



# Round 2

PRESS  
START



## << Round 2 >>

- 가상환경
- Jupyter Lab
- Python 기초



New Assignment



## << Round 2 >>

- 가상환경 <<
- Jupyter Lab
- Python 기초



Let's  
Go

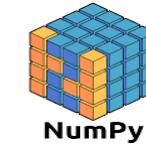




Data Science Project\_1 =



+



NumPy



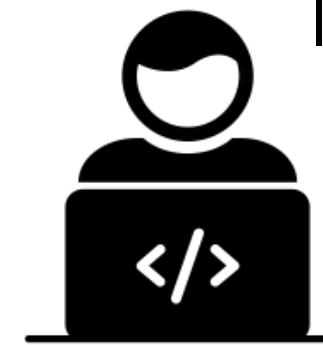
SciPy



Pandas



Seaborn



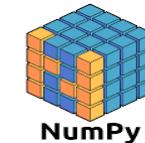
It's OK, but...



Data Science Project\_1 =



+



NumPy



SciPy



Pandas



Seaborn

Deep Learning Project\_1 =



+



TensorFlow



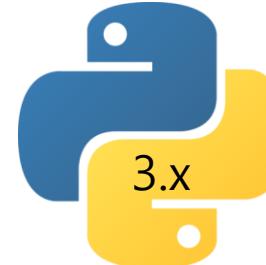
matplotlib

Data Science Project\_2 =

.

.

.



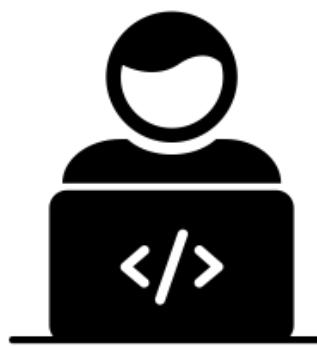
+



BeautifulSoup



Keras



So sad...



# ANACONDA®

- Anaconda는 파이썬의 배포판
- 특정 용도의 소프트웨어 패키지를 묶어서 제공
- 각종 가상환경 구축 가능





# ANACONDA®



가상 환경 1



2.7



Pandas



가상 환경 2



3.7



NumPy



Keras



가상 환경 3



2.7



NumPy



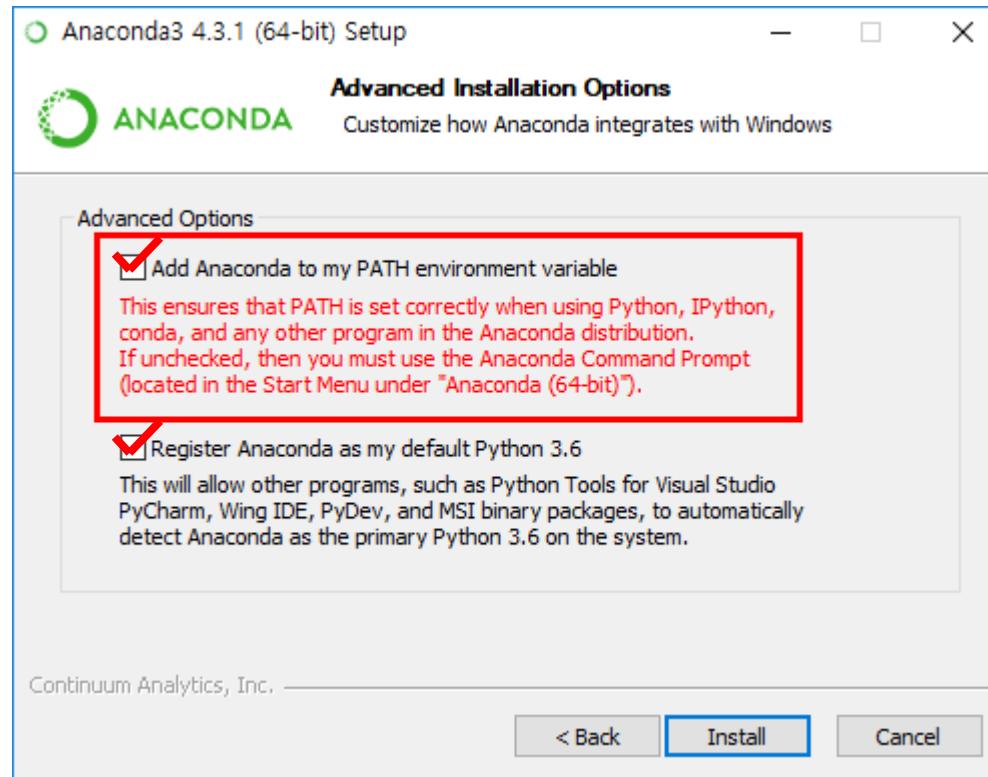
BeautifulSoup

...



