

Python

Data 처리 과정 과 기술

- ❖ Data 처리 및 분석: 데이터의 양이 많아지면서 인간이 하는 것 보다는 컴퓨터 프로그램을 활용하는 것이 효율적이고 편견을 배제할 수 있음



Data 처리 과정 과 기술

❖ 용어

- ✓ 데이터 과학: 정형, 혹은 비정형 데이터를 수집하고 데이터를 정제하고, 준비하고, 분석하는 것과 관련된 모든 것을 포함하는 분야로 데이터로부터 정보 와 Business Insight를 추출하려고 할 때 사용되는 기술들의 포괄적 용어
- ✓ 빅데이터: 기존의 응용 프로그램으로는 효과적으로 처리할 수 없는 엄청난 양 의 데이터 또는 빠른 속도로 생성되는 데이터
- ✓ 데이터 분석: Business Insight 도출을 위해 알고리즘과 수학적 처리 과정을 적용하여 해당 정보에 대한 결론을 도출하고 패턴을 찾기 위한 목적으로 데이터를 다루는 과학



DATA SCIENCE

비정형 혹은 정형 데이터를 정제, 준비, 분석하는 활동



BIG DATA

엄청난 양의 데이터를 말하며 다양한 형태의 정보 자산



DATA ANALYTICS

알고리즘과 수학적 처리 과정을 적용하여 데이터를 다루는 활동

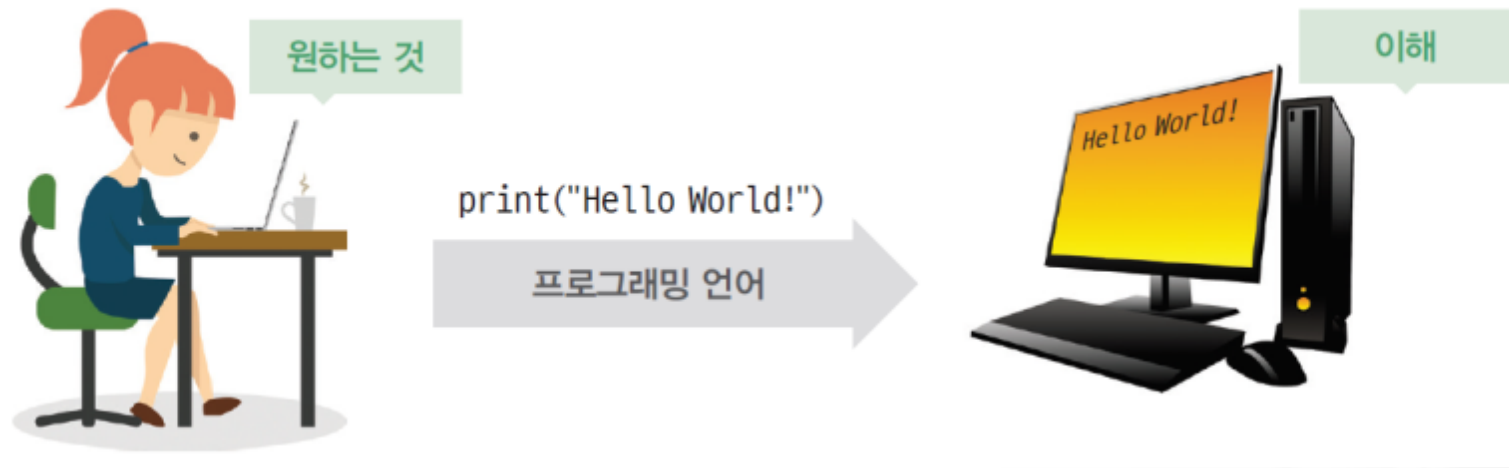
Data 처리 과정 과 기술

- ❖ 프로그래밍 언어(programming language): 인간이 생각하는 것을 컴퓨터로 실행시키기 위해 사용하는 컴퓨터가 이해할 수 있는 언어



