

# How to Write Using Rich Text Format and Markdown in LaTeX and Overleaf

Arindam Basu  
arindam.basu@canterbury.ac.nz

July 24, 2016

# 1 How to use Rich Text Format, Markdown, and Web based LaTeX converters to Write Scholarly Documents

## 1.1 Tools

Over the last two decades between 2000-2015, the world wide web and cloud has emerged as powerful media for collaborative research and writing. In parallel, version control systems, and plain texts that enable scholars to collaborate and share documents have also emerged, along with practice of writing on the web [3]. Luca Merciadri (2012) has argued that eBooks have advantages over paper based books in being lighter and easier to search; among the different formats of authoring and presentation of eBooks and eTexts, LaTeX provides a standard and efficient way of authoring all forms of texts, but particularly texts that are heavy on citations, and presentation of data using tables, charts, and cross-linking [2]. While LaTeX provides a nice environment for structured authoring of text, Markdown is increasingly becoming popular since its inception [1]. Markdown is also easy to author and readable with least hindrance. Using web based LaTeX authoring systems, it is possible to interface Markdown with LaTeX using converters and version control systems. The objective of this tutorial is to show how to use LaTeX effectively in a wysiwyg environment and work using markdown to author scholarly documentation.

## 1.2 Basic Setup

**Overleaf** provides a web-based freemium academic writing environment. By freemium I mean that you can use the basic full version for free, and for more storage and some convenient optional features you pay. You can use this tool to create a range of documents – blog posts, journal articles, posters, slide decks, books, essays and so on. Although the system is based on LaTeX, it provides a mature rich text editor that you can use without knowing LaTeX. I have used Overleaf to write blog posts (publish them using **TheWinnower**), teach, and create powerpoint like presentations using the **Beamer template (PDF)**. Overleaf has a series of tutorials and help files on their site, but all of them assume that you are familiar with and you are going to use LaTeX. You can use Overleaf even if you do not know much about LaTeX. In this tutorial, you will learn how to use the system without much knowledge of LaTeX.

In this tutorial, I assume that you may not be familiar with LaTeX. For essential LaTeX codes that you will need to get things done to write a paper using Overleaf using the system I am about to discuss, I have put together the LaTeX codes. Just copy and paste the codes that I have provided here and you will be set to go. If you combine commonly available web based tools, Overleaf's own template systems, rich text writing environment it offers, and the easy writing environment referred to as Markdown, you will not need to know LaTeX in depth (some commands and usage will still be necessary and I will cover them here) to write scholarly papers and use Overleaf to distribute them to others and submit to journals and blogs. I have provided the steps below.

### 1.3 Sign up for an account

In order to use Overleaf, you will need to create a username and account on Overleaf. You can get an account for yourself for free at the following page: [Overleaf Signup Page](#). You will need your email address to sign up.

### 1.4 Once you signed up, sign in and create a new project

When you sign in, you will see a box that says “New Project”. Click on “New Project” to get started. If you receive an URL to work on a document, and you are asked to click the URL to get to a document, then you already have a document to work on, so just signing in will lead you to that document.

To create a new document, start by creating a new project. A project in Overleaf is a folder containing a number of different files. When you click “Create New Project”, you will see standard templates. Select one. I usually start with blank but you can start with the template titled “Article”. Also note that there are tens of journal article templates, books, theses, etc on the Overleaf site. As in this article, we would like to focus on the basic matter, start with the journal article template. When you do so, you will see a screen, that will look like as follows:

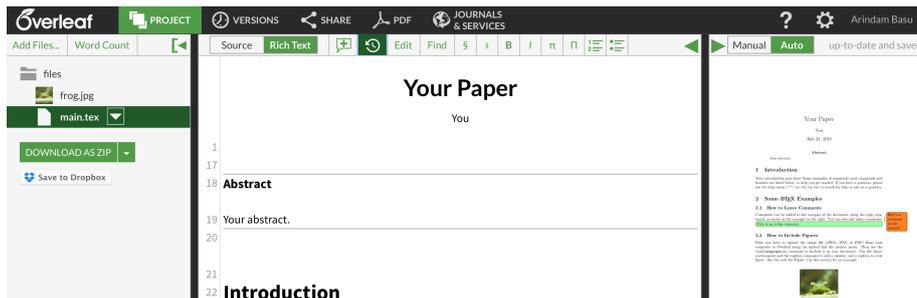


Figure 1: Panels

From left to right, these are the elements:

