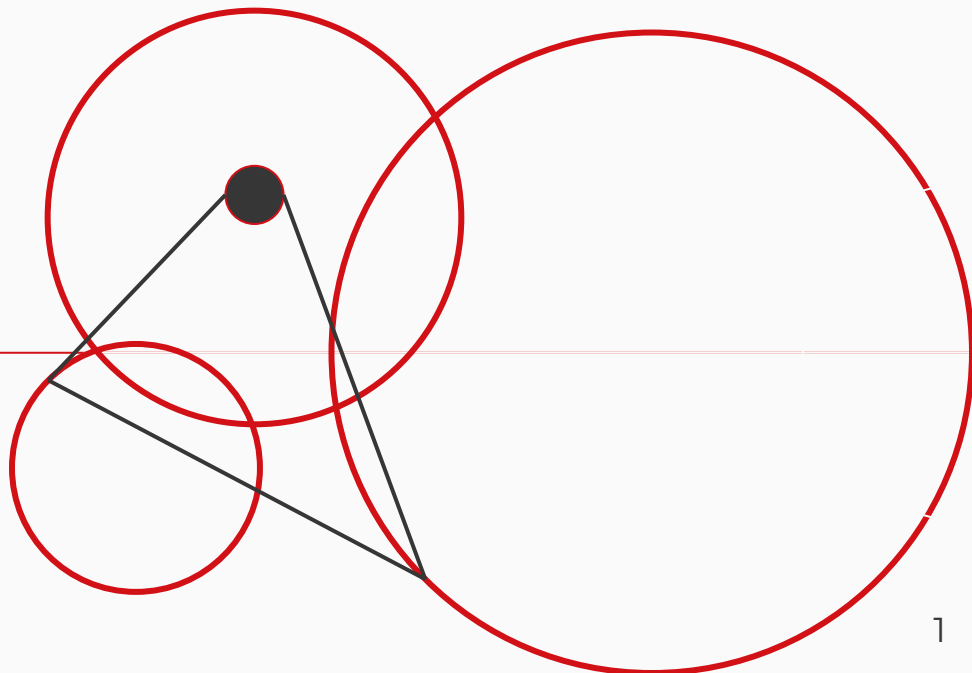


# *Using Google Colab and GitHub*

Fall 2025



*Written by Jessica Huynh-Westfall, PhD*  
SJSU CS 133

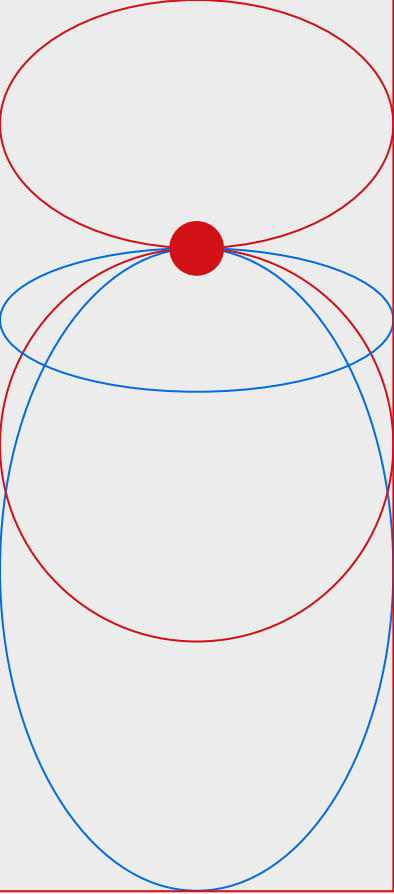


# *Agenda*

---

CS 133 - Data visualization

- Setting up GitHub
- What is Google Colab?
- Let's get some practice in



# *Version control*

*Using git and GitHub to maintain code*



Git is one of the most common version control systems

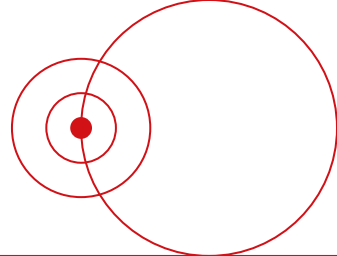
<https://git-scm.com/>



Web-based service that provides hosting of your **git** repository.

