Introduction to Google Colaboratory

A powerful online platform for coding and collaboration. You might be wondering why we are switching from Spyder to Google Colab. Why not just continue in Spyder IDE?

Even though Spyder IDE is still a great tool for initial Data Analysis and cleaning, there are a few reasons why we need Google Colab. Machine learning algorithms require a lot of processing power when dealing with heavy data, and our computers are limited by the hardware. Google Colab provides the required processing power and facilities to run machine learning algorithms seamlessly. It also requires no prior set-up. Sometimes setting up environments and installing dependencies can be a little tricky. Thanks to Google Colab, we don't need to worry about that. We can jump right in and start running our code.

What is Google Colab?

Google Colab, short for Google Colaboratory, is a free cloud-based platform provided by Google that allows users to write and execute Python code in a web-based environment. It offers a user-friendly interface where users can create documents called notebooks, which consist of code cells and text cells. In these notebooks, users can write and execute Python code, visualize data, add explanatory text using Markdown syntax, and include multimedia elements such as images and videos.

It is particularly popular among data scientists and machine learning practitioners for its ability to run Python code in an interface similar to notebook environments, without requiring any setup on the user's machine.

Why Google Colab?

Now you might be wondering why Google Colab over IDEs like Spyder. Google Colab stands out for several reasons that make it a preferred choice over IDEs like Spyder. Here are a few reasons why:

- **Google Colab operates entirely in the cloud, eliminating the need for local installation or setup. Users can start coding immediately without any setup hassle.**

- **Google Colab comes with many popular Python libraries pre-installed, and users can easily install additional libraries using pip or other package managers. This streamlined process simplifies dependency management, especially for beginners.**

- **Google Colab provides seamless real-time sharing and collaboration features, allowing users to share notebooks via URLs and collaborate in real-time. This functionality enhances productivity and facilitates teamwork.**

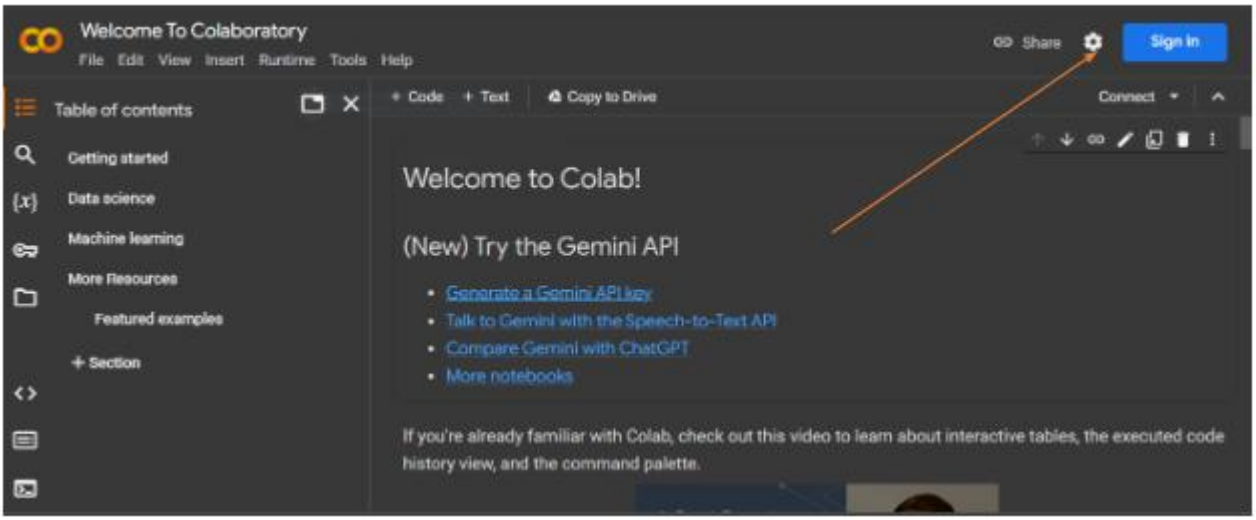
These advantages make Google Colab a convenient and efficient platform for data scientists and machine learning practitioners who prioritize accessibility, scalability, and collaboration in their workflow, especially when compared to traditional local IDEs like Spyder.

How do we start with Google Colab?

The first thing we will need to start our journey with Google Colab is a Google account. So if you don't already have a Google account, you can head to Gmail and register there to create a Google account.