1. Leftmost, you see a file manager looking panel. Next to it, half of the screen is a document that either shows the source LaTeX document or a rich text rendering. If your document shows “Source”, switch to “Rich text Format” by clicking on the box that says “Rich Text”
2. Right half of the screen shows the PDF rendering. Overleaf only shows you the PDF rendering
3. On the top of the window, you will see the following information from left to right: Overleaf, Project, Versions, Share, PDF, Journals and Services, a question mark, a cog wheel, and your name with a small triangle at the top. If you press on Overleaf, it will take you to the Overleaf web page. If you click on “Project”, it will open up a third panel (see the image) and this panel is like a file manager. Clicking on “Versions” will bring you to a version control window. The “Share” link leads to a drop down window with several different sharing options including read only link, read write link, git link (more on this later), and other sharing options. If you click on the PDF button or icon, it will create a PDF document

of the rendered document that shows in the right panel. If you click on Journals and Services, it will lead to a drop down box from where you can send your document to any number of journals or preprint servers. This is usually a one-step process of submitting to a journal or preprint service and very handy. The question mark is about obtaining help in the form of tutorials, and videos (I strongly encourage you use this). The cog-wheel is for customising Overleaf. If you are starting out and you do not know what you are doing, there is no need to touch this and stick with the default. Lastly, your name and the small triangle is about the documents and projects that you have already created on Overleaf or new projects that you can create from that link.

4. If you click on “Project” button (if the Project button is not already highlighted and the panel is not showing), it will open the project panel. On the top of the project panel, the main document writing window, and the PDF panel, there are now other sets of buttons that you will use. These are from left to right: “Add Files”, “Word Count”, a left pointing arrow, source, rich text, a quote symbol with a plus sign in it, a clock symbol, Edit, Find, a large anchor sign, a small anchor sign, B, I, a small pi sign, a big pi sign, a numbered list sign, a bullet point sign, a left pointing arrow, a right pointing arrow, Manual, Auto, and then in grey letters written “up to date and saved”.

## 1.5 The File Manager Panel

Overleaf provides you an in-built file manager. In this file manager, you can store files that you will use for your paper. This can be text or image documents (remember to upload only PDF, image, or text documents). You can also create new folders where you can store your files. If your paper is likely to have a lot of images, you can create separate folders to hold your images.

You can add files from 15 different sources to Overleaf (at the time of writing this tutorial). You can import documents and files from your hard drive, or you import from over the web; you can also use FTP to transfer files to Overleaf (Overleaf itself does not provide an FTP address though). In this tutorial, I will focus on the web version. I only work on the web version (I will write in a later article how you can sync your git with this), I usually add a blank file and name it as `filename.tex` and continue to work on that file. I have found this to be quite efficient. For example, for my paper writing, if I have several sections, say, introduction, methods, background, summary, results, etc, for every section I add a file with a name to indicate the contents. Thus I usually add files named as `introduction.tex`, `methods.tex` and so on. Later, I add the code to the file `main.tex` `\include{introduction}` and repeat this for other files as needed. This code is repeated per line for as many files as I have. The `main.tex` file then takes care of stitching together all the information.

From the file manager, you can also download the entire contents of your document in the form of a zipped file to your desktop and process the files yourself. This is useful if you want to work on the latex files yourself or need to provide the files to a colleague who does not want to work in the Overleaf environment.

The other thing you can do in the file manager panel is to check the word count of your document. You can only do that for the `main.tex` document. So,

if you have a lot of text in your `main.tex` document then you can check the word count this way. I usually use other editors to check the word count. You can also set any document as your main document for processing, and count the words that way (not a good idea).

Now that you have some idea about the file manager space within Overleaf, let us see check out some workflow for writing an academic document and see how Overleaf helps us in writing and formatting an academic document (term paper, journal article, and so on). Early on, it is a good idea to establish a workflow that will help you. Also, let's move on to the middle panel, where most of your writing work will occur.

