

# Python 소개

# CONTENTS

01

Python 소개

02

환경설정

03

Colab 설정



# 01 Python 소개

## 왜 파이썬 인가?

### 장점 1. 읽고 사용하기가 쉽다.

파이썬 개발자이자 PyPL 언어 인덱스를 운영하는 블로거인 피에르 카보넬(Pierre Carbonnelle)은 “파이썬 프로그램의 주요 특성은 읽기 쉽다는 데 있다”고 말했다. 카보넬은 “이점이 많은 개발자에게 혜택을 가져다주었다. 프로그램 작성에 있어서 개발자의 생각을 명확하게 발전시켜줄 뿐만 아니라, 이 프로그램을 유지 보수하거나 개선하는데에도 많은 도움을 준다. 두 사례 모두에서 C++나 자바 같은 언어보다도 파이썬 프로그램은 훨씬 공수(Man-Month, M/M)가 더 적게 들어간다”고 말했다. 즉, 파이썬의 안정성은 오픈소스 개발을 돕는다는 의미다.

PyPL - <https://pypl.github.io/PYPL.html>

## 왜 파이썬 인가?

### 장점 2. 사물인터넷 기회.

파이썬은 **라즈베리파이 (Raspberry pi)**가 기반으로 하고 있는 만큼 사물인터넷에서도 인기를 얻을 것이라고 카보넬은 말한다. 라즈베리파이의 설명서에서는 파이썬 언어를 “사용이 쉬운 (읽고 쓰기 쉬운) 환상적이고 강력한 프로그래밍 언어로, 라즈베리파이와 더불어 사용자의 프로젝트를 실제 세계와 연결해준다”고 이야기한다.

## 왜 파이썬 인가?

### 장점 3. 비동기식 코딩

디벨은 “파이썬은 쓰레딩 대신 단일 이벤트 루프를 사용해 소수 유닛에서 작업하는 비동기식 코드를 작성하는 데 뛰어나다”며, “이 코드는 혼란스러운 자원 경쟁이나 교착상태를 유발하지 않고도 작성과 유지보수를 수월하게 만들어준다고 말했다. “파이썬의 발생자는 이런 방식으로 많은 프로세싱 루프를 끼워 넣는 훌륭한 방식이다.”

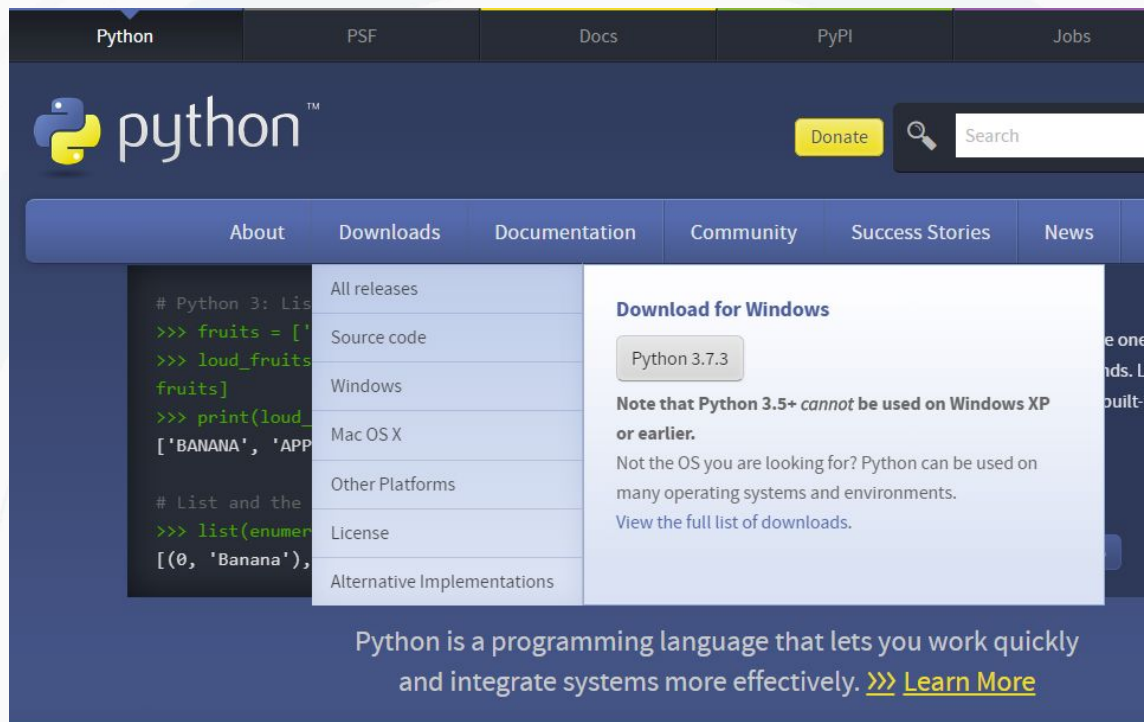




## 02 환경 설정

## 파이썬 설치

링크 <https://www.python.org/> 로 가서, Downloads 클릭하고, 파이썬 최신 버전(3.7.3)을 설치.





