Latex Reference Document

Contents

1 Common Pitfalls and Things to Know

- If regular text is rendering in italics, check that you are **not** in math mode.
- If arithmetic or math operations are rendering with errors, check that you **are** in math mode.
- If superscripts or subscripts are not rendered together, for example: a_sub or 2^99, make sure you enclose the entire subscript or superscript in curly braces:

  ```
  a_{sub} or 2^{99}
  ```

- If you want to force a space in math mode, simply type a backslash followed by a space:

  ```
  $1, 2, 3, 4$
  ```

  renders as

  $1, 2, 3, 4$

  but

  ```
  $1,\ 2,\ 3,\ 4$
  ```

  renders as

  $1,\ 2,\ 3,\ 4$

- Double backslash - \\- forces a newline in your text, but is usually only needed for specific math/align modes.

  For example:

  ```
  text1 \\text2
  ```

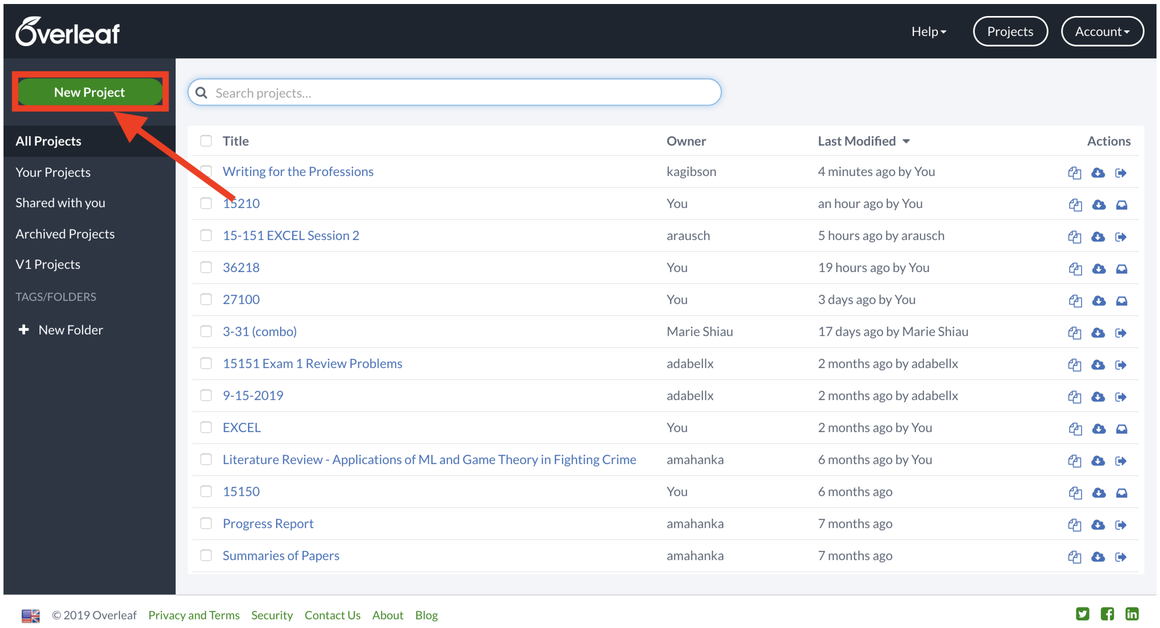
  renders as:

  text1
  text2

- Double click on any section of the rendered PDF to jump to that section of the source code. You can also use the arrow buttons at the top of the division between the editing section and the rendered section to find corresponding text.

