1 Quick Start

This is a quick guide that I'm working on to introduce LaTeX, but Overleaf also has very good documentation of LaTeX: `https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes`
`http://wch.github.io/latexsheet/latexsheet.pdf`

In this section, we'll go over how to set up a very simple LaTeX project. In the first line of your document, declare the document class. For most purposes, "article" is fine. You'll also declare the beginning and end of your document. The lines between declaring the document class and beginning the document are referred to as the *preamble*. This is where you import packages and may define custom functions (commands). It functions somewhat similarly to CSS and the HEAD of an HTML file. Your starting document might look like:

```
\documentclass{article}
\usepackage{firstpackage}
\usepackage{secondpackage}

\begin{document}

The content of my document...

\end{document}
```

1.1 Syntax

Note that some of the following characters must be escaped in order to be used as text characters

```
\           Begins a command
%           Comment out line
{}          Parameters for commands
[]          Brackets may indicate parameters
&           Delimiter for alignment
$..$        Indicates that the contents (between $) are in "math mode"
\\          Line break
*           TODO commands that are similar to other commands with slight changes
```

1.2 Basic Styling

These are some of the most basic LaTeX commands:

```
\textbf{...}        Bolds the text in braces
\textit{...}        Italicizes text in braces
\underline{...}     Underlines the text in braces
\hl{...}            Highlights text in brackets, requires the packages xcolor and soul
\noindent           Prevents automatic indentation
\newpage            Starts a new page
```

By default, the font size for an article is 10pt. LaTeX defines a number of relative font sizes that adjust relative to the global font size if the global size is changed. See `https://www.sascha-frank.com/latex-font-size.html` for details. Note that commands are case-sensitive. Here's an example of changing sizes:

INPUT

```
{\small We're going to start small} and then {\Large get bigger}.
{\LARGE This is really big!}
{\large And this line is in between.}
```

OUTPUT

We're going to start small and then get bigger. This is really big! And this line is in between.

Some features in LaTeX allow a length to be input. LaTeX accepts cm, mm, in, pt, ex, and more. For details, see `https://www.overleaf.com/learn/latex/Lengths_in_LaTeX`

1.3 Images

An image can be included with the **graphicx** package and the file name (or path, if applicable):

```
\documentclass{article}
\usepackage{graphicx}
...
\begin{document}
...
\includegraphics{MyPic.png}
\includegraphics[scale=0.3]{MyPic.png} % optional params include scale
...
\end{document}
```

1.4 Environments

LaTeX also has *environments*, which apply formatting rules to the text within the environment. The syntax for beginning and ending an environment is:

```
\begin{environment}
...
\end{environment}
```

You can also create and define your own custom environments. Some environments have parameters (typically optional). Many environments require specific packages to be used, which will be noted if applicable. Also note that although we'll use tabs to assist with clarity, LaTeX is typically insensitive to tabs.

1.5 Packages

To start, consider using these packages:

```
\usepackage{graphicx}
\usepackage[letterpaper, top=0.75in, rmargin=1in, lmargin=1in]{geometry}
\usepackage[shortlabels]{enumitem}
\usepackage{soul}
\usepackage{xcolor}
\usepackage{lineno}
\usepackage{url}
```

1.6 Troubleshooting

TODO:

- Different versions of compilation
- hbox over/underflow
- Left quotes

2 Alignment

2.1 Line Breaks and Vertical Space

There are multiple ways to add vertical spacing in LaTeX, but some are considered "poor practice". In many cases choosing the "wrong" one will cause "hbox overfull" warnings, but I don't think it's necessary to worry about when just getting started.

Here are some ways to create vertical space with some quick notes on use:

```
\par                        Starts a new paragraph on a new line;
                            automatically indents the new line.

\\                          Inserts a line break; necessary in some enrivonments but
                            not recommended for use in "regular" text

\newline                    Inserts a line break, can't be used for additional whitespace
                            May cause errors if there is "no line to end" according to LaTeX

\vspace{size}               Adds vertical space of the size specified (see allowed units in ch 1)

\vspace{\baselineskip}      Adds vertical space of the size of "baseline line skip"
```

Note: Leaving empty lines in the text will start a new paragraph, but will not add any additional whitespace. Empty lines are essentially equivalent to using the \par command.

I would recommend using new paragraphs "functionally" when a new paragraph is started and using \vspace for "stylistic" line separation. Use \\when inside of a tabbing/tabular environment (covered later this section).

