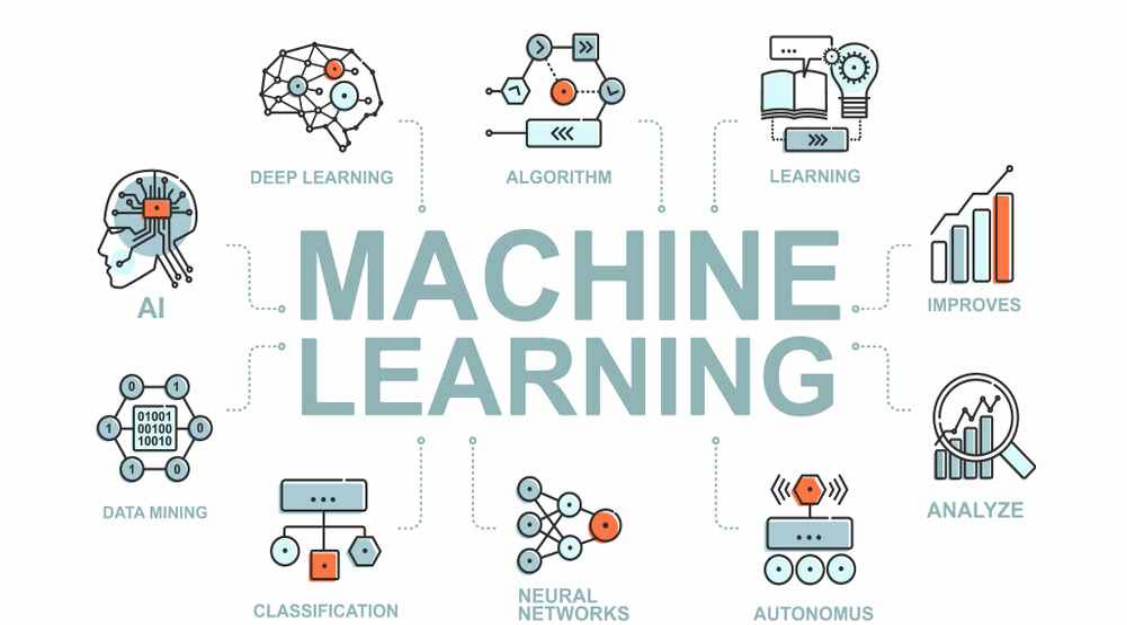
Machine Learning Study



화학안전연구센터

나민주

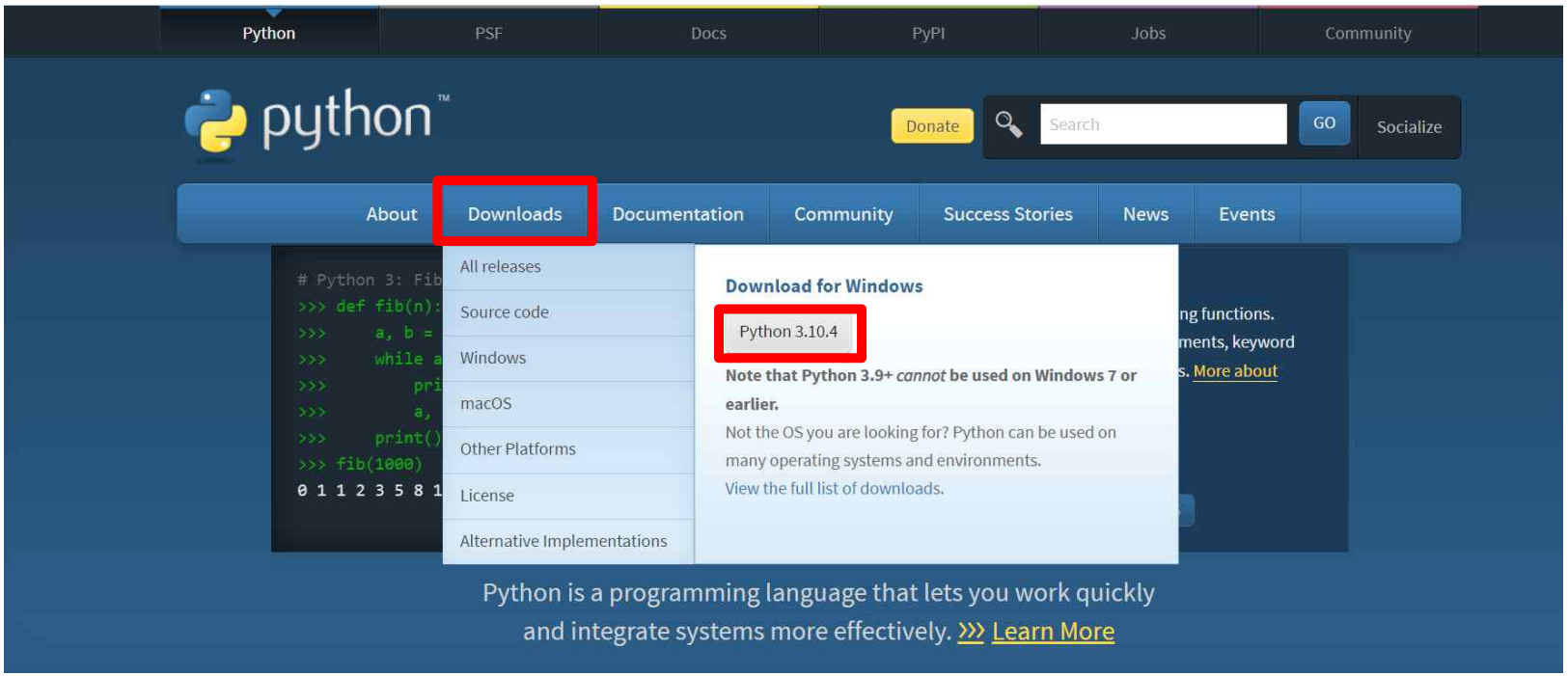
2022.04.12.