1. Go to  
<https://www.anaconda.com/distribution/#download-section>

2. check this box





### 3. how to check my Anaconda is installed well?

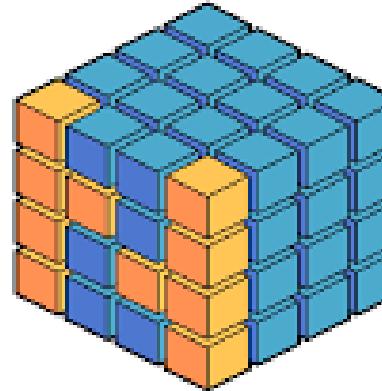
```
C:\ 명령 프롬프트  
Microsoft Windows [Version 10.0.18363.592]  
(c) 2019 Microsoft Corporation. All rights reserved.  
C:\#Users#\user>conda --version  
conda 4.7.12  
C:\#Users#\user>
```

### 4. Creating virtual environments with conda

```
conda create --name test python=3.7
```

### 5. Activating the virtual environments

```
conda activate test / conda deactivate
```



**NumPy**

- 대표적인 파이썬 기반 선형대수 라이브러리
- 다양한 행렬과 배열 연산 함수 지원, 다양한 라이브러리에 기본 내장
- 데이터는 대부분 행렬의 형태를 띠기 때문에 데이터처리에 필수적

# matplotlib

- 자료를 차트나 플롯으로 시각화(visualization)하는 패키지
- 판다스와 연계되어 있으며, EDA를 위한 시각화에 필수적임.



- 다양한 분석과 머신러닝 학습에 사용될 수 있는 파이썬 라이브러리
  - numpy와 Scipy와 상호 운영되도록 디자인 됨



## Pandas

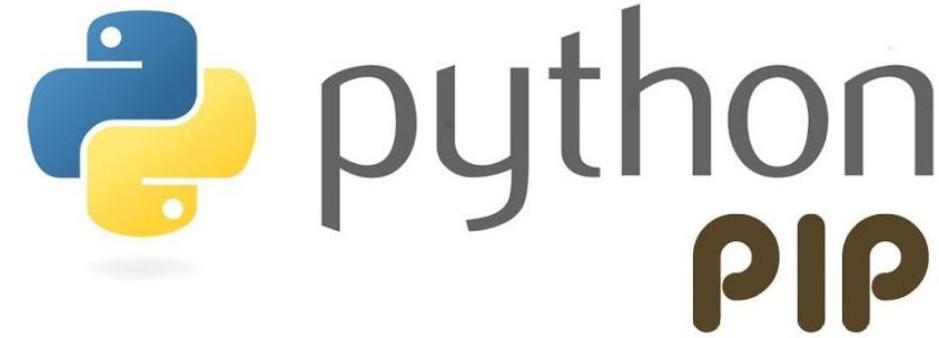


- 데이터 분석 및 조작을 위한 라이브러리
- R과 비슷한 자료형을 사용함(Dataframe)

이렇게 다양한 라이브러리를  
하나하나 관리해야 한다면...?



So sad...



- 파이썬 라이브러리 패키지를 손쉽게 관리 할 수 있게 해주는 시스템
  - 파이썬 3.4이후 버전부터 기본 탑재!
  - 짧은 명령어 한 줄로 손쉽게 설치 가능



## 1. 패키지를 설치해줄 가상환경 활성화

```
C:\Users\user>conda activate test
```

## 2. pip install numpy로 numpy 설치

```
(test) C:\Users\user>pip install numpy
Collecting numpy
  Using cached https://files.pythonhosted.org/packages/a9/3f/numpy-1.18.1-cp37m-win_amd64.whl
Installing collected packages: numpy
Successfully installed numpy-1.18.1
```

## 3. pip freeze로 현재 설치된 패키지 확인

```
(test) C:\Users\user>pip freeze
certifi==2019.11.28
numpy==1.18.1
wincertstore==0.2
```