## 윈도우즈에 아나콘다, 주피터 노트북 설치하기

### 필요한 것

- 파이썬(Python) - 개발 환경 (파이썬 공식 사이트에서 인스톨파일을 다운받아 설치)
- 과학 계산을 위한 여러 파이썬 패키지를 따로 설치.
- 대표적으로 **pandas, NumPy, scikit-learn** 등
- 이런 패키지들은 설치할 때 잘 안 돌아가는 어려움
- 운영체제 환경에 맞게 **패키지들을 미리 준비해 놓은 배포판**을 더 선호함

## 아나콘다 설치

브라우저로 아나콘다  
다운로드 페이지에  
접속합니다.

<https://www.anaconda.com/distribution/>

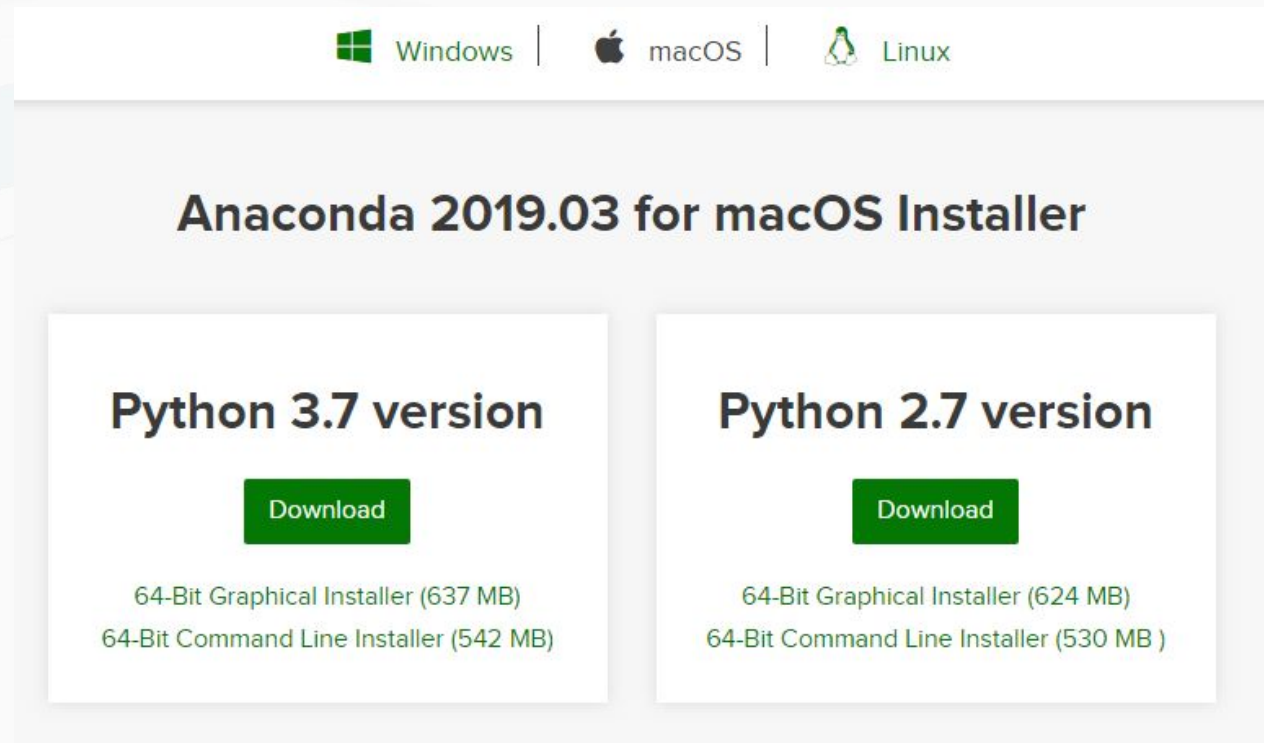
The open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 11 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download 1,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with **Conda**
- Develop and train machine learning and deep learning models with **scikit-learn**, **TensorFlow**, and **Theano**
- Analyze data with scalability and performance with **Dask**, **NumPy**, **pandas**, and **Numba**
- Visualize results with **Matplotlib**, **Bokeh**, **Datashader**, and **Holoviews**

## 아나콘다 설치

파이썬 2.7과 3.7 버전의 Windows용 설치 파일이 있습니다.

Windows용 텐서플로 바이너리 패키지는 현재 파이썬 3.5, 3.6, 3.7버전을 지원하므로 아나콘다의 파이썬 3.7 버전을 설치해야 합니다.



Windows | macOS | Linux

### Anaconda 2019.03 for macOS Installer

#### Python 3.7 version

[Download](#)

64-Bit Graphical Installer (637 MB)  
64-Bit Command Line Installer (542 MB)

#### Python 2.7 version

[Download](#)

64-Bit Graphical Installer (624 MB)  
64-Bit Command Line Installer (530 MB)

