

---

# ML and DL - Warm Up

Python Installation

&

Deep Learning Environment

---

# Python Installation

- Anaconda
  - <https://www.anaconda.com/distribution/>
  - Test Anaconda Environment
    - sumeirude-MacBook-Pro:~ sumeiru\$ which python  
/anaconda3/bin/python  
sumeirude-MacBook-Pro:~ sumeiru\$ which pip  
/anaconda3/bin/pip  
sumeirude-MacBook-Pro:~ sumeiru\$ which conda  
/anaconda3/bin/conda
    - sumeirude-MacBook-Pro:~ sumeiru\$ python  
Python 3.6.5 |Anaconda, Inc.| (default, Apr 26 2018, 08:42:37)  
[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE\_401/final)] on darwin  
Type "help", "copyright", "credits" or "license" for more information.  
>>>

- Miniconda
  - <https://docs.conda.io/en/latest/miniconda.html>

# Command on python

- Go to Terminal (Mac/Linux) or Anaconda Prompt (Windows)
- Install Package
  - pip install {package name}
  - conda install {package name}
- Check how many packages you have
  - pip freeze

# Virtual Environment

- Develop projects
  - Manage various needs effectively
  - Avoid packages version collision

Anaconda Navigator

Sign in to Anaconda Cloud

Home

Environments (highlighted with a red box)

Learning

Community

Documentation

Developer Blog

Search Environments

Installed  Channels  Update index...  Search Packages

base (root)

hw4

tensorflow

Create new environment

Name:

Location:

Packages:

Python

R

Name	Description	Version
_jupyter_nb_ex...	A configuration metapackage for enabling anaconda-bundled jupyter extensions	0.1.0
alabaster	Configurable, python 2+3 compatible sphinx theme.	0.7.10
anaconda	Simplifies package management and deployment of anaconda	5.2.0
ipython	Producing data science projects	0.1.0
ipython_genutils	Utilities for ipython	0.8.2
jupyter	Interactive computing and pythonic api	0.1.0
notebook	IPython notebook	0.24.0
nbformat	IPython notebook	1.0.1
nbconvert	Convert notebooks to various formats	1.6.3
astropy	Community-developed python library for astronomy	3.0.2
attrs	Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	18.1.0
babel	Utilities to internationalize and localize python applications	2.5.3
backcall	Specifications for callback functions passed in to an api	0.1.0
backports		1.0
backports.shutil_g...	A backport of the get_terminal_size function from python 3.3's shutil.	1.0.0
beautifulsoup4	Python library designed for screen-scraping	4.6.0
lxml	DOM, SAX, and Iterator API for XML and HTML	0.0.1

266 packages available

Anaconda Navigator

## Choose your virtual environment

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Documentation

Developer Blog

Twitter YouTube GitHub

**Applications on hw4**

base (root)

hw4

tensorflow

/anaconda3/envs/hw4

VS Code

1.26.1

Streamlined code editor with support for development operations like debugging, task running and version control.

Launch

Gluviz

0.13.3

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Install

**IDE**

lab

JupyterLab

0.35.5

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Install

jupyter

Notebook

5.7.8

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Install

Orange 3

3.19.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Install

IPyConsole

4.3.1

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Install

RStudio

1.1.456

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Install

Spyder

3.3.5

Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Install

**If you have your own GPU**

# Download Cuda

<https://developer.nvidia.com/cuda-downloads>

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

Operating System      Windows      Linux      Mac OSX

Architecture      x86\_64

Version      10      8.1      7      Server 2019      Server 2016      Server 2012 R2

Installer Type      exe (network)      exe (local)

Download Installer for Windows 10 x86\_64

The base installer is available for download below.

Base Installer

Installation Instructions:

1. Double click cuda\_10.1.243\_win10\_network.exe
2. Follow on-screen prompts

Download (19.9 MB)

The checksums for the installer and patches can be found in [Installer Checksums](#).  
For further information, see the [Installation Guide for Microsoft Windows](#) and the [CUDA Quick Start Guide](#).

