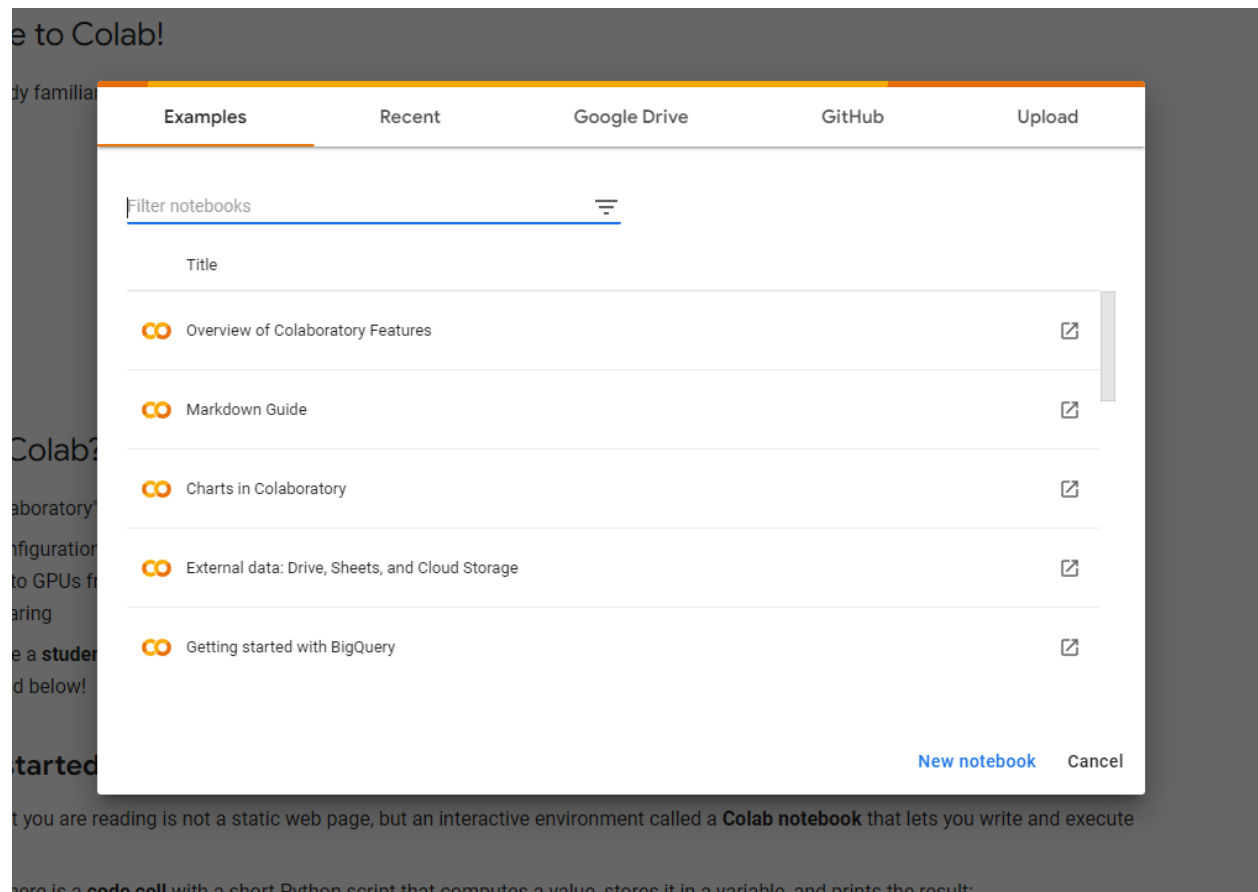


Getting Started with Colab

Google offers a free service called Colaboratory (or Colab), a Jupyter notebook in a browser environment where users may write and execute Python code with the only requirement being access to a Google account. Jupyter is the open-source project that Colab is based on. Colab allows users to share Jupyter notebooks with others without downloading, installing, or running anything on a local machine. Colab provides free GPUs and is especially well suited to machine learning, data analysis and educational purposes. Let's get started!