## << Round 2 >>

- 가상환경
- Jupyter Lab <<
- Python 기초



Let's  
Go





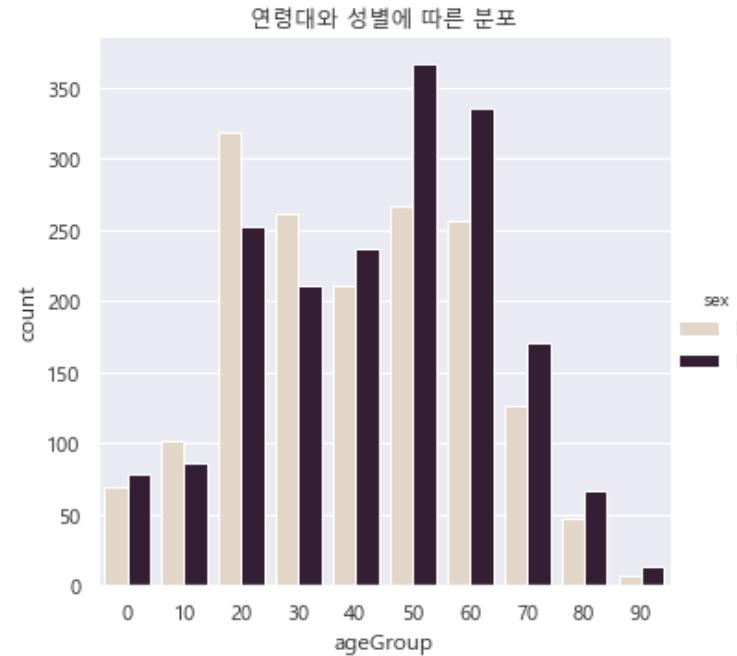
# Jupyter Lab?

- Visual studio, eclipse 같은 프로그래밍용 IDE
  - 웹 기반 IDE로 가볍고 가독성이 좋음
  - 파이썬 말고도 R과 같은 다양한 언어를 사용가능
  - Anaconda를 이용해서 다양한 가상환경에서 실행가능
- Google Colaboratory와 연동하여 클라우드 GPU 무료 이용가능

# Jupyter Lab의 장점

- 코드를 블록 단위로 실행시킬 수 있고 결과에 코멘트를 달 수 있음.

```
sns.catplot(x="ageGroup", kind="count", hue="sex", palette="ch:.25", data=cov_df)
plt.title("연령대와 성별에 따른 분포")
plt.show()
```



- 20대와 50대, 60대가 월등히 많았다.
- 내 생각엔 20대는 놀기 위해서 많이 돌아다녔기 때문에고, 50~60대는 교회 때문이 아닐지???



Applications on base (root) Channels

**JupyterLab** 1.1.4

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

**Launch**

**CMD.exe Prompt** 0.1.1

Run a cmd.exe terminal with your current environment from Navigator activated

**Launch**

**Notebook** 6.0.1

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

**Launch**

**Powershell Prompt** 0.0.1

Run a Powershell terminal with your current environment from Navigator activated

**Launch**

**Glueviz** 0.15.2

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

**Install**

**Orange 3** 3.26.0

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

**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**



Shift + Enter

The screenshot shows the Jupyter Notebook interface. At the top, there are two tabs: 'Notebook' and 'Untitled.ipynb'. The 'Untitled.ipynb' tab is active, indicated by a red box around its icon. A red arrow points from the 'Notebook' tab to the 'Untitled.ipynb' tab. Below the tabs is a toolbar with icons for file operations, a plus sign for creating new cells, and a play button for running cells. The play button is also highlighted with a red box. To the right of the toolbar is a dropdown menu labeled 'Code'. The main area displays a code cell [1]: `print("Hello world")`, which has been run and output 'Hello world'. A red box highlights the 'Shift + Enter' keyboard shortcut above the cell. In the bottom right corner of the notebook window, there is a red box highlighting the 'Markdown' dropdown menu.

