

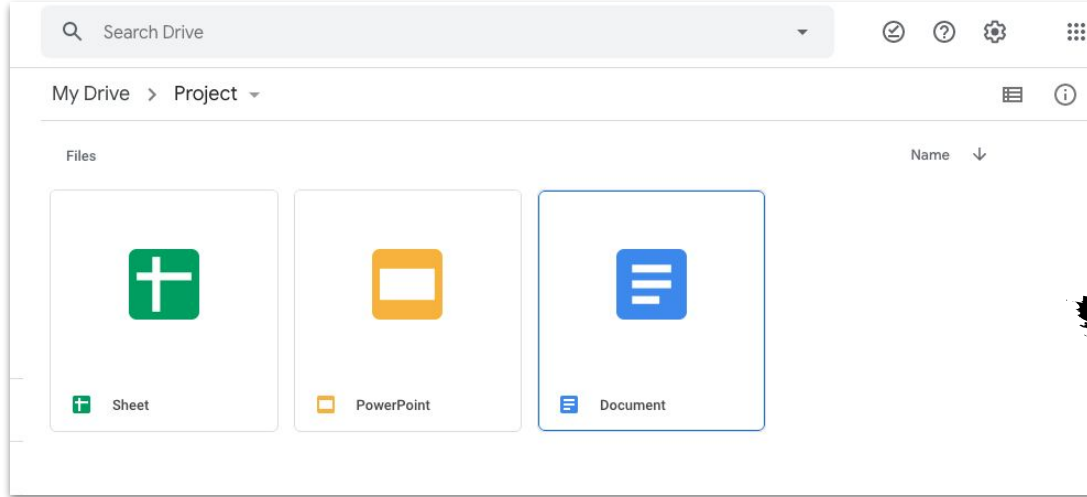
# Colab

or

**Now You Can Share Your Notebook Files on Google Drive**



# Almost Nothing New...

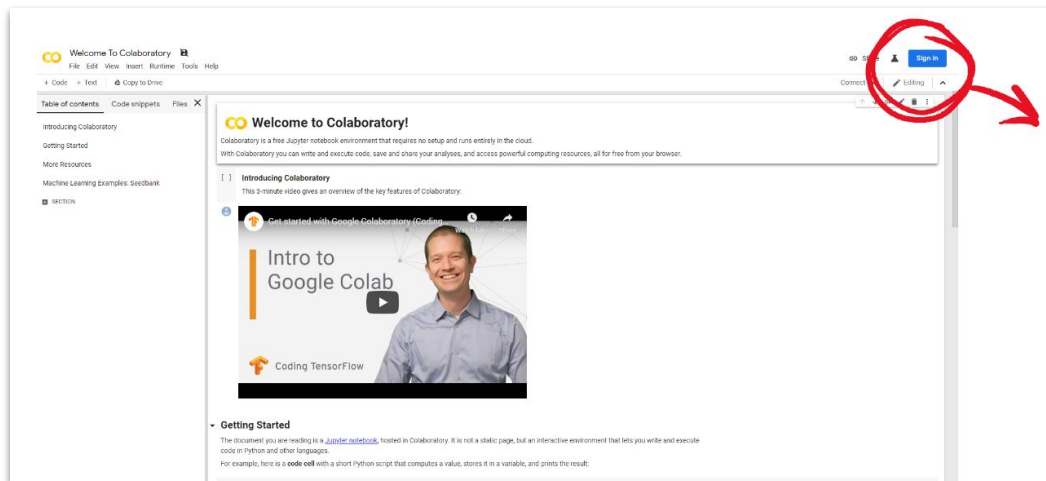


Now you can share your Jupyter Notebook files on Google Drive, just like the way you share documents, sheets, and powerpoints.

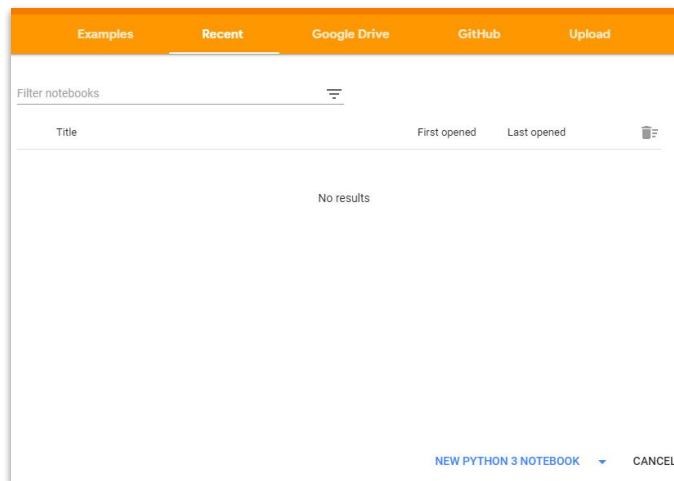
# Where to Start?