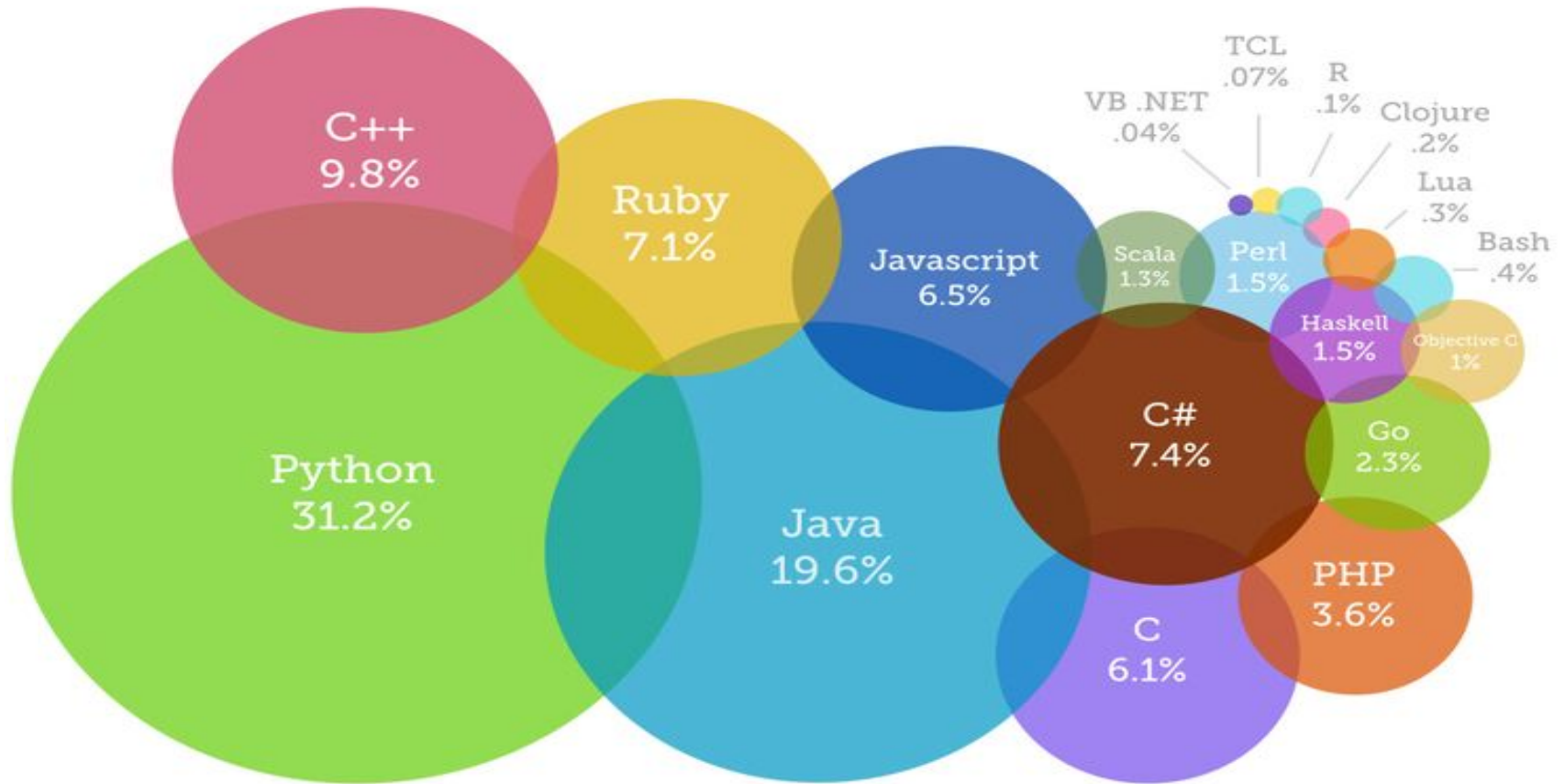
Data 처리 과정 과 기술

- ❖ 프로그래밍(programming): 프로그래밍 언어를 사용하여 프로그램을 개발하는 것



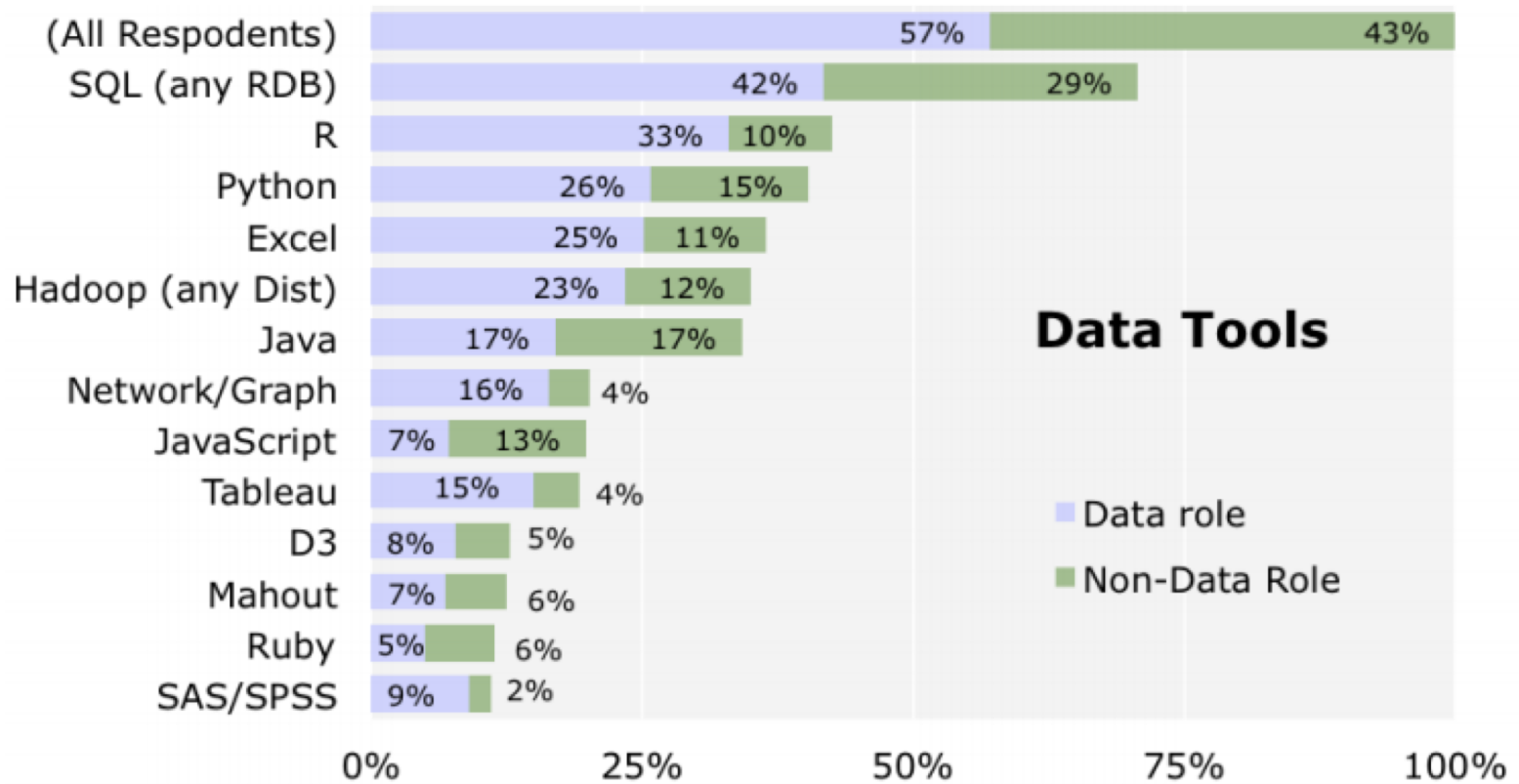
Data 처리 과정 과 기술

❖ 프로그래밍 언어



Data 처리 과정 과 기술

❖ Data Tools



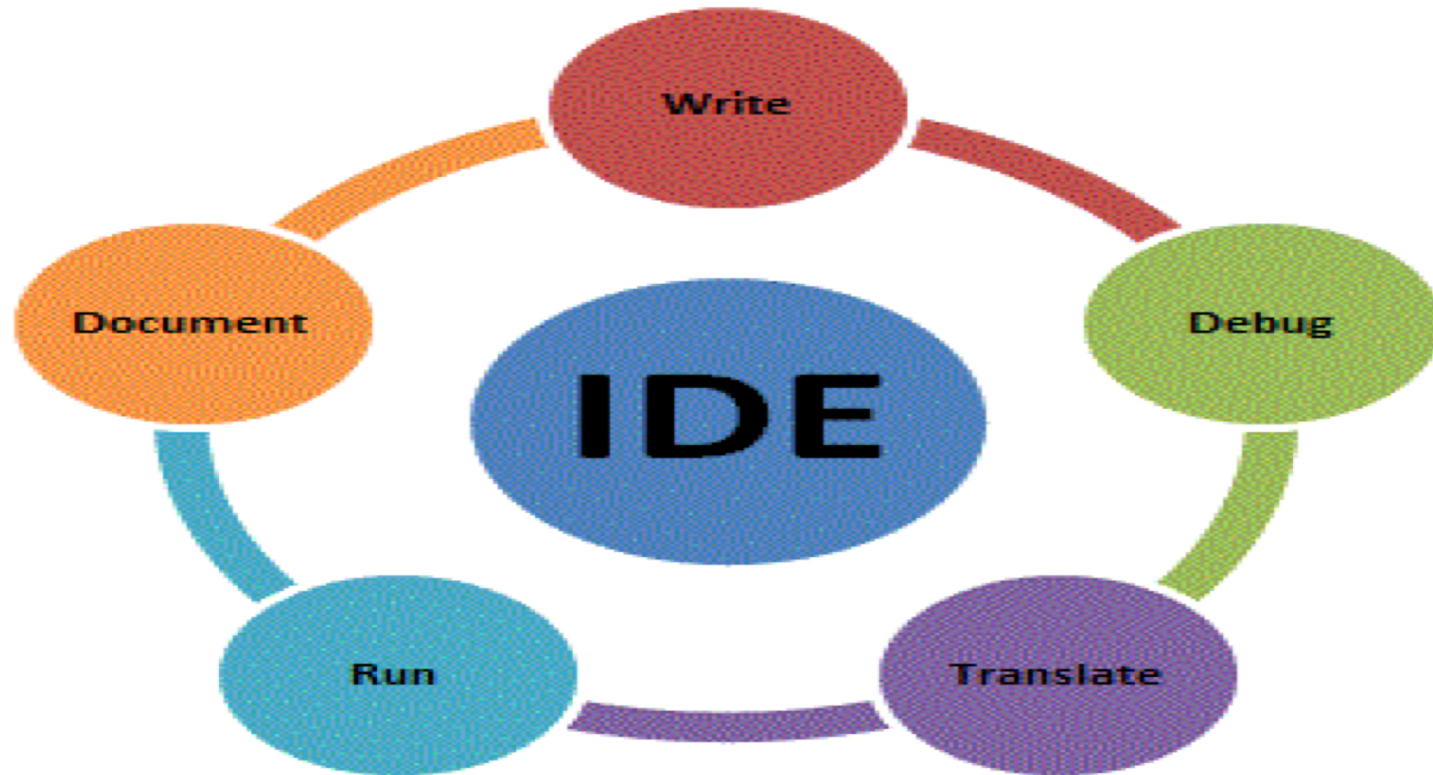
Data 처리 과정 과 기술

- ❖ 컴파일러 & 인터프리터: 인간이 고급 프로그래밍 언어(Source Code)로 작성한 것을 컴퓨터가 실행 할 수 있도록 운영 체제나 가상 머신(Virtual Machine)이 이해할 수 있는 코드로 번역해주는 프로그램



Data 처리 과정 과 기술

- ❖ IDE(integrated development environment – 통합 개발 환경): 코딩, 디버그, 컴파일, 배포 등 프로그램 개발에 관련된 모든 작업을 하나의 프로그램 안에서 처리하는 환경을 제공하는 프로그램



python

❖ python

- ✓ 1991년 프로그래머인 Guido van Rossum이 개발한 플랫폼 독립적인 고급 프로그래밍 언어
- ✓ 인터프리터 언어이며 플랫폼에 독립적인 언어
- ✓ 객체 지향 언어이며 동적 타이핑 언어
- ✓ {} 등의 괄호를 넣지 않고 들여쓰기를 이용하기 때문에 코드가 읽기 편함(가독성)
- ✓ 생태계가 좋음 – 3rd Party 개발자가 만든 라이브러리가 많음



python

❖ python

✓ python의 구현

- Cpython: C로 작성된 인터프리터를 사용하는 일반적인 python
- Stackless python: C Stack을 사용하지 않는 인터프리터
- Jython : 자바 가상 머신 용 인터프리터로 과거에는 제이python(Jpython)이라고 불리기도 함
- Ironpython : .NET 플랫폼 용 인터프리터
- PyPy : python으로 작성된 python 인터프리터