<https://github.com/>

# Version control



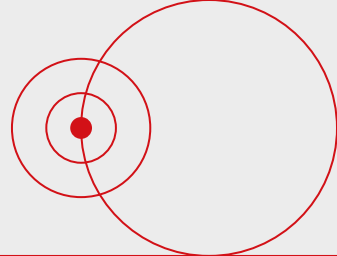
## Version control track changes

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later

- track changes with “snapshots”
- allows user to revert to an earlier snapshot

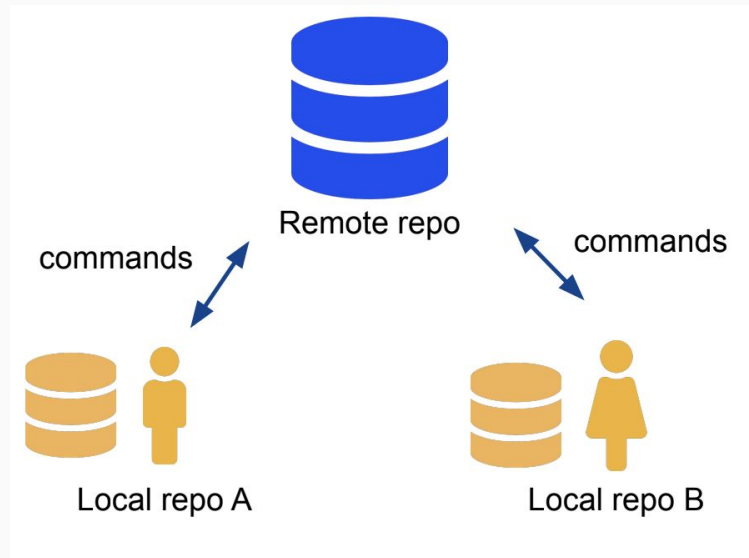


# Git and version control



## Using Git to manage version control

- ❖ Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later
- ❖ Git is one of the most common version control systems
- ❖ GitHub is a web-based service that provides hosting, utilizing the Git



# *Using GitHub to store files and code*

## **Use GitHub to manage your code and changes**

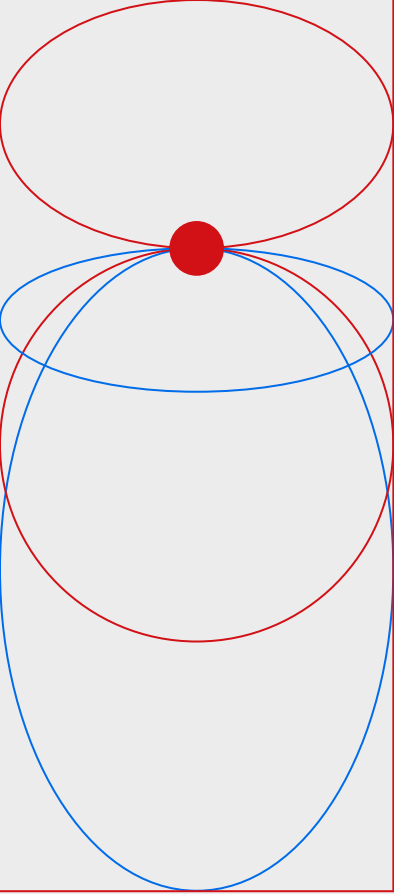
- ❖ Store, change, merge and collaborate on files or code
- ❖ Use Git for version control
- ❖ Visibly tracking iterative changes
- ❖ Team can look back at previous versions of the code to track changes and reverse if necessary
- ❖ Work with unlimited collaborators on project

# *Course GitHub to store files and code*

## **Git repo for class files**

- ❖ Clone the repo  
<https://github.com/CS133-DataVisualization/CS133-classfiles>
- ❖ This repo will be use to download data files, collab notebooks, and other class related files.

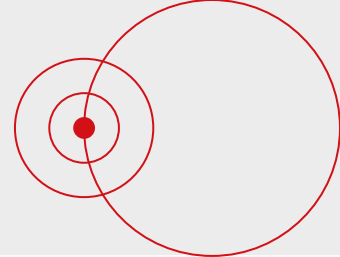




# Google Colab

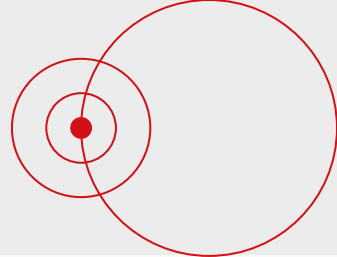
*Notebook that allows for rich text and  
executable code*