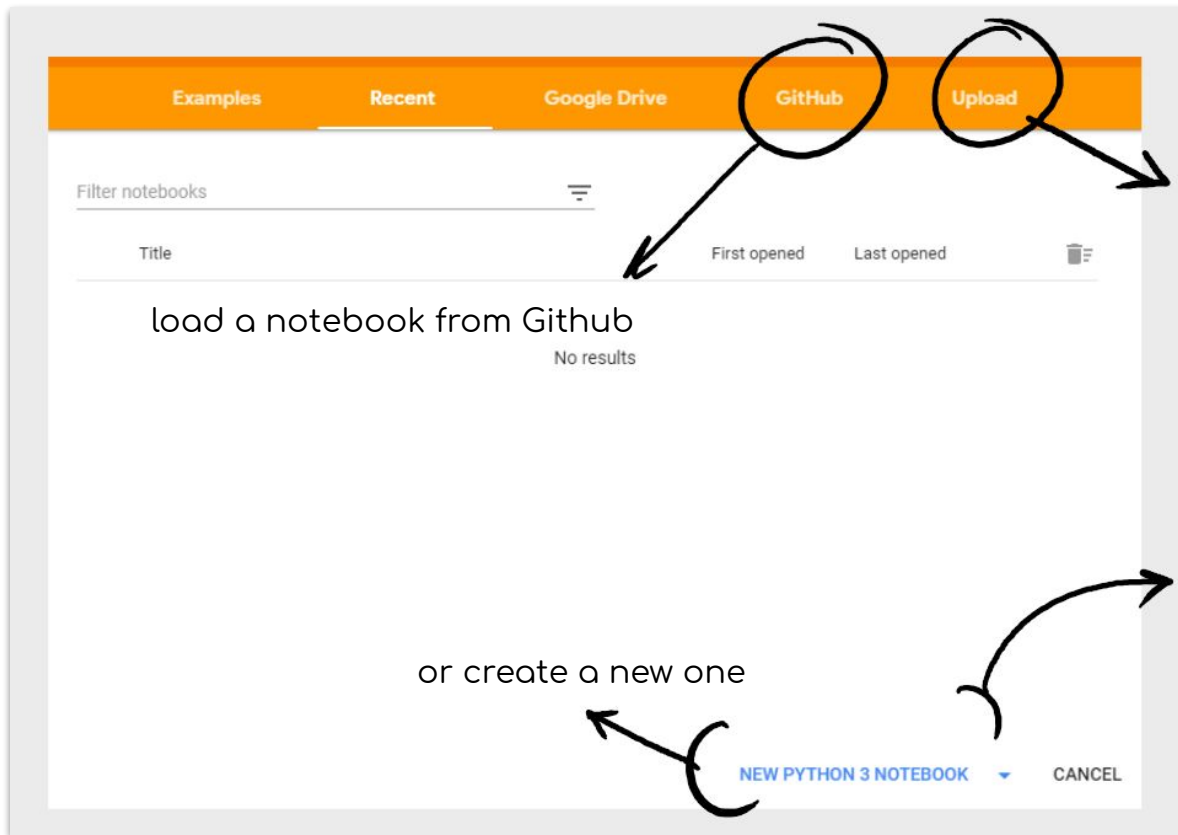
The first time, you need to visit this link <https://colab.research.google.com/>

In case you are not signed in your Google account:

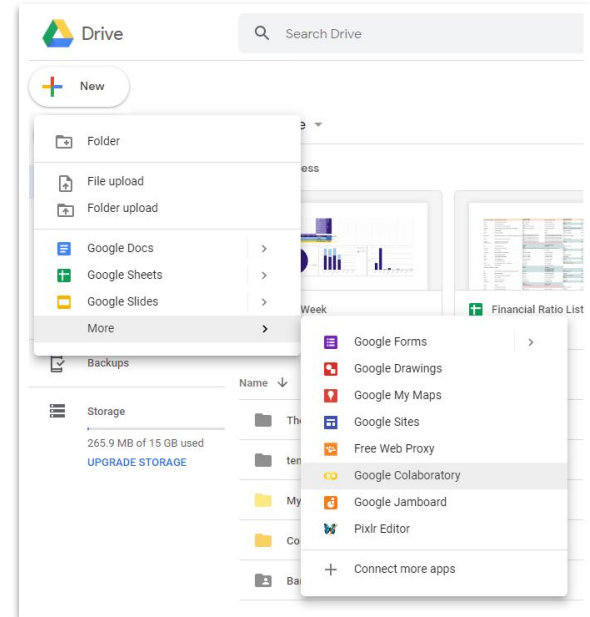


and after signing in:

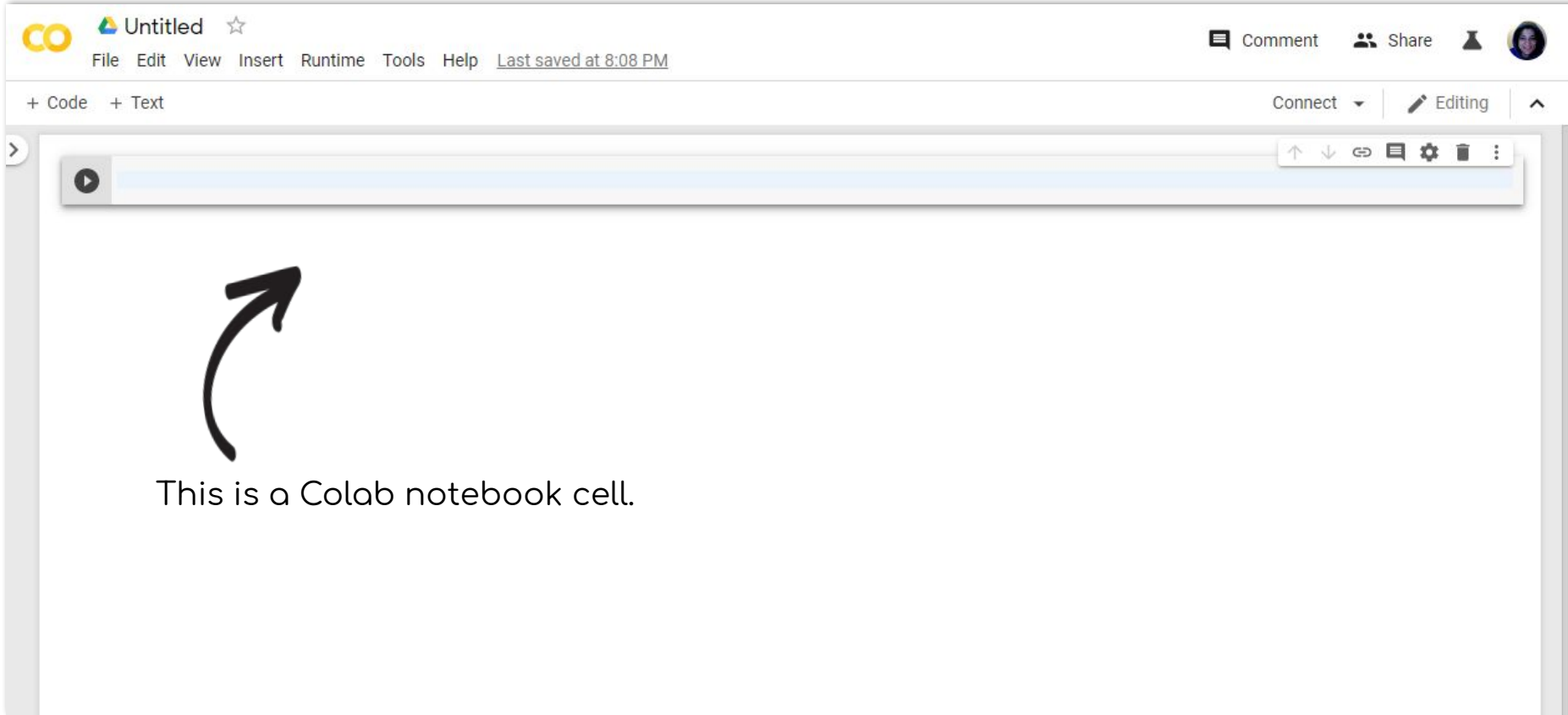




upload a local notebook



This is what your newly created notebook file looks like





+ Code

Connect ▾

Editing



Locate in Drive

Open in playground mode

New Python 3 notebook

New Python 2 notebook

Open notebook... ⌘/Ctrl+O

Upload notebook...