Additional information:

https://www.overleaf.com/learn/latex/Paragraphs_and_new_lines
https://www.overleaf.com/learn/latex/Line_breaks_and_blank_spaces

TODO: setspace package

2.2 Tabbing

The environment *tabbing* can be used to align text. Note that the **tabbing** environment is **not** the same as the **tabular** environment. The tabbing environment is for creating sections of aligned text, while the tabular environment is for creating tables. This is covered in ?

Below is a small example of the tabbing environment. The first line determines the placement of the tabs and the following text is what's actually displayed. Within the tabular environment, use \\for line breaks.

INPUT

```
\begin{tabbing}
    \hspace{1in} \= \hspace{1.5in} \= \kill         % \= Define the tab size(s)
    First \> This is my description \\               % \\ breaks the line
    More \> This is even \> more description \\     % \> adds a tab
    \> \> This is extra tabbed
\end{tabbing}
```

OUTPUT

First	This is my description	
More	This is even	more description
		This is extra tabbed

2.3 Centering

INPUT

```
\begin{center}
This will be centered.

And this line will also be centered
\end{center}
```

OUTPUT

This will be centered.
And this line will also be centered

3 References and Line Numbers

3.1

One of the most useful features of LaTeX is its ability to handle references to other parts of the text. Creating a label for part of the text and referencing the label will maintain consistency even if other parts of the text are changed. For example:

INPUT

```
\subsection{Basic Styling}
\label{subsection:basic-styles}
...
As discussed in Section \ref{subsection:basic-styles},
we can extend this to...
```

OUTPUT

As discussed in Section 1.2, we can extend this to...

If additional sections are added before Section 1.2 or the title of the section is changed, the reference will update and remain correct. This is very useful for coordinating any references within the text. For PDF generation of legal documents, this is particularly useful for tracking line numbers

3.2 Line Numbering

We can use the **lineno** package to automatically create numbered lines and then reference them later. For example:

INPUT

```
\begin{linenumbers}
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin tempor, erat in
eleifend convallis, nulla neque semper magna, id aliquam risus ex posuere lectus.
Curabitur tincidunt enim sit amet purus aliquet tempor. Vestibulum ac sem aliquet,
congue diam a, pretium nibh. \linelabel{important-line} \hl{Aliquam sodales
tincidunt ligula non varius.} Donec vestibulum sodales maximus. Integer lacinia,
quam in consequat ultricies, sem leo pellentesque elit, eget porta eros risus id
nulla. Fusce at sodales neque, eget aliquet velit. Duis non finibus ex.

Curabitur ligula nisi, convallis at ante sit amet, tristique hendrerit felis. Fusce
viverra dui id tellus rhoncus dignissim. Etiam pretium elit vel sem sagittis, id
sollicitudin justo lobortis. Vestibulum ultrices maximus quam vitae pellentesque,
\textbf{as described on line \ref{important-line}}.
\end{linenumbers}
```

OUTPUT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin tempor, erat in eleifend convallis, nulla neque semper magna, id aliquam risus ex posuere lectus. Curabitur tincidunt enim sit amet purus aliquet tempor. Vestibulum ac sem aliquet, congue diam a, pretium nibh. Aliquam sodales tincidunt ligula non varius. Donec vestibulum sodales maximus. Integer lacinia, quam in consequat ultricies, sem leo pellentesque elit, eget porta eros risus id nulla. Fusce at sodales neque, eget aliquet velit. Duis non finibus ex.

Curabitur ligula nisi, convallis at ante sit amet, tristique hendrerit felis. Fusce viverra dui id tellus rhoncus dignissim. Etiam pretium elit vel sem sagittis, id sollicitudin justo lobortis. Vestibulum ultrices maximus quam vitae pellentesque, **as described on line 3**.

If we increase the font size, the line numbering will change but the reference will remain consistent. Note that the same label should not be used twice in the same document, but it can be referenced multiple times. Line numbers will increment over the course of the entire document unless reset.