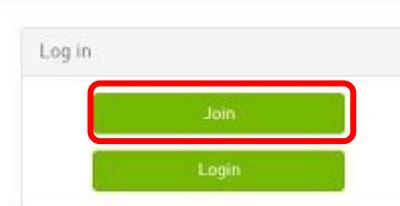
# Download Cudnn

- <https://developer.nvidia.com/cudnn>
- Register an account

## Membership Required

The downloadable file or page you have requested, requires membership of the NVIDIA Developer Program. Please login to gain access or use the button below and complete the short application for this free to join program. Thank you.

[Join now](#)



- Go to E-mail to activate your account

## cuDNN Download

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

I Agree To the Terms of the [cuDNN Software License Agreement](#)

Note: Please refer to the [Installation Guide](#) for release prerequisites, including supported GPU architectures and compute capabilities, before downloading.

For more information, refer to the cuDNN Developer Guide, Installation Guide and Release Notes on the [Deep Learning SDK Documentation](#) web page.

[Download cuDNN v7.6.5 \(November 5th, 2019\), for CUDA 10.1](#)

[Download cuDNN v7.6.5 \(November 5th, 2019\), for CUDA 10.0](#)

[Download cuDNN v7.6.5 \(November 5th, 2019\), for CUDA 9.2](#)

[Download cuDNN v7.6.5 \(November 5th, 2019\), for CUDA 9.0](#)

[Archived cuDNN Releases](#)

- Download cudnn for your CUDA version

[Download cuDNN v7.6.5 \[November 5th, 2019\], for CUDA 10.1](#)

## Library for Windows, Mac, Linux, Ubuntu and RedHat/Centos(x86\_64architecture)

[cuDNN Library for Windows 7](#)

[cuDNN Library for Windows 10](#)

[cuDNN Library for Linux](#)

[cuDNN Library for OSX](#)

[cuDNN Runtime Library for Ubuntu18.04 \(Deb\)](#)

[cuDNN Developer Library for Ubuntu18.04 \(Deb\)](#)

[cuDNN Code Samples and User Guide for Ubuntu18.04 \(Deb\)](#)

[cuDNN Runtime Library for Ubuntu16.04 \(Deb\)](#)

[cuDNN Developer Library for Ubuntu16.04 \(Deb\)](#)

[cuDNN Code Samples and User Guide for Ubuntu16.04 \(Deb\)](#)

[cuDNN Runtime Library for Ubuntu14.04 \(Deb\)](#)

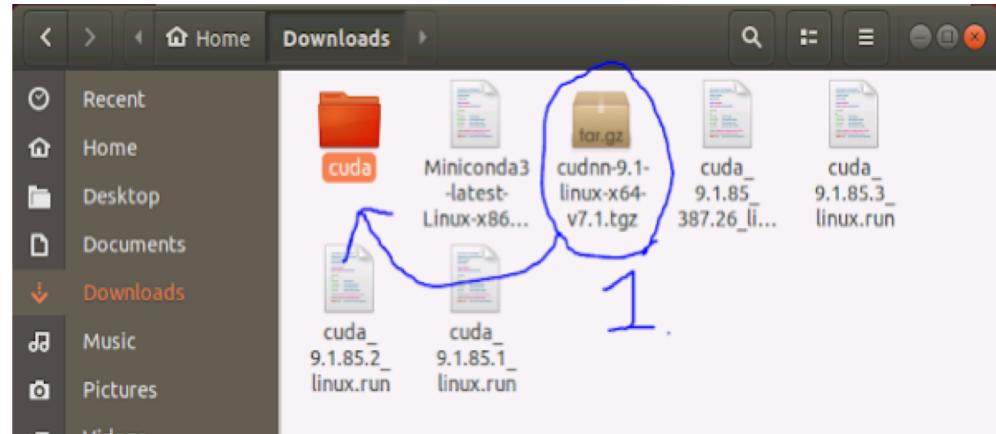
[cuDNN Developer Library for Ubuntu14.04 \(Deb\)](#)

[cuDNN Code Samples and User Guide for Ubuntu14.04 \(Deb\)](#)