2 Creating a Document

Upon logging into the main screen, the following will appear with your list of existing projects:



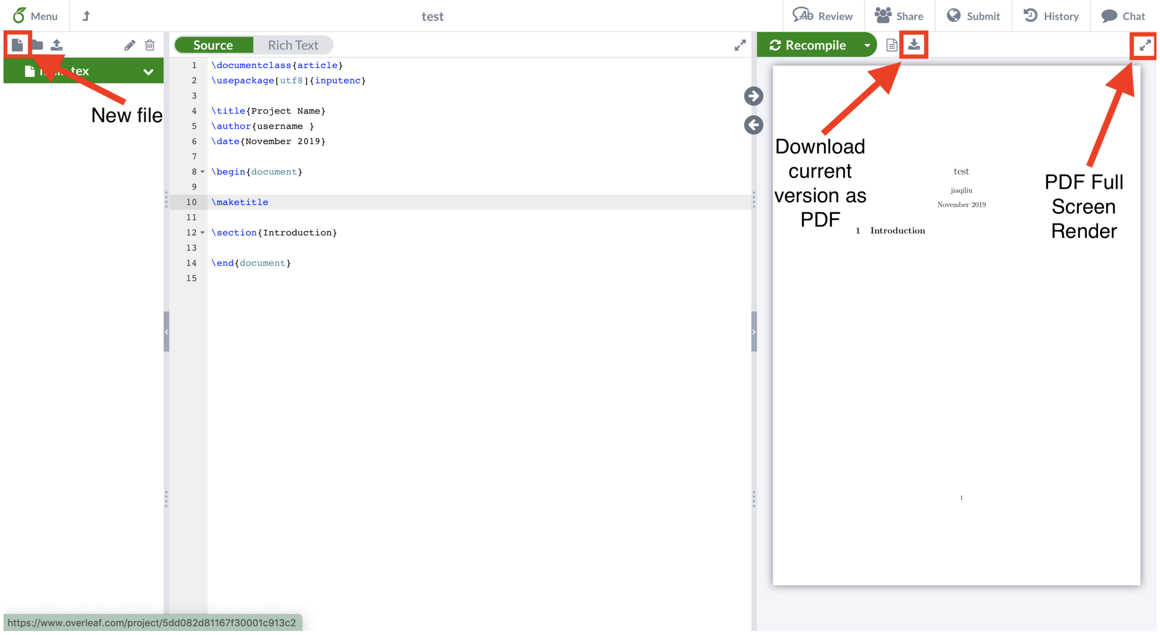
Title	Owner	Last Modified	Actions
Writing for the Professions	kagibson	4 minutes ago by You	
15210	You	an hour ago by You	
15-151 EXCEL Session 2	arausch	5 hours ago by arausch	
36218	You	19 hours ago by You	
27100	You	3 days ago by You	
3-31 (combo)	Marie Shiau	17 days ago by Marie Shiau	
15151 Exam 1 Review Problems	adabellx	2 months ago by adabellx	
9-15-2019	adabellx	2 months ago by adabellx	
EXCEL	You	2 months ago by You	
Literature Review - Applications of ML and Game Theory in Fighting Crime	amahanka	6 months ago by You	
15150	You	6 months ago	
Progress Report	amahanka	7 months ago	
Summaries of Papers	amahanka	7 months ago	

1. Click on "New Project"

2. Select "Blank Project" from the drop-down menu.

3. Type in the project name and click "Create".

The following markdown editor will appear:



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

\title{Project Name}
\author{username }
\date{November 2019}

\begin{document}

\maketitle

\section{Introduction}

\end{document}

```

The layout is divided as follows:

- The left column allows access to various files in the project.

- The middle section is the editor where you write the source code for your document.

- The right-most section is a PDF rendering of your current source code.

Note that arrows point to commonly used buttons for adding a new file in the project, downloading the current file as a PDF, and entering full screen.

3 Formatting

3.1 Sections

The sections commands are built into Latex and do **not** require any additional packages.

- Numbered Sections Hierarchy:

Code	**Output**
\section{Section Name}	**1 Section Name**
\subsection{Section Name}	**1.1 Section Name**
\subsubsection{Section Name}	**1.1.1 Section Name**

- Unnumbered Sections Hierarchy:

Code	**Output**
\section*{Section Name}	**Section Name**
\subsection*{Section Name}	**Section Name**
\subsubsection*{Section Name}	**Section Name**

3.2 Alignment and Lists

Some environments allow you to create bulleted or numbered lists, as well as rows of text/math aligned at specific points.

- Enumerated Lists