Notebook

Untitled.ipynb

Python 3

Console

Python 3

Shift + Enter

[1]: `print("Hello world")`

Hello world

Untitled.ipynb

+

Markdown

[1]: `print("Hello world")`

Hello world

## 첫 쥬피터 노트북 사용하기

- Hello world 출력

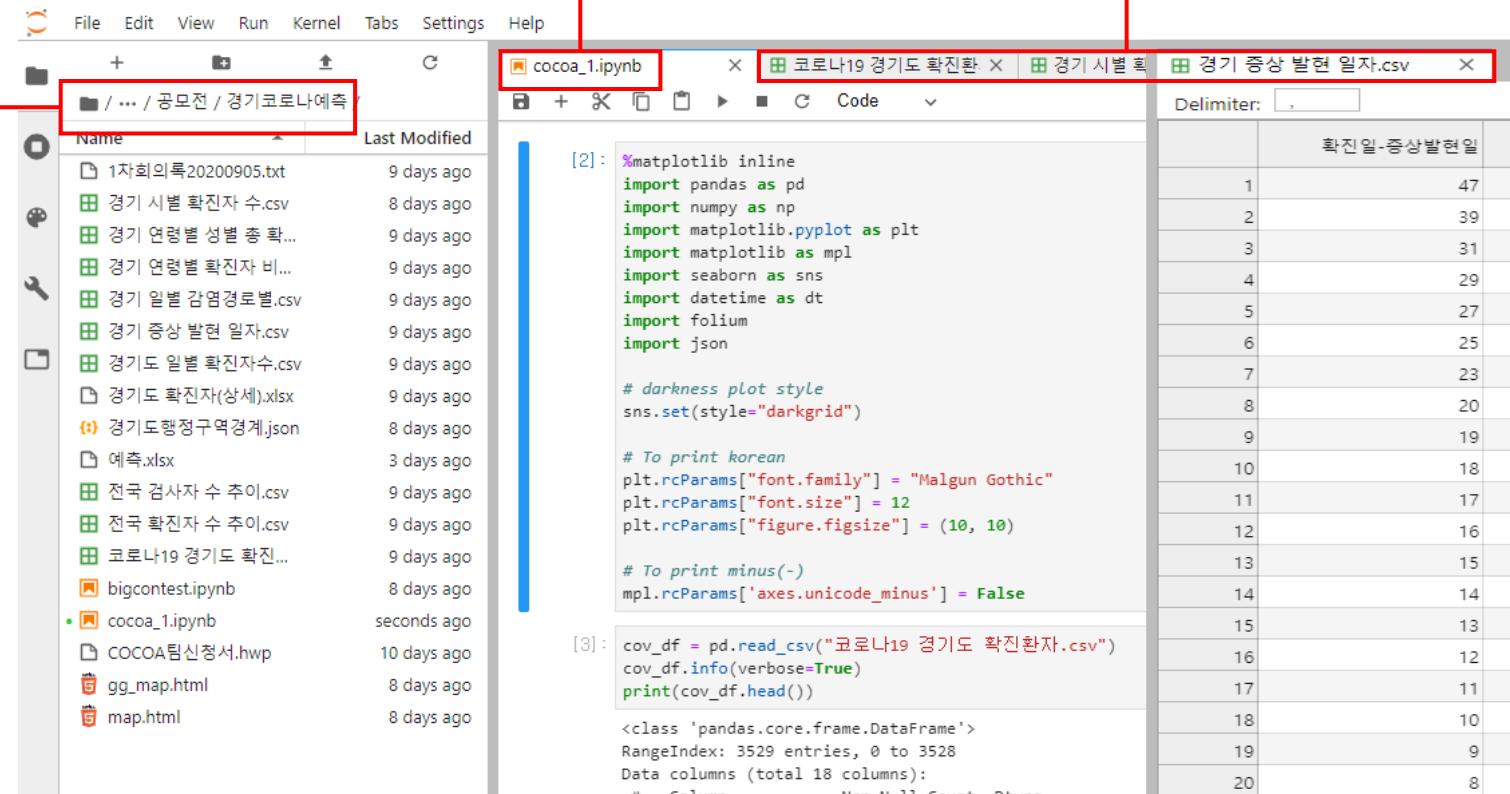
첫 쥬피터 노트북 사용하기

- Hello world 출력

Jupyter의 확장자는 .ipynb

여러 탭을 띄울 수 있고 여러 창을 띄울 수 있음

Directory



The screenshot shows the Jupyter Notebook environment. On the left, there is a file browser window titled 'Directory' with a red box around its title bar. It lists several files and folders, including 'cocoa\_1.ipynb' which is highlighted with a red box. The main area contains three tabs:

- [2]: A code cell containing Python code for importing various libraries like matplotlib, pandas, numpy, etc., and setting plot styles.
- [3]: A code cell showing the execution of a command to read a CSV file ('코로나19 경기도 확진환자.csv') into a DataFrame named 'cov\_df'. The output shows the DataFrame's structure: 3529 entries, 0 to 3528 columns, and a total of 18 columns.
- [4]: A data preview window showing a table with two columns: '확진일-증상발현일' and a numerical column. The data consists of 20 rows, with the last row being 8.