INPUT

```
\begin{linenumbers}
\Large
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin tempor, erat in
eleifend convallis, nulla neque semper magna, id aliquam risus ex posuere lectus.
Curabitur tincidunt enim sit amet purus aliquet tempor. Vestibulum ac sem aliquet,
congue diam a, pretium nibh. \linelabel{important-line} \hl{Aliquam sodales
tincidunt ligula non varius.} Donec vestibulum sodales maximus. Integer lacinia,
quam in consequat ultricies, sem leo pellentesque elit, eget porta eros risus id
nulla. Fusce at sodales neque, eget aliquet velit. Duis non finibus ex.

Curabitur ligula nisi, convallis at ante sit amet, tristique hendrerit felis. Fusce
viverra dui id tellus rhoncus dignissim. Etiam pretium elit vel sem sagittis, id
sollicitudin justo lobortis. Vestibulum ultrices maximus quam vitae pellentesque,
\textbf{as described on line \ref{important-line}}.
\end{linenumbers}
```

OUTPUT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin tempor, erat in eleifend convallis, nulla neque semper magna, id aliquam risus ex posuere lectus. Curabitur tincidunt enim sit amet purus aliquet tempor. Vestibulum ac sem aliquet, congue diam a, pretium nibh. Aliquam sodales tincidunt ligula non varius. Donec vestibulum sodales maximus. Integer lacinia, quam in consequat ultricies, sem leo pellentesque elit, eget porta eros risus id nulla. Fusce at sodales neque, eget aliquet velit. Duis non finibus ex.

Curabitur ligula nisi, convallis at ante sit amet, tristique hendrerit felis. Fusce viverra dui id tellus rhoncus dignissim. Etiam pretium elit vel sem sagittis, id sollicitudin justo lobortis. Vestibulum ultrices maximus quam vitae pellentesque, **as described on line 4**.

4 Lists

4.1 Itemize

The environment *itemize* can be used to make bulleted lists:

INPUT

```
\begin{itemize}
    \item My first point
    \item My second point
\end{itemize}
```

OUTPUT

- My first point
- My second point

4.2 Customized lists

Requires **\usepackage[shortlabels]{enumitem}**

INPUT

```
\begin{enumerate}[label*=\arabic*]
    \item First is here
        \begin{enumerate}[label*=\alph*]
        \item We can label sub-items with letters
        \end{enumerate}
    \item Second is here
        \begin{enumerate}[label*=.\arabic*]
        \item Or we can label sub-items with numbers
        \end{enumerate}
    \item Third is here
\end{enumerate}
```

OUTPUT

1 First is here

1a We can label sub-items with letters

2 Second is here

2.1 Or we can label sub-items with numbers

3 Third is here

5 Tables

Tables are simple to make, but can be challenging to format precisely. Multiple packages exist that may be used for table creation. Creating a table typically uses either the **tabular** package or the **tabularx** package. A table may or may not use the **table** environment. We'll be starting with only **tabular**.

5.1 Simple Tables

For a very simple table, we need to specify the type of column to use and whether or not to include vertical rules. The contents of each cell are separated by & within a row and \\for a column. Horizontal rules are denoted by \hline. It's also generally recommended to center tables. Here's a simple example:

INPUT

```
\begin{center}
    \begin{tabular}{ | c | c c c |} % defines columns and vertical rules
        \hline
        \textbf{First} & \textbf{Second} & \textbf{Third} & \textbf{Fourth} \\
        \hline
        A & B & C & D \\
        E & F & G & H \\
        \hline
    \end{tabular}
\end{center}
```

OUTPUT

First	Second	Third	Fourth
A	B	C	D
E	F	G	H

The default row padding is more narrow than I prefer. I add the following command at the top of my document to add a bit of extra spacing:

```
\setlength\extrarowheight{4pt}
```

OUTPUT

First	Second	Third	Fourth
A	B	C	D
E	F	G	H

5.2 Column Options

Standard column types:

```
c               centred column
l               left-justified column
r               right-justified column
p{width}        paragraph column with text vertically aligned at the top
m{width}        paragraph column with text vertically aligned in the middle (req array package)
b{width}        paragraph column with text vertically aligned at the bottom (req array package)
```

You can also create your own custom columns. This one creates a left-aligned column of a custom width:

```
\newcolumntype{F}[1]{>{\raggedright\arraybackslash}p{#1}}
```

The [1] indicates the number of parameters, and #1 indicates where the first parameter will be inserted. \raggedright makes the column left-aligned, and \is needed on the last column to prevent errors (?). Here's an example:

INPUT

```
\begin{center}
    \begin{tabular}{ | F{1.5in} | F{1in} | F{0.5in}   F{0.5in} | }
        \hline
        \textbf{First} & \textbf{Second} & \textbf{Third} & \textbf{Fourth} \\
        \hline
        A & B & C & D \\
        \hline
        E & F & G & H \\
        \hline
    \end{tabular}
\end{center}
```

OUTPUT

First	**Second**	**Third**	**Fourth**
A	B	C	D
E	F	G	H

The text will also wrap by default for this type of column:

First	**Second**	**Third**	**Fourth**
Here is some text. There's so much text that it needs to wrap! But it will wrap instead of running into another column.	B	C	D
E	F	G	H

You can also use the **multirow** package to combine columns and rows. The function \multicolumn takes three parameters: the number of columns to span, the column style to use for the merged column (including vertical rules), and the text of the column. Here's an example of the multicolumn package:

INPUT

```
\begin{center}
    \begin{tabular}{ | c | c c c |}
        \hline
        \multicolumn{4}{|c|}{The title of my table} \\
        \hline
        \textbf{First} & \textbf{Second} & \textbf{Third} & \textbf{Fourth} \\
        \hline
        A & B & C & D \\
        E & F & \multicolumn{2}{c|}{Either G or H} \\
        \hline
```

```
        \multicolumn{3}{|c|}{Any of I, J, or K}  & L \\
        \hline
    \end{tabular}
\end{center}
```

OUTPUT

The title of my table			
First	**Second**	**Third**	**Fourth**
A	B	C	D
E	F	Either G or H	
Any of I, J, or K			L

Be careful when combining multicolumn with fixed-width columns. Combining multicolumn with fixed-width columns can cause some odd behavior because of the way table sizing works in LaTeX, which will be discussed later in this section.

5.3 Table Sizing: Issues

One of the most frustrating aspects of LaTeX can be making tables or columns *exactly* the size that you want them. This subsection explains the details of why table sizing may not behave as expected in LaTeX, while the next one outlines a solution. Feel free to skip to there, but consider at least skimming this section.

LaTeX tables may behave in unexpected ways because fixed-width columns only set the width of the text in the column, and does not include the default padding on each column. The size of the padding between the text and the borders of the cell is referred to as \tabcolsep. This means that changing the number of columns will change the total width of your table even if the sum of the column widths is unchanged.

For a fixed-width column p{x}, the actual width of the column will be x + 2 * \tabcolsep

Here's an example of how this can cause unexpected behavior:

INPUT

```
\begin{center}
    \begin{tabular}{ | p{1.5in} | p{1in} | p{0.5in}   p{0.5in} | }
        \hline
        \textbf{First} & \textbf{Second} & \textbf{Third} & \textbf{Fourth} \\
        \hline
        A & B & C & D \\
        \hline
    \end{tabular}
\end{center}

\begin{center}
    \begin{tabular}{ | p{1.5in} | p{1in} | p{1in} | }
        \hline
        \textbf{First} & \textbf{Second} & \textbf{Third} \\
        \hline
        A & B & C \\
        \hline
    \end{tabular}
\end{center}
```

OUTPUT

First	Second	Third	Fourth
A	B	C	D

First	Second	Third
A	B	C

You might expect these tables to be the same width, but they're not, despite the column widths summing to 4.5in for both. The width of the first table is actually 4.5in + 4(2 * \tabcolsep), while the width of the second table is only 4.5in + 3(2 * \tabcolsep). The default value of \tabcolsep is 6pt, which is 0.0833... inches. I personally prefer to set it to 0.08in to make any column math easier, since we may want to keep sizes consistent. If we add 0.16in to the third column in our second table, it should now match the first:

INPUT

```
\begin{center}
    \begin{tabular}{ | p{1.5in} | p{1in} | p{1.16in} | }
        \hline
        \textbf{First} & \textbf{Second} & \textbf{Third} \\
        \hline
        A & B & C \\
        \hline
    \end{tabular}
\end{center}
```

OUTPUT

First	Second	Third
A	B	C

This is also why multicolumn can cause unexpected behavior with fixed rows. Consider our example from before if we had some longer text:

The title of my table			
First	**Second**	**Third**	**Fourth**
A	B	C	D
E	F	Either G or H	
Any of I, J, or K, or also some other options, there are a lot of possibilities			L

Because our multicolumn command used column type *c*, the column text width is no longer fixed and the table expands to accommodate the text. We can try F{1.5in} in place of *c*, but this still gives us some odd behavior.

