Google

Colaboratory

# Frequently Asked Questions

## The Basics

**What is Colaboratory?**

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

**Is it really free to use?**

Yes. Colab is free to use.

**Seems too good to be true. What are the limitations?**

Colab resources are not guaranteed and not unlimited, and the usage limits sometimes fluctuate. This is necessary for Colab to be able to provide resources for free. For more details, see Resource Limits

Users who are interested in more reliable access to better resources may be interested in Colab Pro.

**What is the difference between Jupyter and Colab?**

Jupyter is the open source project on which Colab is based. Colab allows you to use and share Jupyter notebooks with others without having to download, install, or run anything.

## Using Colab

**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 you’re using, including any custom

files and libraries that you've 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.

**How can I get my data out?**

You can download any Colab notebook that you've created from Google Drive following these instructions, or from within Colab's File menu. All Colab notebooks are stored in the open source Jupyter notebook format ( .ipynb).

**How can I reset the virtual machine(s) my code runs on, and why is this sometimes unavailable?**

Selecting **Runtime > Factory reset runtime** to return all managed virtual machines assigned to you to their original state. This can be helpful in cases where a virtual machine has become unhealthy e.g. due to accidental overwrite of system files, or installation of incompatible software. Colab limits how often this can be done to prevent undue resource consumption. If an attempt fails, please try again later.

**Why does `drive.mount()` sometimes fail saying "timed out", and why do I/O operations in `drive.mount()`-mounted folders sometimes fail?**

Google Drive operations can time out when the number of files or subfolders in a folder grows too large. If thousands of items are directly contained in the top-level "My Drive" folder then mounting the drive will likely time out. Repeated attempts may eventually succeed as failed attempts cache partial state locally before timing out. If you encounter this problem, try moving files and folders directly contained in "My Drive" into sub-folders. A similar problem can occur when reading from other folders after a successful `drive.mount()`. Accessing items in any folder containing many items can cause errors like `OSError: [Errno 5] Input/output error`. Again, you can fix this problem by moving directly contained items into sub-folders.
Note that "deleting" files or subfolders by moving them to the Trash may not be enough; if that doesn't seem to help, make sure to also Empty your Trash.

**Why do Drive operations sometimes fail due to quota?**

Google Drive enforces various limits, including per-user and per-file operation count and bandwidth quotas. Exceeding these limits will trigger `Input/output error` as above, and show a notification in the Colab UI. A typical cause is accessing a popular shared file, or accessing too many distinct files too quickly. Workarounds include:

- Copy the file using drive.google.com and don't share it widely so that other users don't use up its limits.
- Avoid making many small I/O reads, instead opting to copy data from Drive to the Colab VM in an archive format (e.g. `.zip` or `.tar.gz` files) and unarchive the data locally on the VM instead of in the mounted Drive directory.
- Wait a day for quota limits to reset.

**Why do Drive operations sometimes fail due to storage quota?**

Google Drive imposes a limit on how much data can be stored in it by each user. If Drive operations are failing with `Input/output error` and a notification says storage quota has been exceeded, delete some files using drive.google.com and Empty your Trash to reclaim the space. It might take a little while for the reclaimed space to be available in Colab.

If you'd like to purchase more Drive space, visit Google Drive. Note that purchasing more space on Drive will not increase the amount of disk available on Colab VMs. Subscribing to Colab Pro will.

# Resource Limits

**Why aren’t resources guaranteed in Colab?**

In order to be able to offer computational resources for free, Colab needs to maintain the flexibility to adjust usage limits and hardware availability on the fly. Resources available in Colab vary over time to accommodate fluctuations in demand, as well as to accommodate overall growth and other factors.

Some users want to be able to do more in Colab than the resource limits allow. We have heard from many users who want faster GPUs, longer running notebooks and more memory, as well as usage limits that are higher and don’t fluctuate as much. Introducing Colab Pro is the first step we are taking towards serving users who want to do more in Colab. Our long term goal is to continue providing a free version of Colab, while also growing in a sustainable fashion to meet the needs of our users. If you are interested in doing more in Colab than the resource limits of the free version of Colab allow, please try out Colab Pro and let us know what you think.

**What are the usage limits of Colab?**

Colab is able to provide free resources in part by having dynamic usage limits that sometimes fluctuate, and by not providing guaranteed or unlimited resources. This means that overall usage limits as well as idle timeout periods, maximum VM lifetime, GPU types available, and other factors vary over time. Colab does not publish these limits, in part because they can (and sometimes do) vary quickly.