2.3 데이터 가져오기

2.3.1 작업환경 만들기

Install python

http:www.python.org/



2.3 데이터 가져오기

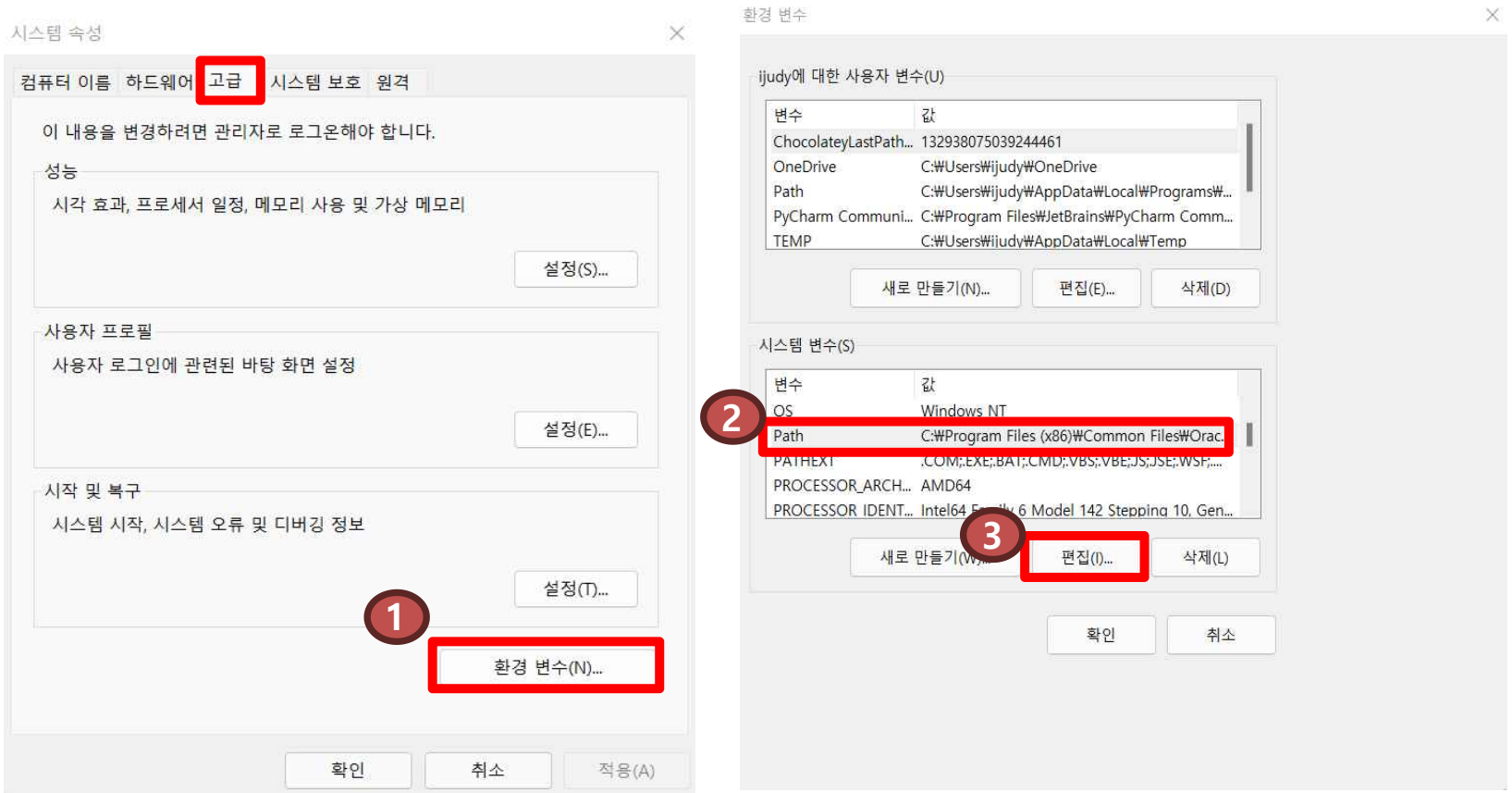
2.3.1 작업환경 만들기

In Mac

$ export ML_PATH="$HOME/ml" (원하는 경로)