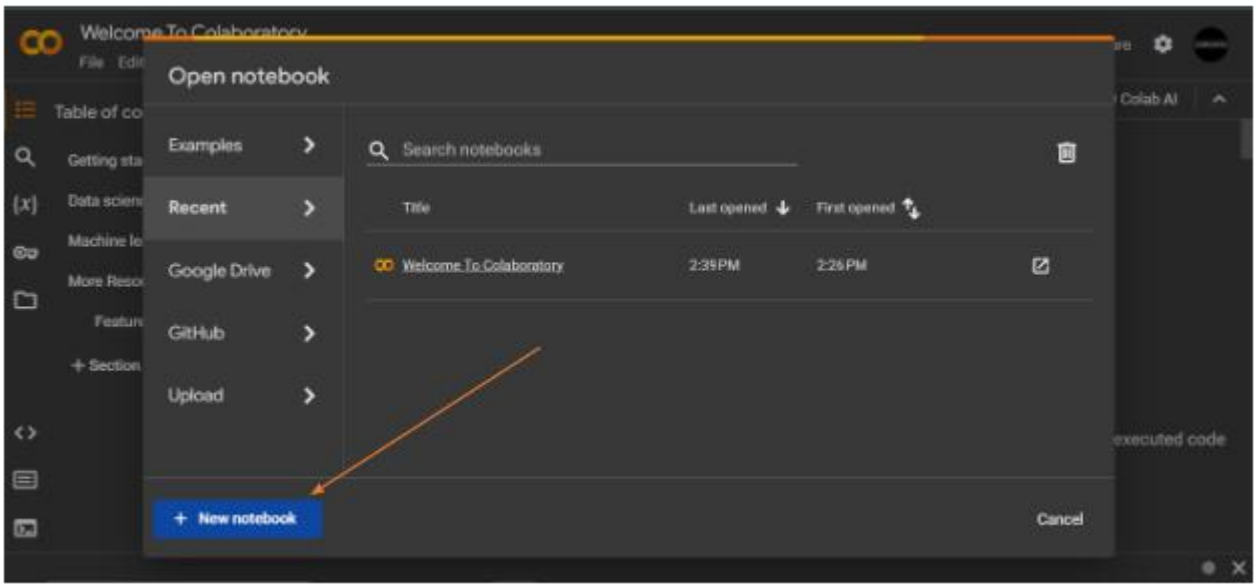
After we have our Google account all set up we can visit https://colab.research.google.com/ to start our journey.

This is something we will see:



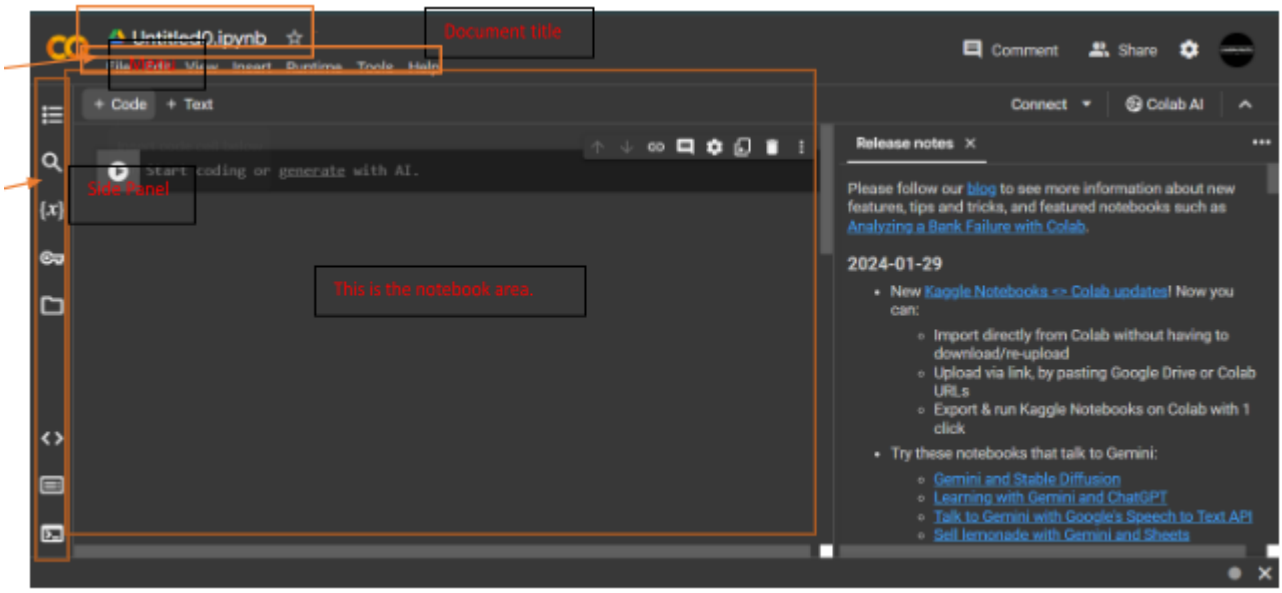
If you are not signed in you will see an option to sign in on the top right corner. You need to sign in so that you can save your file on the associated Google Drive.

After signing in, we will see an interface like this:



There is already a notebook named `Welcome to Colaboratory`. This is a sample notebook provided for us to understand and observe the different features of Google Colab.

