{article}
\begin{document}

\begin{abstract}
This is a simple paragraph at the beginning of the
document. A brief introduction about the main subject.
\end{abstract}

After our abstract we can begin the first paragraph, then press
''enter'' twice to start the second one.

This line will start a second paragraph.

I will start the third paragraph and then add \\ a manual line
break which causes this text to start on a new line but remains
part of the same paragraph. Alternatively, I can use the \verb|
\newline|\newline command to start a new line, which is also
part of the same paragraph.
\end{document}
```

which outputs:

Abstract

This is a simple paragraph at the beginning of the document. A brief introduction about the main subject.

After our abstract we can begin the first paragraph, then press “enter” twice to start the second one.

This line will start a second paragraph.

I will start the third paragraph and then add
a manual line break which causes this text to start on a new line but remains part of the same paragraph. Alternatively, I can use the `\newline`
command to start a new line, which is also part of the same paragraph.

Note how LaTeXautomatically indents paragraphs—except immediately after document headings such as section and subsection—as we see here.

New users are advised that multiple \\or \newlines should not used to "simulate" paragraphs with larger spacing between them because this can interfere with LaTeX's typesetting algorithms. The recommended method is to continue using blank lines for creating new paragraphs, without any \\, and load the parskip package by adding \usepackage{parskip} to the preamble.

Further information on paragraphs can be found in the following articles:

- Paragraphs and new lines
- How to change paragraph spacing in LaTeX
- LaTeX Error: There's no line here to end provides additional advice and guidance on using \\.

6.3 Chapters and sections

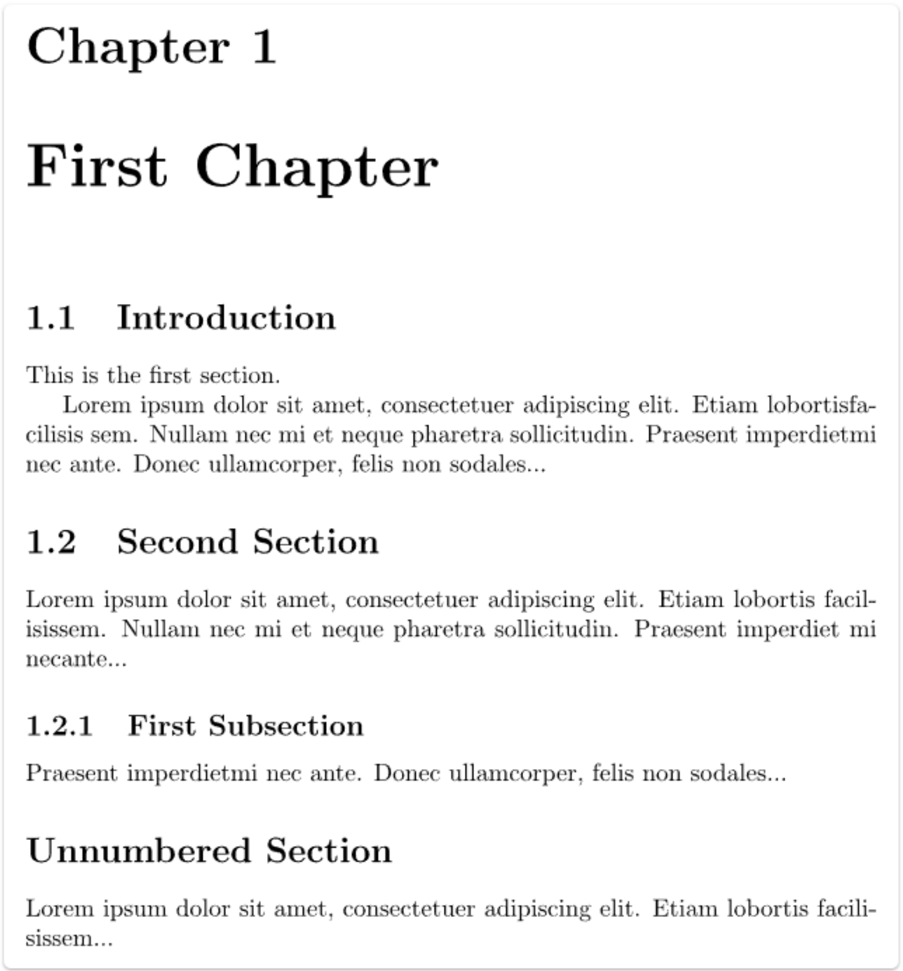
Longer documents, irrespective of authoring software, are usually partitioned into parts, chapters, sections, subsections and so forth. LaTeXalso provides document-structuring commands but the available commands, and their implementations (what they do), can depend on the document class being used. By way of example, documents created using the book class can be split into parts, chapters, sections, subsections and so forth but the letter class does not provide (support) any commands to do that.

This next example demonstrates commands used to structure a document based on the book class:

```
\documentclass{book}
\begin{document}

\chapter{First Chapter}
```

```
\section{Introduction}

This is the first section.

Lorem  ipsum  dolor  sit  amet,  consectetuer  adipiscing
elit. Etiam  lobortisfacilisis sem.  Nullam nec mi et
neque pharetra sollicitudin.  Praesent imperdietmi nec ante.
Donec ullamcorper, felis non sodales...

\section{Second Section}

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Etiam lobortis facilisissem.  Nullam nec mi et neque pharetra
sollicitudin.  Praesent imperdiet mi necante...

\subsection{First Subsection}
Praesent imperdietmi nec ante. Donec ullamcorper, felis non
sodales...

\section*{Unnumbered Section}
Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Etiam lobortis facilisissem...
\\end{document}
```

This example produces the following output:


Chapter 1

First Chapter

1.1 Introduction

This is the first section.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortisfacilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdietmi nec ante. Donec ullamcorper, felis non sodales...

1.2 Second Section

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisissem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi necante...

1.2.1 First Subsection

Praesent imperdietmi nec ante. Donec ullamcorper, felis non sodales...

Unnumbered Section

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisissem...

The names of sectioning commands are mostly self-explanatory; for example, \chapter{First Chapter} creates a new chapter titled First Chapter, \section{Introduction} produces a section titled Introduction, and so forth. Sections can be further divided into \subsection{...} and even \subsubsection{...}. The numbering of sections, subsections etc. is automatic but can be disabled by using the so-called starred version of the appropriate command which has an asterisk (*) at the end, such as \section*{...} and \subsection*{...}.

Collectively, LaTeXdocument classes provide the following sectioning commands, with specific classes each supporting a relevant subset:

```
\part{part}
\chapter{chapter}
\section{section}
```

```
\subsection{subsection}
\subsubsection{subsubsection}
\paragraph{paragraph}
\subparagraph{subparagraph}
```

In particular, the \part and \chapter commands are only available in the report and book document classes. For more information about document structure commands, feel free to visit article about sections and chapters.

7 Creating tables

The following examples show how to create tables in LaTeX, including the addition of lines (rules) and captions.

7.1 Creating a basic table in LaTeX

We start with an example showing how to typeset a basic table:

```
\begin{center}
\begin{tabular}{c c c}
 cell1 & cell2 & cell3 \\
 cell4 & cell5 & cell6 \\
 cell7 & cell8 & cell9
\end{tabular}
\end{center}
```

which produces:

cell1	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

The tabular environment is the default LaTeXmethod to create tables. You must specify a parameter to this environment, in this case {c c c} which advises LaTeXthat there will be three columns and the text inside each one must be centred. You can also use r to right-align the text and l to left-align it. The alignment symbol & is used to demarcate individual table cells within a table row. To end a table row use the new line command \\. Our table is contained within a center environment to make it centred within the text width of the page.

7.2 Adding borders

The tabular environment supports horizontal and vertical lines (rules) as part of the table:

to add horizontal rules, above and below rows, use the \hline command to add vertical rules, between columns, use the vertical line parameter — In this example the argument is —c—c—c— which declares three (centred) columns each separated by a vertical line (rule); in addition, we use \hline to place a horizontal rule above the first row and below the final row:

```
\begin{center}
\begin{tabular}{|c|c|c|}
 \hline
 cell1 & cell2 & cell3 \\
 cell4 & cell5 & cell6 \\
 cell7 & cell8 & cell9 \\
 \hline
\end{tabular}
\end{center}
```

The output is:

cell1	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

Here is a further example:

```
\begin{center}
 \begin{tabular}{||c c c c||}
 \hline
 Col1 & Col2 & Col2 & Col3 \\ [0.5ex]
 \hline\hline
 1 & 6 & 87837 & 787 \\
 \hline
 2 & 7 & 78 & 5415 \\
 \hline
 3 & 545 & 778 & 7507 \\
 \hline
 4 & 545 & 18744 & 7560 \\
 \hline
 5 & 88 & 788 & 6344 \\ [1ex]
 \hline
\end{tabular}
\end{center}
```

which output:

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

Tip!

Creating tables in LaTeXcan be time-consuming so you may want to use the TablesGenerator.com online tool to export LaTeXcode for tabulars.

7.3 Captions, Labels and References

You can caption and reference tables in much the same way as images. The only difference is that instead of the figure environment, you use the table environment.

```
Table \ref{table:data} shows how to add a table caption and reference a table.
\begin{table}[h!]
\centering
\begin{tabular}{||c c c c||}
 \hline
 Col1 & Col2 & Col2 & Col3 \\ [0.5ex]
 \hline\hline
 1 & 6 & 87837 & 787 \\
 2 & 7 & 78 & 5415 \\
 3 & 545 & 778 & 7507 \\
 4 & 545 & 18744 & 7560 \\
 5 & 88 & 788 & 6344 \\ [1ex]
 \hline
\end{tabular}
\caption{Table to test captions and labels.}
\label{table:data}
\end{table}
```

The output is:

Table 1 shows how to add a table caption and reference a table.

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

Table 1: Table to test captions and labels.

8 Adding a Table of Contents

Creating a table of contents is straightforward because the command \tableofcontents does almost all the work for you:

```
\documentclass{article}
\title{Sections and Chapters}
\author{Gubert Farnsworth}
\date{August 2022}
\begin{document}

\maketitle

\tableofcontents

\section{Introduction}

This is the first section.
```

```
Lorem  ipsum  dolor  sit  amet,  consectetuer  adipiscing
elit.   Etiam  lobortisfacilisis sem.  Nullam nec mi et
neque pharetra sollicitudin.  Praesent imperdietmi nec ante.
Donec ullamcorper, felis non sodales...

\section*{Unnumbered Section}
\addcontentsline{toc}{section}{Unnumbered Section}

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Etiam lobortis facilisissem.  Nullam nec mi et neque pharetra
sollicitudin.  Praesent imperdiet mi necante...

\section{Second Section}

Lorem ipsum dolor sit amet, consectetuer adipiscing elit.
Etiam lobortis facilisissem.  Nullam nec mi et neque pharetra
sollicitudin.  Praesent imperdiet mi necante...
\end{document}
```

with the output as seen in this document's TOC after \maketitle command.

Sections, subsections and chapters are automatically included in the table of contents. To manually add entries, such as an unnumbered section, use the command \addcontentsline as shown in the commandline example above.

9 Finding and Using LaTeXPackages

LaTeXnot only delivers significant typesetting capabilities but also provides a framework for extensibility through the use of add-on packages. Rather than attempting to provide commands and features that "try to do everything", LaTeXis designed to be extensible, allowing users to load external bodies of code (packages) that provide more specialist typesetting capabilities or extend LaTeX's built-in features—such as typesetting tables. As observed in the section Adding images, the graphicx package extends LaTeXby providing commands to import graphics files and was loaded (in the preamble) by writing \usepackage{graphicx} command.

9.1 Loading Packages

As noted above, packages are loaded in the document preamble via the \usepackage command but because (many) LaTeXpackages provide a set of options, which can be used to configure their behaviour, the \usepackage command often looks like this:

```
\usepackage[options]{somepackage}
```

The square brackets "[...]" inform LaTeXwhich set of options should be applied when it loads somepackage. Within the set of options requested by the user, individual options, or settings, are typically separated by a comma; for example, the geometry package provides many options to configure page layout in LaTeX, so a typical use of geometry might look like this:

```
\usepackage[total={6.5in,8.75in},
top=1.2in, left=0.9in, includefoot]{geometry}
```

The geometry package is one example of a package written and contributed by members of the global LaTeXcommunity and made available, for free, to anyone who wants to use it.

If a LaTeXpackage does not provide any options, or the user wants to use the default values of a package's options, it would be loaded like this:

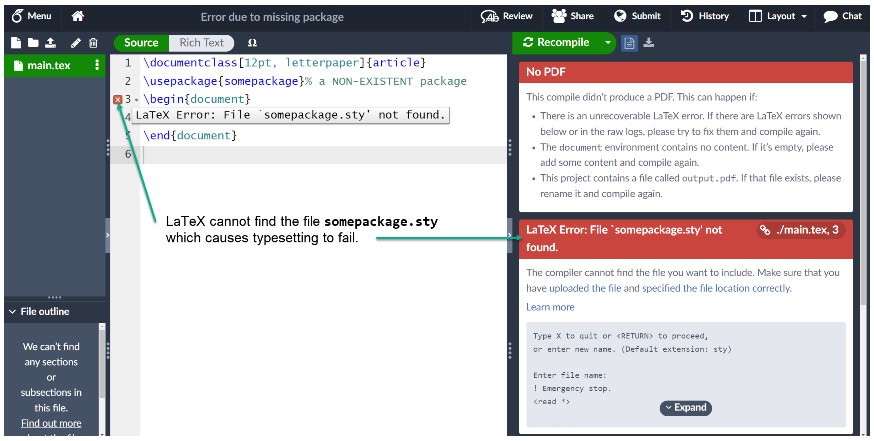
```
\usepackage{somepackage}
```

When you write \usepackage[...]{somepackage} LaTeXlooks for a corresponding file called *somepackage.sty*, which it needs to load and process—to make the package commands available and execute any other code provided by that package. If LaTeXcannot find *somepackage.sty* it will terminate with an error, as demonstrated in the following Overleaf example:

10 Finding Information about Packages: CTAN

Packages are distributed through the Comprehensive TeX Archive Network, usually referred to as CTAN, which, at the time of writing, hosts 6287 packages from 2881 contributors. CTAN describes itself as

> ... a set of Internet sites around the world that offer TEX-related material for download.



```
\documentclass[12pt, letterpaper]{article}
\usepackage{somepackage}% a NON-EXISTENT package
\begin{document}
LaTeX Error: File `somepackage.sty' not found.
\end{document}
```

LaTeX cannot find the file **somepackage.sty** which causes typesetting to fail.

No PDF

This compile didn't produce a PDF. This can happen if:

- There is an unrecoverable LaTeX error. If there are LaTeX errors shown below or in the raw logs, please try to fix them and compile again.
- The document environment contains no content. If it's empty, please add some content and compile again.
- This project contains a file called output.pdf. If that file exists, please rename it and compile again.

LaTeX Error: File `somepackage.sty' not found. ./main.tex, 3

The compiler cannot find the file you want to include. Make sure that you have uploaded the file and specified the file location correctly.

Learn more

```
Type X to quit or <RETURN> to proceed,
or enter new name. (Default extension: sty)

Enter file name:
! Emergency stop.
<read *>
```

Figure 4: Package fails to load.

You can browse CTAN to look for useful packages; for example:

- by topic
- alphabetically (useful if you know the package name)
- You can also use the search facility (at the top of the page).

Packages available on Overleaf: Introducing TEXLive

Once per year a (large) subset of packages hosted on CTAN, plus LATEX-related fonts and other software, is collated and distributed as a system called TEXLive, which can be used to install your own (local) LATEXsetup. In fact, Overleaf's servers also use TEXLive and are updated when a new version of TeX Live is released. Overleaf's TEXLive updates are not immediate but take place a few months post-release, giving us time to perform compatibility tests of the new TEXLive version with the thousands of templates contained in our gallery. For example, here is our TEXLive 2020 upgrade announcement.

Although TEXLive contains a (large) subset of CTAN packages it is possible to find an interesting package, such as igo for typesetting Go diagrams, which is hosted on CTAN but not included in (distributed by) TEXLive and thus unavailable on Overleaf. Some packages hosted on CTAN are not part of TEXLive due to a variety of reasons: perhaps a package is obsolete, has licensing problems, is extremely new (recently uploaded) or has platform dependencies, such as working on Windows but not Linux.

New packages, and updates to existing ones, are uploaded to CTAN all year round but updates to TEXLive are distributed annually; consequently, packages contained in the current version of TEXLive will not be as up-to-date as those hosted on CTAN. Because Overleaf's servers use TEXLive it is possible that packages installed on our servers—i.e., ones available to our users—might not be the very latest versions available on CTAN but, generally, this is unlikely to be problematic.