$ mkdir -p $ML_PATH * mkdir: directory 생성 command

In Windows

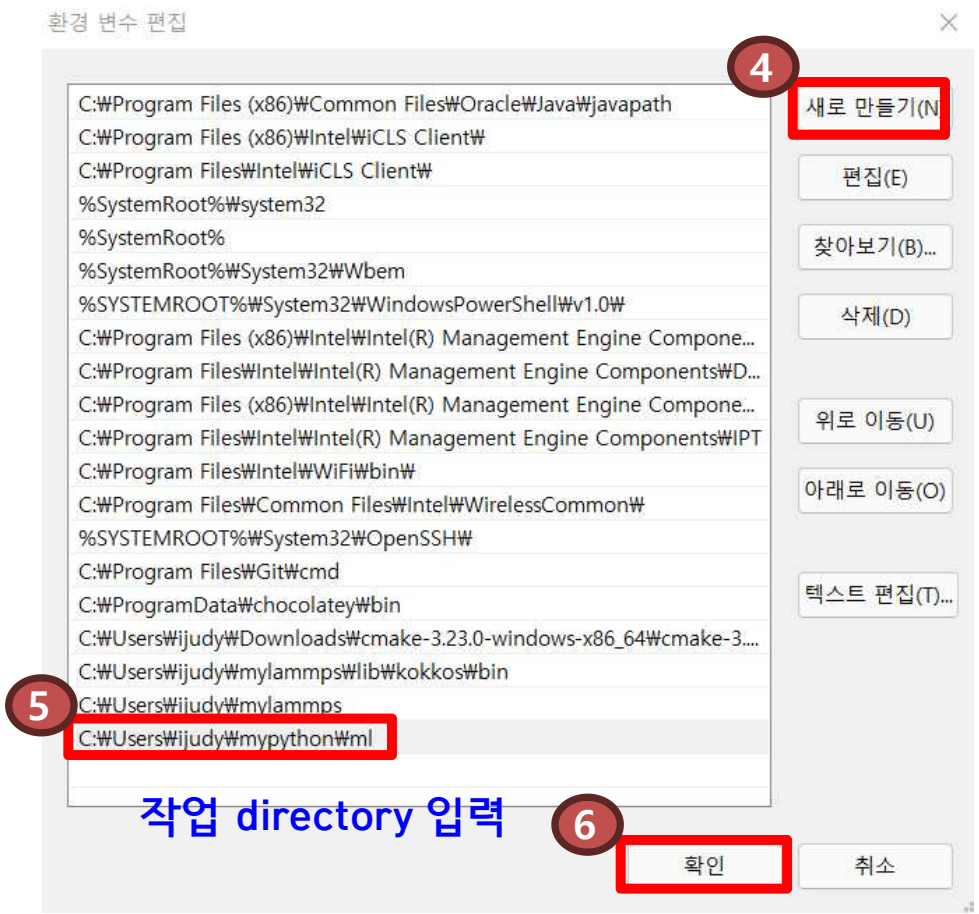
제어판 >> "고급 시스템 설정 보기" 검색
>> 시스템 속성 >> 고급 >> "환경변수" 클릭