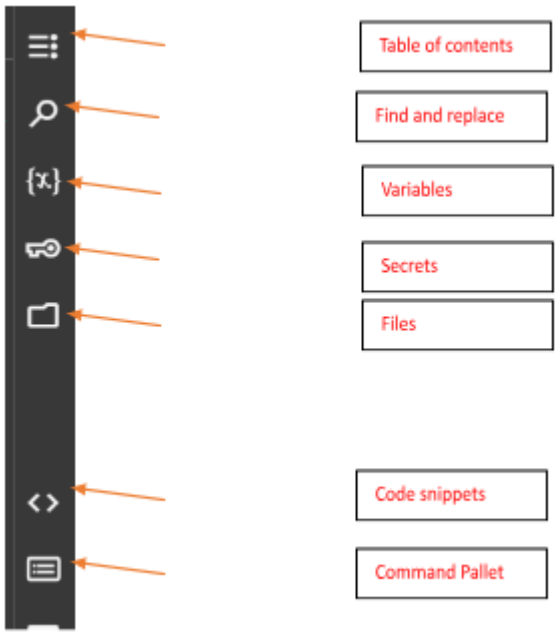
Here, if we click on the New Notebook option to create our new Google Colab Notebook. This is what our new notebook will look like:



As we can see it has a Document title, Menu, Side panel, and text editor. These are the main components that we need to understand. Let's understand the side menu first by taking the example of the sample notebook provided by Google:

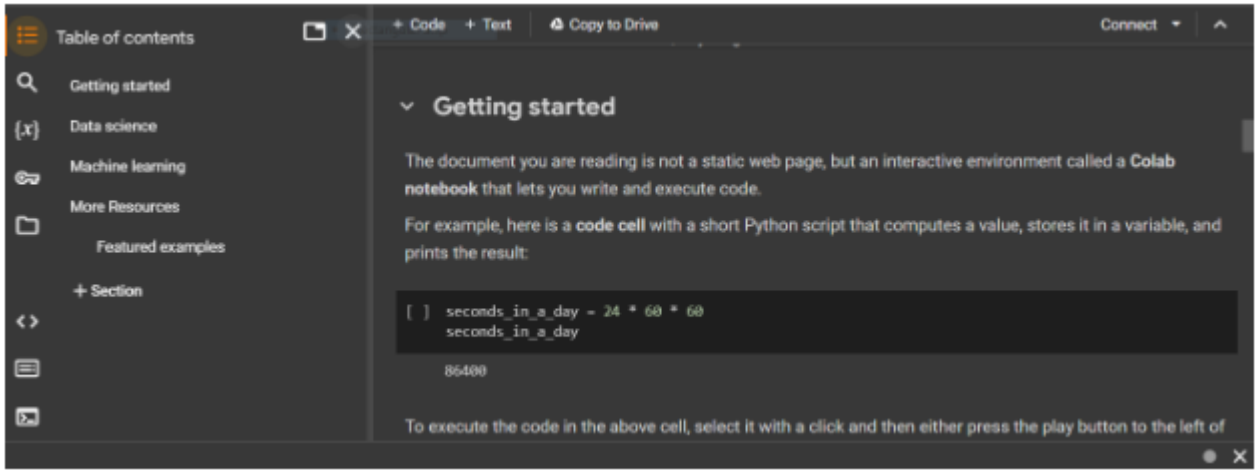
Table of contents:

The Table of Contents in Google Colab is an automatically generated index of headings and sections within a notebook, facilitating easy navigation and overview of the notebook's structure. It appears as a collapsible sidebar, allowing users to click on entries to navigate directly to corresponding sections. Interactive and customizable, it enhances usability by helping users quickly find and explore relevant sections within longer documents.



Find and Replace:

With the find and replace feature in Google Colab, we can find and replace things as required within our Colab notebook. This feature is important as we will not only be writing our codes in Google Colab but also inserting texts and explanations for our code. We might need to find and replace text as we do in a Word document. However, this feature is not only limited to text parts only. We can find and replace parts of our code as well.