1. First need to **log in to your google account**.
2. Go to this link <https://colab.research.google.com>.
3. **Opening Jupyter Notebook:**

On opening the website you will see a pop-up containing the following tabs:



At this point, you may either open a new notebook on the lower right corner of the pop-up or you may upload a file by clicking the top right corner. You may also enter a GitHub URL or search by organization or user. Similarly, you can open any notebook stored in your drive.

4. Examples: Overview of Colaboratory Features:

You can click the “Overview of Colaboratory Features” in the Examples tab of the popup to see what an example notebook looks like. The example notebook has a good overview of features.

▼ Cells

A notebook is a list of cells. Cells contain either explanatory text or executable code and its output. Click a cell to select it.

▼ Code cells

Below is a **code cell**. Once the toolbar button indicates CONNECTED, click in the cell to select it and execute the contents in the following ways:

- Click the **Play icon** in the left gutter of the cell;
- Type **Cmd/Ctrl+Enter** to run the cell in place;
- Type **Shift+Enter** to run the cell and move focus to the next cell (adding one if none exists); or
- Type **Alt+Enter** to run the cell and insert a new code cell immediately below it.

There are additional options for running some or all cells in the **Runtime** menu.

```
[ ] 1 a = 10
    2 a
```

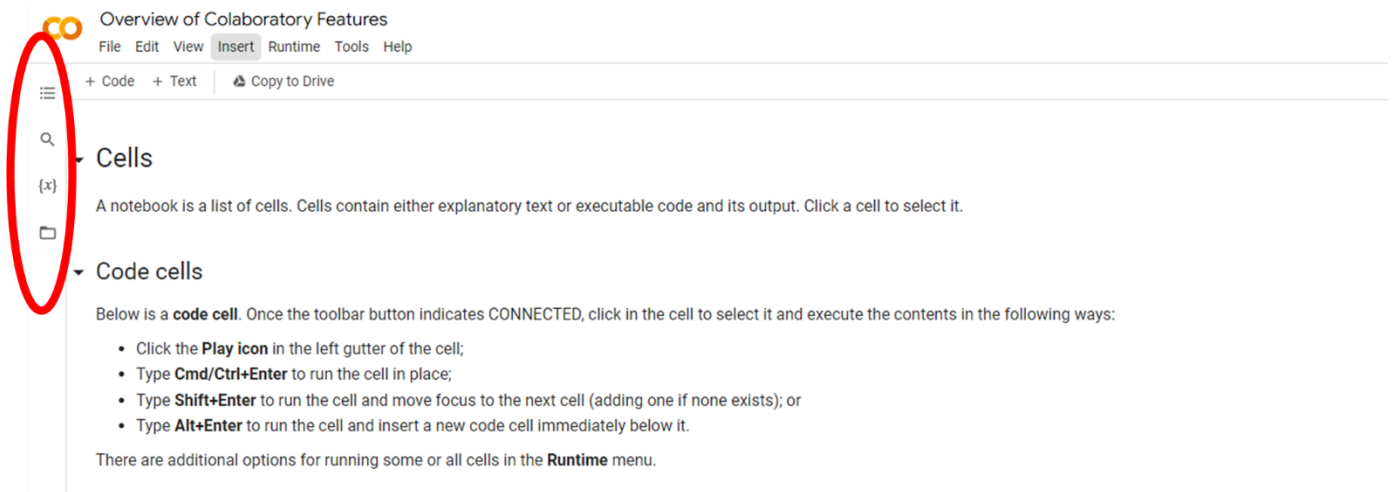
10

Text cells

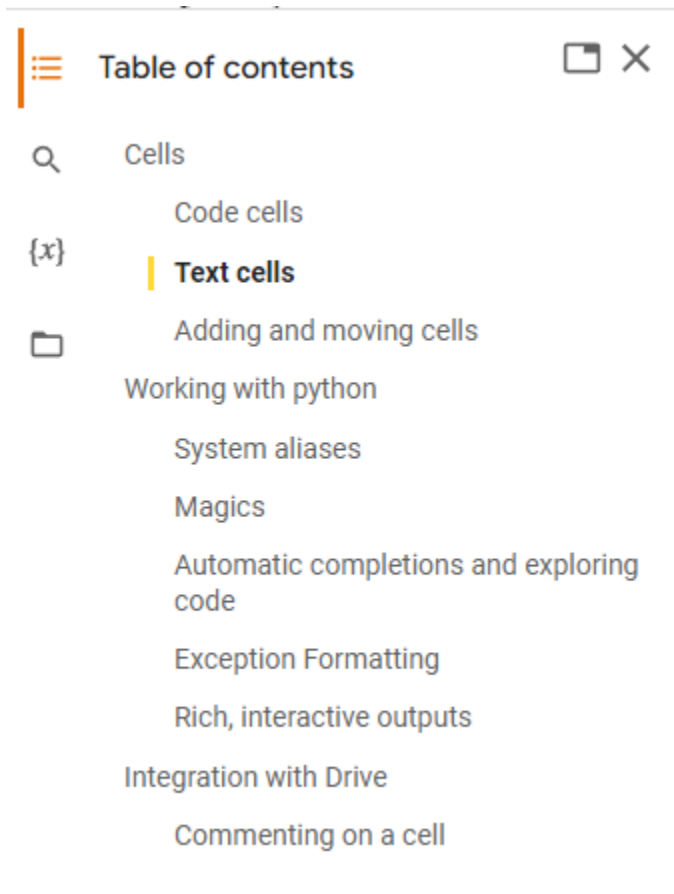
This is a **text cell**. You can **double-click** to edit this cell. Text cells use markdown syntax. To learn more, see our [markdown guide](#).

You can also add math to text cells using [LaTeX](#) to be rendered by [MathJax](#). Just place the statement within a pair of $$ signs. For example $$ becomes $$.