## 1.6 Configure a workflow

In general, an academic document such as research paper, a journal article, a term paper at a university course etc have some common elements and structural elements to them. First, you need to provide a title and indicate the authorship. In LaTeX documents, this is taken care of in what is known as the “Preamble” of the document. In Overleaf, we place the preamble in the `main.tex` document, and you will find that somewhere in the beginning of the document if you press “Source” in the middle panel. Second, most scholarly articles usually display information in the form of tables and figures. In a LaTeX document, these are referred to as special environments where tables and figures are placed. If you have many tables and figures, I recommend that you place the figures and tables in a separate file labelled as `figures.tex` and `tables.tex`. That way, you can either refer to them from within the document or you can copy and paste codes of tables and figures from those documents to the specific part of the document that you are writing. Third, all scholarly article authors depend on citations and bibliography while writing their articles and essays. Every fact posted at an article or a book or a document, needs to be properly attributed in terms of where this is located. This is why you need citations and citations are linked to more detailed information, usually in footnotes, endnotes, and bibliography. Besides, endnotes and footnotes are used to denote specific chunks of information that cannot be included in the body of the main text. In this tutorial, I will not focus on endnotes and footnotes, but only discuss using bibliography. The format of bibliography in a LaTeX document is done by using BibTeX. In overleaf, you can upload a bibtex document, or you can import a bibtex formatted document from your hard drive, or you can import a bibtex document from other web based services such as [CiteULike](#), [Mendeley](#), and others. You can also import bibtex documents that are exported by reference management software such as Endnote and Papers, and others. Lastly, you can directly write Bibtex formatted entries and work from there.

Remember that you will always find ‘`main.tex`’ file in the file manager panel for all templates. This `main.tex` contains all the essential codes that will compile the paper for you and will render it in the right hand panel for you to watch. Therefore, the “Word Count” that you get to see in the crumb above in the file manager panel will give you word count for all the words contained in the `main.tex` file. In the file manager, I also create the following three files:

- A `references.bib` file. This will contain my references in the bibtex format (this is the only technical bit that you will need to learn here)

- A figures.tex file. This file contains codes of all my figures and images for the paper
- A tables.tex file. Similar to the one for figures, it contains all my tables codes for the paper.

In most cases, you will not have to alter the contents of the main.tex file. This file is also the only one where you will need to click on the “Source” in the central panel and work. I suggest the following: (see the following figure)

1. Click on the “Source” button on the main.tex file
2. Clean up the Source of the main.tex document so that it will only contain the following:

```

\documentclassa4paper{article}
\usepackageenglish{babel}
\usepackageutf8{inputenc}
\usepackage{amsmath}
\usepackage{graphicx}
\usepackagecolorinlistoftodos{todonotes}
\usepackage{booktabs}

title{A Sample Research Paper}

author{Your Name}

date{today}

begin{document}
maketitle
% now add what sections you would like to add
%include{summary}
%include{introduction}
%include{tables}
%include{figures}

% sign indicate comments. Uncomment to add them.

bibliography{references}
bibliographystyle{plain}
end{document}

```

Once you set up this structure, all you need to do is to write your paper in the central pane. You can at this point, minimise distractions by clicking on the arrows on the left and right side of the central panel. For each section of the paper, you select that section and write in rich text in the central panel. You can occasionally check how the document is being rendered.

## 1.7 Style of writing and individual elements to configure

In this step, you need to do a few things to get your figures, tables, and bibliography sorted.

### 1.7.1 How to add tables to your document

For putting together tables, I recommend you use [Tablesgenerator.com](https://www.tablesgenerator.com) site.

Steps:

- Visit the site and configure using LaTeX tables.
- Use the bookends theme to style your tables.
- Then copy the resulting code and
- paste to either the place in the document where you want the table to appear or paste it to the tables.tex file.

This way, you can create professional quality tables without writing a single line of code yourself. Once you have studied the process of writing LaTeX tables yourself, it will be intuitive for you to write professional quality latex tables.

### 1.7.2 How to add figures

Steps: - Upload the figure (either a jpg or a png figure) to the file space - Use the following code to add the figure either to the figures.tex file or directly in the document where you want the figures to appear.

```
\begin{figure}[ht]
\includegraphics{name_of_the_file}[width=\textwidth]
\end{figure}
```

## 1.8 Add citations and bibliography using bibtex

As Overleaf is about writing in LaTeX, your citations and bibliography must be configured as bibtex. The simplest and perhaps the most basic way is to add bibtex formatted citations in the references.bib file and then use the `\cite{}` commands to add the citations. As soon as you start typing the `cite{}` command, you will see a dropdown box that will appear with the citation ID. Select the citation ID of the citation that you would like to add to your document. Depending on your bibliography style, the reference list will be automatically generated for your document.