INPUT

```
\begin{center}
    \begin{tabular}{ | F{0.5in} | F{0.5in} F{0.5in} F{0.5in} |}
        \hline
        \multicolumn{4}{|c|}{The title of my table} \\
        \hline
        \textbf{First} & \textbf{Second} & \textbf{Third} & \textbf{Fourth} \\
        \hline
        A & B & C & D \\
        E & F & \multicolumn{2}{c|}{Either G or H} \\
```

```
        \hline
        \multicolumn{3}{|F{1.5in}|}{Any of I, J, or K, or also some other options, there are a lot of p
        \hline
    \end{tabular}
\end{center}
```

OUTPUT

The title of my table			
First	**Second**	**Third**	**Fourth**
A	B	C	D
E	F	Either G or H	
Any of I, J, or K, or also some other options, there are a lot of possibilities			L

Now it's wrapped, but the text doesn't fill the cell because the first three columns actually have a width of 1.5in + 3(2 * 0.08in) = 1.98in. For a single column with a total width of 1.98in, the text width is 1.98 - 2(0.08) = 1.82in. So our table should actually look like:

INPUT

```
\begin{center}
    \begin{tabular}{ | F{0.5in} | F{0.5in} F{0.5in} F{0.5in} |}
        \hline
        \multicolumn{4}{|c|}{The title of my table} \\
        \hline
        \textbf{First} & \textbf{Second} & \textbf{Third} & \textbf{Fourth} \\
        \hline
        A & B & C & D \\
        E & F & \multicolumn{2}{c|}{Either G or H} \\
        \hline
        \multicolumn{3}{|F{1.82in}|}{Any of I, J, or K, or also some other
        options, there are a lot of possibilities}  & L \\
        \hline
    \end{tabular}
\end{center}
```

OUTPUT

The title of my table			
First	**Second**	**Third**	**Fourth**
A	B	C	D
E	F	Either G or H	
Any of I, J, or K, or also some other options, there are a lot of possibilities			L

5.4 Table Sizing: Solutions

In order to keep a consistent width among tables and for correct multicolumn handling, I recommend defining the following column type: (Note that this requires the **calc** package)

```
\newcolumntype{A}[1]{>{\raggedright\arraybackslash}p{#1 - 2\tabcolsep}}
```

This will create a column with a *total* size of the input. This allows us to simply sum the values for tables with different numbers of columns that should have the same total width.

INPUT

```
\begin{center}
    \begin{tabular}{ | A{1.5in} | A{1in} | A{0.75in}   A{0.75in} | }
        \hline
        \textbf{First} & \textbf{Second} & \textbf{Third} & \textbf{Fourth} \\
        \hline
        A & B & C & D \\
        \hline
    \end{tabular}
\end{center}

\begin{center}
    \begin{tabular}{ | A{1.5in} | A{1in} | A{1.5in} | }
        \hline
        \textbf{First} & \textbf{Second} & \textbf{Third} \\
        \hline
        A & B & C \\
        \hline
    \end{tabular}
\end{center}
```

OUTPUT

First	**Second**	**Third**	**Fourth**
A	B	C	D

First	**Second**	**Third**
A	B	C

This also simplifies multicolumn handling:

INPUT

```
\begin{center}
    \begin{tabular}{ | A{0.75in} | A{0.75in} A{0.75in} A{0.75in} |}
        \hline
        \multicolumn{4}{|c|}{The title of my table} \\
        \hline
        \textbf{First} & \textbf{Second} & \textbf{Third} & \textbf{Fourth} \\
        \hline
        A & B & C & D \\
        E & F & \multicolumn{2}{A{1.5in}|}{Either G or H} \\
        \hline
        \multicolumn{3}{|A{2.25in}|}{Any of I, J, or K, or also some other
        options, there are a lot of possibilities}  & L \\
        \hline
    \end{tabular}
\end{center}
```

The title of my table			
First	**Second**	**Third**	**Fourth**
A	B	C	D
E	F	Either G or H	
Any of I, J, or K, or also some other options, there are a lot of possibilities			L

5.5 The table environment

TODO, but in summary: it's not needed and can cause unexpected behavior due to LaTeX trying to place your table "nicely"

6 Page Layout

6.1 Geometry package

The **geometry** package can be used to set the page layout. For this document, we're using:

```
\usepackage[letterpaper, top=0.75in, rmargin=1in, lmargin=1in]{geometry}
```

6.2 Headers and footers

7 Advanced

7.1 Custom commands

7.2 Constant variables

7.3 If/then

7.4 Iterating

7.5 Special Characters

```
\           Begins a command
%           Comment out line
{}          Parameters for commands
[]          Brackets may indicate parameters but do not typically need to be escaped
&           Delimiter for alignment
$..$        Indicates that the contents are in "math mode"
\\          Line break
#           TODO
@           TODO
_           TODO
^           TODO
~           TODO
*           TODO
http://www.emerson.emory.edu/services/latex/latex_toc.html#SEC155
```

7.6 Imports and inputs

7.7 Different versions of TeX