Rename...

Move to trash

Save a copy in Drive...

Save a copy as a GitHub Gist...

Save a copy in GitHub...

Save ⌘/Ctrl+S

Save and pin revision ⌘/Ctrl+M S

Revision history

Download .ipynb

Download .py

Update Drive preview

Print ⌘/Ctrl+P

And when you're done, you can save the file on your Drive,  
upload it as a snippet or  
a Notebook file on Github,

or save it locally as either a Notebook,  
or a Python file.

# So, Why Colab?

- It's a full option!
- You can share
- Cloud computing for free
- It's the fun version of Jupyter Notebook

# So, Why Colab?

- It's a full option!
- You can share
- Cloud computing for free
- It's the fun version of Jupyter Notebook

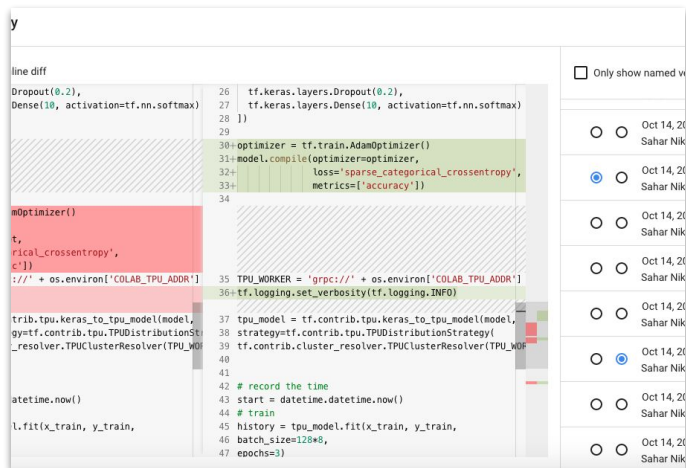


You don't need to install ANYTHING (at least for this course); nor you need to deal with different environments on your system.



# So, Why Colab?

- It's a full option!
- You can share
- Cloud computing for free
- It's the fun version of Jupyter Notebook



```
line diff
Dropout(0.2),
Dense(10, activation=tf.nn.softmax)
26 tf.keras.layers.Dropout(0.2),
27 tf.keras.layers.Dense(10, activation=tf.nn.softmax)
28 })
29
30 optimizer = tf.train.AdamOptimizer()
31 model.compile(optimizer=optimizer,
32               loss='sparse_categorical_crossentropy',
33               metrics=['accuracy'])
34
35 tpu_worker = 'grpc://' + os.environ['COLAB_TPU_ADDR']
36 tf.logging.set_verbosity(tf.logging.INFO)
37 tpu_model = tf.contrib.tpu.keras_to_tpu_model(model,
38 strategy=tf.contrib.tpu.TPUDistributionStrategy(
39 tf.contrib.cluster_resolver.TPUCClusterResolver(TPU_WORKER,
40
41 # record the time
42 start = datetime.datetime.now()
43 # train
44 history = tpu_model.fit(x_train, y_train,
45 batch_size=128*8,
46 epochs=3)
```

<input type="radio"/>	<input type="radio"/>	Oct 14, 20	Sahar Nik
<input checked="" type="radio"/>	<input type="radio"/>	Oct 14, 20	Sahar Nik
<input type="radio"/>	<input type="radio"/>	Oct 14, 20	Sahar Nik
<input type="radio"/>	<input type="radio"/>	Oct 14, 20	Sahar Nik
<input type="radio"/>	<input type="radio"/>	Oct 14, 20	Sahar Nik
<input type="radio"/>	<input checked="" type="radio"/>	Oct 14, 20	Sahar Nik
<input type="radio"/>	<input type="radio"/>	Oct 14, 20	Sahar Nik
<input type="radio"/>	<input type="radio"/>	Oct 14, 20	Sahar Nik

You can collaborate in Notebook framework, while having access to an amazing version control interface, unlike Github's mess with Notebook files.