# Cudnn

Cudnn is a compressed file

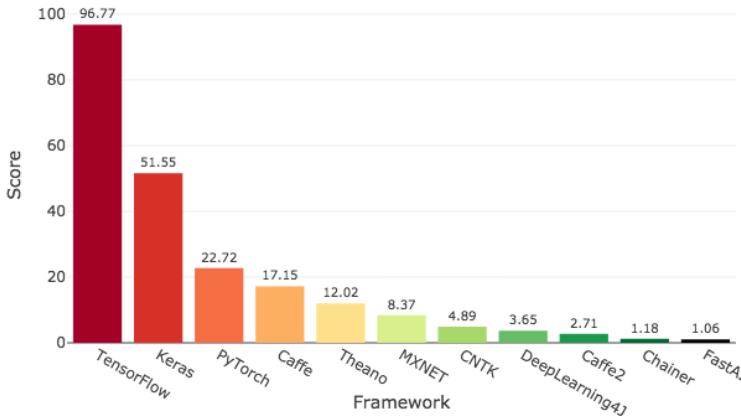
- Unzip the Cudnn file
- Find the Cuda file (previously downloaded)
  - cd cuda
  - Put Cudnn unzipped file into Cuda file





There are a lot of deep learning packages, such as Theano\*, Caffe, TensorFlow, PyTorch, Keras, and so on.

Deep Learning Framework Power Scores 2018



Theano 已死，有事烧纸



1人赞同了该文章

简单点儿说，MILA 的头出来说，继续搞 Theano 已经不大可能出啥象样的学术成果了，所以就不再折腾了……

[groups.google.com/forum...](https://groups.google.com/forum/)

# Deep Learning Package - Pytorch

- <https://pytorch.org/>

The image shows a screenshot of the PyTorch website's installation configuration interface. It features a grid of dropdown menus with various options. The options are:

- PyTorch Build: Stable (1.3) (highlighted in red)
- Your OS: Linux, Mac (highlighted in red), Windows
- Package: Conda (highlighted in red), Pip, LibTorch, Source
- Language: Python 2.7, Python 3.5, Python 3.6 (highlighted in red), Python 3.7, C++
- CUDA: 9.2, 10.1 (highlighted in red), None

Below the configuration grid, there is a section labeled "Run this Command:" containing the following text:

```
conda install pytorch torchvision -c pytorch
# MacOS Binaries dont support CUDA, install from source if CUDA is needed
```

A red rectangular box highlights the command text.

# Test Pytorch

- Go to your Power Shell / cmd / Terminal

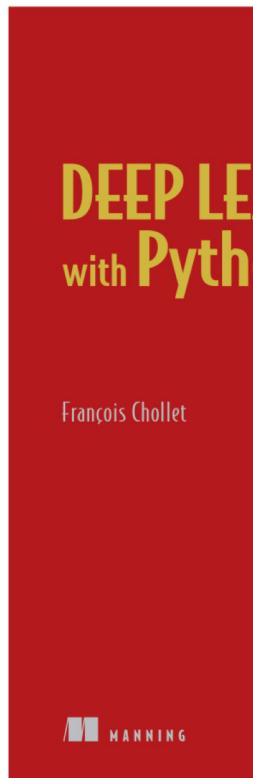
```
[ailab@ailab-1070ti-new-47:~$ python
Python 3.7.5 (default, Oct 25 2019, 15:51:11)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information
```

```
[>>> import torch
[>>> torch.cuda.is_available()
True
```

If output is true, it represents your GPU is loaded

- >>> quit() //you will end of the python program

# Deep Learning Package - Tensorflow & Keras



**You can do your final project from now on**

**If you don't have your own GPU**

# Google Colab

- Login your google drive
- Create a new colab file
  - Right click at empty place, in the menu: More>Colaboratory
  - If not found this option : More>Connect more apps > search and add Colaboratory



Drive

Search Drive

My Drive &gt; colab ▾

Name	Owner	Last modified
Untitled0.ipynb	me	Nov 28, 20

- [New](#)
- [My Drive](#)
- [Computers](#)
- [Shared with me](#)
- [Recent](#)
- [Starred](#)
- [Trash](#)
- [Backups](#)
- [Storage](#)

8.9 GB of 15 GB used

[UPGRADE STORAGE](#)