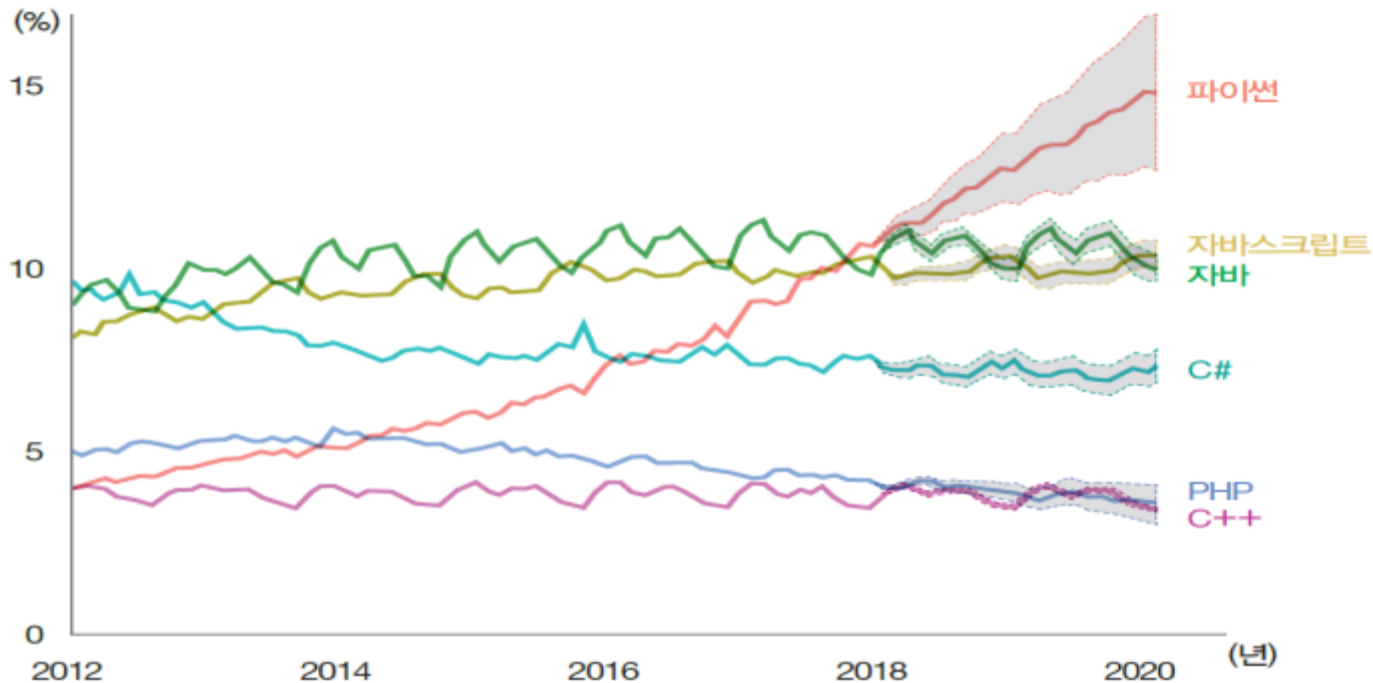
✓ 사용 가능 플랫폼

- MS Windows
- Macintosh
- Unix
- Linux

python

❖ python을 배우는 이유

- ✓ Stack Overflow의 조사 결과 현재 가장 많은 질의 응답을 받는 프로그래밍 언어는 python

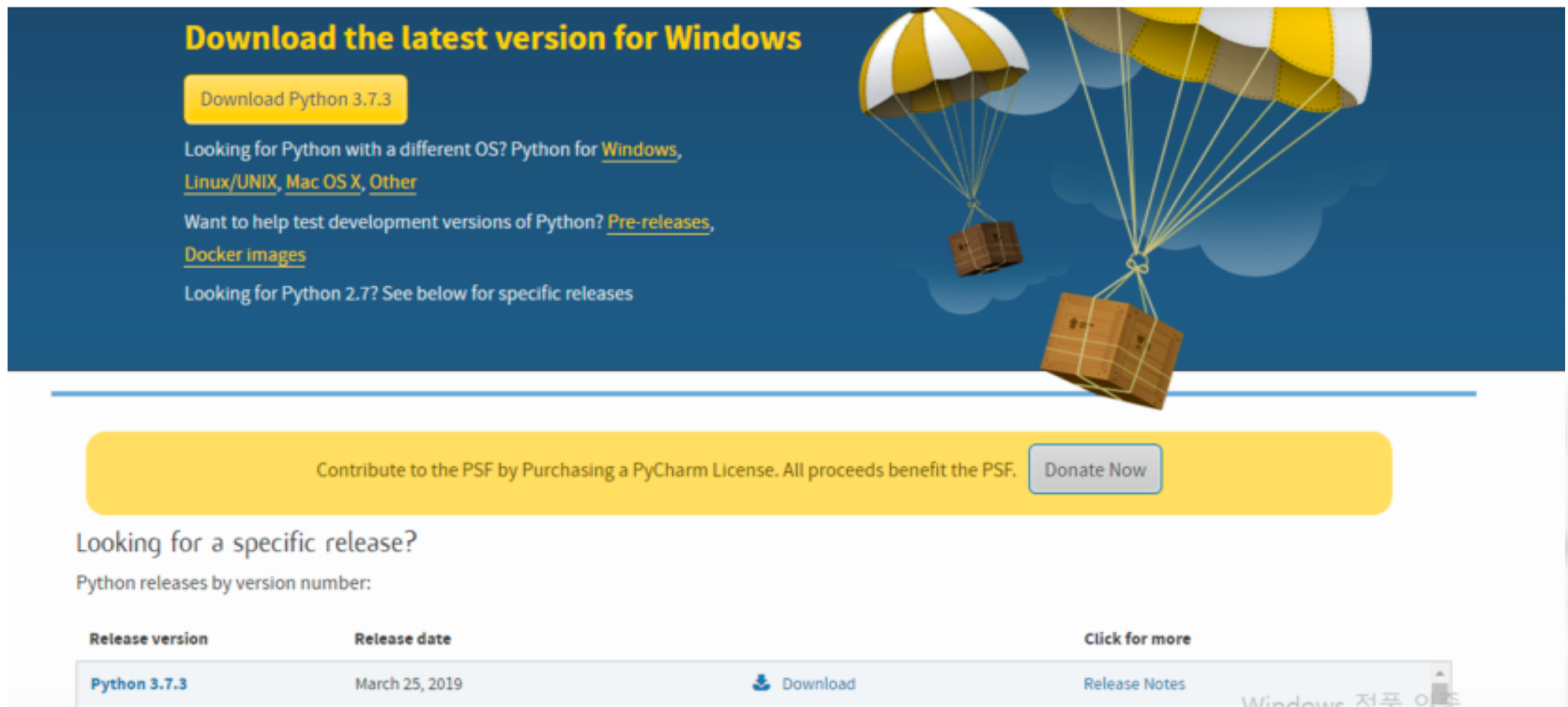


python

❖ 개발 환경

✓ python Interpreter 설치

- python 기본 인터프리터: <https://www.python.org/downloads/>



Download the latest version for Windows

[Download Python 3.7.3](#)

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [Mac OS X](#), [Other](#)

Want to help test development versions of Python? [Pre-releases](#), [Docker images](#)

Looking for Python 2.7? See below for specific releases

Contribute to the PSF by Purchasing a PyCharm License. All proceeds benefit the PSF. [Donate Now](#)

Looking for a specific release?

Python releases by version number:

Release version	Release date	Click for more
Python 3.7.3	March 25, 2019	Download Release Notes

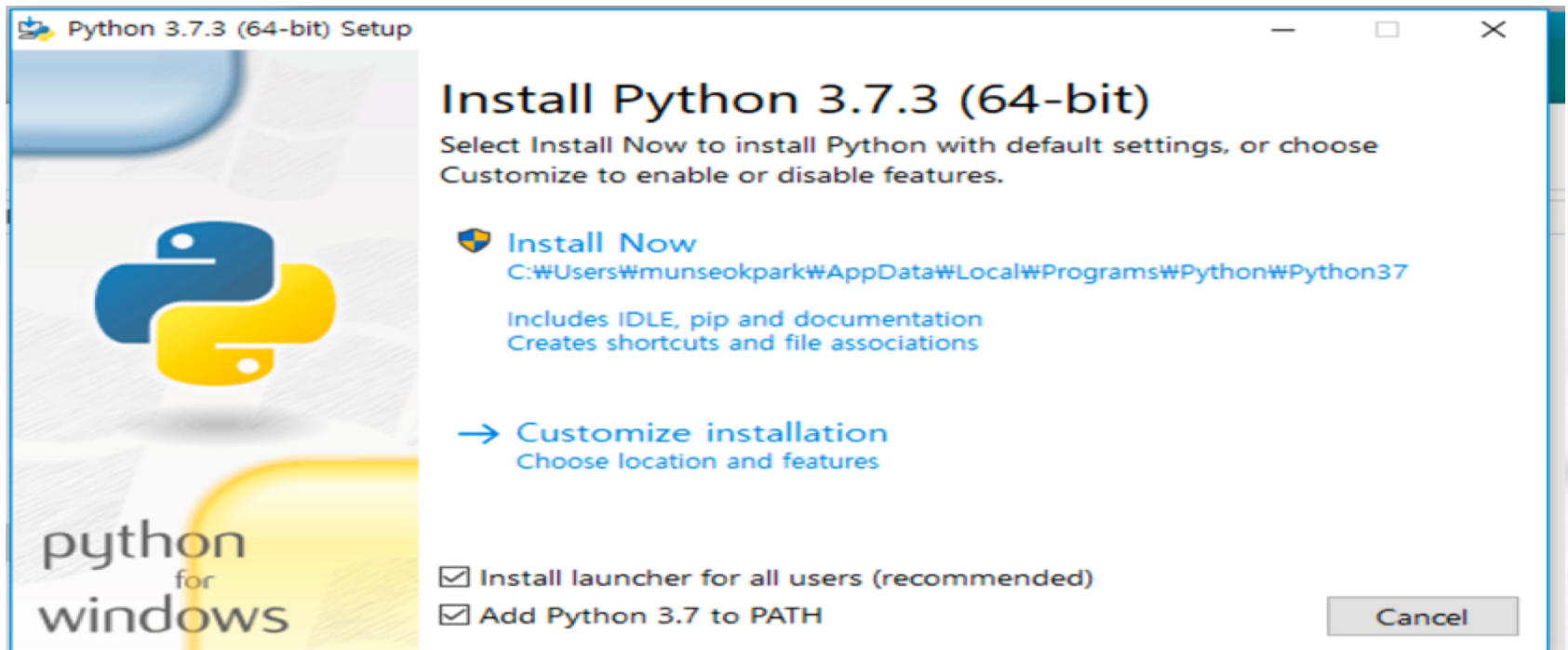
python

❖ 개발 환경

✓ python Interpreter 설치

- python 기본 인터프리터: <https://www.python.org/downloads/>

◆ Windows 에서는 설치할 때 path(명령어를 위치에 상관없이 사용하기 위한 설정)에 python 설치 경로를 추가해주는 옵션을 선택하는 것이 개발에 편리



python

❖ 개발 환경

✓ python Interpreter 설치

- python 기본 인터프리터: <https://www.python.org/downloads/>

◆ 리눅스에서 설치

- 리눅스에서는 종류에 따라 python2 나 python3가 설치가 되어 있음
- python3 설치 여부는 명령 프롬프트에서 아래 명령 실행
\$ python3
- 설치가 안된 경우 아래 명령을 이용해서 설치(Cent OS)
\$ sudo yum install -y epel-release
\$ sudo yum install -y python3*
- Ubuntu 는 sudo apt-get install python3-tk

python

❖ 개발 환경

✓ python Interpreter 설치

- python 기본 인터프리터: <https://www.python.org/downloads/>

◆ 설치 확인: 윈도우는 cmd 나 power shell 맥이나 리눅스에서는 터미널에서 확인 –
python --version

```
[(base) adam@ADAMui-MacBookPro ~ % python --version  
Python 3.8.8  
(base) adam@ADAMui-MacBookPro ~ % █
```

python

❖ 개발 환경

✓ python Interpreter 설치

● anaconda

- ◆ python 배포판으로 Ipython, Jupyter notebook, Qt Console 등 을 포함
- ◆ 데이터 분석, 수학, 과학 관련 다양한 라이브러리 포함
- ◆ python 인터프리터 와 라이브러리를 같이 설치함
- ◆ <https://www.anaconda.com/download/>

Get Superpowers with Anaconda

Anaconda is the leading open data science platform powered by Python. The open source version of Anaconda is a high performance distribution of Python and R and includes over 100 of the most popular [Python](#), R and Scala packages for data science. Additionally, you'll have access to over 720 packages that can easily be installed with conda, our renowned package, dependency and environment manager, that is included in Anaconda. Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution. See [the packages included with Anaconda](#) and [the Anaconda changelog](#).

Which version should I download and install?

Because Anaconda includes installers for Python 2.7 and 3.5, either is fine. Using either version, you can use Python 3.4 with the conda command. You can create a 3.5 environment with the conda command if you've downloaded 2.7 — and vice versa.

If you don't have time or disk space for the entire distribution, try [Miniconda](#), which contains only conda and Python. Then install just the individual packages you want through the conda command.

Anaconda for Windows

PYTHON 2.7	PYTHON 3.5
WINDOWS 64-BIT GRAPHICAL INSTALLER 342M	WINDOWS 64-BIT GRAPHICAL INSTALLER 382M
Windows 32-bit Graphical Installer 280M	Windows 32-bit Graphical Installer 280M

Behind a firewall? Use these [zipped Windows installers](#).

cloudera

GET NOW

VIEW OUR
WEBINARS

See upcoming and previous
webinars from our experts.

