

# GETTING STARTED IN GOOGLE COLAB

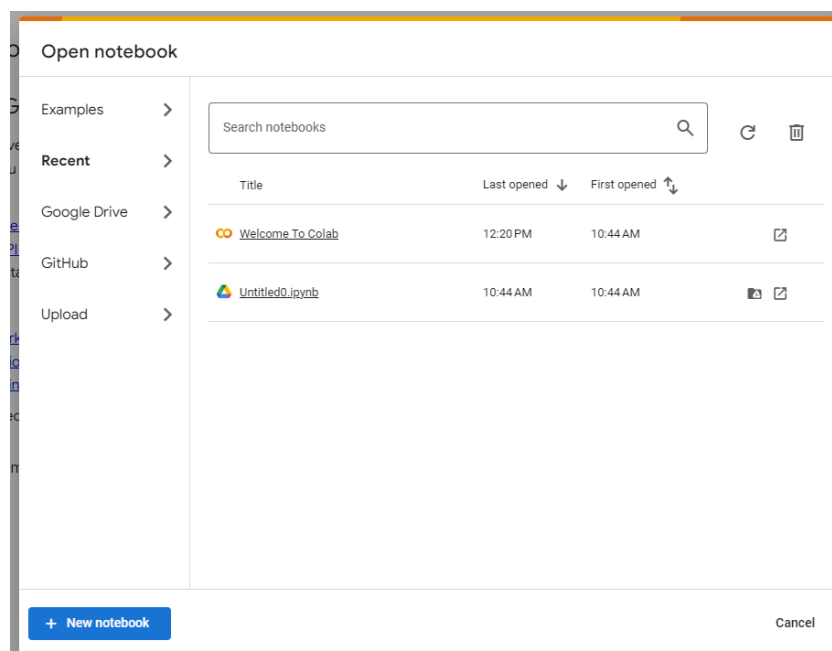


## Accessing Google Colab

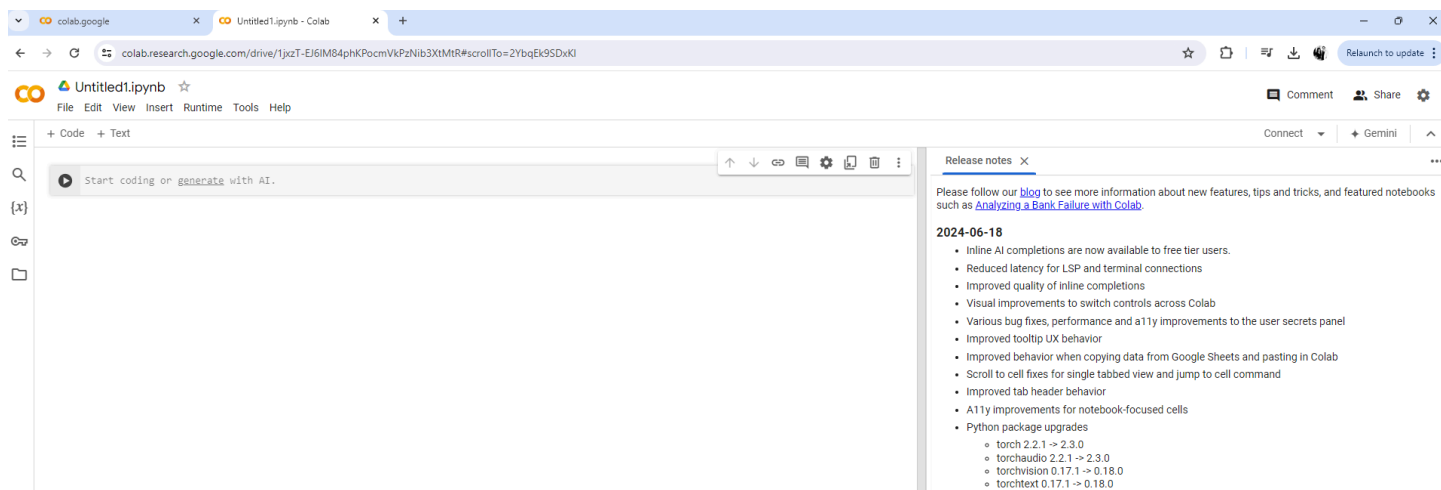
- You can freely access Google Colab at this [link](#).