GPUs and TPUs are sometimes prioritized for users who use Colab interactively rather than for long-running computations, or for users who have recently used less resources in Colab. As a result, users who use Colab for long-running computations, or users who have recently used more resources in Colab, are more likely to run into usage limits and have their access to GPUs and TPUs temporarily restricted. Users with high computational needs may be interested in using Colab’s UI with a local runtime running on their own hardware. Users interested in having higher and more stable usage limits may be interested in Colab Pro.

**What types of GPUs are available in Colab?**

The types of GPUs that are available in Colab vary over time. This is necessary for Colab to be able to provide access to these resources for free. The GPUs available in Colab often include Nvidia K80s, T4s, P4s and P100s. There is no way to choose what type of GPU you can connect to in Colab at any given time. Users who are interested in more reliable access to Colab's fastest GPUs may be interested in Colab Pro.

Note that using Colab for cryptocurrency mining is disallowed entirely, and may result in your account being restricted for use with Colab altogether.

**How long can notebooks run in Colab?**

Notebooks run by connecting to virtual machines that have maximum lifetimes that can be as much as 12 hours. Notebooks will also disconnect from VMs when left idle for too long. Maximum VM lifetime and idle timeout behavior may vary over time, or based on your usage. This is necessary for Colab to be able to offer computational resources for free. Users interested in longer VM lifetimes and more lenient idle timeout behaviors that don't vary as much over time may be interested in Colab Pro.

**How much memory is available in Colab?**

The amount of memory available in Colab virtual machines varies over time (but is stable for the lifetime of the VM). (Adjusting memory over time allows us to continue to offer Colab for free.) You may sometimes be automatically assigned a VM with extra memory when Colab detects that you are likely to need it. Users interested in having more memory available to them in Colab, and more reliably, may be interested in Colab Pro.

**How can I get the most out of Colab?**

Resources in Colab are prioritized for users who have recently used less resources, in order to prevent the monopolization of limited resources by a small number of users. To get the most out of Colab, consider closing your Colab tabs when you are done with your work, and avoid opting for a GPU when it is not needed for your work. This will make it less likely that you will run into usage limits in Colab. Users interested in going beyond the resource limits in the free version of Colab may be interested in Colab Pro.

**I saw a message saying my GPU is not being utilized. What should I do?**

Colab offers optional accelerated compute environments, including GPU and TPU. Executing code in a GPU or TPU runtime does not automatically mean that the GPU or TPU is being utilized. To avoid hitting your GPU usage limits, we recommend switching to a standard runtime if you are not utilizing the GPU. Choose **Runtime > Change Runtime Type** and set *Hardware Accelerator* to *None*.

For examples of how to utilize GPU and TPU runtimes in Colab, see the Tensorflow With GPU and TPUs In Colab example notebooks.

# Additional Questions

**What browsers are supported?**

Colab works with most major browsers, and is most thoroughly tested with the latest versions of Chrome, Firefox and Safari.

**How is this related to colaboratory.jupyter.org?**

In 2014 we worked with the Jupyter development team to release an early version of the tool. Since then Colab has continued to evolve, guided by internal usage.

**What about other programming languages?**

Colab focuses on supporting Python and its ecosystem of third-party tools. We're aware that users are interested in support for other Jupyter kernels (eg R or Scala). We would like to support these, but don't yet have any ETA.

**I found a bug or have a question, who do I contact?**

Open any Colab notebook. Then go to the Help menu and select ”Send feedback...”.

**Why prompt to enable third-party cookies?**

Colab uses HTML iframes and service workers hosted on separate origins in order to display rich outputs securely. Browsers require enabling third-party cookies to use the service workers within iframes. An alternative to enabling third-party cookies for all sites is to allow the following hostname in your browser settings: googleusercontent.com.

**How do I change the editor font?**

Colab uses a generic monospace font for the editor. You can configure what font family is used for monospace in most modern browsers. Here's a few common ones:

- In Firefox, follow the steps provided in the Firefox support documents to configure the "Monospace" font.
- In Chrome, navigate to "chrome://settings/fonts" and modify the section labeled "Fixed-width font".

**Does Colab support Python 2?**

The Python development team has declared that Python 2 will no longer be supported after January 1st, 2020. Colab has stopped updating Python 2 runtimes, and is gradually phasing out support for Python 2 notebooks. We suggest migrating important notebooks to Python 3.

To change your Python 2 notebook's runtime to Python 3, choose **Runtime > Change Runtime Type** and select Python 3. Changing the runtime from Python 3 to Python 2 is not supported. For information on migrating your code from Python 2 to Python 3, see Porting Python 2 Code to Python 3.

**Where can I learn more about Colab Pro?**

There is an FAQ for Colab Pro on the Colab Pro sign-up page.

**How does billing work for Colab Pro?**

Information for Colab Pro, including pricing, can be found at the Colab Pro sign-up page.