If you look to the left-hand side of your screen, you will see an expandable menu that may be useful for viewing the table of contents or using the find/replace tool.



Once clicked you will see:



I suggest focusing mostly on the “Cells” section. If you’re new to coding or Python, don’t worry about understanding the “Working with Python” section at this point.

5. Adding and Moving Cells

You can add new cells by using the **+ CODE** and **+ TEXT** buttons that show when you hover between cells. These buttons are also in the toolbar above the notebook where they can be used to add a cell below the currently selected cell.

You can move a cell by selecting it and clicking **Cell Up** or **Cell Down** in the top toolbar.

Adding and moving cells

You can add new cells by using the **+ CODE** and **+ TEXT** buttons that show when you hover between cells. These buttons are also in the toolbar above the notebook where they can be used to add a cell below the currently selected cell.

You can move a cell by selecting it and clicking **Cell Up** or **Cell Down** in the top toolbar.

Consecutive cells can be selected by “lasso selection” by dragging from outside one cell and through the group. Non-adjacent cells can be selected concurrently by clicking one and then holding down Ctrl while clicking another. Similarly, using Shift instead of Ctrl will select all intermediate cells.



6. Settings, Dark Mode, Font Size, Show Line Numbers

Dark mode can be accessed through clicking “Tools” → “Settings”. on the top menu, which will allow you to change the theme, font size, and link a GitHub account.

Settings

Site	Theme light
Editor	<input type="checkbox"/> Show desktop notifications for completed executions <input type="checkbox"/> New notebooks use private outputs (omit outputs when saving)
Colab Pro	Default page layout horizontal
GitHub	
Miscellaneous	Custom snippet notebook URL <input type="checkbox"/> Use a temporary scratch notebook as the default landing page.