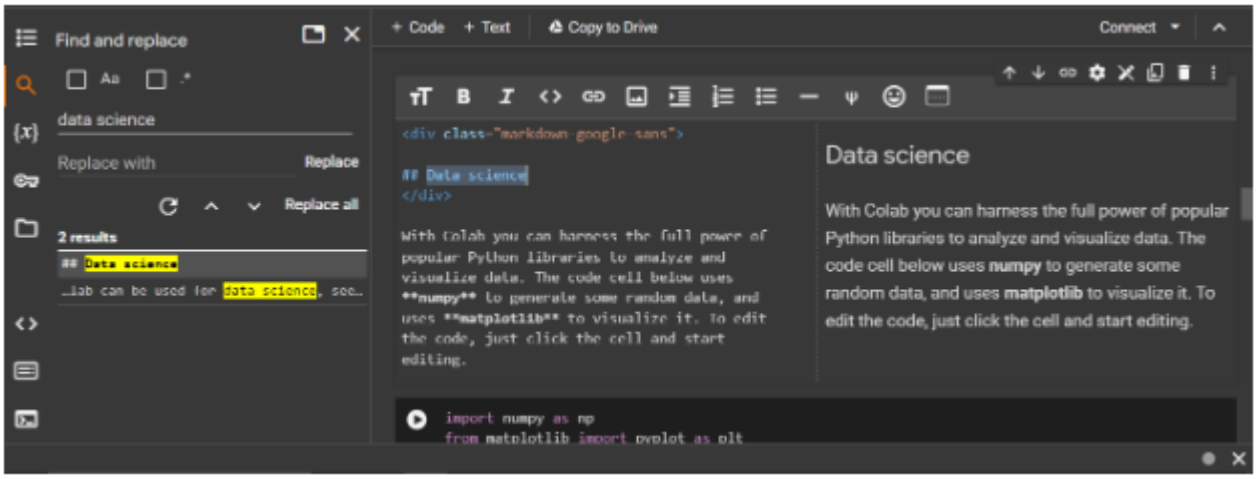


Variables:

The "Variables" tab in Google Colab is a feature that enhances debugging capabilities by allowing users to inspect and interact with

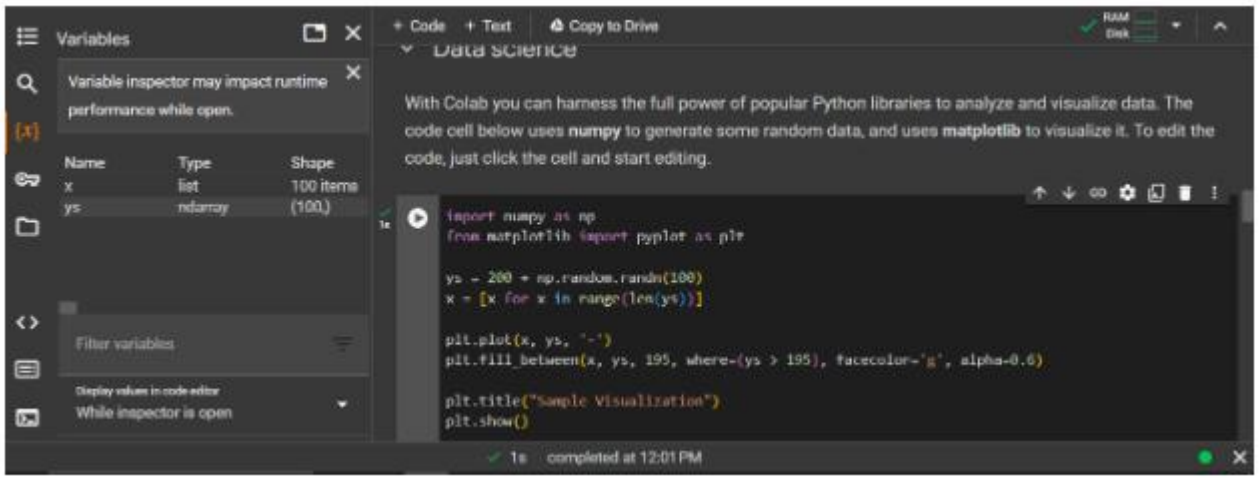
variables in real-time while their code is running. It is similar to the variables tab in spyder IDE. It provides a list of variables currently in memory, along with their values and data types, offering a quick overview of the state of variables during code execution.

It's particularly useful for debugging purposes, providing users with insight into the state of their variables at different points in their code.



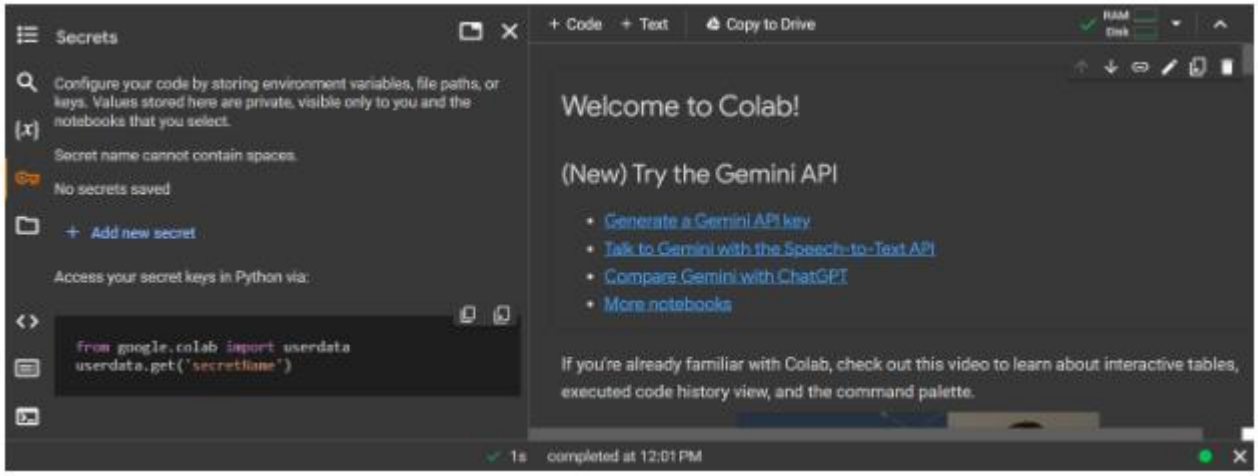
Secrets:

The Secrets tab in Google Colab is a secure and convenient way to store and access sensitive information like API keys, passwords, credentials, or any other data you don't want to hardcode directly into your notebook cells. This enhances security and prevents accidental exposure of sensitive values when sharing or collaborating.