You can add Colab Notebooks directly to Git repos.

# So, Why Colab?

- It's a full option!
- You can share
- Cloud computing for free
- It's the fun version of Jupyter Notebook



A screenshot of the Google Colab environment information page. The page is divided into three sections: Software, Hardware, and Environment. The Hardware section is highlighted with a yellow background and a red circle. A red speech bubble points to the Hardware section with the text "12GB RAM". The Hardware section lists the following information:

```
=== Hardware ===  
nvidia gpus : 1  
torch devices : 1  
- gpu0 : 11441MB | Tesla K80
```

The Environment section lists the following information:

```
=== Environment ===  
platform : Linux-4.15.0-101-gcp-with-Ubuntu-18.04-bionic  
distro : #1 SMP Thu Aug 8 02:47:02 PDT 2019  
conda env : Unknown  
python : /usr/bin/python3  
sys.path :  
/env/python  
/usr/lib/python3.6.zip  
/usr/lib/python3.6  
/usr/lib/python3.6/lib-dynload  
/usr/local/lib/python3.6/dist-packages  
/usr/lib/python3/dist-packages
```

You have free access to 12GB of RAM and can work with the NVIDIA Tesla K80 GPU.

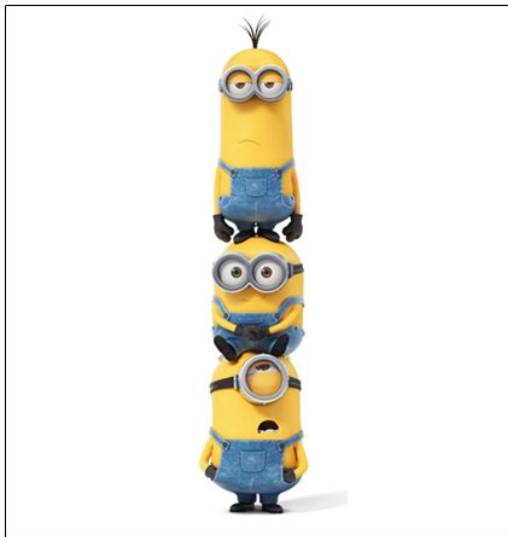
# CPU, GPU, and TPU



**CPU**

One guy, capable of doing any task, but, obviously, only one at a time

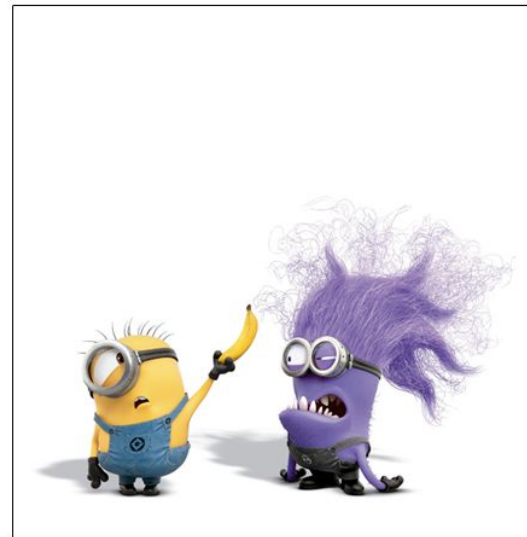
Working unit: Scaler



**GPU**

Bunch of guys, doing a few specialized tasks; can do the same task with many different values, simultaneously

Working unit: Vector



**TPU**

Special unit, called MXU, designed for matrix multiplication

Working unit: Matrix

The screenshot shows the Google Colab interface. The 'Runtime' menu is open, displaying options such as 'Run all', 'Run before', 'Run the focused cell', 'Run selection', 'Run after', 'Interrupt execution', 'Restart runtime...', 'Restart and run all...', 'Reset all runtimes...', 'Change runtime type', 'Manage sessions', and 'View runtime logs'. The 'Change runtime type' option is highlighted. An orange arrow points from this option to the 'Notebook settings' dialog box. Another orange arrow points from the 'Connect' button in the top right to a dropdown menu showing 'Connect to hosted runtime' and 'Connect to local runtime...'. A third orange arrow points from the 'Connect to local runtime...' option to the 'Notebook settings' dialog box. The code editor on the left contains the following code:

```
[ ] import numpy as np
import tensorflow

[ ] mat1 = np.random.r
mat2 = np.random.r

mat3 = tf.matmul(m

mat3

<tf.Tensor 'MatMul

[ ] from fastai utils show install import *
```

You can simply switch between CPU, GPU, and TPU.