# *Why are we using Google Colab?*



- ❖ Cloud-based Jupyter notebook environment
- ❖ No setup, easy sharing/collaborating
- ❖ Write and execute Python code
- ❖ Free GPU and TPU (eg., TensorFlow, PyTorch)
- ❖ Import datasets from Kaggle
- ❖ Import to GitHub
- ❖ Notebooks are ideal for data analysis because they can include:
  - Nicely formatted text (e.g., text that describes the data)
  - Code and code output
  - Spaces for displaying plots

# *Markdown language*

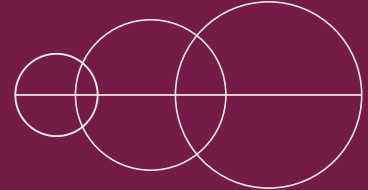


## **What is markdown?**

Formatting rich text cells as simple markup language. The Markdown source shows the source text and the rendered version.

Colab has two type of cells; text and code.

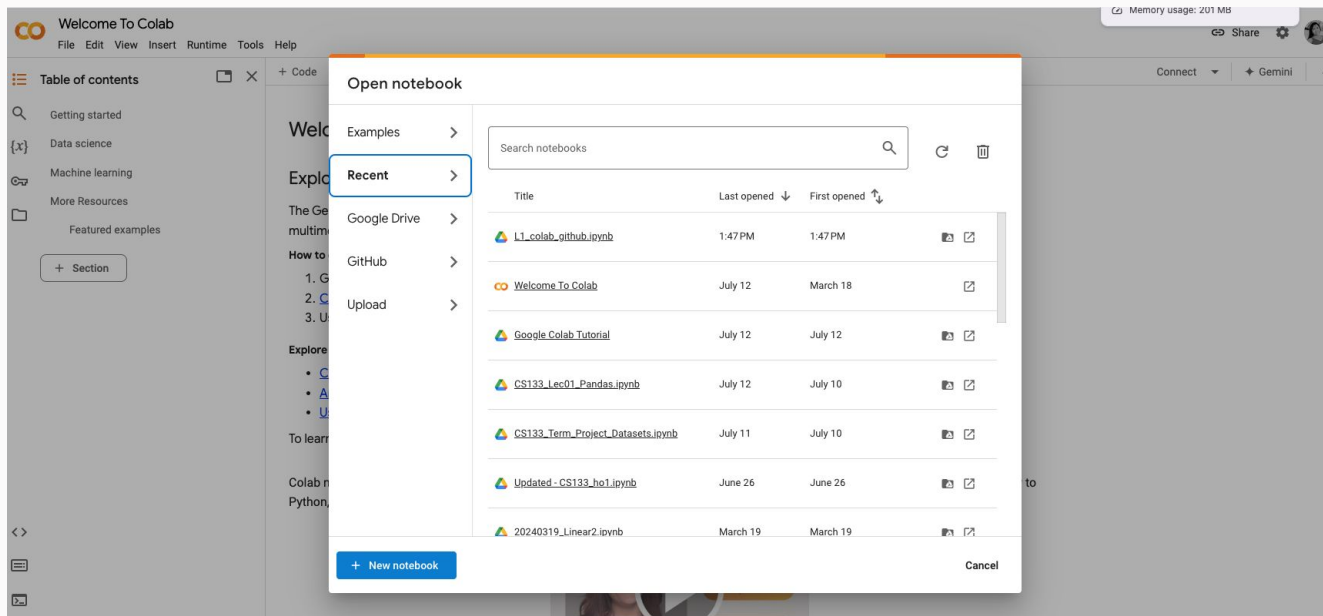
# Getting started



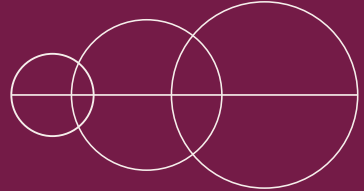
❖ Need Google account, GitHub account and Kaggle (optional)

❖ Open <https://colab.research.google.com> and click on

+ New notebook



# Notebook example



- ❖ Pull the GitHub repository to make a copy of the in-class hands-on notebook.
- ❖ Each bit of text or code is referred to as a **cell**

Note that code cells can also contain **comments**, which are lines of text that are *not* evaluated as code. Comments are used to write brief notes about specific lines of code, and are preceded by the `#` symbol. In the example below I've included a comment and some code.