- [New folder...](#)
- [Upload files...](#)
- [Upload folder...](#)
- [Google Docs](#)
- [Google Sheets](#)
- [Google Slides](#)
- [More](#)

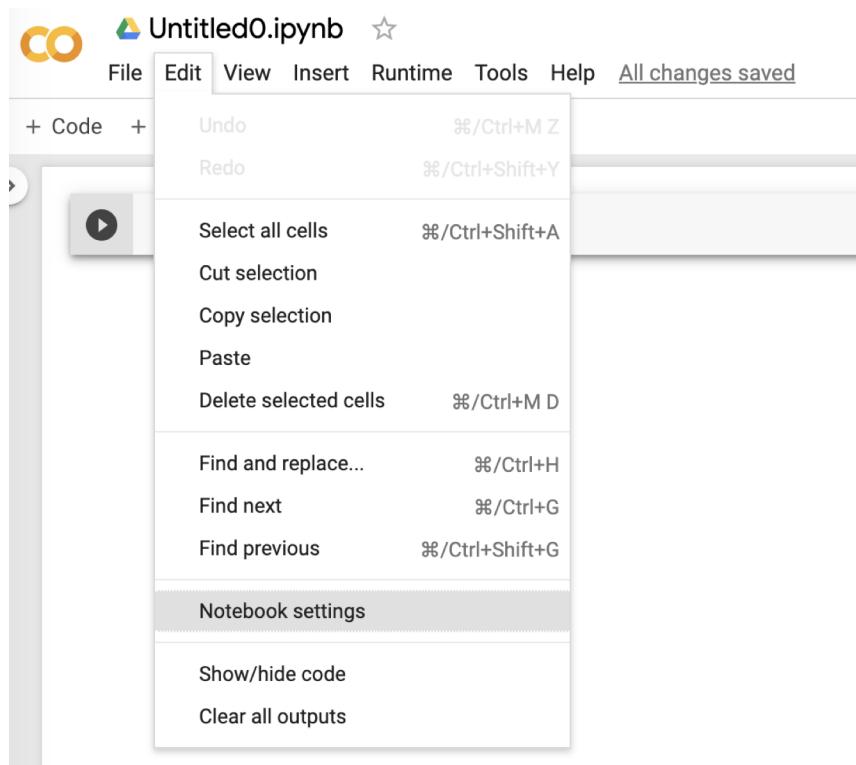
- [Google Forms](#)
- [Google Drawings](#)
- [Google My Maps](#)
- [Google Sites](#)
- [Colaboratory](#)
- [Gantter for Google Drive](#)
- [MindMeister](#)
- [Mindomo](#)
- [Pixlr Editor](#)
- [Connect more apps](#)

DV

- Here you could input some code, then press the play button or shift+enter to execute the code.
- You could just consider it as a cloud jupyter notebook with tensorflow in your google drive.

A screenshot of a Jupyter Notebook interface titled "Untitled0.ipynb". The notebook has a dark theme with white text. At the top, there's a toolbar with icons for file operations like "文件" (File), "修改" (Edit), "视图" (View), "插入" (Insert), "代码执行程序" (Code Interpreter), "工具" (Tools), and "帮助" (Help). Below the toolbar, there are tabs for "代码" (Code) and "文字" (Text), and buttons for "单元格" (Cell) operations: "↑ 单元格" (Up Cell) and "↓ 单元格" (Down Cell). A pink handwritten annotation "or" is placed above the play button icon. Another pink handwritten annotation "Shift + ↲" is placed above the "Shift + Enter" key combination. The main area shows a code cell containing the Python command `print("haha")`. The output of this cell is "haha", displayed in a separate row below the code. The entire interface is framed by a thick black border.

# Select GPU to Train Model



## Notebook settings

Runtime type

Python 3

Hardware accelerator

GPU



Omit code cell output when saving this notebook

CANCEL

SAVE

# Tensorflow Demo on Colab

- a good example to let you study how to visualize the result of your model
  1. Copy the test example code from [here](#)

This is a simple linear regression model optimized by gradient descent.

