



Round 2

**PRESS
START**



<< Round 2 >>

- 가상환경
- Jupyter Lab
- Python 기초



New
Assignment



<< Round 2 >>

- 가상환경 <<
- Jupyter Lab
- Python 기초



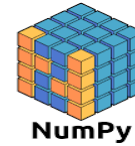
Let's
Go



Data Science Project_1 =



+



NumPy



SciPy

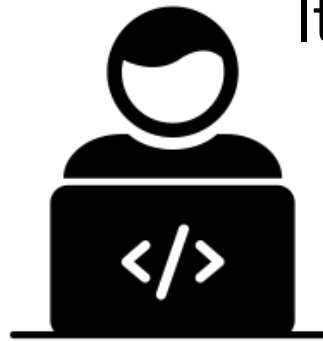
Pandas



Seaborn



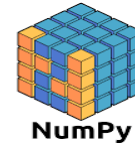
It's OK, but...



Data Science Project_1 =



+



SciPy

Pandas



Seaborn



Deep Learning Project_1 =



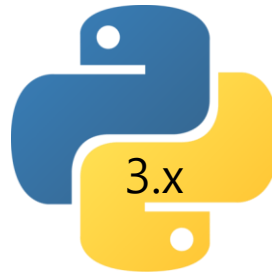
+



TensorFlow

matplotlib

Data Science Project_2 =



+

BeautifulSoup



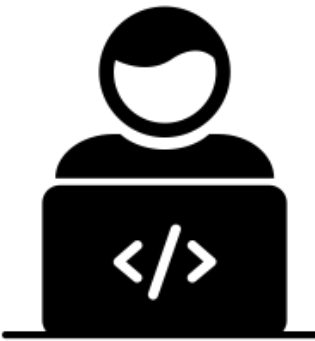
scikit-learn



Keras

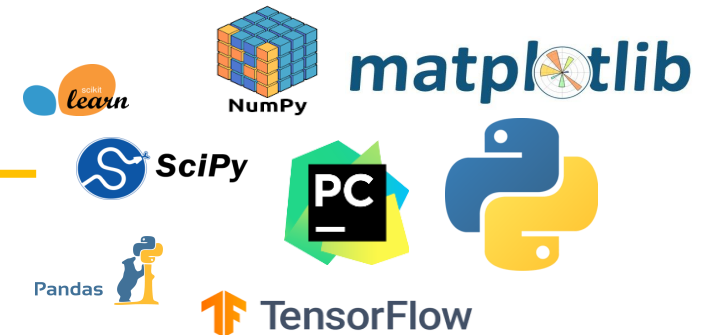
.
. .
.

So sad...