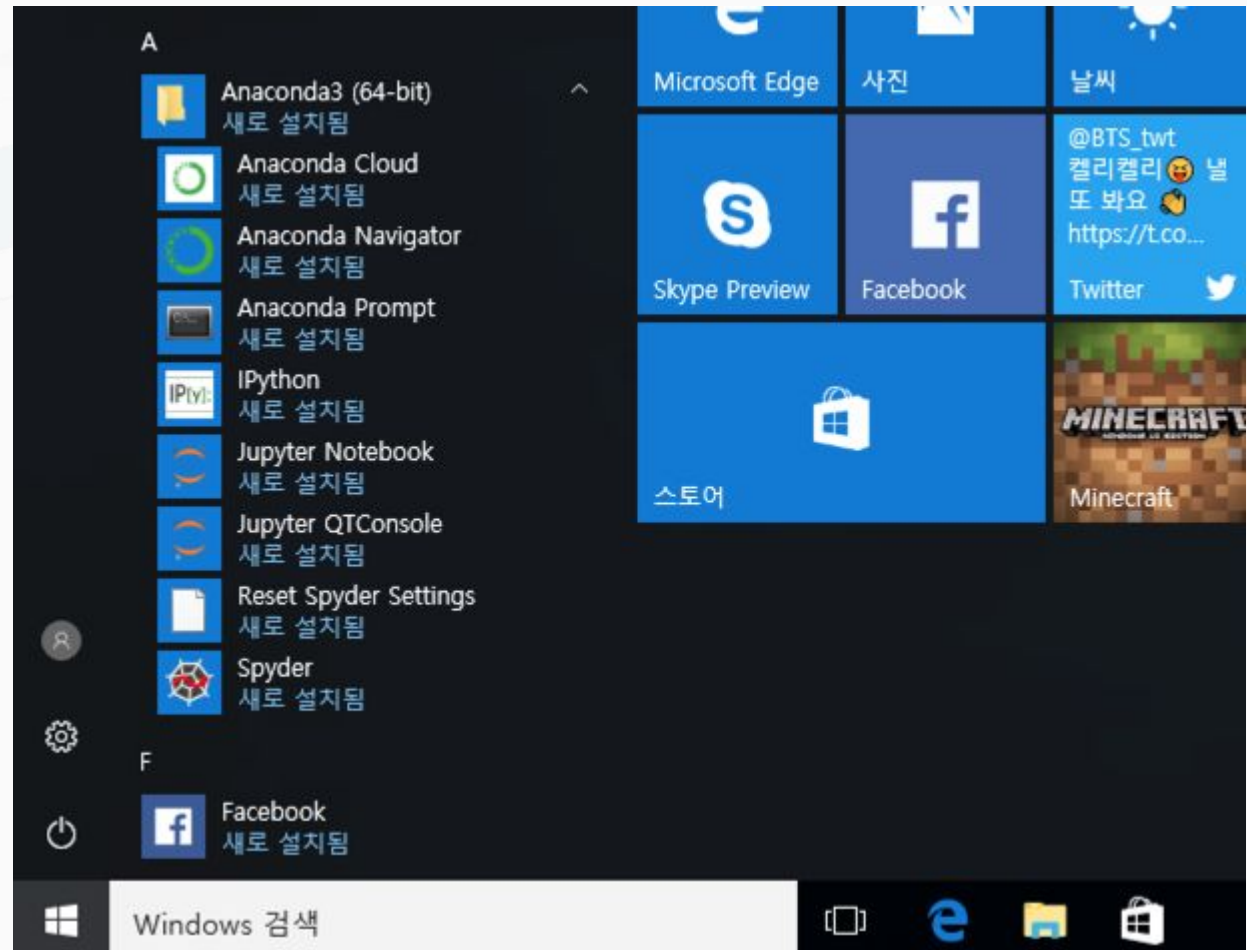
## 아나콘다 설치

Setup을 시작합니다.



## 아나콘다 설치

이 다음부터는  
Anaconda Prompt에서  
명령행 인터페이스를  
사용해서  
진행하겠습니다



## 아나콘다 설치

Anaconda Prompt에 들어오면 conda 패키지 관리자를 사용할 수 있습니다.

먼저 conda 자체를 업데이트 합니다.

```
>conda update -n base conda
```

다음엔 설치된 파이썬 패키지를 모두 최신 버전으로 업데이트 합니다.

```
>conda update --all
```



## 아나콘다 설치

```

Administrator: Anaconda Prompt
verify
(base) C:\WINDOWS\system32>conda update -n base conda
Collecting package metadata: done
Solving environment: done

## Package Plan ##

environment location: C:\Anaconda
added / updated specs:
- conda

The following packages will be downloaded:

package | build | size
-----|-----|-----
certifi-2019.3.9 | py37_0 | 155 KB
conda-4.7.5 | py37_0 | 3.0 MB
conda-package-handling-1.3.10 | py37_0 | 280 KB
-----|-----|-----
Total: | 3.5 MB

The following NEW packages will be INSTALLED:
conda-package-handling-1.3.10-py37_0

The following packages will be UPDATED:
conda | conda-forge::conda-4.6.14-py37_0 --> pkgs/main::conda-4.7.5-py37_0

The following packages will be SUPERSEDED by a higher-priority channel:
certifi | conda-forge --> pkgs/main

Proceed ([y]/n)? y

Downloading and Extracting Packages
conda-4.7.5 | 3.0 MB | ##### | 100%
conda-package-handling-1.3.10 | 280 KB | ##### | 100%
certifi-2019.3.9 | 155 KB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
  
