

Google Colab Instructions

In this instruction, we get familiar with how to create and use Python notebooks in Google colab.

What is Colab?

Google colab is an environment that allows developers to have access to an interactive IDE. There are some advantages of this colab. For example,

You have access to both Python 2 and 3.

You have access to CPU, GPU, and TPU

You can write Linux commands in IDE environment

Most of the required libraries are pre-installed

You have access to cloud for storing and retrieving your data



Figure 1- Google colab and connection to other tools

You can either start coding in a notebook or upload from github ipython notebook. Let's start from writing a code from scratch. Here are the steps:

1- Go to <https://colab.research.google.com>. This is the page that you will see:

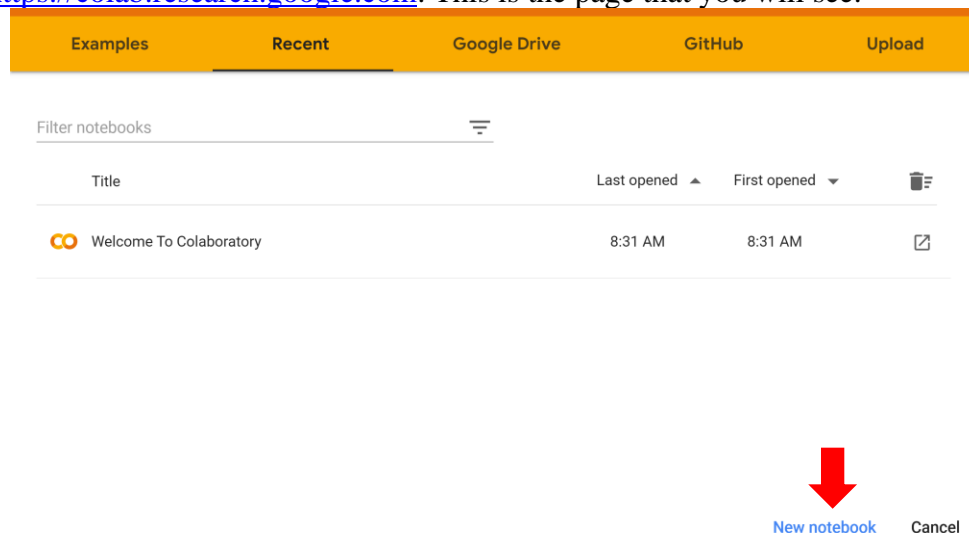


Figure 2- Main page of colab

2- You may go to “GOOGLE DRIVE” tab to store your code on google drive. Click on arrow next to “NEW PYTHON NOTEBOOK 3” to choose the version of the language that you want to use. This is the environment that you will see which is very similar to Python notebook:

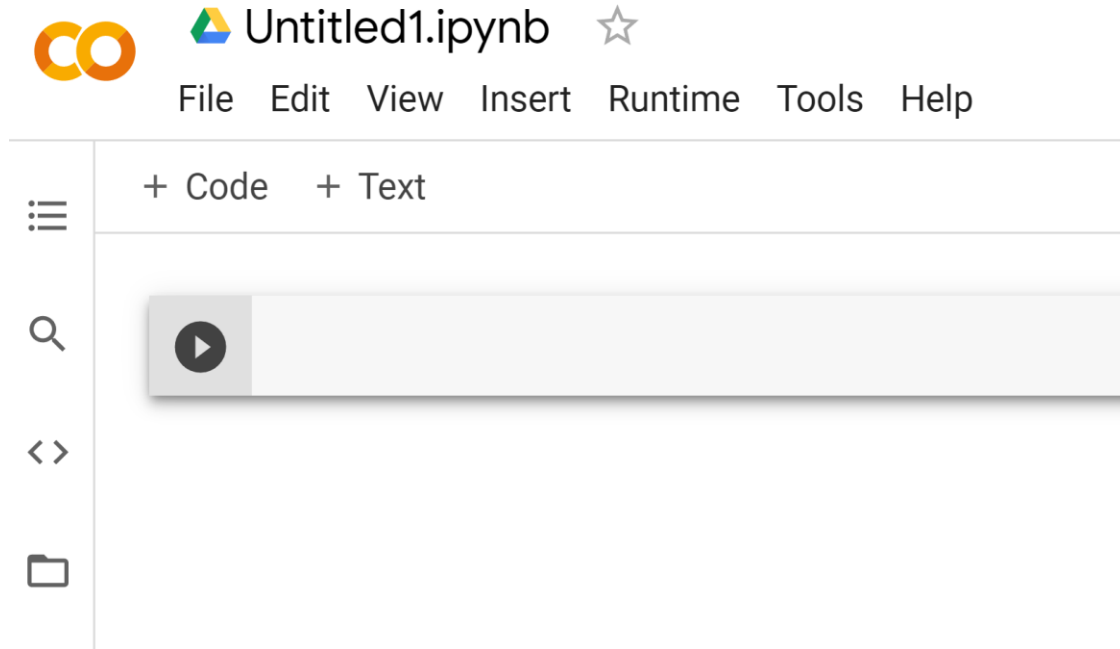


Figure 3- Colab coding environment

Before start actual coding, click on the “Runtime\Change runtime type” to choose between available sources.