## 1.9 Add Tables, Figures, Bibtex

LaTeX is very intuitive and easy to learn. The interesting and somewhat time consuming part of learning LaTeX is how to format tables. It is not difficult but it is more of an art to correctly format tables in LaTeX. Having said that, there are a number of different websites where you can format tables for LaTeX and then copy and paste the code into Overleaf. Here are the steps:

1. Prepare your tabular data in the form of spreadsheets, ideally in a csv ('comma separated value') file. You can also keep your tabular in the form of a spreadsheet. If you work with spreadsheets such as Apple Numbers, Microsoft Excel, LibreOffice Calc, Google Sheets, and other similar apps, all of these will allow you to take data from the spreadsheet in the form of csv.
2. If you work with R or another statistical data analysis software, most software will let you export your tables from the app to a csv file format.
3. Once you have your data in the form of csv file, head over to [Tablesgenerator.com website](https://www.tablesgenerator.com)

4. Read the csv file in that webpage, and create a table. I recommend you use the booktabs style of table creation. For more information about booktabs package in LaTeX, read their [user manual \(PDF\)](#)
5. Then generate the code and copy it to "clipboard"
6. Return to your Overleaf document and paste. Remember to add a proper caption and a label for the table. Each label must be unique for each table; else Overleaf will complain during compilation.

These six steps will be sufficient for most tables. You can also put the tables in a special file, call it tables.tex in the file manager and subsequently copy and paste the table codes to the relevant sections of the document. Remember to add the following code next to each table environment:

This is needed so that the table will be placed where you want to place it, or at least at the top of the page. Also, make sure to add a nice caption to each table.

## 1.10 Add figures in Overleaf

If you want to figures in an Overleaf document, do:

1. First upload the figure to the filesystem. Use only jpg or png (or pdf) extensions for your figures.
2. Then copy and paste the following code:

```
\begin{figure}[ht]
\centering
\includegraphics[width=\textwidth]{imagefile.jpg}\caption{\label{fig:image} Write the capt
\end{figure}
```

Remember to include a proper caption and a proper label for the image. Each label must be unique to each image. As before [ht] stands for here and top.

## 1.11 Add citations and references to Overleaf

In Overleaf, citations are stored in a bibtex file. A bibtex file is a plain text file that contains information about your references (the title of the paper, the author names, the issue, the journal title, the pages, abstract, etc). Each reference, depending on the type of the resource, will have a different entry, and the specific information that you should enter for each type of entry is different as well. For more information about bibtex, please see [How to Use Bibtex](#)

Essentially, most academic search engines, and reference management software can provide you with a copy of their reference libraries in the bibtex format. The bibtex file ends in a .bib file format. I recommend that you start with an empty 'references.bib' file in your file manager where you will store your references. This is a plain text file and will contain the bibliographic entries as you go on conducting your research. Every bibliography entry in the bibtex file has an identifier. For example, see the following code:

```
@misc{ Nobody06,  
author = "Nobody Jr",  
title = "My Article",  
year = "2006" }‘
```

Here, ‘Nobody06’ is the ID for this entry. When you type to insert a specific bibliographic information as a citation to a fact, use the following code to enter the data

```
\cite{ID code}
```

Where ID code is the ID code of the specific entry of the bibliographic entry. For example, if you wanted to insert the Nobody06 citation in a LaTeX document, you’d simply type `\cite{Nobody06}` and so on.

You must also place the following two entries in the main.tex file in Overleaf just above the `\end{document}` code:

```
\bibliography{name of the bib file}  
\bibliographystyle{name of the style}
```

That’s it. LaTeX will then insert the citations and the reference list correctly at the end of the document. In the next part, I will show you how you can use Markdown to write in Overleaf without using LaTeX.

## 1.12 Use Git, Pandoc, and Markdown with Overleaf

It’s great to be able to write in LaTeX and use Overleaf. Basically, all you have to do is to write in LaTeX anywhere and just upload the tex file to Overleaf. You can also use the Wysiwyg editor in Overleaf to add content. However, there is another way: using Markdown in a plain text editor and build up Overleaf and push everywhere through Overleaf.