```

## 텐서플로우 설치

아나콘다 배포판에는 텐서플로 최신 버전이 늦게 포함되므로 파이썬 기본 패키지 관리자인 pip로 텐서플로를 설치합니다.

```
>pip install tensorflow
```

## 텐서플로우 설치

```
>pip install tensorflow
```

```
Anaconda Prompt
unicodecsv: 0.14.1-py35_0
vs2015_runtime: 14.0.25123-0
wcmwidth: 0.1.7-py35_0
werkzeug: 0.11.15-py35_0
wheel: 0.29.0-py35_0
widgetsnbextension: 1.2.6-py35_0
win_unicode_console: 0.5-py35_0
wrapit: 1.10.8-py35_0
xlrd: 1.0.0-py35_0
xlsxwriter: 0.9.6-py35_0
xlwings: 0.10.2-py35_0
xlwt: 1.2.0-py35_0
zlib: 1.2.8-vc14_3 [vc14]

Proceed ([y]/n)? y
#
# To activate this environment, use:
# > activate py35
#
# To deactivate this environment, use:
# > deactivate py35
#
# * for power-users using bash, you must source
#

C:\Users\Ricky Park>activate py35
(py35) C:\Users\Ricky Park>pip install tensorflow
```

## 텐서플로우 설치

```
>pip install tensorflow
```

이미 설치가 된 경우는 'Requirement already satisfied'라는 메시지가 보여집니다.

```
Administrator: Anaconda Prompt - jupyter notebook

(base) C:\WINDOWS\system32>pip install tensorflow
Requirement already satisfied: tensorflow in c:\anaconda\lib\site-packages (1.13.1)
Requirement already satisfied: six>=1.10.0 in c:\anaconda\lib\site-packages (from tensorflow) (1.12.0)
Requirement already satisfied: keras-preprocessing>=1.0.5 in c:\anaconda\lib\site-packages (from tensorflow) (1.0.9)
Requirement already satisfied: tensorboard<1.14.0,>=1.13.0 in c:\anaconda\lib\site-packages (from tensorflow) (1.13.1)
Requirement already satisfied: wheel>=0.26 in c:\anaconda\lib\site-packages (from tensorflow) (0.33.1)
Requirement already satisfied: tensorflow-estimator<1.14.0rc0,>=1.13.0 in c:\anaconda\lib\site-packages (from tensorflow) (1.13.0)
Requirement already satisfied: keras-applications>=1.0.6 in c:\anaconda\lib\site-packages (from tensorflow) (1.0.7)
Requirement already satisfied: grpcio>=1.8.6 in c:\anaconda\lib\site-packages (from tensorflow) (1.16.1)
Requirement already satisfied: absl-py>=0.1.6 in c:\anaconda\lib\site-packages (from tensorflow) (0.7.1)
Requirement already satisfied: numpy>=1.13.3 in c:\anaconda\lib\site-packages (from tensorflow) (1.16.2)
Requirement already satisfied: astor>=0.6.0 in c:\anaconda\lib\site-packages (from tensorflow) (0.7.1)
Requirement already satisfied: protobuf>=3.6.1 in c:\anaconda\lib\site-packages (from tensorflow) (3.7.1)
Requirement already satisfied: termcolor>=1.1.0 in c:\anaconda\lib\site-packages (from tensorflow) (1.1.0)
Requirement already satisfied: gast>=0.2.0 in c:\anaconda\lib\site-packages (from tensorflow) (0.2.2)
Requirement already satisfied: werkzeug>=0.11.15 in c:\anaconda\lib\site-packages (from tensorflow) (0.14.1)
Requirement already satisfied: markdown>=2.6.8 in c:\anaconda\lib\site-packages (from tensorflow) (2.6.11)
Collecting mock>=2.0.0 (from tensorflow-estimator<1.14.0rc0,>=1.13.0->tensorflow)
  Downloading https://files.pythonhosted.org/packages/05/d2/f94e68be6b17f46d2c353564da56e6fb89ef09faeeff3313a046cb810ca9/mock-3.0.5-py2.py3-none-any.whl
Requirement already satisfied: h5py in c:\anaconda\lib\site-packages (from keras-applications>=1.0.6->tensorflow) (2.9.0)
Requirement already satisfied: setuptools in c:\anaconda\lib\site-packages (from protobuf>=3.6.1->tensorflow) (40.8.0)
Installing collected packages: mock
Successfully installed mock-3.0.5
```

## 주피터 실행

주피터 노트북을 실행하려면 아나콘다 프롬프트에서 `jupyter notebook` 명령을 사용합니다.

```
>jupyter notebook
```

주피터 노트북이 실행되면 자동으로 기본 브라우저가 실행되어 주피터 노트북 서버에 접속합니다.