You can picture the secrets tab as a locked box within your Colab notebook. It lets you store sensitive info like passwords or API keys

without revealing them in your code cells. This keeps things secure, even if you share your notebook.



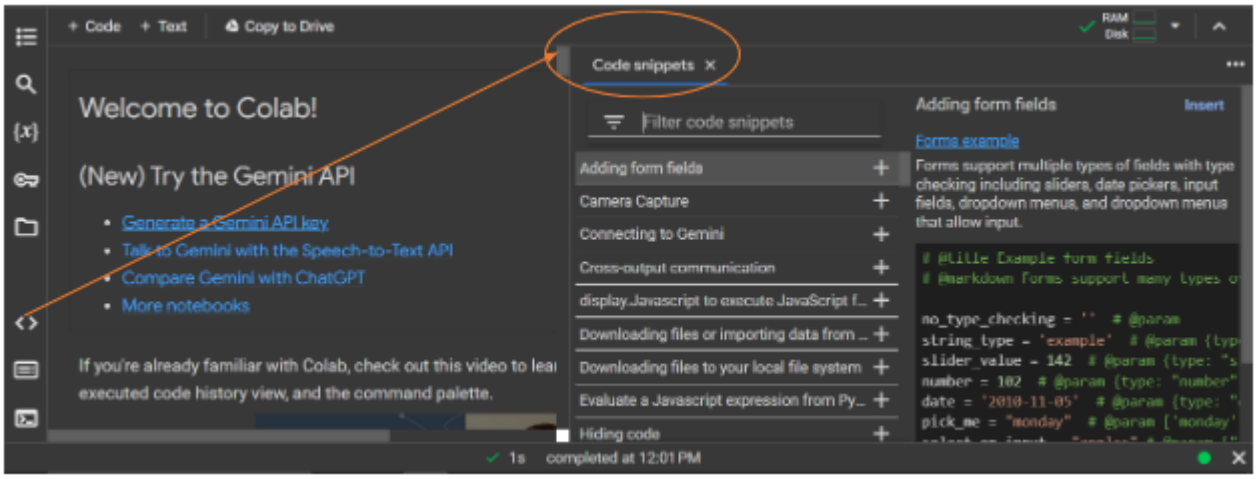
Files:

The Files tab in Colab is like a temporary desk drawer for your projects. Toss in files from your computer, code them fresh in Colab, or organize things with folders - it's all there. Click to open files, use code to interact with them, and even share with teammates. Remember, this drawer empties when you leave Colab, so connect your Google Drive or a similar cloud buddy for long-term storage. Think of it as a handy workspace, not a permanent filing cabinet, and your Colab projects will be clutter-free and accessible!

Code Snippets

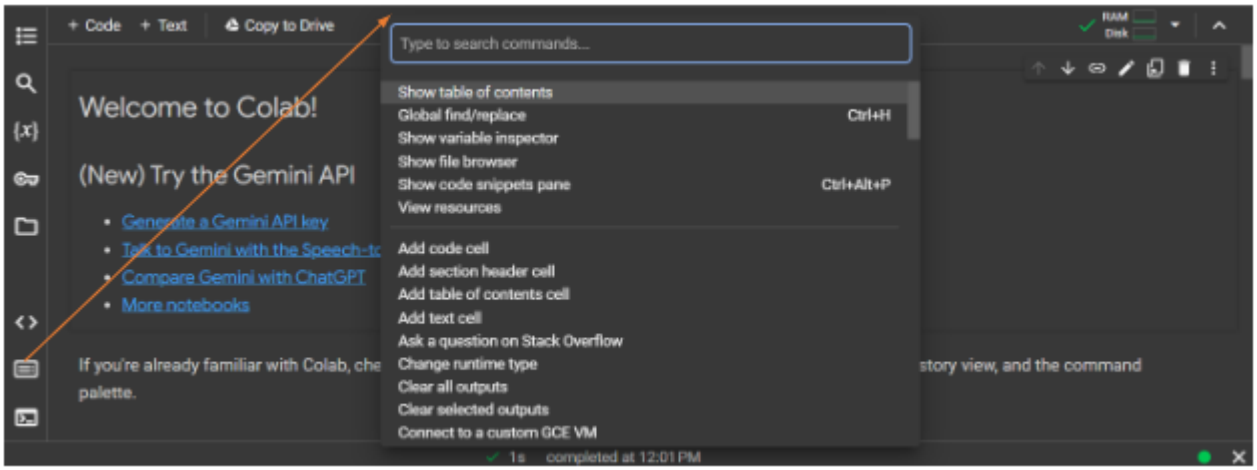
In Google Colab, code snippets are like pre-built Lego blocks for your code. Instead of typing the same stuff over and over, you can just grab a snippet and plop it in, saving you time and reducing typos. They come in all shapes and sizes, from simple loops to complex functions, and you can find tons of collections online. Think of them as handy shortcuts for common coding tasks, making your Colab projects faster and smoother.