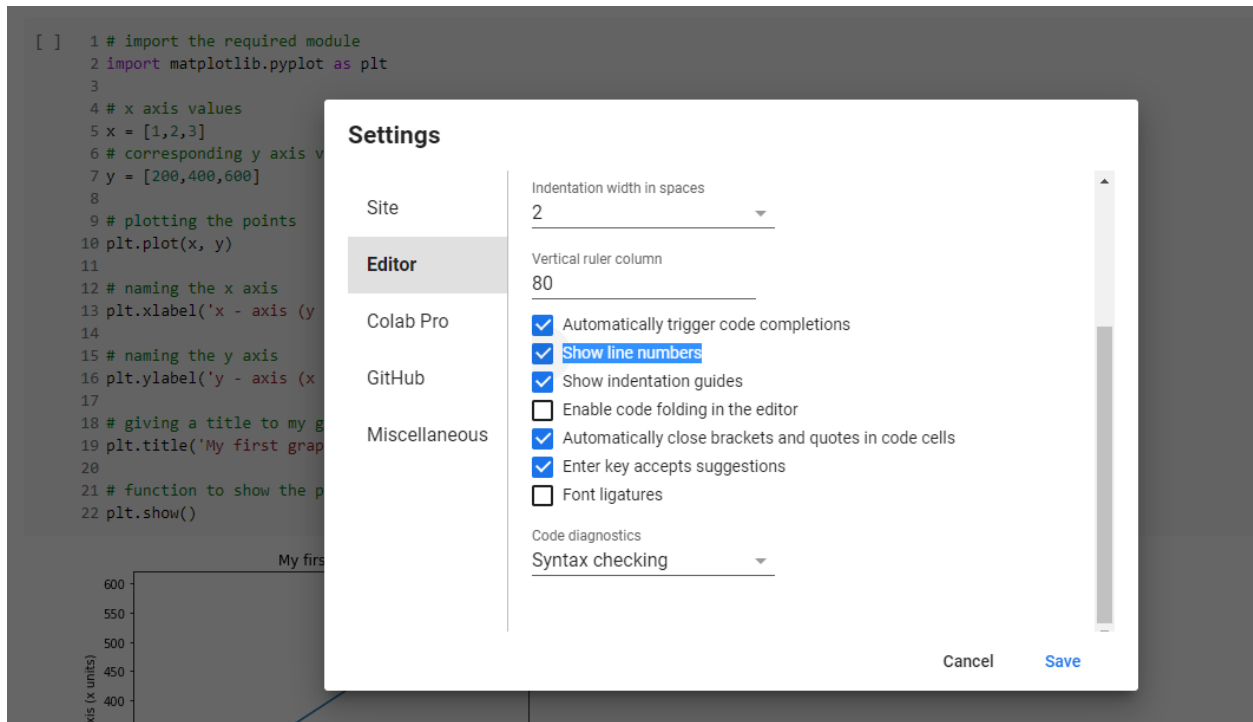
Cancel Save

Settings

Site	Editor key bindings default
Editor	Font size 14 px Font family used when rendering code monospace Indentation width in spaces 2 Vertical ruler column 80 <input checked="" type="checkbox"/> Automatically trigger code completions <input checked="" type="checkbox"/> Show line numbers <input checked="" type="checkbox"/> Show indentation guides <input type="checkbox"/> Enable code folding in the editor
Colab Pro	
GitHub	
Miscellaneous	

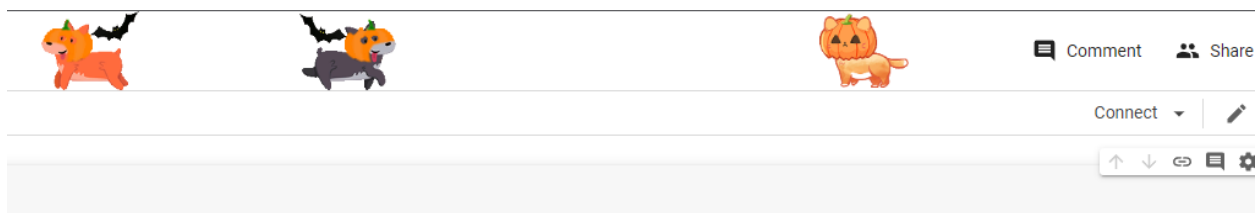
Cancel Save

Another helpful feature is showing the line numbers in the code cell, which can be toggled in “Tools” → “Settings → Editor” by clicking the show line numbers box.