  ```
  \begin{enumerate}
          \item Milk
          \item Eggs
          \item Bread
  \end{enumerate}
  ```

  renders as,

  1. Milk
  2. Eggs
  3. Bread

  Note that each \item automatically gives the next number in the list.

- Bulleted Lists

  ```
  \begin{itemize}
          \item Milk
          \item Eggs
          \item Bread
  \end{itemize}
  ```

  renders as,

  - Milk
  - Eggs
  - Bread

  Note that each \item gives the next bullet in the list.

- Aligned Text:

  This environment allows you to align multiple lines in specific places. Since this is a math environment, you must surround text prose in a \text{} command.

```
\begin{align}
        \text{text } & \{math alignment text}\\
        & \text{math alignment text}
\end{align}
```

Note that the & creates the column of alignment and text can go before or after. However, the align environment uses math mode by default.
The above when tranferred to text mode renders as:

text alignment text
math alignment text

3.3 Bold and Italics

Creating bold and italic text is as simple as wrapping text in \textbf{} and \textit{} commands, respectfuly.

```
This was the most \textit{crazy} winter \textbf{ever}.
```

renders as

This was the most *crazy* winter **ever**.

3.4 Quotation Marks

Suppose you want to render quotation marks. You may try writing

```
''Hello, neighbor!''
```

and realize it renders as:

”Hello, neighbor!”

The LaTeXway to write quotes is to write two backticks for the begin quote (``) and two single quotes for the end quote (' '). This way you can control the direction of your quotes.

```
``Hello, neighbor!''
```

renders as

“Hello, neighbor!”

4 Various Modes and Environments

4.1 Math Mode

Knowing how to format math symbols and equations is invaluable knowledge to have when using LaTeX. There are several ways to create math environments, which we will cover below. Remember to always include the **amsmath** package, which includes many useful math symbols.

To use math symbols, write the following command at the top of your document under the `documentclass` command.

```
\usepackage{amsmath}
```

4.1.1 Zoo of Math Modes

1. $$ - Single Dollars

   Inserting text in between two single dollars renders math symbols in place. For example:

   ```
   The equation is $1 + 2 = 3$, and that’s cool!
   ```

   renders as:
   The equation is $1 + 2 = 3$, and that’s cool!

2. $$$$ - Double Dollars

   Inserting text in between two double dollars renders math symbols in the center of the page. For example:

   ```
   The equation is $$1 + 2 = 3.$$
   ```

   renders as:
   The equation is

   $$1 + 2 = 3.$$

   It is customary to put punctuation like periods or commas before the ending $$.

3. the **equation*** environment Inserting text in a **equation*** environment renders that text similar to double dollars (separate line and centered).

   ```
   \begin{equation*}
   1 + 1 = 2
   \end{equation*}
   ```

renders as

$$1 + 1 = 2$$

Note that you can also label your equation by removing the asterisk *.

```
\begin{equation}
1 + 1 = 2
\end{equation}
```

renders as

$$1 + 1 = 2 \tag{1}$$

4. The \[\] environment is a shorthand for the **equation*** environment, and is considered better practice over double-dollars.

```
\[
1 + 1 = 2
\]
```

renders as

$$1 + 1 = 2$$

5. the **align*** environment is special. You can align multiple lines of math equations with the & symbol.

```
\begin{align*}
1 + 1 + 1 + 1 &= 2 + 1 + 1\\
&= 2 + 2\\
&= 4\\
\end{align*}
```

renders as

$$\begin{aligned}
1 + 1 + 1 + 1 &= 2 + 1 + 1\\
&= 2 + 2\\
&= 4
\end{aligned}$$

4.1.2 Spacing in Math Mode

If you want to force a space in math mode, simply type a backslash followed by a space:

```
$1, 2, 3, 4$
```

renders as

$1, 2, 3, 4$

but

```
$1,\ 2,\ 3,\ 4$
```

renders as

$1,\ 2,\ 3,\ 4$

4.1.3 Prose Text in Math Mode

If you need to write text inside some math mode as if it were outside math mode, then you should wrap your text inside the \text{} command.