Markdown was designed by John Gruber in 2004. It is a great way to write clean, styled document using only plain text and nothing more. The structure of writing Markdown documents is highly standardised, and very intuitive. A Markdown document is also very clean to read and share with people who do not know anything about Markdown at all. You should be able to edit Markdown documents in any text editor. In Windows Operating System, you can use a notepad to write in markdown. In other operating systems, you can use a simple text editor to edit a markdown document. There are excellent web based tools to write markdown as well. Learn more about Markdown here:

<https://daringfireball.net/projects/markdown/>

The scope of Markdown is now extended. Markdown can now include tables, and citations. Thus, you can use Markdown to write your document and use Overleaf to produce a neat stylised PDF document. This has several advantages. First, Markdown is very simple to use; practically nothing to learn yet you can use all the benefits of using Overleaf without knowing practically anything about how to write LaTeX. Second, as Markdown is a plain text based system, you can use it everywhere exactly like you can use Overleaf in every device. Third, there are scores of Apps, several of them free, that will let you use Markdown and write in nice stylised markdown and produce beautiful document. You can use any search engines to search for Markdown Apps for your operating system. There are web based instances where you can write in Markdown and download the document on your computer. Later you can upload this document. I’d

encourage you to use Markdown to write your paper and publish using Overleaf in real time.

### 1.13 Why would you like to use Markdown with Overleaf?

It is the best of both worlds of Overleaf and Markdown. While Overleaf is versatile as a file manager and an app from where you can push your paper or a number of different directions, its main “Engine”, that of, LaTeX can be daunting for some, and is not very easy to read due to its syntax structure. On the other hand, markdown is easy to write and has a very simple syntax, therefore it is also easy to read. As it is based on plain text, you can give your copy of Markdown formatted document to anyone for editing, without any fuss about the structure that a LaTeX code embedded document. People who only use Word/wordprocessors and nothing else, or would not like to use Overleaf or are fussy about other applications, will not have any problem about opening Markdown formatted documents (just provide them with an extension of .txt). You can open such a document in any text processing application. Plus it is great for authoring Web pages. This means you write once in Markdown and purpose it for webpages, slide decks, PDFs, whatever. Here, I am going to show you you can use it to work with Overleaf as well without knowing LaTeX.

In order to get it work, you will need a bit of set up. If you do not know how to use Markdown, here’s a Markdown cheat sheet to get you started:

<https://warpedvisions.org/projects/markdown-cheat-sheet/>

In order to use Markdown and write using Overleaf, in addition to a plain text editor or an Editor with specialised Markdown syntax checks (run a search on the Web to find out which these apps are), you will also need Git and Pandoc. A full discussion of these apps is beyond the scope of this short tutorial, but I strongly encourage you to look up and learn how to use these.

### 1.14 Install Git and Pandoc first

1. Install Pandoc (a free, open-source app) on your machine. Get Pandoc from here: <http://pandoc.org/index.html>
2. Install Git. Like Pandoc, Git is also free and Open Source Software. It is used for version control. Git interfaces with Overleaf). You can download Git from [Git](#)
3. In order to work with Git, Pandoc and Overleaf, always start with Overleaf. In Overleaf, create your new project and give it a name. Fill in the details
4. Then, head over to main.tex file and add the following code at the preamble:

```
\documentclass[a4paper]{article}
\usepackage[english]{babel}
\usepackage[utf8x]{inputenc}
\usepackage{amsmath}
\usepackage{graphicx}
\usepackage[colorinlistoftodos]{todonotes}

\usepackage{booktabs}
```

```

\usepackage[colorlinks=false]{hyperref}
\usepackage{enumitem}
\usepackage{longtable}
\setlist{nosep, noitemsep}
\providecommand{\tightlist}{%
\setlength{\itemsep}{0pt}\setlength{\parskip}{0pt}}\title{Write your Title}

\author{Your Name \ \ Other Information}
\date{\today}

\begin{document}
\maketitle

%\include{segments of the paper
}

\bibliography{references}
\bibliographystyle{plain}
\end{document}

```

You will see that these new sets of packages are just more additions to the ones that you already set up for the main.tex we described earlier.