Notebook settings

Hardware accelerator

None



None

GPU

TPU

Output when saving this notebook

Cancel

Save

Figure 4- Available resources

Here again you can choose between available types of python. As you see here, you can choose to run your code on GPU rather than CPU to speed up your computations. You can also click on the name on top of the page and changed it to your desired name.

If you like to see the power of GPU resources, you may type following commands in the cell:

```

from tensorflow.python.client import device_lib

print("Show System RAM Memory:\n\n")
!cat /proc/meminfo | egrep "MemTotal*"

print("\n\nShow Devices:\n\n"+str(device_lib.list_local_devices()))

```

To run the command, click on the arrow key on the left hand side of the cell. This will show you a page like this:

Show System RAM Memory:

```
MemTotal:          13302920 kB
```

Show Devices:

```

[name: "/device:CPU:0"
 device_type: "CPU"
 memory_limit: 268435456
 locality {
 }
 incarnation: 10900234812275313976
 xla_global_id: -1
 , name: "/device:GPU:0"
 device_type: "GPU"
 memory_limit: 14465892352
 locality {
   bus_id: 1
   links {
   }
 }
 incarnation: 1781117250928914076
 physical_device_desc: "device: 0, name: Tesla T4, pci bus id: 0000:00:04.0, compute capability: 7.5"
 xla_global_id: 416903419
]

```

It looks we have a Tesla T4 GPU. With 16GB of RAM which is good for our experiments. We need Tensorflow ver. 2.7 for this lab so first check the version of Tensorflow using:

```

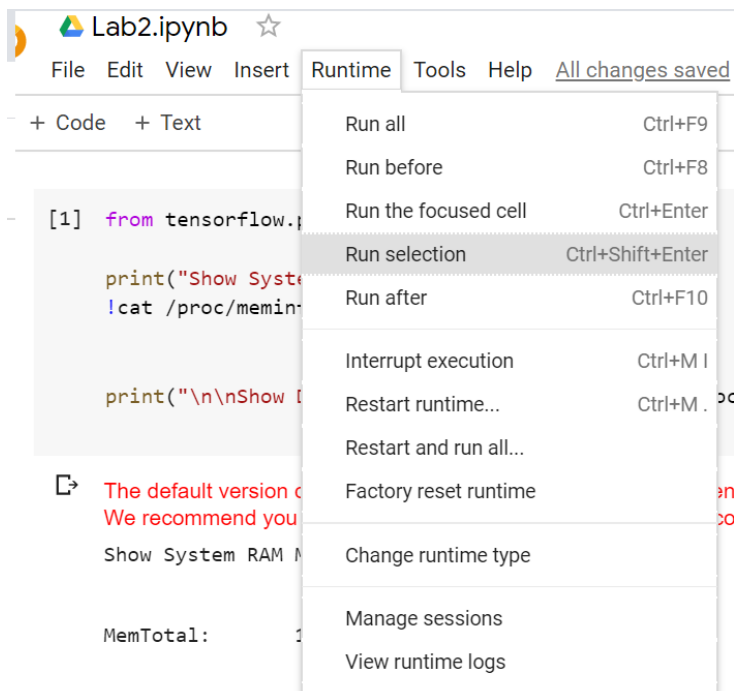
import tensorflow as tf
tf.__version__

```

If you see any other versions, you can upgrade it using:

```
!pip install q tensorflow-gpu==2.7.0
```

And restart the code from here:



We can share a google drive into Colab environment using:

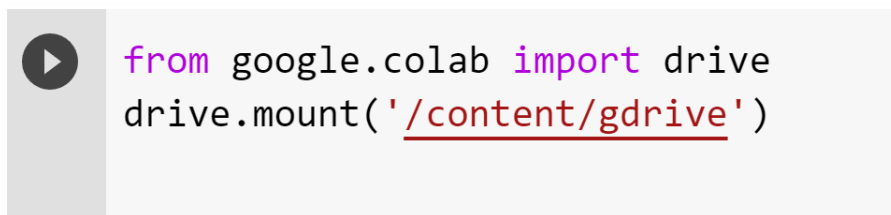
```
from google.colab import drive
drive.mount('/content/gdrive')
```

This will show you a permit notification, click on “connect to Google Drive” and login to your google account:

Permit this notebook to access your Google Drive files?

This notebook is requesting access to your Google Drive files. Granting access to Google Drive will permit code executed in the notebook to modify files in your Google Drive. Make sure to review notebook code prior to allowing this access.

No thanks [Connect to Google Drive](#)



Mounted at /content/gdrive

Now we can load codes our datasets from the google drive.

There are other resources like Amazon (AWS), Microsoft (Azure) or Floydhub but unfortunately, they are not free.

For final submission, you may use following code to convert the ipynb to pdf using:

```
%shell jupyter nbconvert --to pdf 'filename.ipynb'
```

Or for html submission:

```
%shell jupyter nbconvert --to html 'filename.ipynb'
```