## 주피터 실행

```
(base) C:\WINDOWS\system32>jupyter notebook
[I 22:29:13.504 NotebookApp] JupyterLab extension loaded from C:\Anaconda\lib\site-packages\jupyterlab
[I 22:29:13.504 NotebookApp] JupyterLab application directory is C:\Anaconda\share\jupyter\lab
[I 22:29:13.506 NotebookApp] Serving notebooks from local directory: C:\WINDOWS\system32
[I 22:29:13.506 NotebookApp] The Jupyter Notebook is running at:
[I 22:29:13.506 NotebookApp] http://localhost:8888/?token=de5df78579814b8d052c643860741ee85b6ef1b13311efc9
[I 22:29:13.506 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 22:29:13.742 NotebookApp]
```

To access the notebook, open this file in a browser:

file:///C:/Users/yyoo%20DL/AppData/Roaming/jupyter/runtime/nbserver-14600-open.html

Or copy and paste one of these URLs:

http://localhost:8888/?token=de5df78579814b8d052c643860741ee85b6ef1b13311efc9



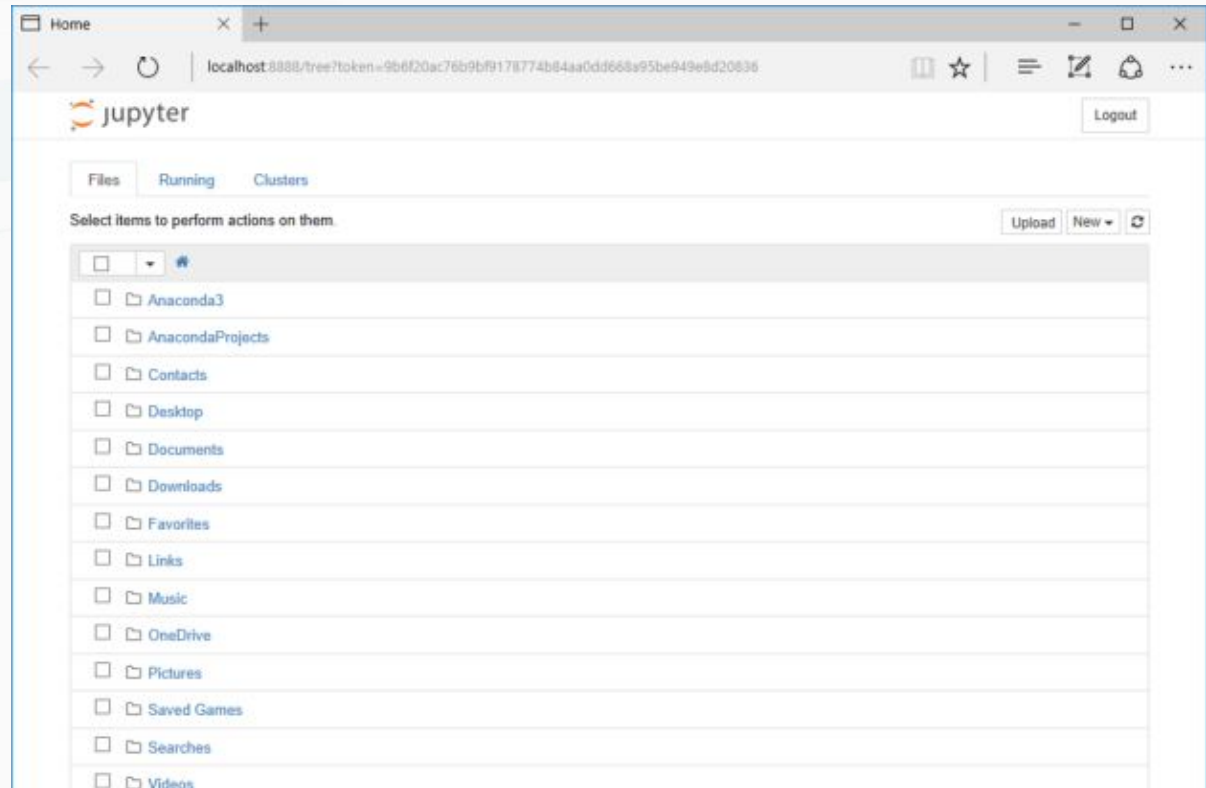
## 주피터 실행

로컬 컴퓨터의 주피터 노트북  
서버 주소는

<http://localhost:8888/>입니다.