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





ANACONDA®

가상 환경 1



ANACONDA®

가상 환경 2



ANACONDA®

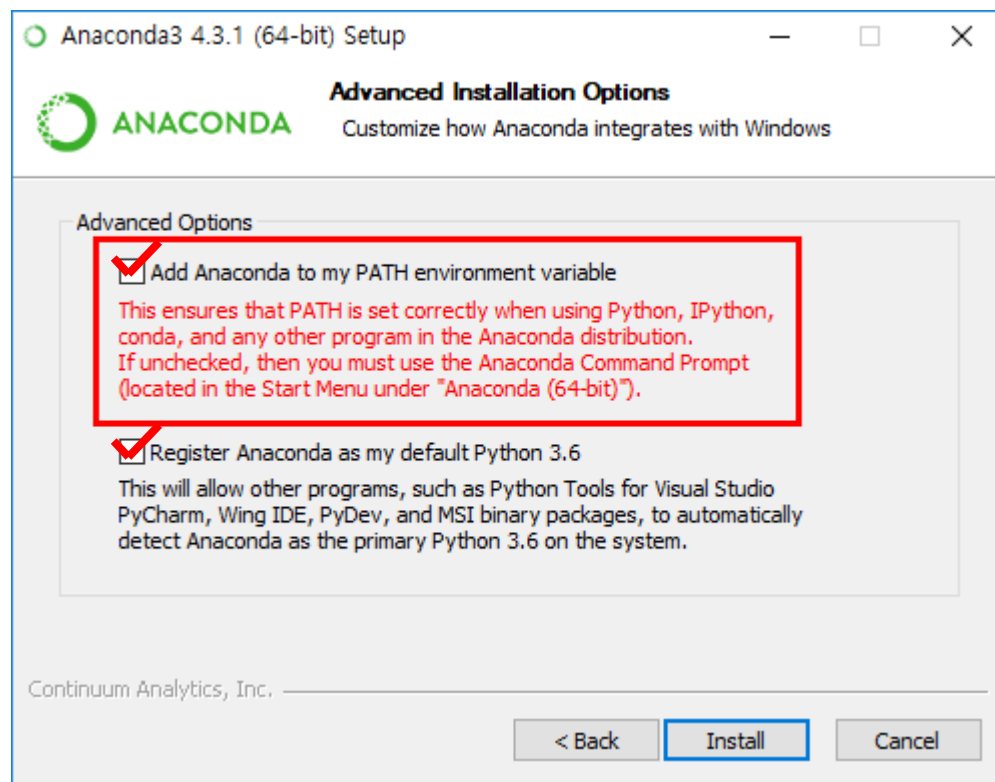
가상 환경 3



...

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

2. check this box



3. how to check my Anaconda is installed well?

```
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

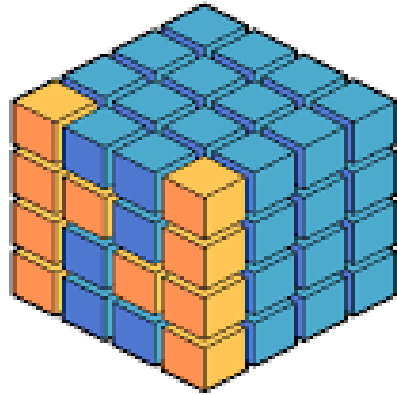
C:\Users\User>
```

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 띄기 때문에 데이터처리에 필수적



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



- 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

<< Round 2 >>

- 가상환경
- Jupyter Lab <<
- Python 기초



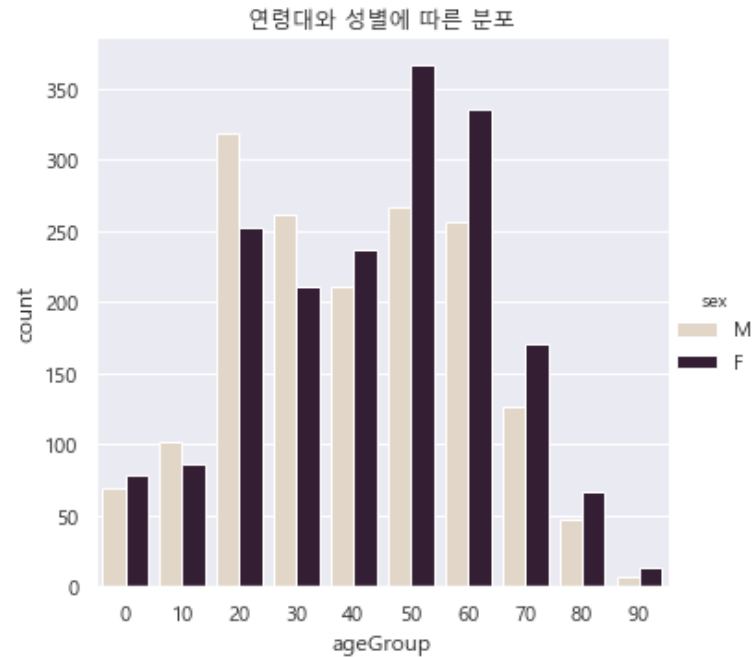
Let's
Go



Jupyter Lab의 장점





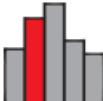


- 코드를 블록 단위로 실행시킬 수 있고 결과에 코멘트를 달 수 있음.

```
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 때문이고, 50~60대는 교회때문이 아닐지??

Applications on base (root) Channels

 CMD.exe Prompt 0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated Launch	 JupyterLab 1.1.4 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. Launch	 Jupyter 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	

Notebook



Console



Shift + Enter

Untitled.ipynb



Code



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

Hello world

Untitled.ipynb



Markdown

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

Hello world

첫 주피터 노트북 사용하기

- Hello world 출력

Untitled.ipynb



Markdown

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

Hello world