#### Exercise 14

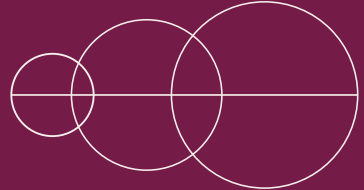
Run the cell to see what happens.

```
[ ] 1 # This is a comment: 10 - 5
    2 10 - 2
```

Text cell

Code cell

# Google Colab sample notebook



Each bit of text or code is referred to as a **cell**

## ▼ Sample notebook

This is text! You can make things **bold** or *italic*, you can include [links](#), and you can make bulleted lists:

- The is the first bullet
- This is the second
- Last one

```
[1] 1 2 + 2
```

```
↳ 4
```

```
[3] 1 a = 3
```

```
2 b = 5
```

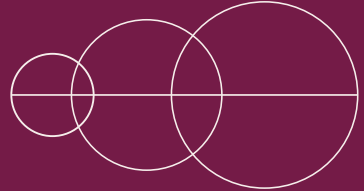
```
3 a > b
```

```
↳ False
```

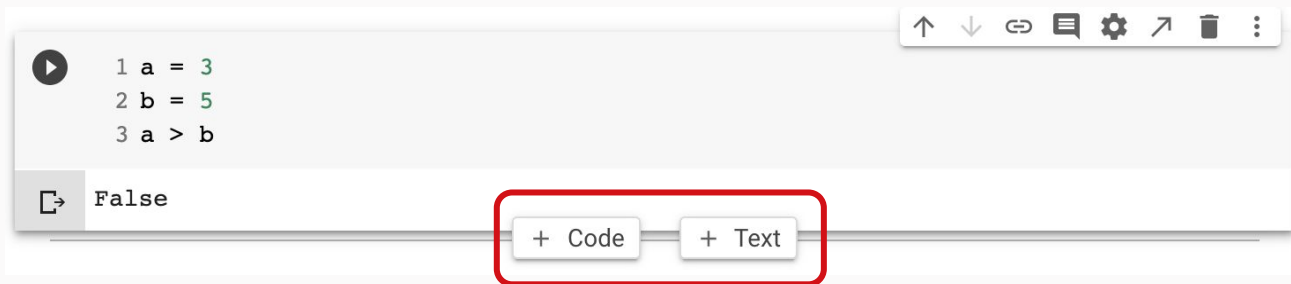
Text cell

Code cell

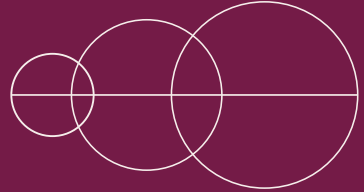
# Creating new cells



- ❖ Some exercises will ask you to create a new cell
  - Single click on an existing text/code cell, and then hover your mouse towards the bottom edge of the cell, in the center
  - You can select either a Code cell or Text cell



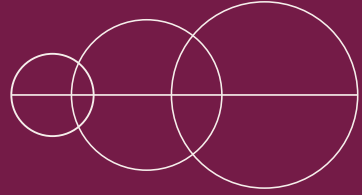
# Formatting text cells



- ❖ Colab has buttons for things like:
  - Changing font **size**
  - Making things **bold** and *italic*
  - Creating numbered/bulleted lists
  - Creating links
  
- ❖ Note: you can make these formatting changes by typing, i.e., without clicking buttons
  - As you work through the hands-on, you'll learn how to do this



# Formatting text cells



The screenshot shows a text editor window with a toolbar at the top. The toolbar includes icons for text formatting (bold, italic, code), alignment, link, image, list, and other functions. The main text area on the left contains the following text:

```
# My message  
I hope *everyone* is staying safe and that no one's home has  
been affected by the **terrible** fires! |
```

The text on the right is a preview of the formatted text:

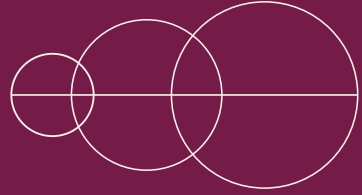
**My message**

I hope *everyone* is staying safe and that no one's home has been affected by the **terrible** fires!