The 'Notebook settings' dialog box is shown. It has a title bar 'Notebook settings'. Below the title, there are three settings:

- Runtime type:** A dropdown menu currently set to 'Python 3'.
- Hardware accelerator:** A dropdown menu currently set to 'None'. A question mark icon is next to it.
- Omit code cell output:** A checkbox that is currently unchecked.

Below these settings, there is a button labeled 'Saving this notebook'. At the bottom right of the dialog, there are two buttons: 'CANCEL' and 'SAVE'.

The Colab status bar is shown. It includes a 'RAM' indicator with a green checkmark and a 'Disk' indicator with a grey bar. To the right of these indicators is a dropdown menu and an 'Edit' button. Below these indicators, there is a dark grey box with white text that reads: 'Connected to "Python 3 Google Compute Engine backend" RAM: 0.78 GB/12.72 GB Disk: 25.10 GB/48.97 GB'.

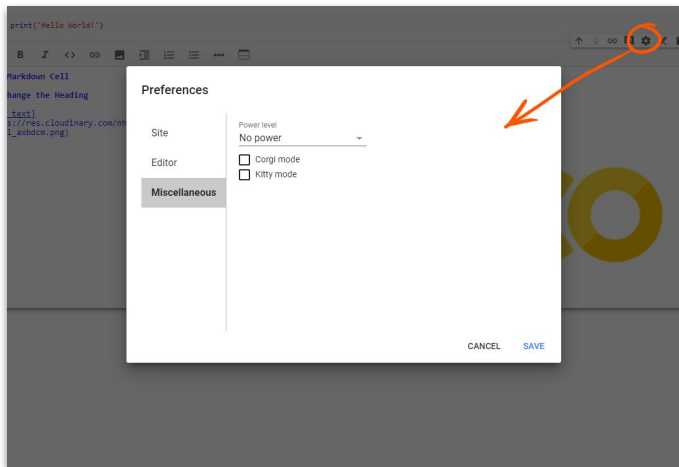
# Drawbacks

- Don't leave the Colab open, for lunch if your lunchtime is more than 90 minutes
- Even if you skip lunch and also dinner, *runtime* will reset after 12 hours
- Employing maximum TPU power needs you to change your model
- Heavy computations? Google will restrict your access temporarily

Conclusion: Not good for research, but wonderful for school  
Use it wisely, and it's probably enough for your final project

# So, Why Colab?

- It's a full option!
- You can share
- Cloud computing for free
- It's the fun version of Jupyter Notebook



- Link to cells
- Comments
- User-friendly markdowns with realtime preview
- Automate search offer on Stack Overflow
- Automate offer for code samples
- ...

And whenever you're bored, stuck, and tired, don't forget to play with **Miscellaneous** options in Preference

# How to Submit Homework Using Colab?

- For each interested Group, a folder will be created and shared with the members on Google Drive
  - Every week, the homework Colab notebook will be uploaded in those folders
  - You work on those files and that's about it!
  - After the deadline, your share mode will be changed to view only
  - After the grading is done, you'll be back to editor mode
  - Local notebook files and Colab notebooks are easily and neatly convertible, so even if one member of a group is interested to work with Colab, it is still possible for the group to submit the homework using Colab
- 
- If interested, send an email including, at least, one Google account username plus the group number to Luke

# TensorFlow 2.0.0

```
import tensorflow as tf  
print(tf.__version__)
```

The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.  
We recommend you [upgrade](#) now or ensure your notebook will continue to use TensorFlow 1.x via the %tensorflow\_version 1.x magic: [more info](#).

1.15.0

You should run this line magic before importing tensorflow.

So if you already ran the cell for importing TF, you need to reset the runtime, then run the line magic and import TF afterward.

```
[1] %tensorflow_version 2.x
```

TensorFlow 2.x selected.

```
import tensorflow as tf  
print(tf.__version__)
```

2.0.0



# Thanks and Questions