```
## 첫 주피터 노트북 사용하기
```

```
- Hello world 출력
```

Jupyter의 확장자는 .ipynb

여러 탭을 띄울 수 있고 여러 창을 띄울 수 있음

Directory

The screenshot shows the JupyterLab interface with the following components:

- File Browser (Left):** Displays a directory structure. The path `/ ... / 공모전 / 경기코로나예측` is highlighted with a red box. The file `cocoa_1.ipynb` is selected.
- Code Editor (Center):** Shows the content of `cocoa_1.ipynb`. The code includes imports for `matplotlib`, `pandas`, `numpy`, `seaborn`, `datetime`, `folium`, and `json`. It also sets the plot style to `darkgrid` and configures the font and figure size. The code is as follows:


```
[2]: %matplotlib inline
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
import seaborn as sns
import datetime as dt
import folium
import json

# darkness plot style
sns.set(style="darkgrid")

# To print korean
plt.rcParams["font.family"] = "Malgun Gothic"
plt.rcParams["font.size"] = 12
plt.rcParams["figure.figsize"] = (10, 10)

# To print minus(-)
mpl.rcParams['axes.unicode_minus'] = False

[3]: cov_df = pd.read_csv("코로나19 경기도 확진환자.csv")
cov_df.info(verbose=True)
print(cov_df.head())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3529 entries, 0 to 3528
Data columns (total 18 columns):
# Column Non-Null Count Dtype
```
- Table (Right):** Displays a table with two columns: `확진일-증상발현일`. The table contains 20 rows of data, with the first row showing values 1 and 47. The table is titled `경기 증상 발현 일자.csv`.

<< 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
짤다면...??



in java && C...

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

10 팩토리얼을
for문을 이용해
짤다면...??



```
# 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 표현가능 !!!

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

```
# 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]]

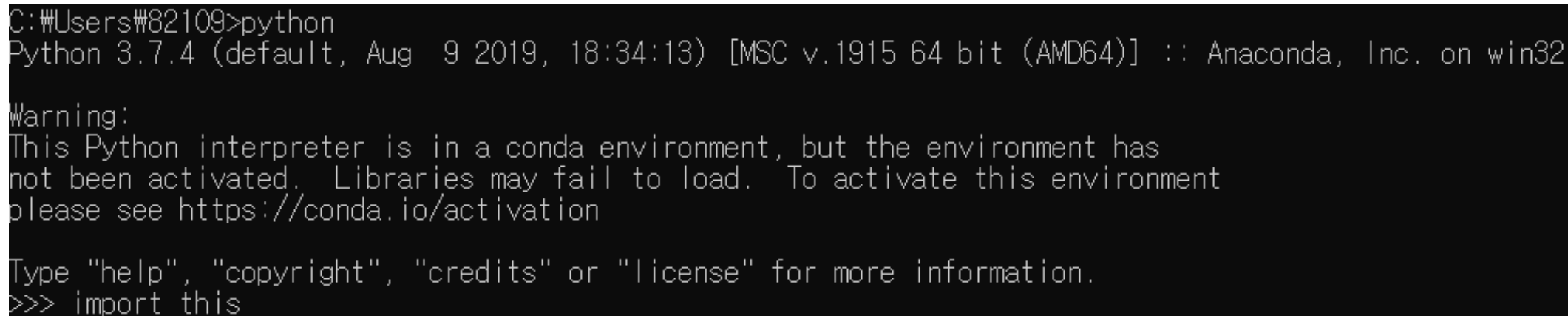
# 튜플 자료형
ex_tuple = (1, "이", [3])

# 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?





파이썬은 하나의 간결하고 아름다운 코드를 지향해!

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