VIEW NOW

python

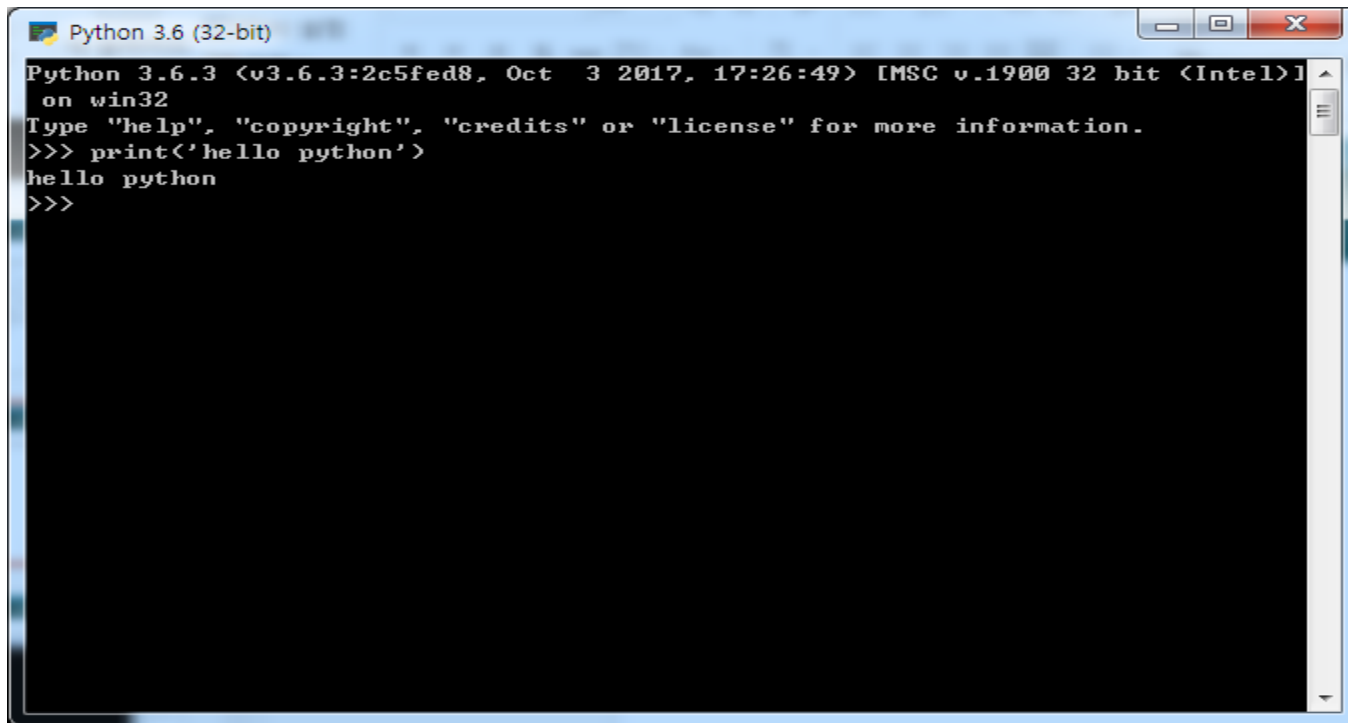
❖ 개발 환경

- ✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구
 - python console
 - python IDLE
 - ipython: 아나콘다를 설치하면 같이 설치
 - jupyter notebook: 아나콘다를 설치하면 같이 설치
 - spyder: 아나콘다를 설치하면 같이 설치
 - Eclipse: PyDev 플러그 인을 설치
 - Visual Studio의 python 도구
 - PyCharm
 - VSCode
 - Google Colab 과 같은 클라우드 플랫폼

python

❖ 개발 환경

- ✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구
 - Python Console: CMD 나 Terminal에서 python 이라고 입력하고 Enter를 눌러서 실행시켜서 줄 단위로 실행

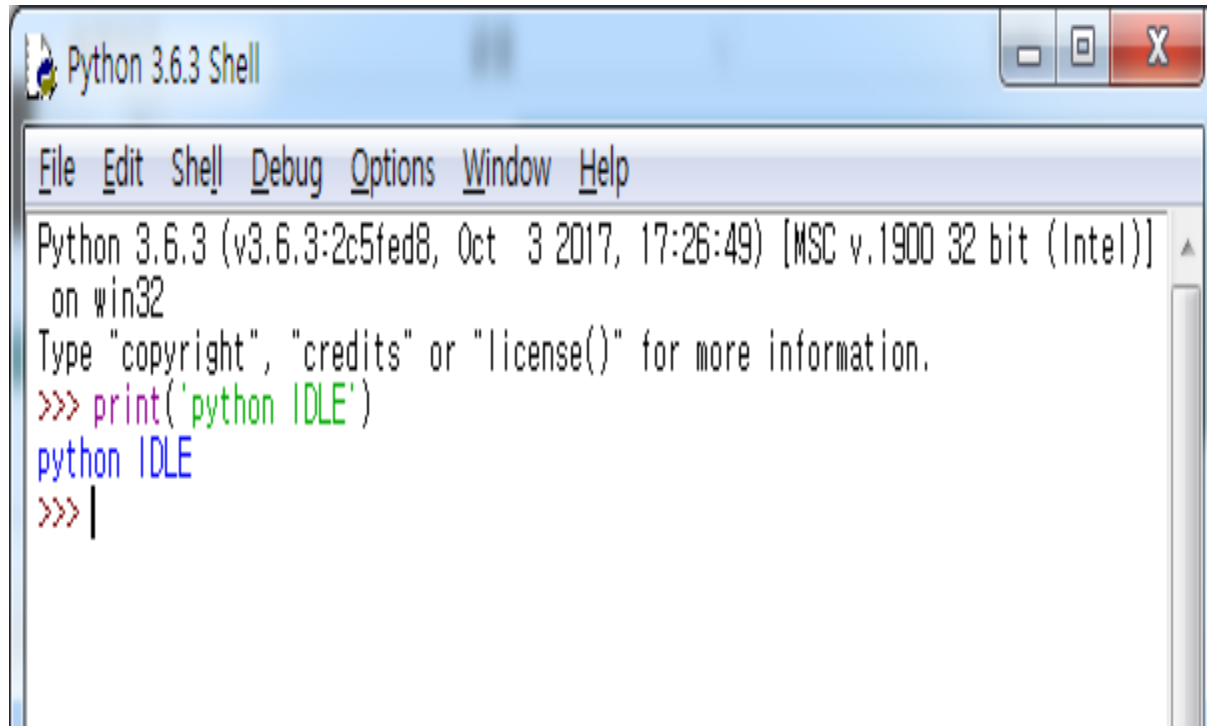


```
Python 3.6 (32-bit)
Python 3.6.3 (<v3.6.3:2c5fed8, Oct 3 2017, 17:26:49> [MSC v.1900 32 bit <Intel>])
on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('hello python')
hello python
>>>
```

python

❖ 개발 환경

- ✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구
 - Python IDLE



```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('python IDLE')
python IDLE
>>> |
```

python

❖ 개발 환경

- ✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구
 - Jupyter Notebook
 - ◆ 브라우저 기반이라서 인터페이스가 친숙
 - ◆ 보고서 작성이 쉬움
 - ◆ 여러가지 프로그래밍 언어를 지원: python, R, Scala 등
 - ◆ 지원하는 언어 확인: <https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>
 - ◆ email, dropbox, github 에 공유 가능
 - ◆ Code를 image, Video, Latex 그리고 Javascript로 생성 가능
 - ◆ Apache Spark 와 함께 사용 가능
 - ◆ 코드를 중간 중간 실행해 둔 상태에서 다른 코드를 추가해서 실행 가능
 - ◆ 그래프를 화면 하단에 출력