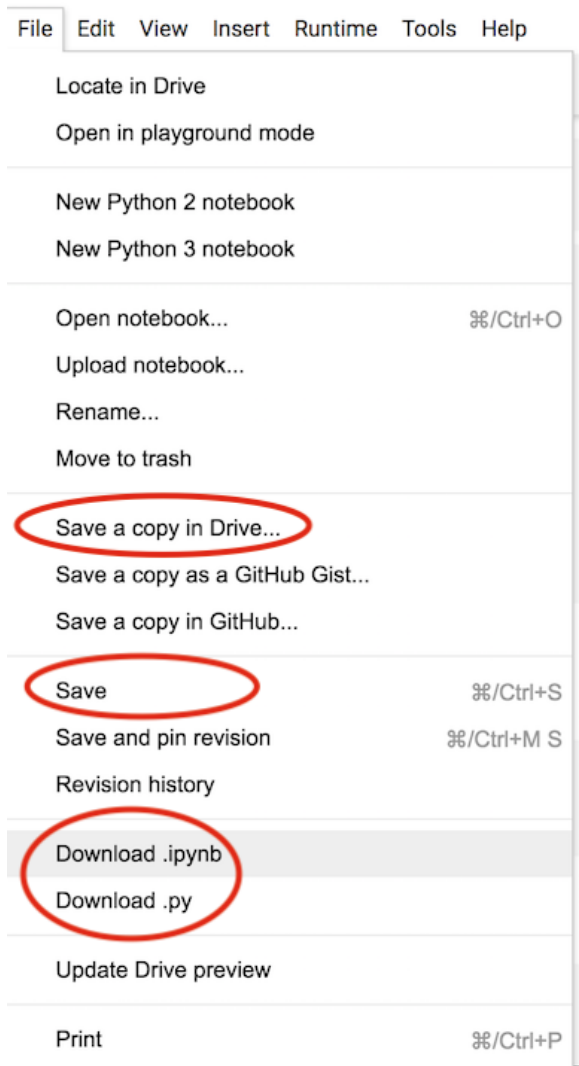
7. Cat and Corgi Mode

Perhaps most importantly, under miscellaneous you can add animated animals in the header through “Tools” → “Settings” → “Miscellaneous”. Here, you can select the mode, and you will see your cats and dogs keeping you company. Note that they are seasonal too!



8. Save or download your work

You can use “command-s” or drop the “File” menu down to save. You can create a copy of your notebook by dropping “File” -> “Save a Copy in Drive.” All Colab notebooks are stored in the open source Jupyter notebook format (.ipynb). You can also download your workbook by going from “File” -> “download .ipynb” :



9. Notebook Keyboard Shortcuts

Shortcuts from Jupyter Notebook don't work directly in Colab, but most work by just adding Ctrl + M before the Jupyter keyboard shortcut.

Action	Jupyter Notebook	Google Colab
Add a cell above	A	Ctrl + M + A
Add a cell below	B	Ctrl + M + B
See all keyboard shortcuts	H	Ctrl + M + H
Change cell to code	Y	Ctrl + M + Y
Change cell to markdown	M	Ctrl + M + M
Interrupt the kernel	II	Ctrl + M + I
Delete a cell	DD	Ctrl + M + D
Checkpoint notebook	Ctrl + S	Ctrl + M + S

Below are some notable exceptions to this rule for which either the shortcut is changed completely or kept the same.