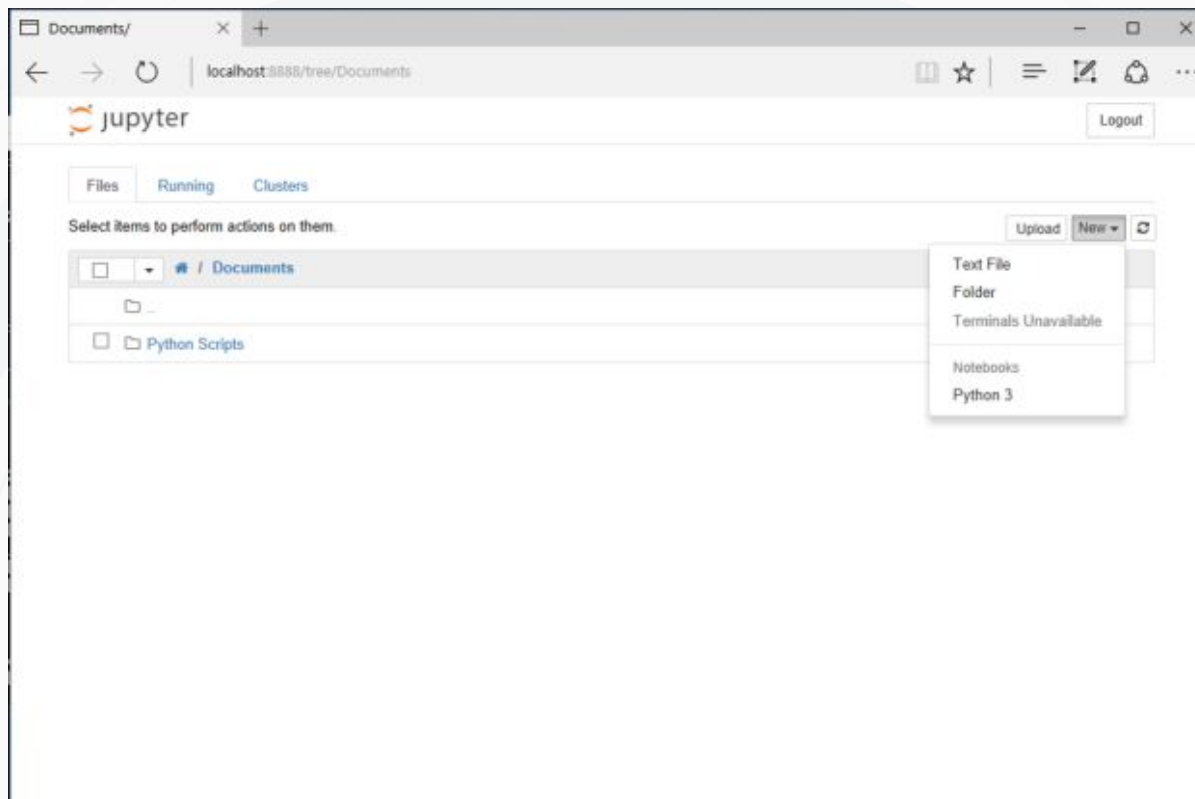
주피터 노트북을 실행한 현재  
폴더를 기본 홈 페이지로  
설정됩니다.

이 폴더 하위에 파이썬 주피터  
노트북을 만들고 실행할 수  
있습니다.



## 주피터 실행

Documents 폴더로 들어가서 새로운 파이썬 노트북을 만들어 보겠습니다. 오른쪽 위에 있는 New 버튼을 누르면 새로운 파이썬 3 주피터 노트북을 생성할 수 있습니다.



## 주피터 실행

새로운 브라우저 탭이 열리면서 Untitled 노트북이 생성됩니다.

첫번째 코드 셀(cell)에 `import tensorflow as tf`를 입력하고 **Shift+엔터 키**를 입력합니다.

아무런 메시지가 나오지 않으면 텐서플로를 주피터 노트북에서 사용할 수 있도록 설치에 성공한 것입니다.

또한, `tf.__version__`로 현재 텐서플로우 버전을 알 수 있습니다.



# 03 Colab 설정

## Google Colab

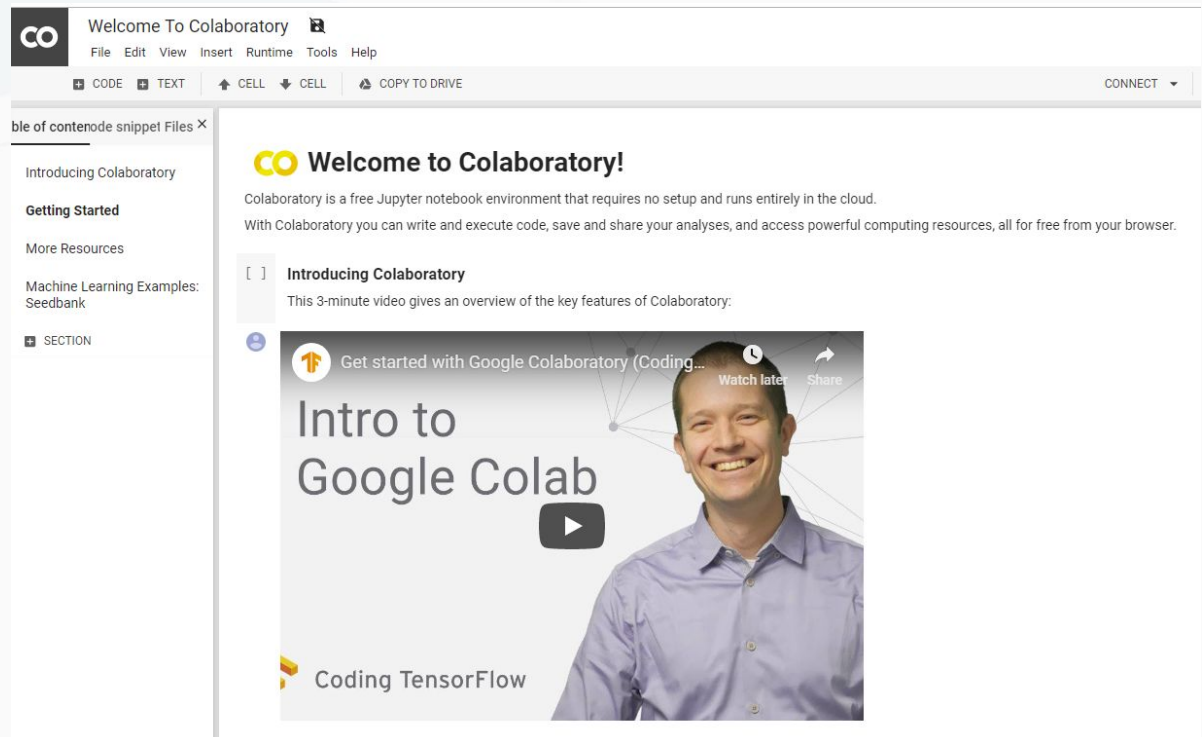
- Colaboratory는 설치가 필요없고 전적으로 클라우드에서 실행되는 무료 Jupyter 노트북 환경
- Colaboratory를 사용하면 브라우저에서 무료로 코드를 작성하고 실행하고 분석을 저장 및 공유하며 강력한 컴퓨팅 리소스에 액세스 가능
- 코랩에선 파이썬을 자유롭게 코딩할 수 있고, 텐서플로우나 케라스 등 딥러닝 라이브러리도 쉽게 사용가능
- 데이터 관련 처리를 하다보면 자신의 데이터를 클라우드 상에서 올려서 테스트 해야할 때가 있는데, 로컬 디스크에서 코랩 클라우드 상에 바로 업로드 가능.
- 또한, 구글 드라이브 상에 올린다음 연동 후 사용 가능