## Jupyter Tip

- 코드 작성 중 Tab을 누르면 code navigation 작동
- Jupyter kernel을 바탕화면이나 상태표시줄에 바로가기 등록 가능
- command/edit mode가 존재, 이를 이용해서 단축키를 활용한 코딩 가능
  - 필요한 패키지는 conda prompt를 켜고 pip install로 설치!
  - 저장(Ctrl + s)을 생활화 합시다...!

## << Round 2 >>

- 가상환경
- Jupyter Lab
- Python 기초 <<



Let's  
Go





<https://wikidocs.net/5>

```
if 4 in [1,2,3,4] : print("4가 있습니다")
```

-> 만약 4가 1, 2, 3, 4 중에 있으면 "4가 있습니다"를 출력한다.

파이썬의 문법은 인간의 언어와 닮아 있다.





?

```
23 def start() :
24     print("-Calculator v1.0-")
25     while 1 :
26         global result
27         if result == 0 :
28             result = int(input("주연산자(수)를 입력하세요."))
29         else :
30             print(result)
31         unResult = result
32
33         sign = input("부호를 입력하세요. ! 를 입력하면 종료합니다.")
34         if sign == '!' :
35             break
36         try :
37             num = int(input("피연산자(수)를 입력하세요."))
38         except :
39             print("입력이 잘못되었습니다.")
40
41         if sign == '+' :
42             add(num)
43             print(unResult, " ", sign, " ", num, " = ", result)
44
45         elif sign == '-' :
46             sub(num)
47             print(unResult, " ", sign, " ", num, " = ", result)
48
```





코드블록을  
구분하는 기준  
은 중괄호가 아  
닌 들여쓰기!!!



```
23 def start() :
24     print("-Calculator v1.0-")
25     while 1 :
26         global result
27         if result == 0 :
28             result = int(input("주연산자(수)를 입력하세요."))
29         else :
30             print(result)
31         unResult = result
32
33         sign = input("부호를 입력하세요. ! 를 입력하면 종료합니다.")
34         if sign == '!' :
35             break
36         try :
37             num = int(input("피연산자(수)를 입력하세요."))
38         except :
39             print("입력이 잘못되었습니다.")
40
41         if sign == '+' :
42             add(num)
43             print(unResult, " ", sign, " ", num, " = ", result)
44
45         elif sign == '-' :
46             sub(num)
47             print(unResult, " ", sign, " ", num, " = ", result)
48
```

Tab



10 팩토리얼을  
for문을 이용해  
짠다면...??





# in java && C...

```
# if you are programming 10! for C/java
for(int i = 1; i <= 10; i++)
{
    result *= i
}
```

10 팩토리얼을  
for문을 이용해  
짠다면...??





```
# if you are programming 10! for C/java  
for(int i = 1; i <= 10; i++)  
{  
    result *= i  
}
```

```
# in python...  
for i in range(10):  
    result *=i
```

파이썬의 기본적인 for문의 형태  
는  
타 언어의 foreach문과 흡사해!



파이썬의 전형적인 for문은  
리스트(배열)를 읽어오는 방식!!!

- in : 리스트를 읽어오는 연산자
- range() : 리스트를 만들어주는 함수
- List Comprehension 을 이용해서  
다양하고 가독성 높은 표현 가능 !!!

1	2	3	4	1	5	9
5	6	7	8	2	6	10
9	10	11	12	3	7	11

```
# list Comprehension  
matrix = [[1,2,3,4],[5,6,7,8],[9,10,11,12]]  
[[row[i] for row in matrix] for i in range(4)]  
# -> [[1, 5, 9], [2, 6, 10], [3, 7, 11], [4, 8, 12]]
```

2차원 행렬 전치가 무려 1줄 컷... 세상에...



```
# 리스트 자료형  
ex_list = [1, "이", , [3]]  
  
# 튜플 자료형  
ex_tuple = (1, "이", [3])  
  
# 딕셔너리 자료형  
ex_dictionary = {1 : 1, 2 : "이", 3 : [3]}  
  
# 집합 자료형  
ex_set = set([1, "이", [3]])
```

이렇듯 파이썬은 자료구조와 밀접한 관련이 있어!  
자료구조를 잘하면 파이썬도 잘할 수 있겠지?





```
C:\#Users\#82109>python
Python 3.7.4 (default, Aug  9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> import this
```

**cmd -> python -> import this  
직접 실행해보자!**

파이썬은 하나의 간결하고 아름다운 코드를 지향  
해!

파이썬 인터프리터엔 파이썬의 철학을 담은  
이스터에그(import this)가 존재할 정도!