2.3 데이터 가져오기

2.3.1 작업환경 만들기

In Windows



2.3 데이터 가져오기

2.3.1 작업환경 만들기

```
# pip 설치 여부 확인
> pip3 --version
pip 22.0.4 fromC:\Users\ijudy\AppData\Local\Programs\Python\Python310\lib\site-packages\pip (python 3.10)

# 최신버전 pip 설치 필요하면, upgrade 진행
> python3 -m pip install --user -U pip
Collecting pip
[...]
Succesfully installed pip-22.0.4
```

2.3 데이터 가져오기

2.3.1 작업환경 만들기

package 설치

> **pip3 install --upgrade jupyter matplotlib numpy pandas scipy scikit-learn**

Collecting jupyter..

Downloding [...]

> # 새로운 "> 프롬프트"가 생성되면 잘 설치된 것!

jupyter 실행

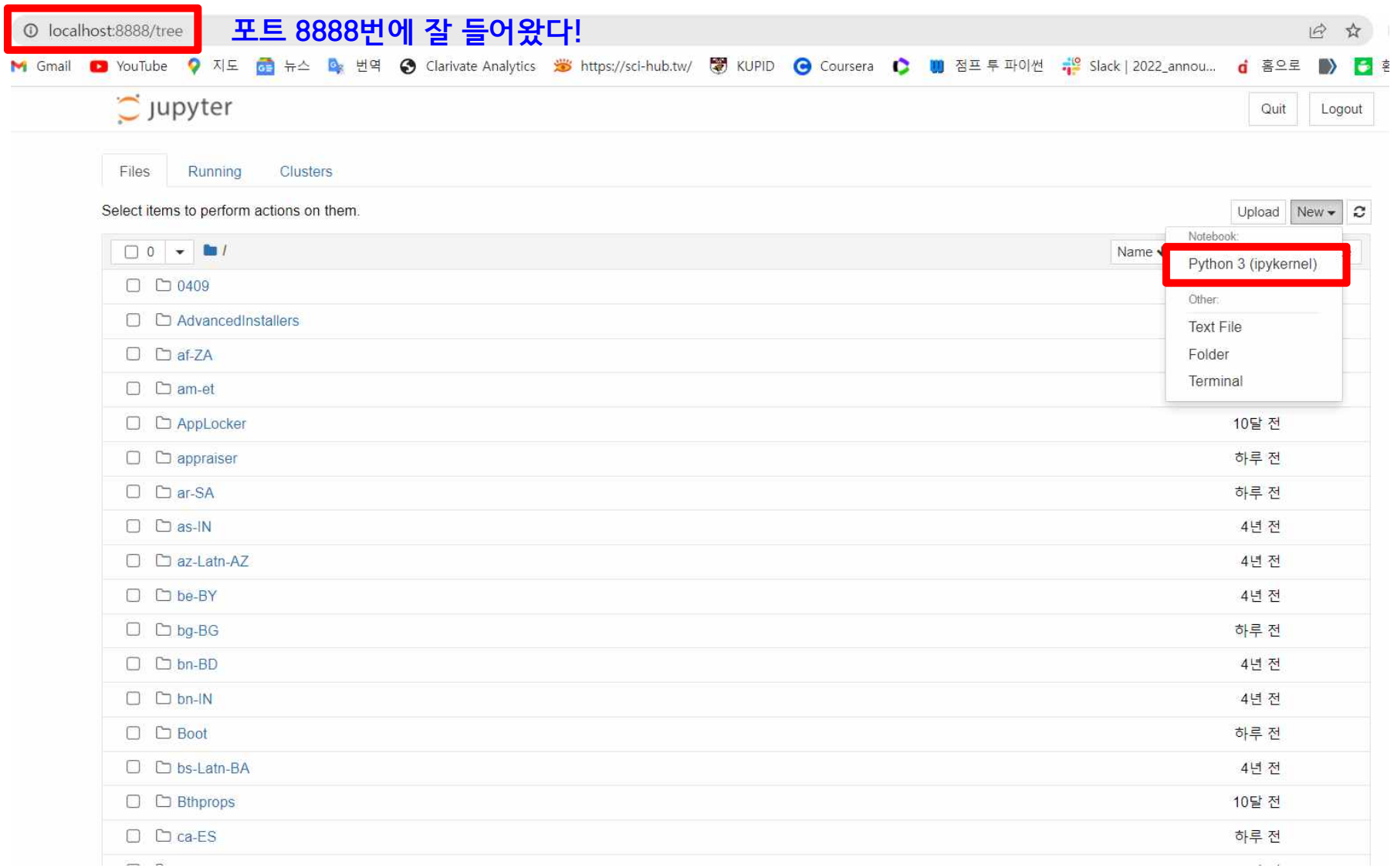
> **jupyter notebook**

```
C:₩WINDOWS₩system32>jupyter notebook
[I 07:21:53.827 NotebookApp] Writing notebook server cookie secret to C:₩Users₩ijudy₩AppData₩
Roaming₩jupyter₩runtime₩notebook_cookie_secret
[I 07:21:54.754 NotebookApp] Serving notebooks from local directory: C:₩WINDOWS₩system32
[I 07:21:54.755 NotebookApp] Jupyter Notebook 6.4.10 is running at:
[I 07:21:54.755 NotebookApp] http://localhost:8888/?token=5e9f51aeeaf3e738b5051689d21e05a8fda
688c0351a0662
[I 07:21:54.755 NotebookApp]  or http://127.0.0.1:8888/?token=5e9f51aeeaf3e738b5051689d21e05a
8fda688c0351a0662
[I 07:21:54.755 NotebookApp] Use Control-C to stop this server and shut down all kernels (twi
ce to skip confirmation).
[C 07:21:54.841 NotebookApp]
    To access the notebook, open this file in a browser:
        file:///C:/Users/ijudy/AppData/Roaming/jupyter/runtime/nbserver-11864-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=5e9f51aeeaf3e738b5051689d21e05a8fda688c0351a0662
     or http://127.0.0.1:8888/?token=5e9f51aeeaf3e738b5051689d21e05a8fda688c0351a0662
```