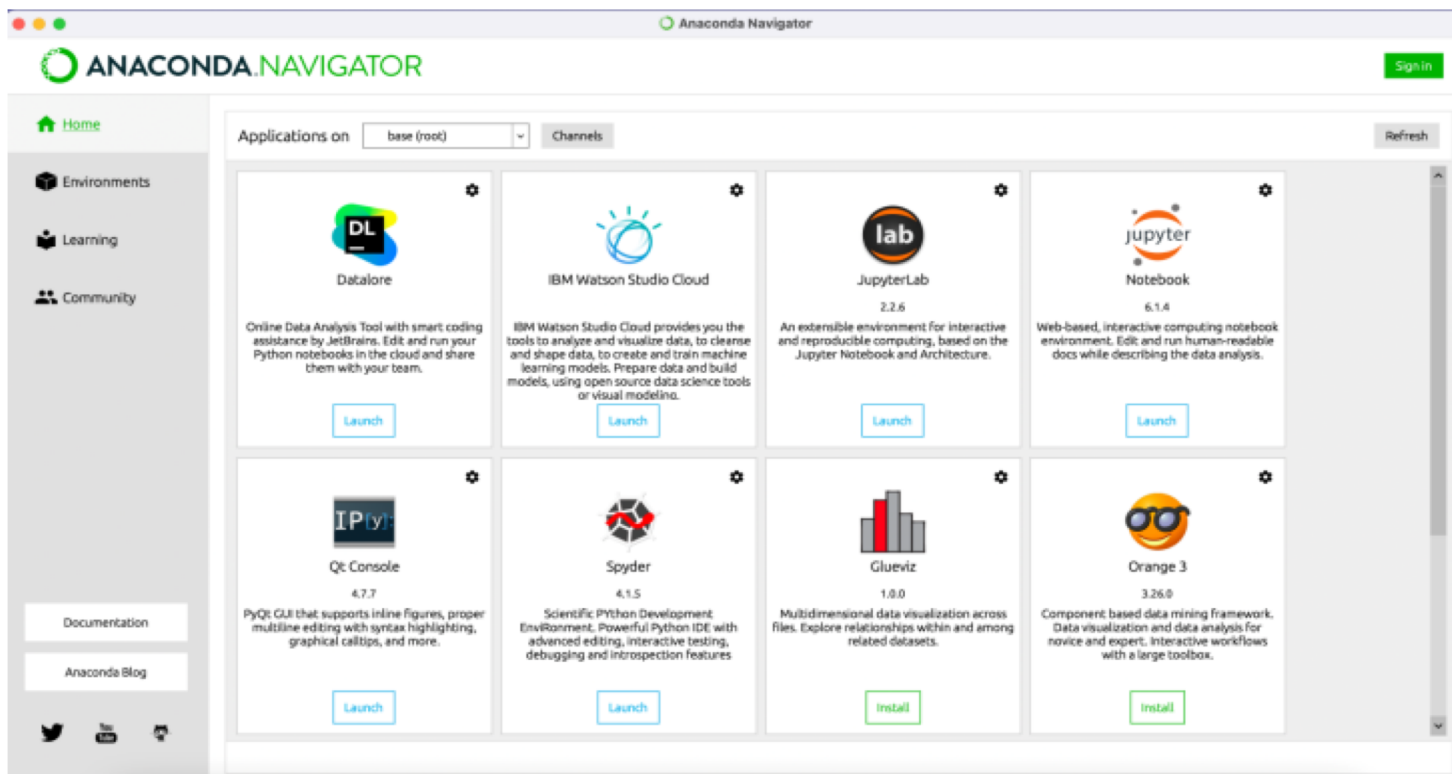
python

❖ 개발 환경

✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구

- Jupyter Notebook

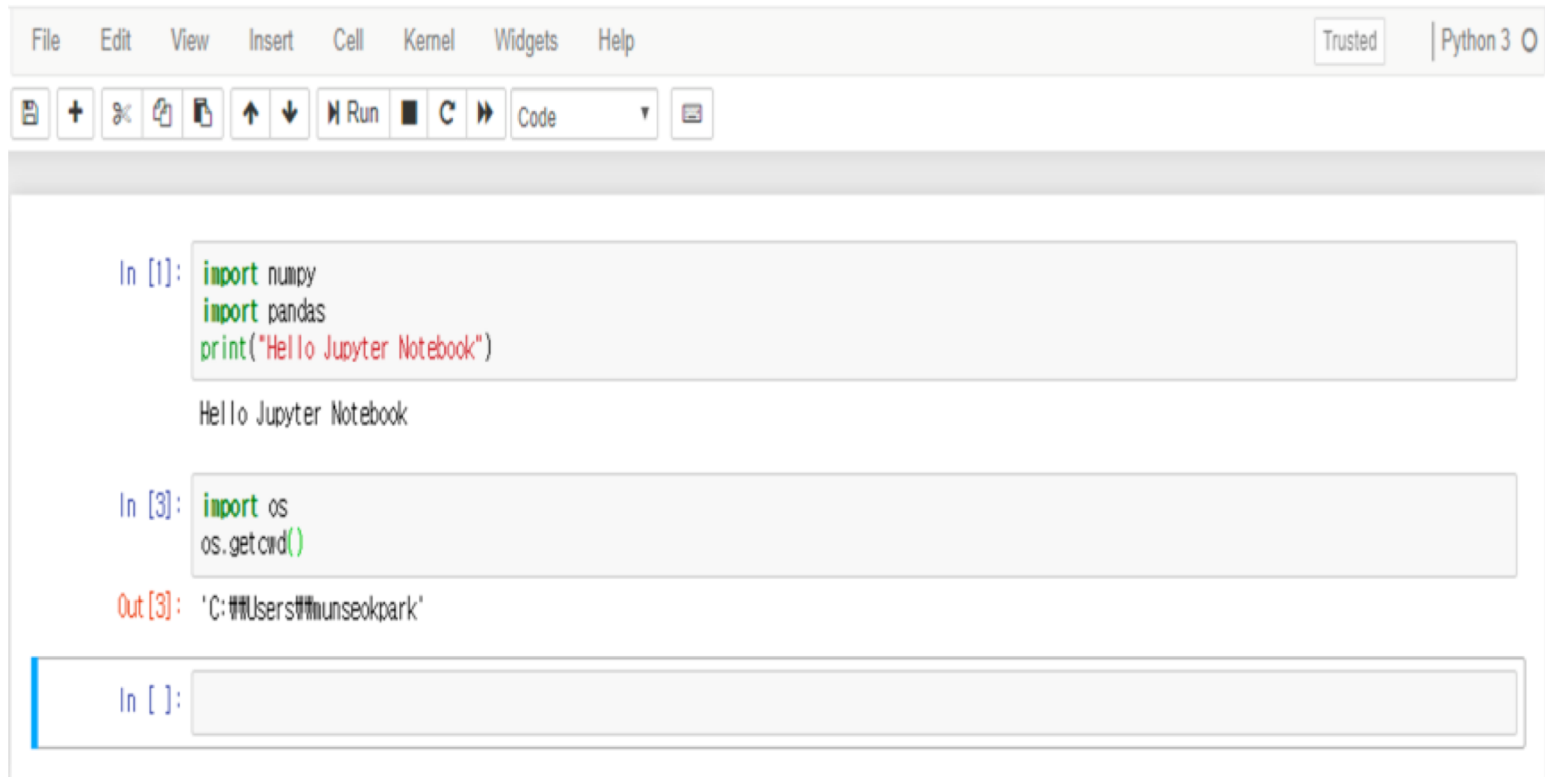
- ◆ 실행: Anaconda Navigator



python

❖ 개발 환경

- ✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구
 - Jupyter Notebook



python

❖ 개발 환경

- ✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구
 - spyder
 - ◆ anaconda를 설치하면 자동 설치
 - ◆ Code Sense가 동작하기 때문에 초보자가 작성하기에 편리
 - ◆ 코딩하는 창과 결과를 보여지는 창이 다름
 - ◆ 오른쪽의 창에서 만들어진 변수 확인 가능
 - ◆ 코드를 길게 작성하는 경우 편리

python

❖ 개발 환경

- ✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구
 - spyder

The screenshot displays the Spyder Python IDE interface. On the left, a code editor window titled 'temp.py' contains the following Python code:

```
1 import matplotlib.pyplot as plt
2 ar = [1,2,3,4]
3 plt.plot(ar)
4 plt.ylabel('some numbers')
5 plt.show()
6
```

On the right, the 'Variable explorer' panel shows a table with the following data:

Name	Type	Size	Value
ar	list	4	[1, 2, 3, 4]

Below the variable explorer, the 'IPython console' panel shows the execution output:

```
Python 3.7.0 (default, Jun 28 2018, 07:39:16)
Type "copyright", "credits" or "license" for more information.

IPython 7.6.1 -- An enhanced Interactive Python.

In [1]: runfile('/Users/munseokpark/.spyder-py3/temp.py', wdir='/Users/munseokpark/.spyder-py3')
```

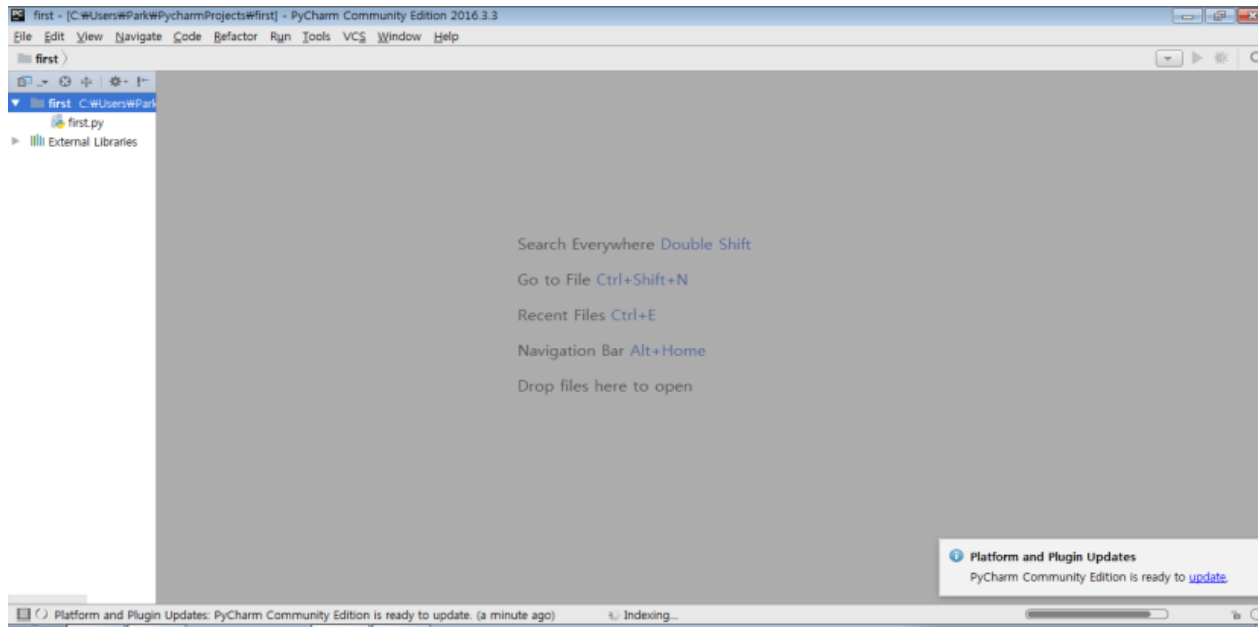
The console also displays a line plot with the y-axis labeled 'some numbers' and the x-axis ranging from 0.0 to 3.0. The plot shows a blue line starting at (0, 1) and ending at (3, 4).

Below the plot, the console shows the prompt 'In [2]:'.

python

❖ 개발 환경

- ✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구
 - PyCharm
 - ◆ IntelliJ를 제공하는 JetBrains에서 배포하는 프로그램
 - ◆ <https://www.jetbrains.com/pycharm/> 에서 Community 버전을 선택해서 다운로드 후 설치