You'll type here,  
on the left

On the right  
is a preview

# Formatting text cells



The image shows a text editor interface with a toolbar at the top. The toolbar contains icons for various formatting actions, each with a red arrow pointing to a label above it:

- Change font size (points to the font size icon)
- Bold (points to the **B** icon)
- Italic (points to the *I* icon)
- Format as code (points to the `<>` icon)
- Link (points to the link icon)
- Insert image (points to the image icon)
- Indent (points to the indent icon)
- Numbered list (points to the numbered list icon)
- Bulleted list (points to the bulleted list icon)
- Add horiz. line (points to the horizontal line icon)
- Reposition preview (points to the reposition preview icon)

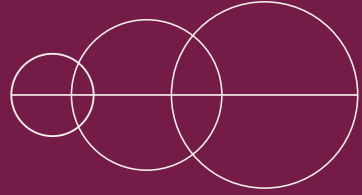
Below the toolbar, the text editor shows the following content:

`# My message`  
I hope *\*everyone\** is staying safe and that no one's home has been affected by the **\*\*terrible\*\*** fires! |

To the right, a preview of the formatted text is shown:

**My message**  
I hope *everyone* is staying safe and that no one's home has been affected by the **terrible** fires!

# Formatting text cells



The screenshot shows a text editor interface with a toolbar at the top. The text area contains the following content:

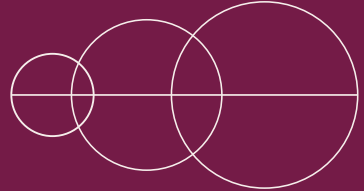
```
# My message
I hope *everyone* is staying safe and that no one's home has
been affected by the **terrible** fires!
Hello.
```

The rendered output on the right is titled "My message" and displays the text with formatting: "I hope *everyone* is staying safe and that no one's home has been affected by the **terrible** fires". The word "Hello." is on a new line, indicated by a red box around it in the original image.

To create a new line inside a text cell, use 2 spaces \_ \_ at the end of the line, then hit enter

This screenshot is identical to the one above, but with a red arrow pointing to the two spaces at the end of the line "been affected by the **\*\*terrible\*\*** fires!". A red box below the arrow contains the text "Added 2 spaces".

# *“Running” a cell*



When editing a text cell, the left side looks rather ugly. To make things look nice (like the preview), we need to “run” the cell.

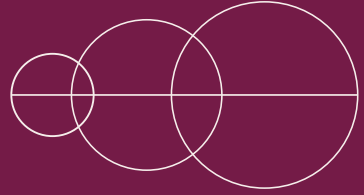
Simultaneously hit **Shift** and **Enter** to run a cell

-or-

Hit the  (play) button

This is true for code cells too.

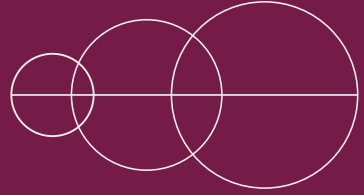
# *Code cells to run python scripts*



Code cells can run simple and complex Python code

An example is using Python as a calculator to run simple mathematical expressions.

- Type in a simple mathematical expression and then run the cell
- The output (found under the code cell) will be the answer



## To run simple mathematical expression

```
[1] 1 2 + 2
```

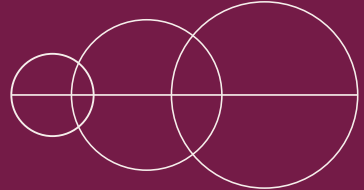
```
↪ 4
```

```
[3] 1 a = 3
```

```
2 b = 5
```

```
3 a > b
```

```
↪ False
```



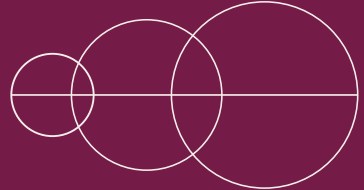
## LaTeX for rendering mathematical equations and symbol