In other words - code snippets are pre-written code that you can use if you're writing a common function or running a loop that is regularly used. Instead of writing the code yourself, you can simply copy and paste the code snippets into your program.



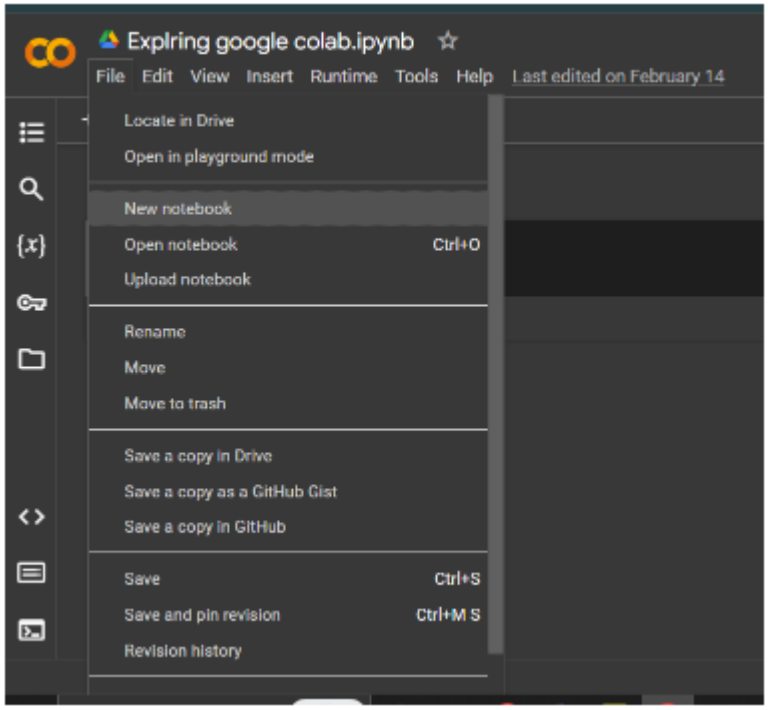
Command Pallet:

In Colab, the Command Palette is your keyboard shortcut hub. Think search bar, but for notebook actions: find code cells, access hidden features, and even insert pre-written code - all at lightning speed! Boost your productivity, discover cool stuff, and feel like a coding ninja with keyboard power!



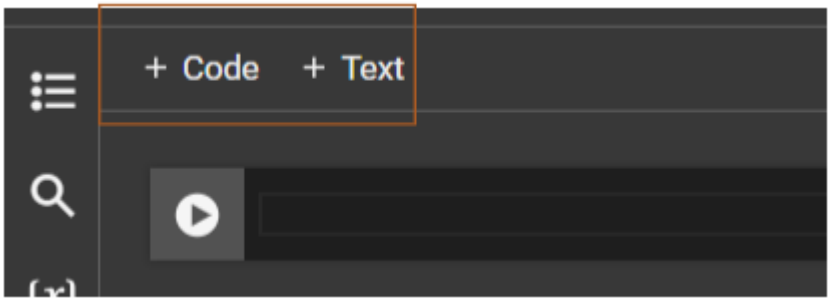
The Notebook editor:

To understand the editor, let's create a new notebook by navigating to the file menu and creating a new notebook.



Once the notebook is created, you'll see an empty cell where you can start typing. This cell is a code cell by default.

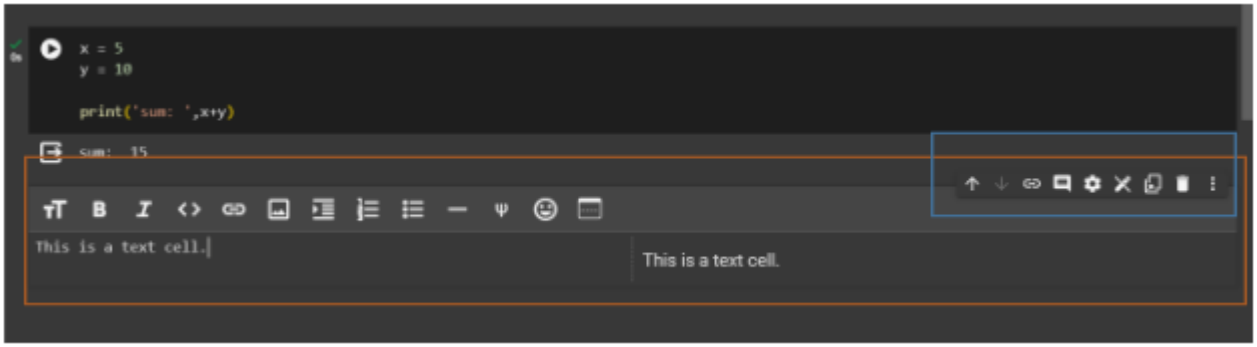
The code cells can be used to write and execute Python code simultaneously. It serves as a designated area where users can input, write, execute, and test snippets of code written in Python. Within a code cell, users can write any valid code, including variable assignments, function definitions, import statements, loops, conditional statements, and more.