아래 링크에서 시작

<https://colab.research.google.com/>

## Google Colab(Colaboratory)

- Google 계정
- 실시간 저장
- Share
- Collaboration
- 풍부한 resource

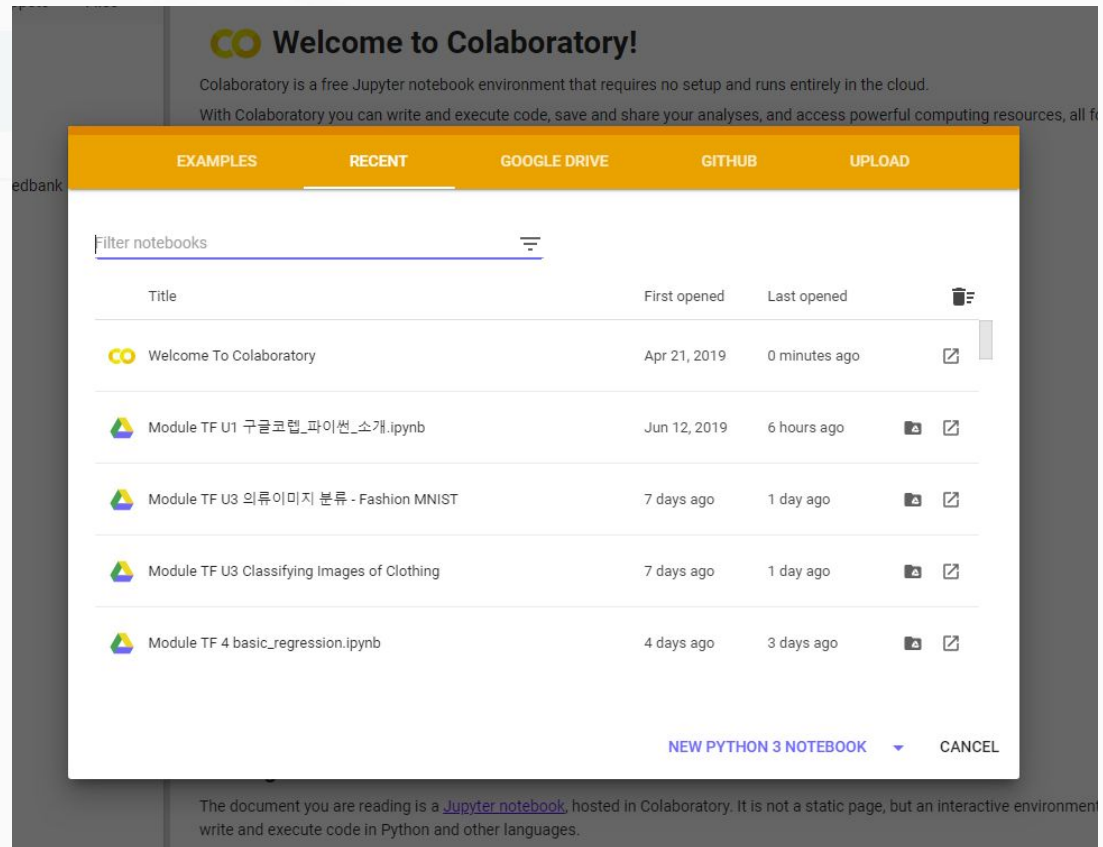




## Google Colab(Colaboratory)

풍부한 resource

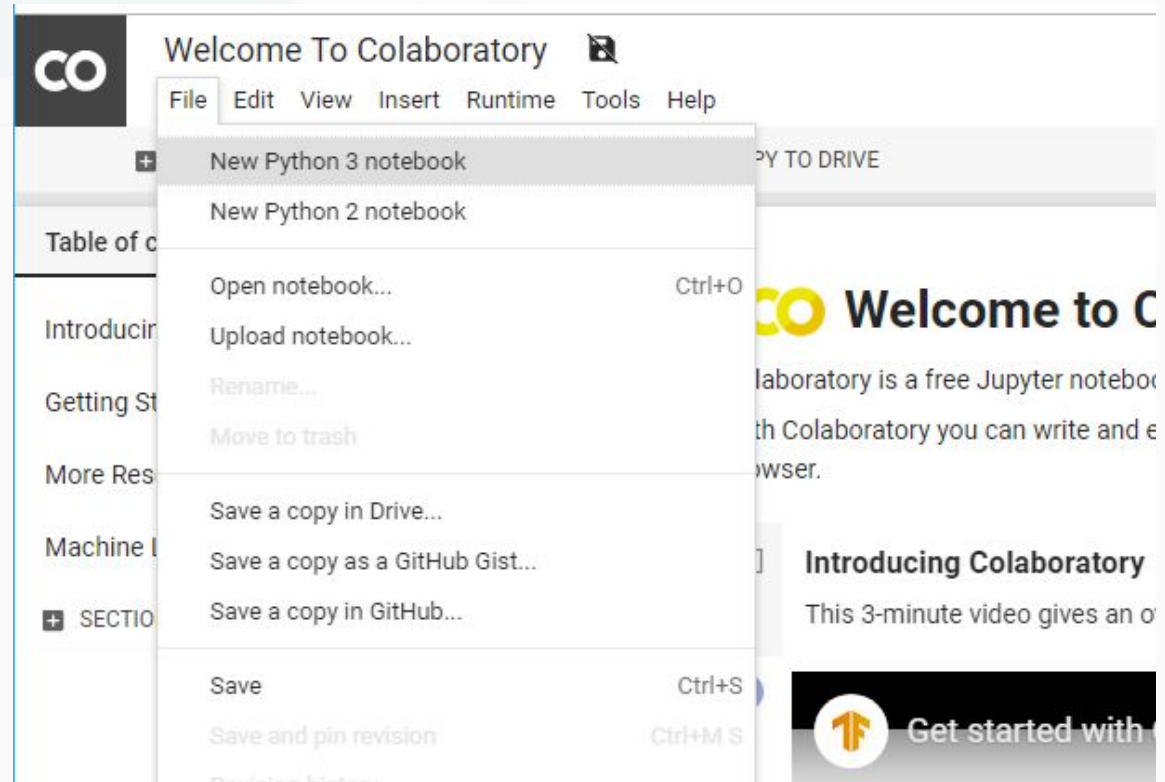
- Examples / Recent
- Google Drive
- 깃허브(Github)
- Upload



## Google Colab(Colaboratory)

새로운 노트북 시작