jupyter server가 terminal에서 실행되었고, 포트 888번에 대기 중

2.3 데이터 가져오기

2.3.1 작업환경 만들기



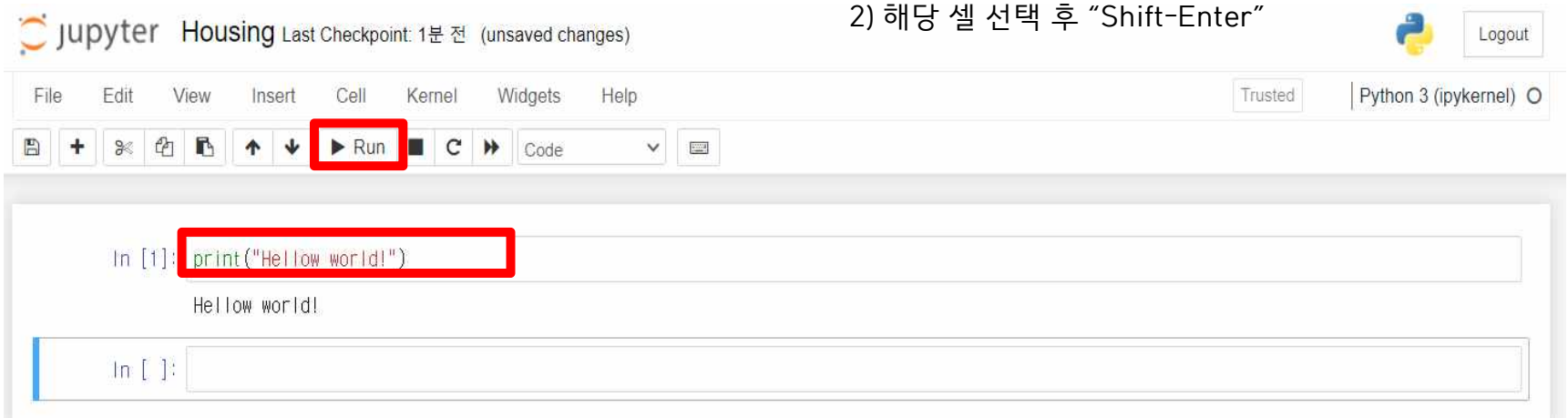
2.3 데이터 가져오기

2.3.1 작업환경 만들기

1) 이름을 Housing으로 바꿔준다. (Housing.ipynb)

2) python의 시작을 알리는 "Hello world!"

실행 방법 1) Click "**Run**"
2) 해당 셀 선택 후 "Shift-Enter"



2.3 데이터 가져오기

2.3.2 데이터 다운로드

1) 데이터 추출

```
import os
import tarfile
import urllib.request

DOWNLOAD_ROOT = "https://raw.githubusercontent.com/rickiepark/handson-ml2/master/"
HOUSING_PATH = os.path.join("datasets", "housing")
HOUSING_URL = DOWNLOAD_ROOT + "datasets/housing/housing.tgz"

def fetch_housing_data(housing_url=HOUSING_URL, housing_path=HOUSING_PATH):
    if not os.path.isdir(housing_path):
        os.makedirs(housing_path)
    tgz_path = os.path.join(housing_path, "housing.tgz")
    urllib.request.urlretrieve(housing_url, tgz_path)
    housing_tgz = tarfile.open(tgz_path)
    housing_tgz.extractall(path=housing_path)
    housing_tgz.close()
```

2) fetch_housing_data() 호출
--> 작업공간에 datasets/housing directory 생성 & .tgz 파일 다운 후 압축풀어 .csv 파일로 생성

```
fetch_housing_data()
```

3) pandas 함수로 데이터 read

```
# Pandas 사용하여 데이터 read
import pandas as pd

def load_housing_data(housing_path=HOUSING_PATH):
    csv_path = os.path.join(housing_path, "housing.csv")
    return pd.read_csv(csv_path)
```

2.3 데이터 가져오기

2.3.3 데이터 구조 훑어보기

1) **head()** : Data의 위쪽 5행을 보여줌

```
In [7]: housing = load_housing_data()
        housing.head()
```

Out[7]:

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR BAY
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR BAY
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR BAY
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR BAY
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR BAY

2) **info()** : Data에 대한 간략한 설명, 전체 행 수, 데이터 타입, NULL이 아닌 값의 개수 확인 가능

```
In [9]: housing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  float64
 3   total_rooms         20640 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  float64
 6   households          20640 non-null  float64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  float64
 9   ocean_proximity     20640 non-null  object
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
```

housing data의 경우,

총 20,640개의 샘플이 있으며,
총 10개 column을 가짐
total_bedrooms만 20,433개만 NULL이 아님 -> 즉, 207개는 없음

data type 다름
-> 범주형(categorical) 데이터인지 확인

2.3 데이터 가져오기

2.3.3 데이터 구조 훑어보기

3) **Data["column"].value_counts()**:
Data에서 특정 column 지정하여, column이 가지는 category과 각 category의 수 확인

```
In [10]: housing["ocean_proximity"].value_counts()
Out[10]: <1H OCEAN     9136
         INLAND        6551
         NEAR OCEAN    2658
         NEAR BAY      2290
         ISLAND           5
         Name: ocean_proximity, dtype: int64
```