2. Paste the code to an empty colab notebook and execute.

Notice that tensorflow, numpy and matplotlib are all pre-installed in colab.

co Untitled0.ipynb ☆

文件 修改 视图 插入 代码执行程序 工具 帮助

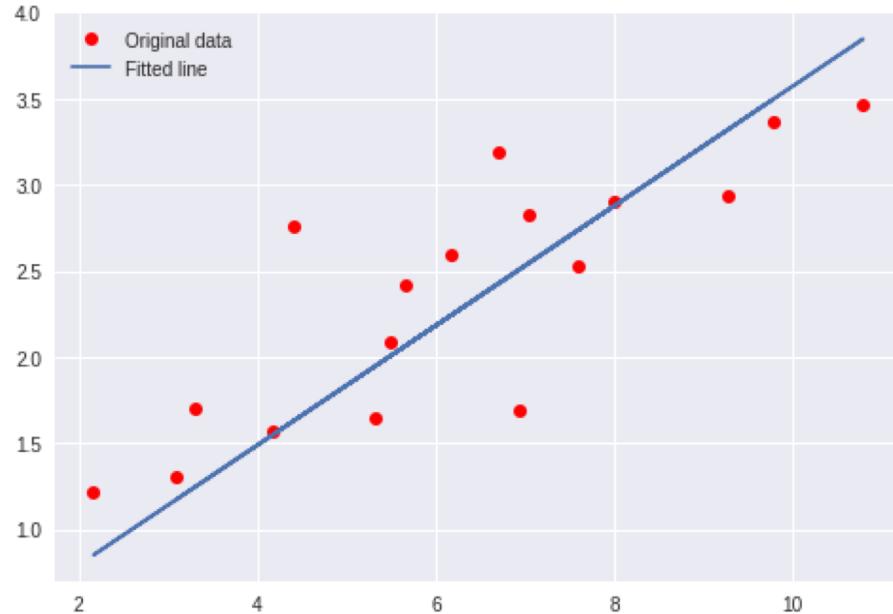
+ 代码 + 文字 | ↑ 单元格 ↓ 单元格

```
...  
A linear regression learning algorithm example using TensorFlow library.  
Author: Aymeric Damien  
Project: https://github.com/aymericdamien/TensorFlow-Examples/  
...  
  
from __future__ import print_function  
  
import tensorflow as tf  
import numpy  
import matplotlib.pyplot as plt  
rng = numpy.random  
  
# Parameters  
learning_rate = 0.01  
training_epochs = 1000  
display_step = 50  
  
# Training Data  
train_X = numpy.asarray([3.3,4.4,5.5,6.71,6.93,4.168,9.779,6.182,7.59,2.167,  
7.042,10.791,5.313,7.997,5.654,9.27,3.1])  
train_Y = numpy.asarray([1.7,2.76,2.09,3.19,1.694,1.573,3.366,2.596,2.53,1.221,  
2.827,3.465,1.65,2.904,2.42,2.94,1.3])  
n_samples = train_X.shape[0]  
  
# tf Graph Input  
X = tf.placeholder("float")  
Y = tf.placeholder("float")  
  
# Set model weights  
W = tf.Variable(rng.randn(), name="weight")  
b = tf.Variable(rng.randn(), name="bias")  
  
# Construct a linear model  
pred = tf.add(tf.multiply(X, W), b)  
  
# Mean squared error  
cost = tf.reduce_sum(tf.pow(pred-Y, 2))/(2*n_samples)  
# Gradient descent
```

代码 文字 单元格 单元格

Optimization Finished!

Training cost= 0.10748411 W= 0.34734625 b= 0.098240584



Testing... (Mean square loss Comparison)

Testing cost= 0.093680486

Absolute mean square loss difference: 0.013803624

3. The output can be easily show by the plot library: matplotlib

# Install Pytorch on Colab

You can use pytorch instead of tensorflow.