Upon execution, the code cell processes the code input and produces the corresponding output, which may include numerical results, textual outputs, visualizations, or errors. Code cells play a crucial role in facilitating iterative development and experimentation, allowing users to refine and debug their code in an interactive manner.

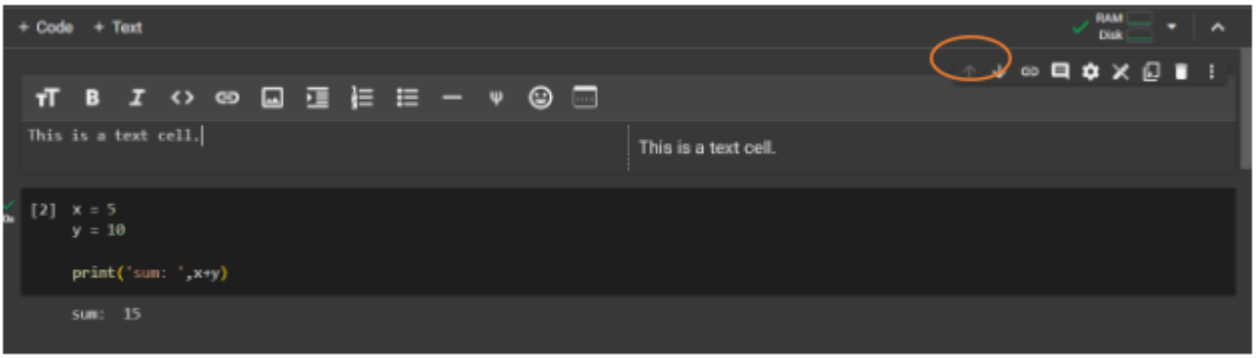
So in a code cell, we can write python code and execute it using the play button on the left of the cell. The output is shown below the code snippet when we execute the code.

We have the option to add code and text cells on the top left side of the editor.

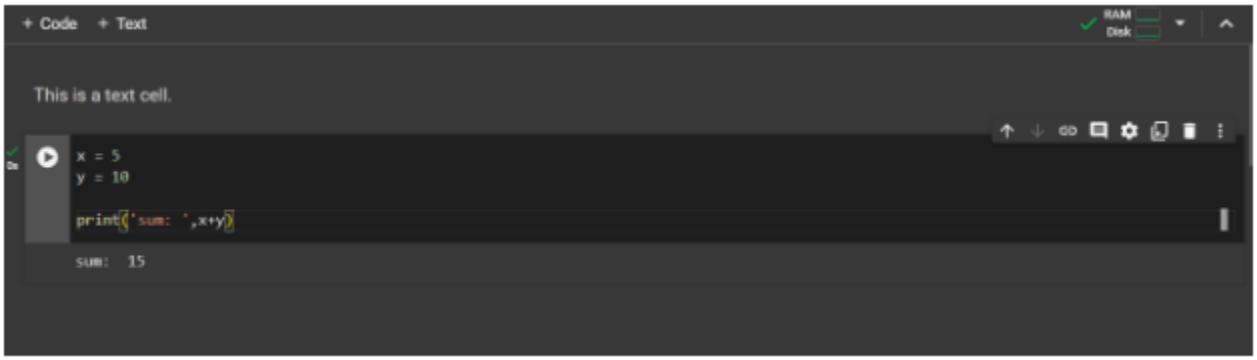


We can also add text cells. a text cell, also known as a Markdown cell, provides a space within a notebook for users to input and format textual content using Markdown syntax. Markdown is a lightweight

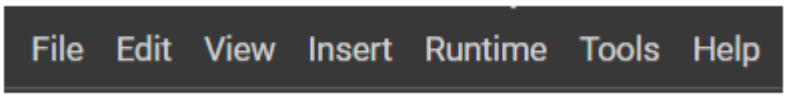
markup language that enables us to add structure, emphasis, and formatting to plain text without relying on complex tools like HTML.