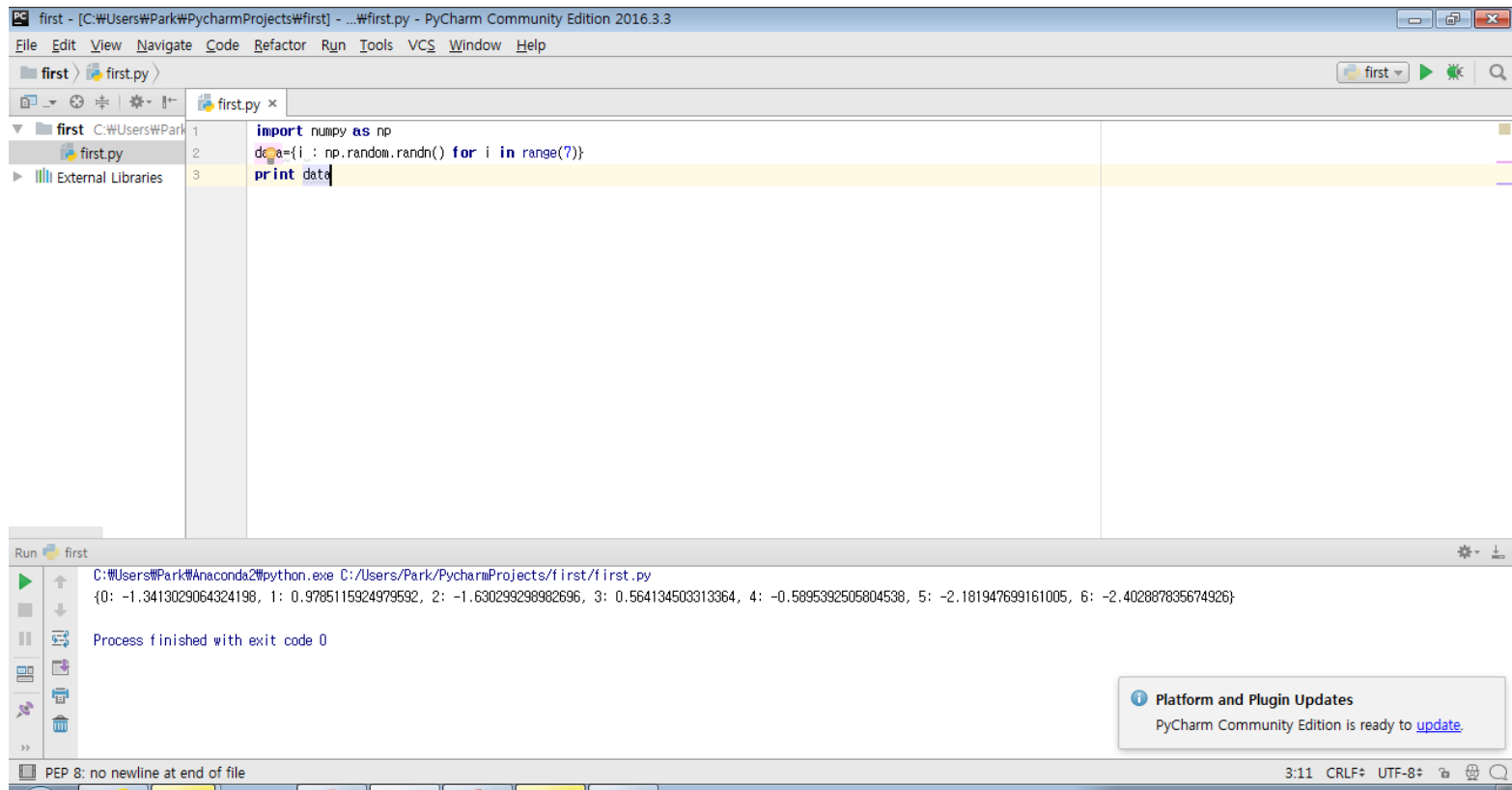


python

❖ 개발 환경

✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구

● PyCharm



python

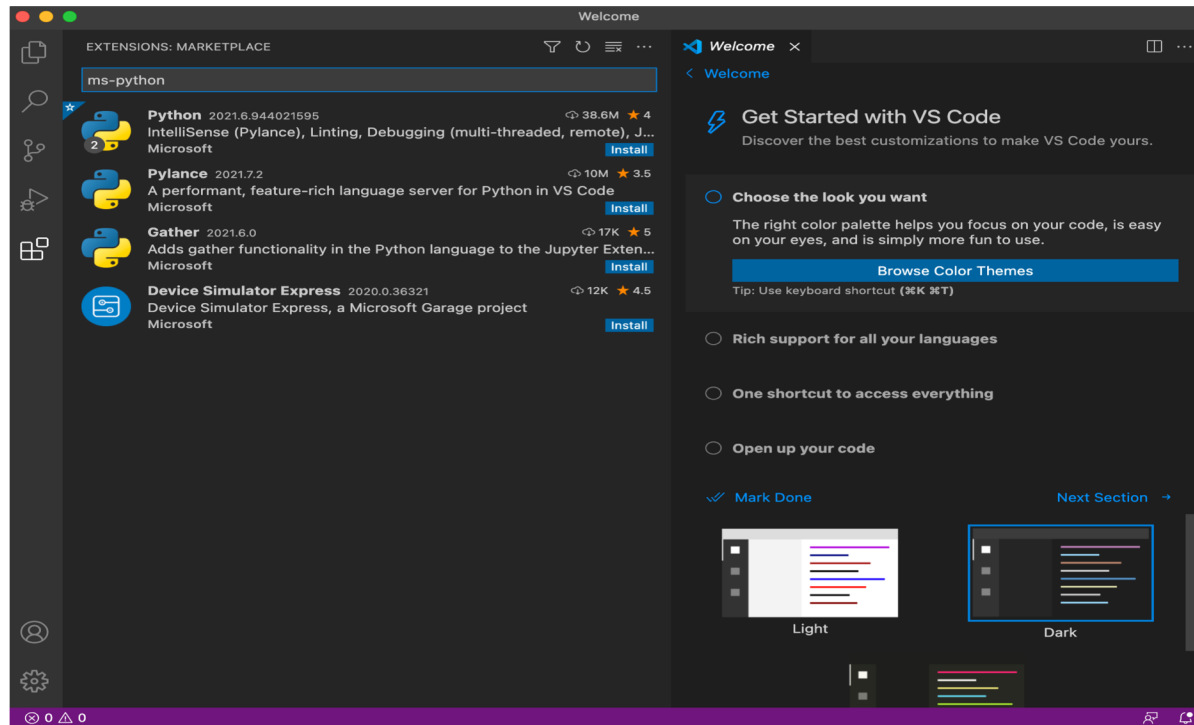
❖ 개발 환경

✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구

- VS-Code

- ◆ <https://code.visualstudio.com/download> 에서 다운로드

- ◆ 설치 한 후 실행을 하고 왼쪽의 아이콘 중에서 Extension을 눌러서 ms-python을 검색해서 설치

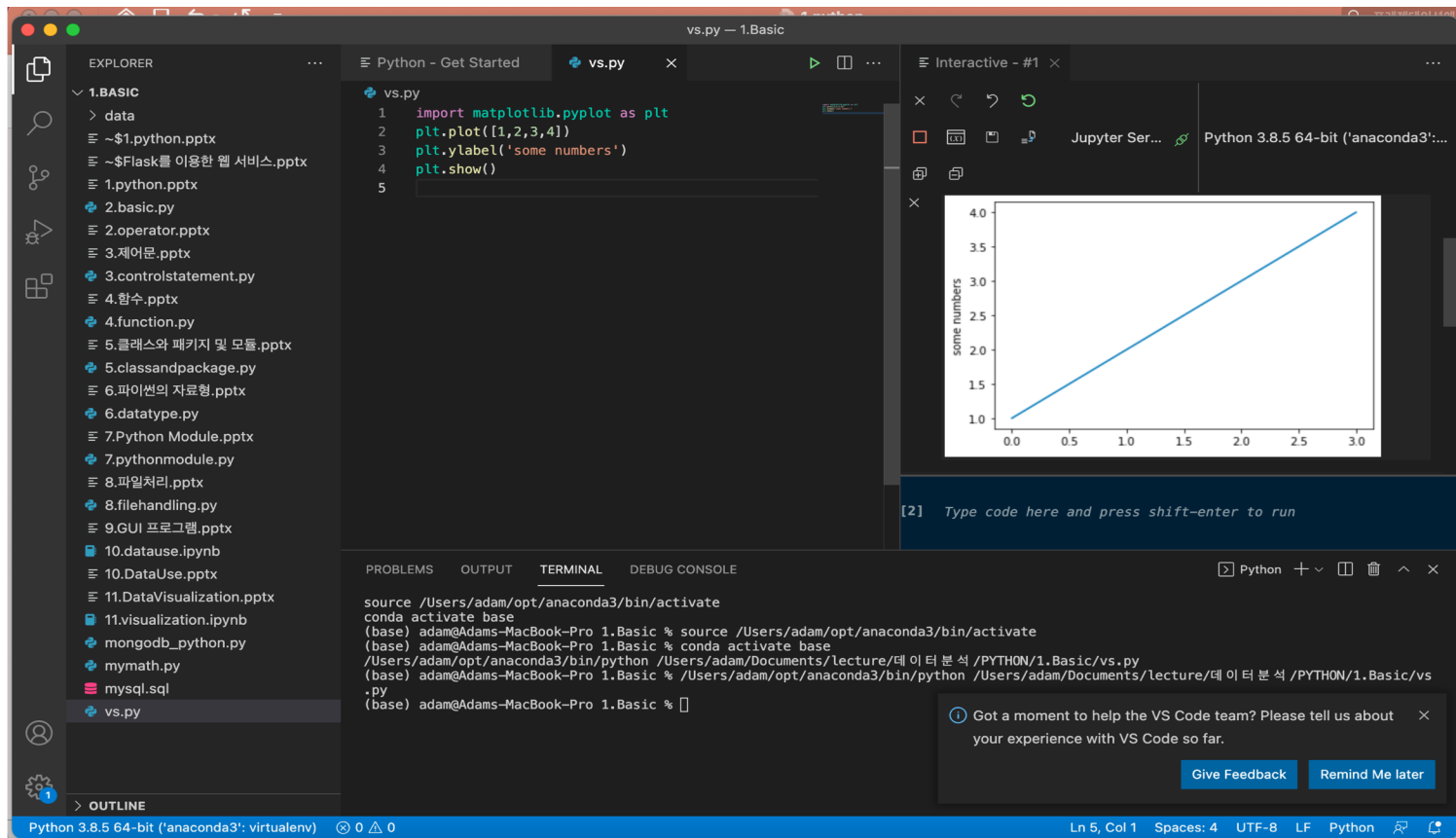


python

❖ 개발 환경

✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구

● VS-Code



python

❖ 개발 환경

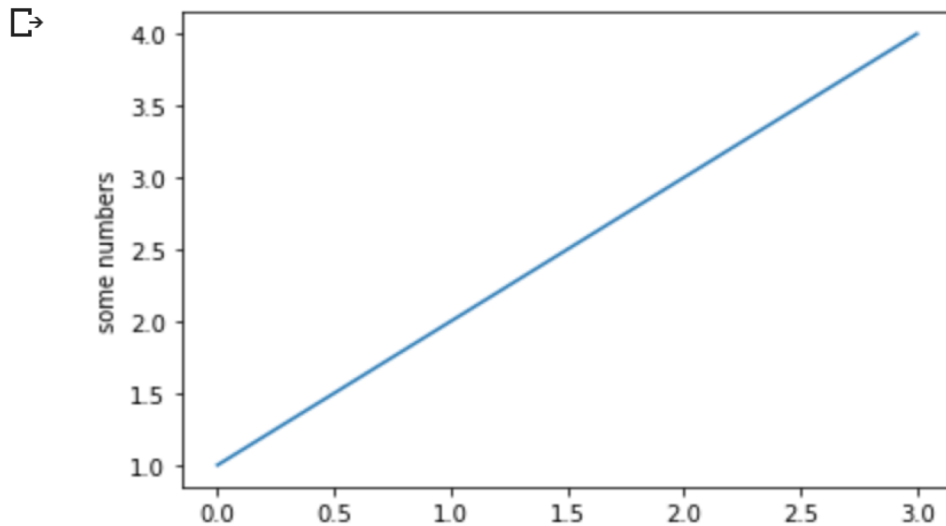
- ✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구
 - Colab
 - ◆ Colab은 클라우드 기반의 Jupyter 노트북 개발 환경
 - ◆ 내부적으로는 코랩 + 구글드라이브 + 도커 + 리눅스 + 구글 클라우드 기술 스택으로 이루어진 것으로 알려져 있음
 - ◆ <https://colab.research.google.com/notebooks/intro.ipynb?hl=ko#recent=true>
 - ◆ python을 설치할 필요 없고 IDE도 설치할 필요 없음
 - ◆ GPU를 사용할 수 있음
 - ◆ 로컬의 데이터를 호출할 때는 contents 디렉토리에 업로드 해야 함
 - ◆ 다른 패키지를 설치할 때 나 데이터를 다운로드 받을 때 ubuntu 명령을 이용해야 함

python

❖ 개발 환경

- ✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구
 - Colab

```
▶ import matplotlib.pyplot as plt  
plt.plot([1,2,3,4])  
plt.ylabel('some numbers')  
plt.show()
```



python

❖ 개발 환경

✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구

- Colab

- ◆ 현재 작업 디렉토리 확인

- #현재 작업 디렉토리 확인

- import os

- os.getcwd()