## Creating a New Notebook

- When you navigate to colab.google, click on the “Open Colab” button.
- When you do so, a navigation wizard opens up asking you to open a notebook. Your window should look like the below:



- Click the “New notebook” button at the bottom left and a new window will open up which looks like this:



- You can change the name of your project from “Untitled1” to anything you like by clicking on the name “Untitled1” at the top of your screen. I’ll call mine “First Coding Project”.

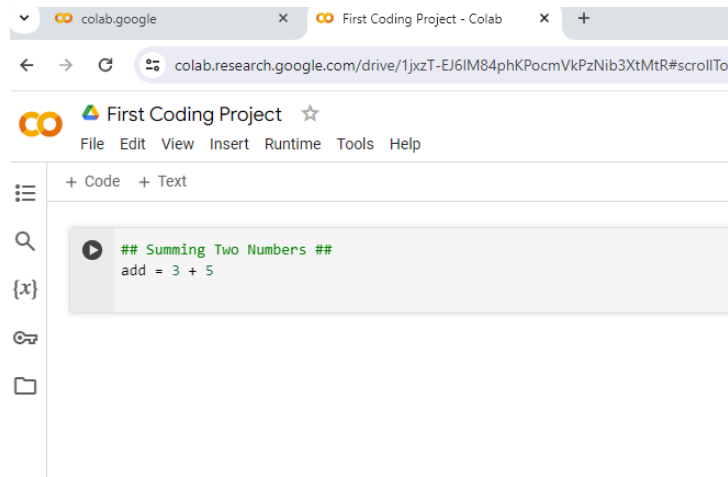
## Understanding Google Colab

- There are lots of interfaces we can use to run Python code. So why do we use Google Colab? Well here are a few reasons:
  - **Cloud-Based Jupyter Notebooks:** Google Colab provides an interface for creating and running Jupyter notebooks, which are documents that can contain live code, equations, visualizations, and narrative text.
  - **Free Access to Computing Resources:** Users get free access to powerful computing resources, including GPUs and TPUs, which are essential for tasks like machine learning and deep learning.
  - **Collaboration and Sharing:** You can share your Colab notebooks with others easily, allowing multiple people to collaborate on the same document in real-time, similar to Google Docs.
  - **Integration with Google Drive:** Colab is tightly integrated with Google Drive, enabling you to save and manage your notebooks in your Drive. This also makes it easy to import and export files to and from your notebooks.
  - **Free and Easy to Use:** Google Colab is free to use, and you only need a Google account to get started. There is no need for any complex setup or installation.

## Writing Python Code

- If we want to start writing Python code, we can simply click the “+ Code” button in the upper lefthand corner of our Google Colab window.
- When we do so, we get a grey rectangle with a Play button to the left which implores us to “Start coding”. This rectangle is called a “Code Chunk”.

- I recommend making comments/notes to ourselves about what the code is doing (or supposed to be doing). This is especially useful when working in teams.
  - We can do so within the code by using `#` symbols at the beginning of a new line
  - We can also add commentary about our code and its output by clicking the “+ Text” button.
- Suppose I want to write code to save the result of  $3 + 5$  as “add”
  - But I also want to make a note that I’m summing two values
- To do so, I will write the below code:



- Again, placing a `#` symbol at the beginning of a new line indicates to Python that you are making a note or comment.
  - Note, you don’t have to place two at the beginning nor do you have to place two at the end. That is my own personal style/preference

## Executing Code

- Now that we’ve written a little bit of code, how do we get Google Colab to actually run it? There are multiple ways but the easiest is simply to click the “Play” button to the left of our code chunk.
- With your cursor on a given line of code, you can also use the keyboard shortcut:
  - CNTRL + ENTER (PC)
  - CMND + RETURN (Mac)

## What Happens When Executing Code

- When we run code, like we just did to save the “add” object, a few different things can occur down in the console:
  - The code we just ran executed and output nothing.
- This means that the code executed successfully! Nothing is output because we are assigning the output to a new object called “add”

- We also know it executed successfully because we didn't receive an **ERROR** or **WARNING** message, which are also two things that can occur when executing code:
  - An **ERROR** message means the code did not execute at all
  - A **WARNING** message means the code executed, but there is something about the way it executed that we should be mindful of
- When the code executes, we can also receive output of some kind
  - For example, if you highlight just  $3 + 5$  and execute that code, the value 8 is printed to the console.