```
$ X = \{ x :\\text{is infinite} \} $
```

renders as

$X = \{ x : x \text{ is infinite} \}$

4.2 Common Math Symbols

We have listed common math symbols that you can enter into any math environment listed above.

Do not copy-paste the code below into your document. Symbols like the carrot ˆ will not render properly.

Symbol	**Example**	**Code**
Fractions	$\frac{5}{4}$	\frac{5}{4}
Square Root	$\sqrt{2}$	\sqrt{2}
Superscript	a^{2x}	a^{2x}
Subscript	x_{15}	x_{15}
Equivalent	$\equiv$	\equiv
Not equal to	$\neq$	\neq
Less than equal to	$\leq$	\leq
Greater than equal to	$\geq$	\geq
Summation	$\sum$	\sum
Product	$\prod$	\prod
Integral	$\int_{a}^{b}$	\int_{a}^{b}
Infinity	∞	\infty
Exists	$\exists$	\exists
For all	$\forall$	\forall
Element of	$\in$	\in
Set union	$\cup$	\cup
Set intersection	$\wedge$	\wedge
Subset	$\subset$	\subset
Integers	$\mathbb{Z}$	\mathbb{Z}
Real numbers	$\mathbb{R}$	\mathbb{R}
Rational numbers	$\mathbb{Q}$	\mathbb{Q}
Natural numbers	$\mathbb{N}$	\mathbb{N}
Implies	$\implies$	\implies
Implied by	$\impliedby$	\impliedby
If and only if	$\iff$	\iff

4.3 Matrices

To create a matrix, you use a matrix environment *inside* some math environment. We will present a few matrix environments below.

4.3.1 Zoo of Matrix Environments

1. **matrix**

   This environment creates a matrix with no surrounding punctuation.

   ```
   $ \begin{matrix}
   a & b\\
   c & d
   \end{matrix} $
   ```

   $\begin{matrix} a & b \\ c & d \end{matrix}$

2. **pmatrix**

   This environment creates a matrix with parenthesis as surrounding punctuation.

   ```
   $ \begin{pmatrix}
   a & b\\
   c & d
   \end{pmatrix} $
   ```

   $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

3. **bmatrix**

   This environment creates a matrix with brackets as surrounding punctuation.

   ```
   $ \begin{bmatrix}
   a & b\\
   c & d
   \end{bmatrix} $
   ```

   $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$

4.3.2 Creating matrices with Arbitrary Dimensions

Suppose you have a 4×5 sized matrix, you would add &s between every column, like so:

```
$ \begin{bmatrix}
1 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 1 & 0\\
\end{bmatrix} $
```

renders as

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 1 & 0\\
\end{bmatrix}$$

4.4 Inserting Code

Often times in CS classes you will need to write pseudocode or an algorithm in a language such as Python, C, or SML. You could just write your code as prose in LaTeX, but making sure that reserved symbols like % and \ from your code render as their literal representation is hard. To make this easy, there is a text environment that renders any text you write literally.

1. First, under your **documentclass** command at the top of your document, you need to import the **listings** package like this:

   ```
   \usepackage{listings}
   ```

2. Then, suppose you want to insert some text that looks like code. You should create a **lstlisting** environment like this:

   ```
   \begin{lstlisting}
   \end{lstlisting}
   ```

   Then you can enter your code:

   ```
   \begin{lstlisting}
   print(”Hello, World!”)
   a = 0
   b = a + 1
   \{lstlisting}
   ```

   This should render as

   ```
   print(”Hello, World!”)
   a = 0
   b = a + 1
   ```

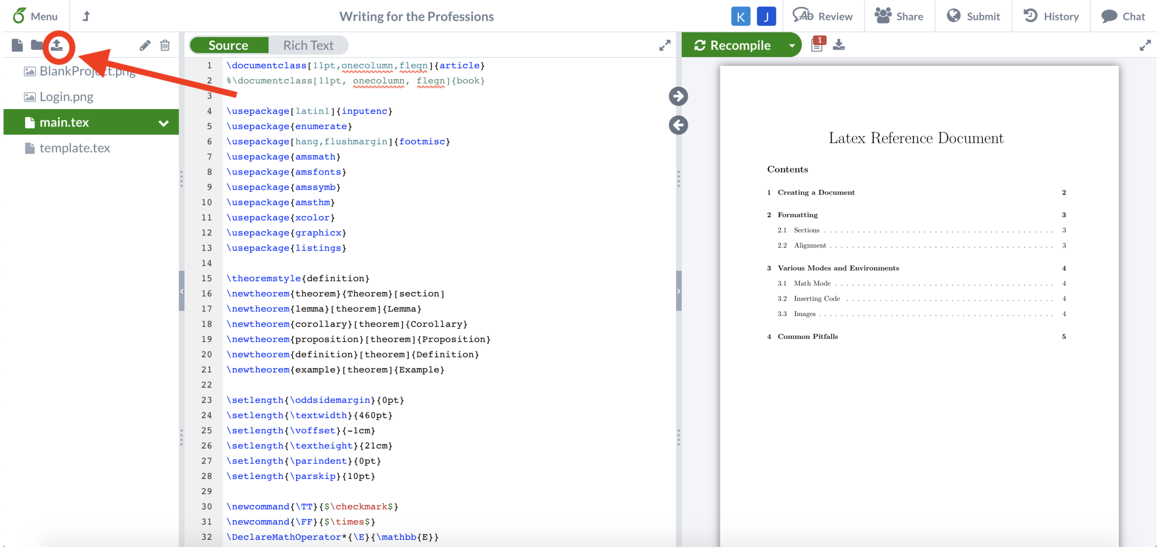
4.5 Images

Images are inserted into documents in latex using the **graphicx** package.
To use this package, include the following line after the \documentclass command:

```
\usepackage{graphicx}
```

Let's say the image you want to insert is in the file *sampleimage.png*.

1. Upload the image to Overleaf
   Click on the upload symbol (shown below) and pick the file to upload



```
\documentclass[11pt,onecolumn,fleqn]{article}
%\documentclass[11pt, onecolumn, fleqn]{book}

\usepackage[latin1]{inputenc}
\usepackage{enumerate}
\usepackage[hang,flushmargin]{footmisc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{amsthm}
\usepackage{xcolor}
\usepackage{graphicx}
\usepackage{listings}

\theoremstyle{definition}
\newtheorem{theorem}{Theorem}[section]
\newtheorem{lemma}[theorem]{Lemma}
\newtheorem{corollary}[theorem]{Corollary}
\newtheorem{proposition}[theorem]{Proposition}
\newtheorem{definition}[theorem]{Definition}
\newtheorem{example}[theorem]{Example}

\setlength{\oddsidemargin}{0pt}
\setlength{\textwidth}{460pt}
\setlength{\voffset}{-1cm}
\setlength{\textheight}{21cm}
\setlength{\parindent}{0pt}
\setlength{\parskip}{10pt}

\newcommand{\TT}{$\checkmark$}
\newcommand{\FF}{$\times$}
\DeclareMathOperator*{\E}{\mathbb{E}}
```

Latex Reference Document

Contents

1 Creating a Document **2**

2 Formatting **3**

2.1 Sections 3

2.2 Alignment 3

3 Various Modes and Environments **4**

3.1 Math Mode 4

3.2 Inserting Code 4

3.3 Images 4

4 Common Pitfalls **5**

2. Insert the uploaded image
   Use the following command to insert the image into the document

```
\includegraphics[scale=0.2]{sampleimage.png}
```

The value assigned to scale decides how large the image is.
(Note: You can also size images by specifying width instead of scale, for example, width=100mm instead of scale=0.2)