- 'content'

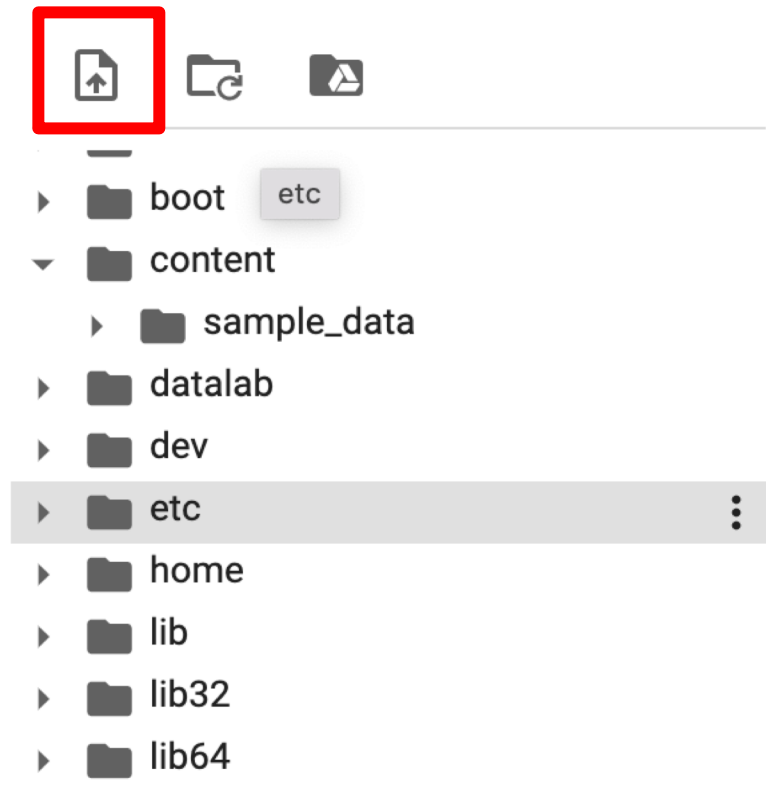
python

❖ 개발 환경

✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구

- Colab

- ◆ 로컬 파일 업로드



python

❖ 개발 환경

✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구

- Colab

- ◆ 로컬 파일 업로드

```
from google.colab import files  
file_uploaded = files.upload()
```

파일 선택 선택된 파일 없음

Cancel upload

파일 선택 korea.png

- **korea.png**(image/png) - 52171 bytes, last modified: 2010. 12. 16. - 100% done
Saving korea.png to korea.png

python

❖ 개발 환경

✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구

- Colab

- ◆ 구글 드라이브 공유

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

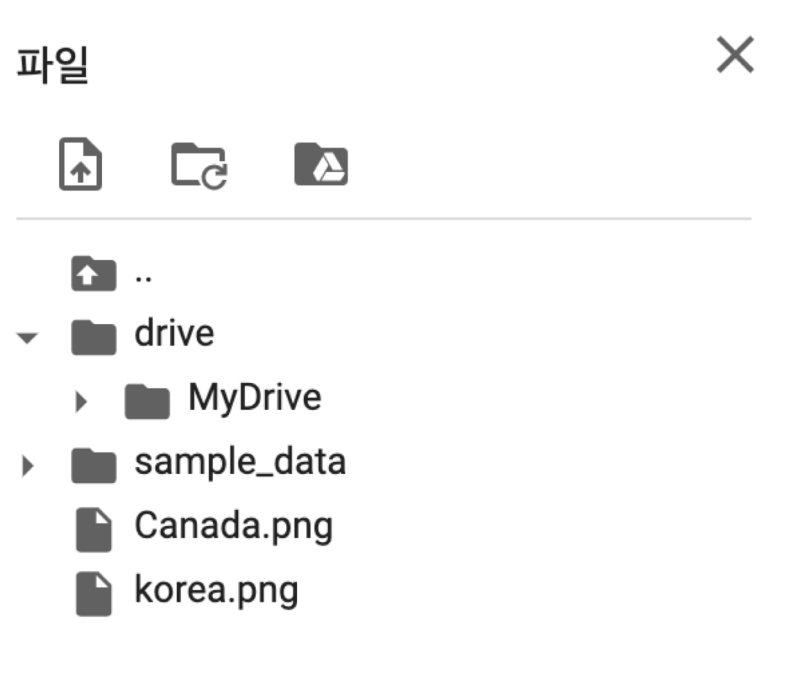
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0

Enter your authorization code:

python

❖ 개발 환경

- ✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구
 - Colab
 - ◆ 구글 드라이브 공유



python

❖ 개발 환경

✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구

- Colab

- ◆ 패키지 영구적 설치

```
import os, sys
from google.colab import drive
drive.mount('/content/drive')
```

```
pg_path = '/content/packages'
```

```
os.symlink('/content/drive/My Drive/Colab Notebooks', pg_path)
sys.path.insert(0, pg_path)
```

python

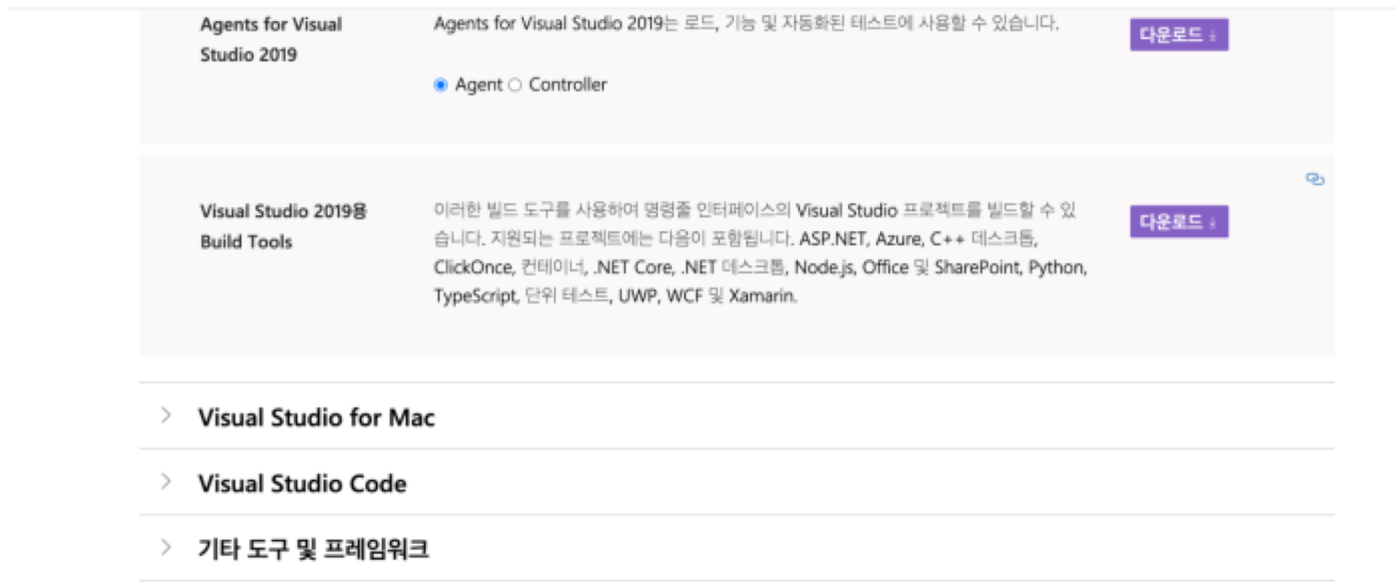
❖ 개발 환경

- ✓ IDE(Integrated Development Environment - 통합 개발 환경) – 코딩과 컴파일 및 실행 도구
 - Colab
 - ◆ 패키지 영구적 설치
 - `!pip install --target=$pg_path selenium`

python

❖ 개발 환경

- ✓ Windows – Visual Studio 재배포 패키지 설치
 - 윈도우 환경에서 서드파티 패키지를 설치할 때 Microsoft Visual Studio Build Tools가 필요한 경우가 있음
 - <https://visualstudio.microsoft.com/ko/downloads/> 에 접속해서 Visual Studio 2019용 Build Tools 를 클릭해서 다운로드 받은 후 설치



python

❖ 개발 환경

- ✓ 자연어 처리를 위해서는 JDK가 필요한 경우가 있음



python

❖ 작업 디렉토리 설정

✓ 작업 디렉토리 확인

```
In [1]: import os
```

```
In [2]: os.getcwd()
```

```
Out[2]: 'C:\Users\WWPark'
```

✓ 작업 디렉토리 변경

```
In [3]: os.chdir('c:\python')
```

python

❖ 가상 환경

✓ 개요

- Python 가상 환경은 Python 프로젝트를 진행할 때 독립된 환경을 만들어 주는 도구
- Python 개발자 A가 2개의 Python 프로젝트를 개발하고 관리한다고 가정했을 때 Python 프로젝트를 각각 P-1, P-2 라고 한 경우 P-1, P-2에 필요한 Python 또는 Python 라이브러리의 버전이 다를 수 있는데 P-1에는 Python 2.7 버전을 P-2에는 Python 3.8 버전이 필요할 수 있는데 이때 하나의 데스크톱에 서로 다른 버전의 Python을 설치해야 하는 문제가 발생
- 서로 다른 구성을 가진 개발 환경은 구축하기도 어렵고 사용하기에도 힘든데 가상 환경이 없던 예전에는 그런 고생을 감수할 수밖에 없었지만 Python 가상 환경을 이용하면 하나의 데스크톱 안에 독립된 가상 환경을 여러 개 만들 수 있는데 프로젝트 P-1을 위해 가상 환경 V-1을 만든 다음 Python 2.7 버전을 설치하고 프로젝트 P-2를 위해 가상 환경 V-2를 만든 다음 Python 3.8버전을 설치해서 사용할 수 있음