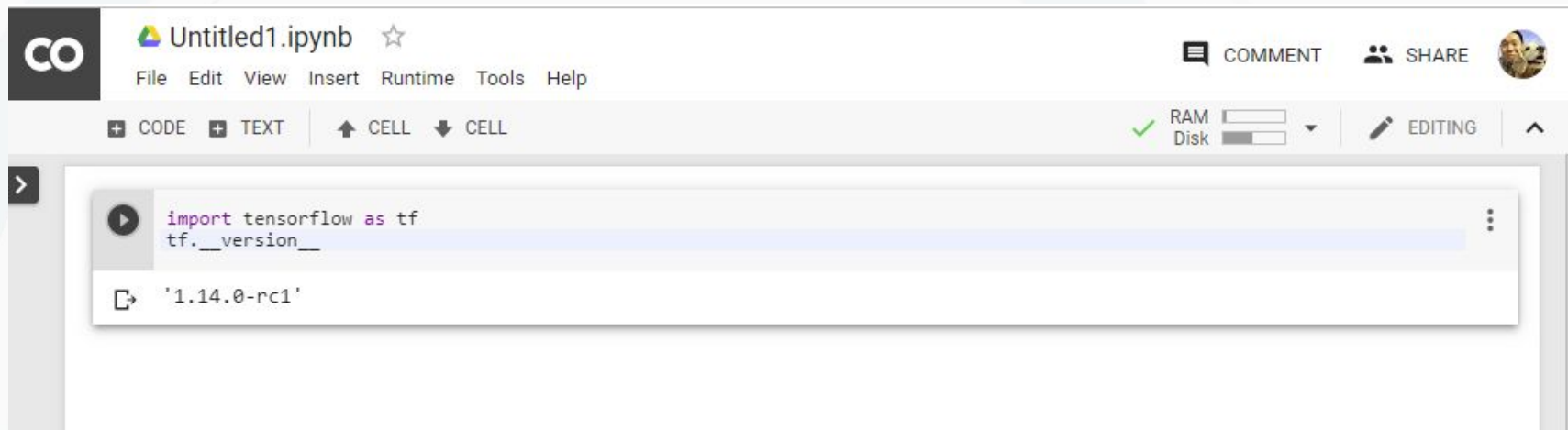
New Python 3 notebook



## Google Colab(Colaboratory)

텐서플로우 불러온 후 버전 확인하기

```
import tensorflow as tf  
  
tf.__version__
```



## Reference

<http://www.itworld.co.kr/news/92103>

[https://tykimos.github.io/2019/01/22/colab\\_getting\\_started/](https://tykimos.github.io/2019/01/22/colab_getting_started/)