## 1.15 Clone the document from Overleaf to a folder and git it

In order to do this, you will need to know a little bit of Git. Git is a version control system and you can learn about working your way with Git from the following resource:

<https://githowto.com>

I suggest you work your way through the tutorial. But for this tutorial, you do not have to read it first. Follow along this tutorial.

Here are the steps to get the git going with Overleaf:

1. First create a new project in Overleaf.
2. Visit the Overleaf project. See where it says “share” and open that window. In that page, you will find a box “Clone with Git”
3. Make the changes in the main.tex file as described above so that the main.tex should look like above
4. On your desktop create a folder where you will store your local files
5. Then, open a terminal (I assume you have Git and Pandoc installed and properly configured) and type:

```
git clone <url of the share>
```

The Git app on your computer will work and put the contents of the Overleaf folder of your project there. You have now created a Git repository (note that it will have a name that similar to the project name of your Overleaf (a name that looks like a code given to it by Overleaf, not the name or title you gave it). This new folder which is now connected to your Overleaf account through git is referred to as Repository. You will add and modify this to talk to Overleaf and work from here. Change into this directory by typing

```
cd <the name of the repository>
```

The filename extension for markdown documents is .md, but as it happens, you will not need markdown documents in Overleaf; you will only need tex files or latex files. So, you may want to keep the computer from uploading the markdown files. Hence, you will need to create what is known as a .gitignore file and indicate that you will keep all files with .md in them and push them to gitignore. This makes sure that they will not be transferred to Overleaf. Here is the manual for gitignore, and I recommend you read it:

<https://git-scm.com/docs/gitignore>

In Mac for instance, in your directory, you create a .gitignore file by doing:

```
touch .gitignore
```

Then, in this gitignore file, in each line, you type the name of the file or pattern that you want Git to ignore while staging changes and then uploading to your Overleaf repository. In our example, we would like to ignore .md files. So, you do as follows:

```
echo \*.md > .gitignore
```

Now that you have created a file .gitignore and added .md to it, you will need to push the .gitignore to the Overleaf repository. For this, write the following code at the command prompt:

```
git add .
```

Do not forget to add the dot after the space after git add. This will make sure that you will not forget to add all files to the Overleaf repository. For now, this code is going to add .gitignore file to the cue that it will be sent to the Overleaf repository. As you have also changed the local repository (this is called “staging”), you will now need to write a message (this is called commit). Write a commit message,

```
git commit -m 'added gitignore' -a
```

Note the structure of this message. It should start with a single quote and end with a single quote. Also, note that you will add a dash and then m and a dash and then an a to the line. This ensures that you will not miss any file accidentally. Now you have to send the file to the Overleaf remote repository. You do that by executing the following:

```
git push
```

We shall later see that this is part of a process. I assume that by now you have already installed Pandoc on your machine. To check that pandoc is installed, do:

```
pandoc --version
```

When you do that, if Pandoc is installed, you will see the version information and packages that are installed. If not, or if it returns an error message, you will need to reinstall Pandoc, or install Pandoc if you have not already done so.

## 1.16 Write a Markdown document

Now is the time to write a markdown document. Remember that in our work, we are going to use two levels of headers (as Overleaf's wysiwyg writing environment allows you two levels; we can include enumerated lists, and bullet point lists; as we are writing a scholarly document, we should ideally include tables, figures, and citations drawn from a bibtex formatted bibliographic reference list. In our case that bibtex formatted bibliographic reference list is references.bib that we have downloaded via git from Overleaf. With these constraints you can write a document in markdown that looks somewhat like this:

```
# A Sample Markdown Document
## This is a second level header
The objective here is to write a simple document with some components we come to accept i

- This is the first element of a bullet point list
- This is the second element of a bullet point list

1. This is the first element of a numbered list
2. This is the second element of a numbered list
```

Writing a table is simple too, just as follows

```
| Header Element | Second Column Header |
|-----|-----|
| First cell    | Second Cell          |
| More info     | More more info       |
```

You can see how easy it is to create tables directly here. You can of course use Tablesgen. Finally, if you want to add citations, you need to add citation ID that you will find in y

```
@article{leijen2015,
  title={Madoko: Scholarly markdown in the cloud},
  author={Leijen, Daan},
  journal={SNAPL'15: The Inaugural Summit on Advances in Programming Languages},
  year={2015}
}
```

This is an article that discusses a markdown processor referred to as madoko. If you have

There are different ways to add citation usually adding an at sign and then the citation a

Let's say we save this document as 'first.md' in the same folder where we have all other files stored.