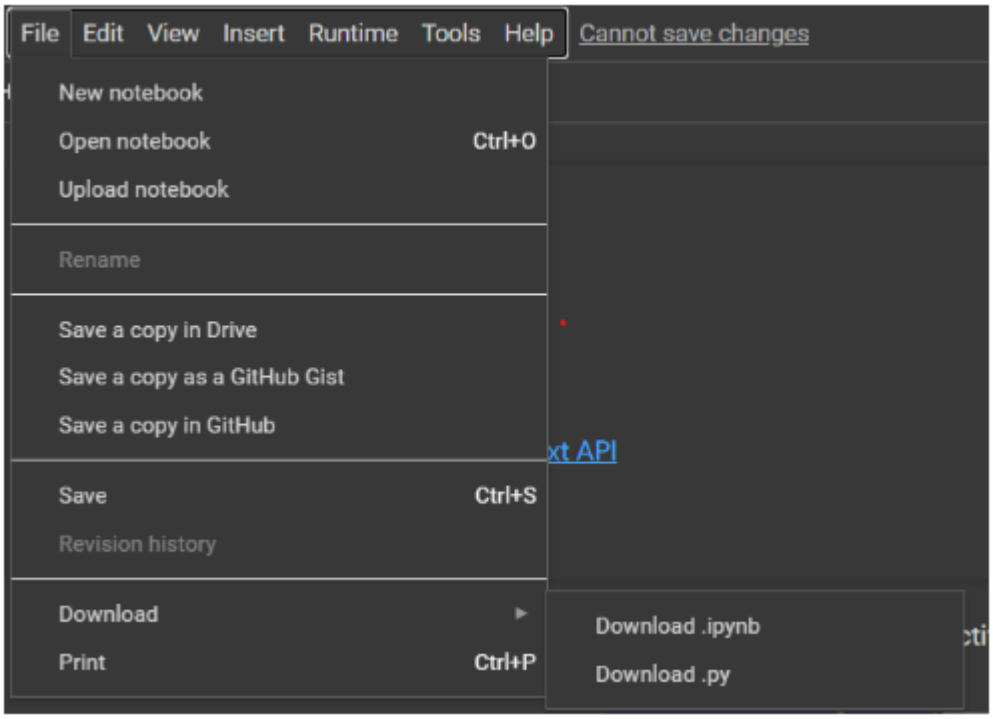
Within a text cell, we can include elements such as headings, paragraphs, lists, links, images, code snippets, and mathematical expressions. Text cells offer flexibility in documentation, annotation, and explanation, allowing users to provide context, instructions, annotations, or descriptions alongside code cells and visualizations.
We can use the arrow key from the menu hovering over the right side of the cell to rearrange the position of cells in the notebook.



The final output looks something like this:



By combining code cells for executable code and text cells for explanatory text, we can create cohesive and informative narratives within the notebooks.
Now let's understand the menu options:



In Google Colab, the menu bar offers options for various aspects of your notebook environment. Here's a quick breakdown:

File :

We'll find tools for managing our notebook in Google Colab's "File" menu.

Here, we can create a new notebook and open an existing notebook or upload a notebook from our local computer. We can also save our notebooks to Google Drive and GitHub. We also have an option for downloading the notebook.

There are two available formats for download `.ipynb` and `.py.`

When we download a file with the ".ipynb" extension, we're saving a notebook that contains both text and code. This file can be uploaded back to platforms like Google Colab or other notebook environments, where it will look and work just like it did when we were working on it. The text we wrote will show up as normal text, and any code we wrote will be ready to run as if we were still in Google Colab. It's a convenient

way to save and share our work while keeping all the formatting and functionality intact.

However, if we download this in `.py` format, we get a python file where the text cells will be in comments, and this python file will be in executable form on the python console; given that all the dependencies are met, which means that all the libraries that are required to run the script are installed.

Edit :

In Colab's "Edit" menu, you have your digital editing toolbox at hand. Suppose you accidentally erased some code? Just hit "Undo" or "Redo" to bring it back.

Need to reuse content across cells? "Cut" and "Copy" are your best friends. "Paste" what you've copied in a flash, saving you precious time.

You can use "Find/Replace" to hunt down specific text and swap it instantly. Whether you're polishing your code or streamlining your workflow, the "Edit" menu keeps your notebook looking sharp and efficient.

View :

In Google Colab, imagine the "View" menu as your personal workspace organizer. It lets you tweak things to make coding more comfortable and efficient. You can select what panels show on the screen. You can also see the executed code history here.

Runtime :

The "runtime" refers to the environment where your code runs and does its work. Think of it like the stage where all the action happens when you're performing a play. In the context of Google Colab, you can control this runtime environment, which means you can decide how your code is executed and managed.

You can control the underlying computing environment with runtime, including restarting the runtime and managing Python environments.

The runtime menu has options such as `restart session`, `disconnect and delete runtime` interrupt execution` these options allow us certain control over the runtime environment.

We can also run the code blocks in our notebook from the runtime menu. It has options such as `run all`, `run before` etc.

Tools :

The tools menu has options for opening command pallets, settings, and a keyboard shortcuts list.

Help :

In the help menu, you can get assistance with Colab features, access official documentation, and explore tutorials or community resources.

This is what we need to know for now. As we progress, we will keep exploring new features.