Action	Jupyter Notebook	Google Colab
Restart runtime	00	Ctrl + M + .
Run cell	Ctrl + Enter	Ctrl + Enter
Run cell and add new cell below	Alt + Enter	Alt + Enter
Run cell and goto the next cell below	Shift + Enter	Shift + Enter
Comment current line	Ctrl + /	Ctrl + /

10. Installing and Importing Libraries

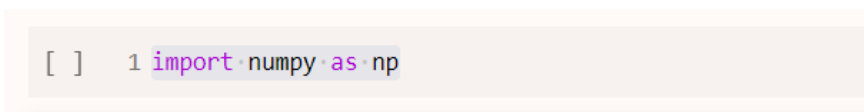
Currently, installations within Google Colaboratory are not persistent; you must reinstall libraries every time you (re-)connect. But since Colab has numerous useful common libraries installed by default, this is less of an issue than it may seem. Installing libraries are easily added in one of a couple different ways. Colab supports both the pip and apt package managers.

We can use the `!` operator to install any package in Colab. What `!` actually does is, *it tells the notebook cell that this line is not a Python code, its a command line script*. So, to run any command line script in Colab, just add a `!` **preceding** the line. The general format is `!pip install package_name`. Let's say you want to install numpy, then you would write the following in a code cell:

A screenshot of a Google Colab code cell. On the left, there is a green checkmark icon and a '3s' timer. Next to it is a play button icon. The code text in the cell is `1 !pip install numpy`.

```
1 !pip install numpy
```

After installation, you just need to import it by using import:

A screenshot of a Google Colab code cell. The code text in the cell is `[] 1 import numpy as np`.

```
[ ] 1 import numpy as np
```

There are many other lesser-known features in Colab, so feel free to do some research on your own and see what other neat things you can find!

FAQs

Where are my notebooks stored, and can I share them?

Colab notebooks are stored in [Google Drive](#), or can be loaded from [GitHub](#). Colab notebooks can be shared just as you would with Google Docs or Sheets. Simply click the Share button at the top right of any Colab notebook, or follow these Google Drive [file sharing instructions](#).

Where are my notebooks stored, and can I share them?

Colab notebooks are stored in [Google Drive](#), or can be loaded from [GitHub](#). Colab notebooks can be shared just as you would with Google Docs or Sheets. Simply click the Share button at the top right of any Colab notebook, or follow these Google Drive [file sharing instructions](#).

If I share my notebook, what will be shared?

If you choose to share a notebook, the full contents of your notebook (text, code, output, and comments) will be shared. You can omit code cell output from being saved or shared by using **Edit > Notebook settings > Omit code cell output when saving this notebook**. The virtual machine that you are using, including any custom files and libraries that you have setup, will not be shared. So it's a good idea to include cells which install and load any custom [libraries](#) or [files](#) that your notebook needs.

Can I import an existing Jupyter/IPython notebook into Colab?

Yes. Choose "Upload notebook" from the File menu.

How can I search Colab notebooks?

You can search Colab notebooks using [Google Drive](#). Clicking on the Colab logo at the top left of the notebook view will show all notebooks in Drive. You can also search for notebooks that you have opened recently using **File > Open notebook**.

Where is my code executed? What happens to my execution state if I close the browser window?

Code is executed in a virtual machine private to your account. Virtual machines are deleted when idle for a while, and have a maximum lifetime enforced by the Colab service.

Does Colab support Python 2?

Python 2 is no longer supported in Colab. For information on migrating your code from Python 2 to Python 3, see [Porting Python 2 Code to Python 3](#).

Useful references

- [Colab FAQ](#)
- [Google Tips for Power Users](#)
- [Visualization with Matplotlib](#)

-

Markdown

- [Github Markdown basics](#)
- [Github flavored Markdown](#)
- [Original Markdown spec: Syntax](#)
- [Original Markdown spec: Basics](#)
- [marked.js library used by Colab](#)
- [LaTeX mathematics for equations](#)