## 1.17 Process this document with Pandoc

We are going to ask Pandoc to do the following:

1. We shall provide the name of the document we want to change from Markdown to LaTeX, so we shall say `pandoc first.md`

2. Then we ask Pandoc to convert it from markdown to latex, so we say  
`pandoc first.md -f markdown -t latex`
3. Finally, we want Pandoc to output a latex document which will end with a tex extension and we decide to name it `first.tex`. Note that at this stage if we are only going to write a part of my paper I won't need a standalone document. At other times, we will need to output a standalone document. If you do, add `-s` to the mix.  
So, the code now looks like this:

```
pandoc first.md -f markdown -t latex -o first.tex
```

If you copy and paste the above code and run in the folder from a command prompt, Pandoc will run the code. If everything goes correctly, then pandoc will return no error message. You can check that your new file `first.tex` exists by typing `ls` at the command prompt and see that it exists in the folder. Next, if you want this file to be added to the `main.tex` for Overleaf to process, just make a small change to your `main.tex` file and include the file with

```
include{first}
```

Note, that we do not add the tex extension to it. That's it.

## 1.18 Now use git to sync the files

Now that you have done the following:

- Created a `first.tex` file
- Added an entry to the References list
- Modified the `main.tex` to add this file

You are ready to push these changes to the Overleaf server. So what you want to do is:

```
git status
git add .
git commit -m -a 'added first, changed files'
git push
```

## 1.19 Open Overleaf site and see the changes take effect

If you now visit the Overleaf site, you will see that the changes immediately take shape. You can check the rich text editor and modify changes there. You can view the rendering in the PDF window. Your document is now ready to be used by the Overleaf system to be pushed to other sites for submission, or adding more changes. You can share the document to your colleagues or collaborators for editing. You may see in the first pass, it is not really perfect, and you will need to make changes. In particular, you will need to make changes to the citations. This is because in our markdown document, we did not use the markdown formatting of citations and used the LaTeX style of writing citations without the backslash signs. You will now need to correct these changes. I usually tidy up the references the last thing after I am satisfied with all my references. You can also add figures, and other elements to the document.

Another thing you may want to do is change things in Overleaf, and then bring those changes back to your computer or to the Markdown editor. Then you can work on the document more and then push the changes to Overleaf

again. This means you will be pulling in changes, and then convert the tex document to a markdown document, and then edit the markdown document and then push it back again. Here is the code that you can use to accomplish those changes.

```
git status
git pull
pandoc file.tex -f latex -t markdown file.md
git add .
git commit -m -a 'Your message goes in here'
git push
```

Notes:

1. The first line will now pull in the changes from your Overleaf document space
2. You may additionally want to convert the tex file that you pulled in to your document space and work there. Note that if you use pandoc, it will not give you back the LaTeX tables to the neat format you used for writing in your markdown document, so be a little tactful about what to do.
3. Then continue to work on your markdown formatted or tex formatted files and add and push.

This way you keep working on your markdown document either on the desktop or on the web, and as you play with git and pandoc, your document gets pushed to Overleaf in a tidy format. You can of course modify and continue to work in Overleaf as well, and pull in the changes when you need to work on them more.

## 1.20 Summary

So the workflow goes like this:

1. Start a document with Overleaf.
2. Use git to sync that document with a folder of your choice on your desktop
3. If the Overleaf document already has some text based tex codes written (not too much tables, etc), then convert the tex document using Pandoc to markdown and work on it.
4. Use Pandoc again to convert the Markdown formatted document to LaTeX and use git to upload back to Overleaf.
5. Let Overleaf compile the tex and you can make changes

Here, I have provided you with a basic workflow to work with Overleaf only using rich text editor and markdown.

## References

- [1] S Leonard. Guidance on markdown: Design philosophies, stability strategies, and select registrations. Technical report, 2016.
- [2] Luca Merciadri. Tex in the ebook era: a simple–yet fully working–approach. *PracTeX journal (The)*, 2012(1), 2012.
- [3] Johannes Wilm and Daniel Frebel. Real-world challenges to collaborative text creation. In *Proceedings of the 2nd International Workshop on (Document) Changes: modeling, detection, storage and visualization*, page 8. ACM, 2014.