✓ 가상 환경 생성

`python -m venv 가상환경이름`

- 가상 환경 이름으로 디렉토리가 생성됨

python

❖ 가상 환경

✓ 가상 환경 진입 과 해제

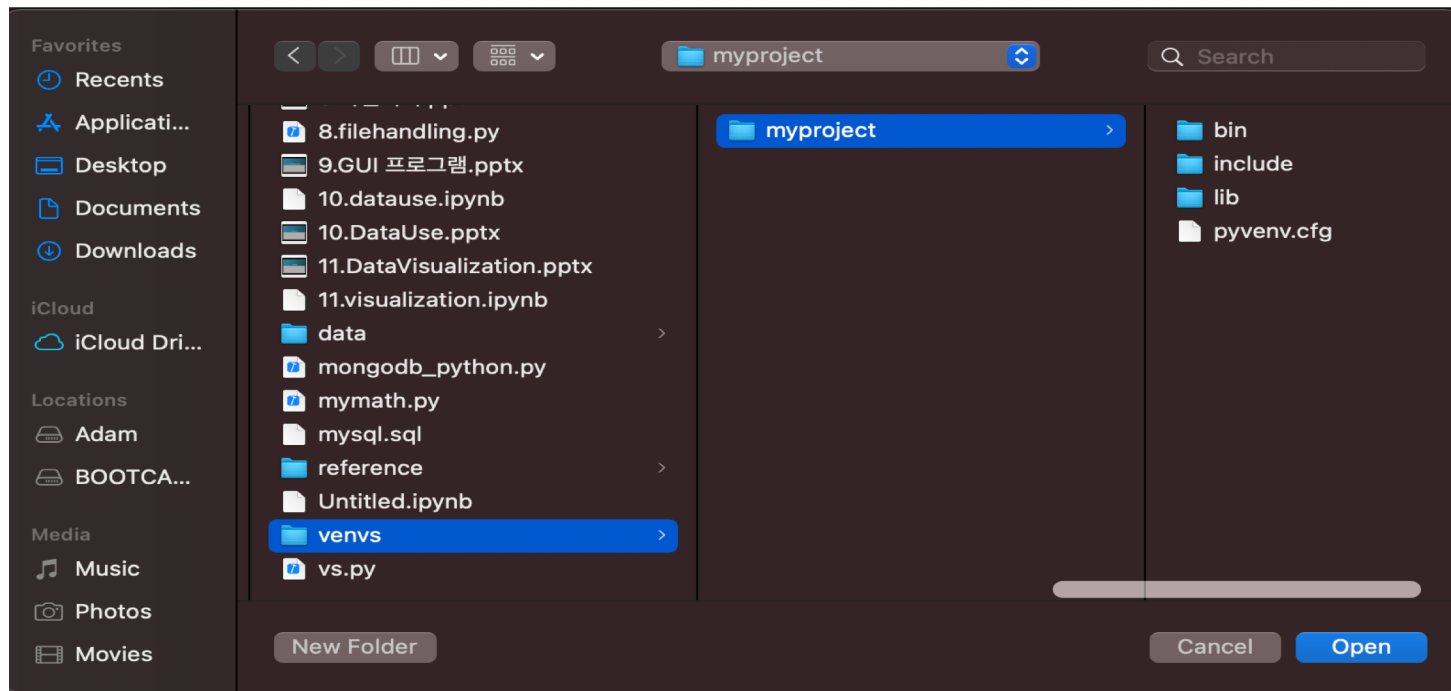
- 가상 환경 진입(사용): 가상 환경에 진입하려면 가상 환경에 있는 Scripts(bin) 디렉터리의 activate라는 명령을 수행
- 가상 환경 벗어나기: 가상 환경에 진입하려면 가상 환경에 있는 Scripts(bin) 디렉터리의 deactivate라는 명령을 수행

python

❖ 가상 환경

✓ PyCharm에서 가상 환경 사용

- [File] - [Open] 메뉴를 실행시켜서 이전에 생성한 프로젝트를 선택

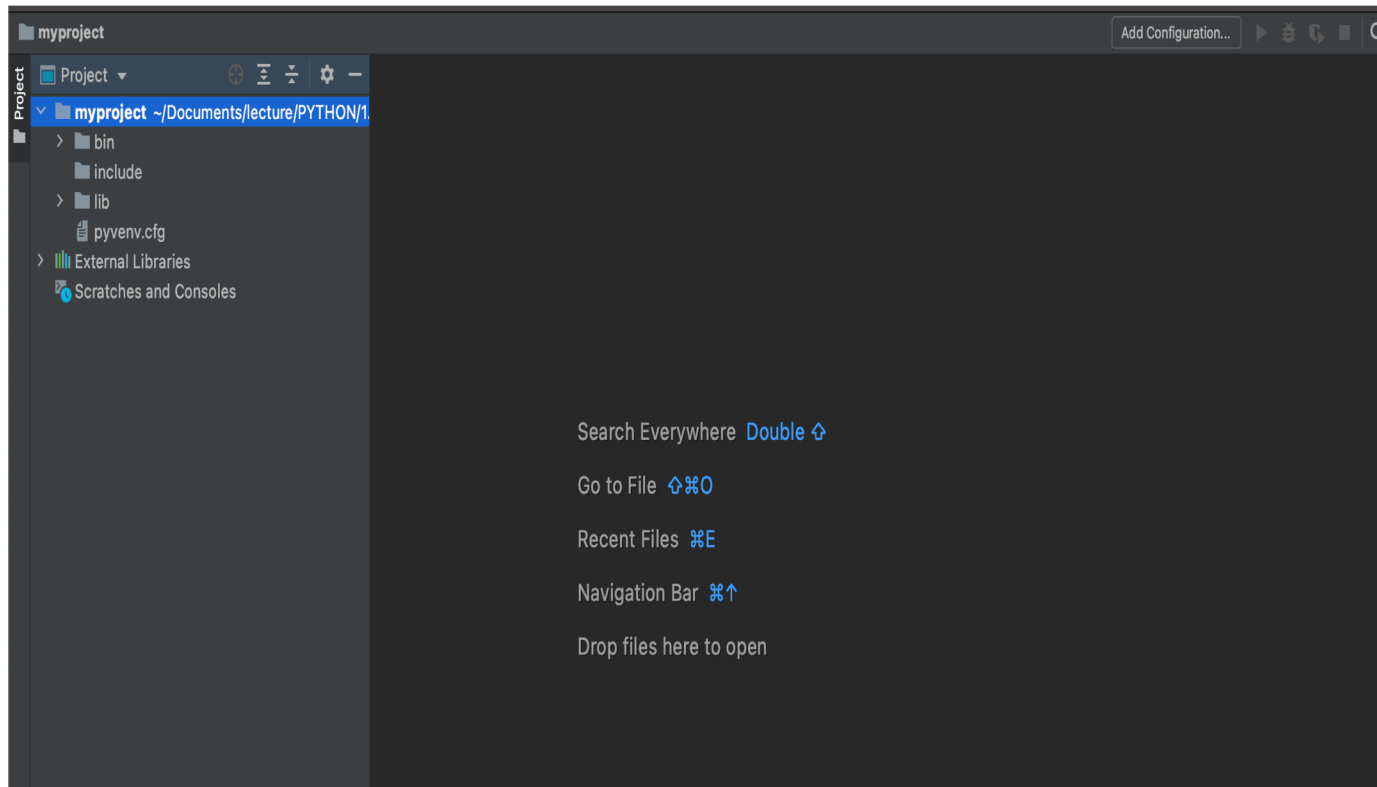


python

❖ 가상 환경

✓ PyCharm에서 가상 환경 사용

- [File] – [Open] 메뉴를 실행시켜서 이전에 생성한 프로젝트를 선택

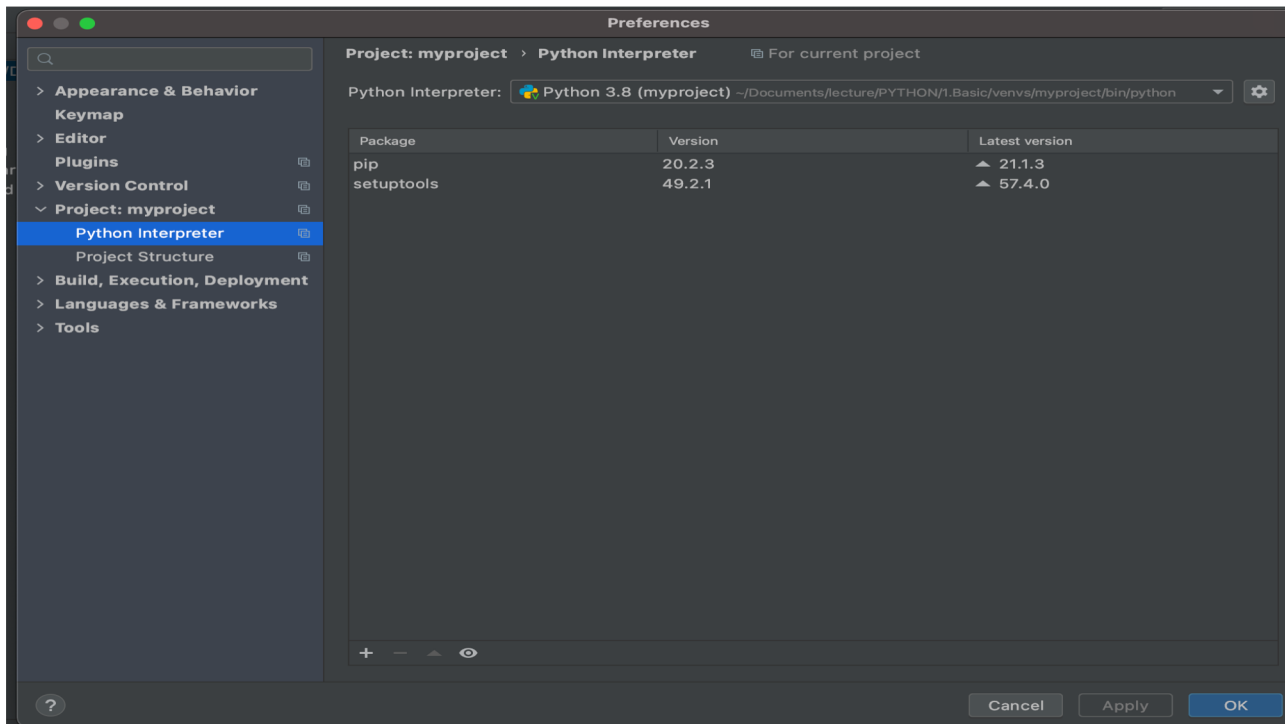


python

❖ 가상 환경

✓ PyCharm에서 가상 환경 사용

- [PyCharm] – [Preference] ([File] – [Settings])메뉴를 실행시켜서 Python 인터프리터를 설정



python

❖ 가상 환경

✓ PyCharm에서 가상 환경 사용

- [File] – [New] – [Python File]을 선택하고 내용을 작성하고 실행