**$\tau$**  **B** *I* <> ↺ ↻ ↶ ↷ —  $\Psi$  😊 📄

### You can also display math symbol using `\$\$`  
`$y = \sqrt{k}$ + $x^2$ - $\sin(y)$`

You can also display math symbol using \$\$  
$$y = \sqrt{k} + x^2 - \sin(y)$$

# Common LaTeX commands



Fraction

`\frac{numerator}{denominator}`

Subscript

`x_i`

Superscript

`x^2`

Greek

`\alpha, \beta, \sigma`

Symbols

`\infty, \int, \sum`

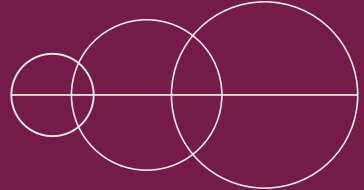
Functions

`\sin, \cos, \log`

Matrix

`\begin{matrix} ... \end{matrix}`





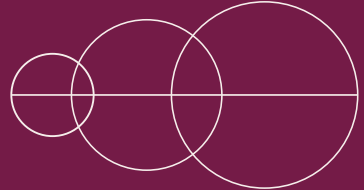
## Write comments about the code

Within a code cell, you can also include comments about the code

**Comments:** Lines of text within a code cell that are not evaluated as code

- ❖ Used for writing brief notes about certain lines of code
- ❖ Preceded by the # symbol
- ❖ For multi-line comments use `''' '''`

# Comments within code cells



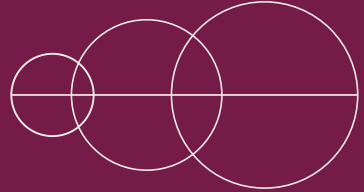
When running the following cell, we see the output of the code (line 2) and not the *comment* (line 1)

```
[5] 1 # 5 + 1  
    2 2.6 * 3.4
```

← comment

```
↳ 8.84
```

Note: Please put a space after the # sign when writing comments



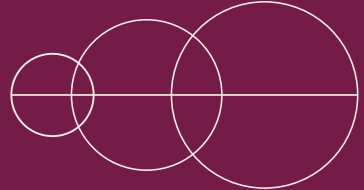
When should you use a text cell, and when should you write a comment in a code cell?

## ❖ **Comments**

- Generally very short (less than a sentence)
- Used to provide notes about specific lines of code

## ❖ **Text cells**

- Appropriate for longer text or text that is formatted



## ▼ Practice notebook

In this notebook, I'll be practicing some simple Python commands. For example:

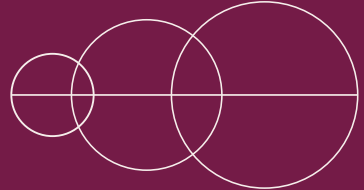
- Conditional execution
- Iteration

You'll be learning both of these skills later in the semester!

```
1 # conditional execution
2 x = 3
3 if x > 1:
4     print('Greater than 1')
5 else:
6     print('Less than or equal to 1')
7
8 # iteration
9 my_list = [1, 2, 3]
10 for num in my_list:
11     print(num)
```

```
☞ Greater than 1
1
2
3
```

# Notebook documentation



Google Colab interface showing a notebook titled "M1\_C1\_markdown.ipynb". The notebook content includes a heading "Introducing Markdown language", a section "# Colab Basics" with creation and update information, and a list of learning outcomes. Below this is a section "Colab Basics" with similar information and a list of learning outcomes.

**Introducing Markdown language**

# Colab Basics  
Created by Jessica Huynh-Westfall (SJSU ID)  
Last updated: August 12, 2025

Learning outcome: Learning how to use Google Colab to write markdown and code

1. Creating and formatting text using Markdown symbol
2. Writing mathematical equations
3. Write simple code and printing/viewing its output

**Colab Basics**

Created by Jessica Huynh-Westfall (SJSU ID) Last updated: August 12, 2025

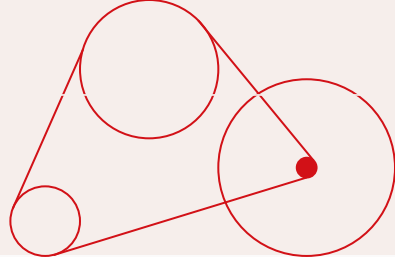
Learning outcome: Learning how to use Google Colab to write markdown and code

1. Creating and formatting text using Markdown symbol
2. Writing mathematical equations
3. Write simple code and printing/viewing its output

At the top of every notebook, it's helpful to include details like:

- ❖ Notebook title, using heading 1
- ❖ Name of person who created the notebook (or your name and SJSU ID, if it's an assignment)
- ❖ Date that you last updated the notebook
- ❖ Purpose of the notebook

# *Class activity*



1. We will work on M1\_C1\_markdown.ipynb. Download from the class repository.
2. Work on the in-class CL1.1 and CL1.2
3. Enter the code you wrote for CL1.2 into the CL1\_2 quiz on Canvas.  
Access code: “math”