1. Import torch directly, then the colab would ask you to install pytorch.
2. Press INSTALL TORCH, then a menu would show up in the left side.
3. Press Insert in the sidemenu, then a new code block would show up
4. Execute the new code block

目录 代码段 文件 X

import torch

Install [pytorch](http://pytorch.org/) →

Install [pytorch](http://pytorch.org/)  
插入 3

```
# http://pytorch.org/
from os.path import exists
from wheel.pep425tags import get_abbr_impl
platform = '{}{}-{}'.format(get_abbr_impl(),
cuda_output = !ldconfig -p|grep cudart.so|awk '{print $3}'|head -1
accelerator = cuda_output[0] if exists('/dev/nvidia0') else 'cpu'
!pip install -q http://download.pytorch.org/torch/torch-0.4.1-{platform}-linux_x86_64.whl
import torch
```

查看来源笔记本

[4] from torch import nn 1

ModuleNotFoundError Traceback (most recent call last)
<ipython-input-4-d986de4e2488> in <module>()
----> 1 from torch import nn

ModuleNotFoundError: No module named 'torch'

NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To install torch, click the button below.

2 INSTALL TORCH SEARCH STACK OVERFLOW

[5] # http://pytorch.org/
from os.path import exists
from wheel.pep425tags import get\_abbr\_impl, get\_impl\_ver, get\_abi\_tag
platform = '{}{}-{}'.format(get\_abbr\_impl(), get\_impl\_ver(), get\_abi\_tag())
cuda\_output = !ldconfig -p|grep cudart.so|sed -e 's/.\*\.\([0-9]\*\)\.\([0-9]\*\)\\$/cu\1\2/'|head -1
accelerator = cuda\_output[0] if exists('/dev/nvidia0') else 'cpu'
!pip install -q http://download.pytorch.org/torch/torch-0.4.1-{platform}-linux\_x86\_64.whl
import torch

Shift + ↴



The screenshot shows a Jupyter Notebook interface with the title "Untitled0.ipynb". The menu bar includes File, Edit, View, Insert, Kernel, Help, and Tools. Below the menu is a toolbar with "代码" (Code) and "文字" (Text) buttons, and "单元格" (Cell) navigation buttons. The notebook contains three code cells:

- [9] import torch
- [11] # torch hello world  
a = torch.Tensor([1,2,3,4])  
a
- ⇒ tensor([1., 2., 3., 4.])

A pink arrow points from the text "torch hello world" in cell [11] to the output "tensor([1., 2., 3., 4.])" in cell [11].

- Delete all block then try torch.
- If success, “torch hello world” should work

# Install other python packages on Colab

just input:

```
!pip install {package name}
```

then execute.

ex. install pandas:

```
[12] !pip install pandas
Requirement already satisfied: pandas in /usr/local/lib/python3.6/dist-packages (0.22.0)
Requirement already satisfied: pytz>=2011k in /usr/local/lib/python3.6/dist-packages (from pandas) (2018.7)
Requirement already satisfied: numpy>=1.9.0 in /usr/local/lib/python3.6/dist-packages (from pandas) (1.14.6)
Requirement already satisfied: python-dateutil>=2 in /usr/local/lib/python3.6/dist-packages (from pandas) (2.5.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.6/dist-packages (from python-dateutil>=2->pandas) (1.11.0)
```

# Mount google drive folder to colab

## 1. Input this and execute

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

#### 2.-4. follow the picture

1

```
D from google.colab import drive  
drive.mount('/content/drive')  
2  
... Go to this URL in a browser: https://accounts.google.com/o/oauth2  
Enter your authorization code:  

```



**5.&6. Paste auth code to colab and press enter**

登入

請複製這組授權碼，然後切換至您的應用程式，再貼上授權碼：

5 copy

```
from google.colab import drive  
drive.mount('/content/drive')  
  
... Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client\_id=  
  
Enter your authorization code:  
[REDACTED] 6 paste
```

## 7. Test the mounted google drive folder by `os.listdir()`

I just show the files in 'Colab Notebooks' here. You could find everything in '`/content/drive/My Drive`'.

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

↳ Go to this URL in a browser: <https://accounts.google.com/o/>

Enter your authorization code:  
.....

Mounted at /content/drive

done

```
[20] import os  
os.listdir("/content/drive/My Drive/Colab Notebooks")
```

↳ ['Untitled0.ipynb']

**So, you can do your project from now on**