4) **describe()** : numeric 특성의 요약정보 제시

```
In [11]: housing.describe()
```

Out[11]:

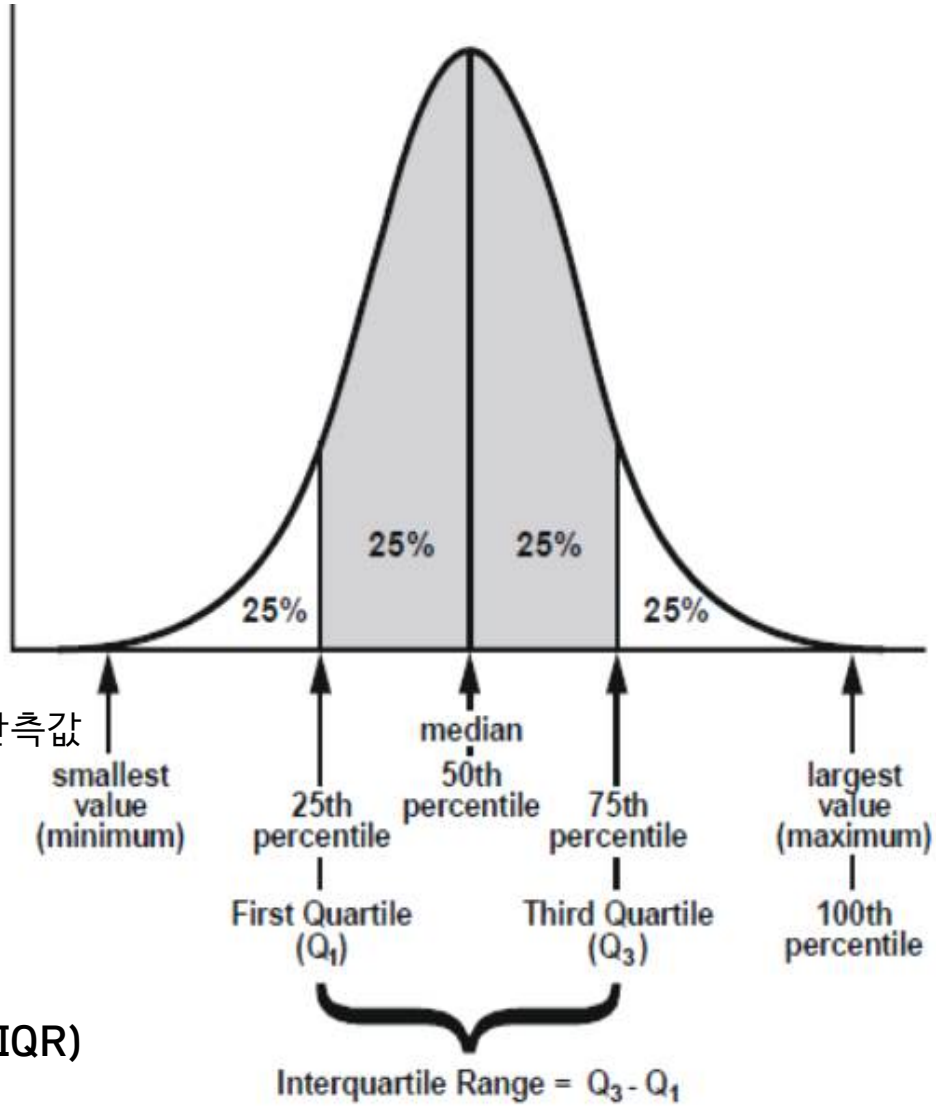
	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000	20640.000000	20640.000000	20640.000000
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476744	499.539680	3.870671	206855.816909
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462122	382.329753	1.899822	115395.615874
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900	14999.000000
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000000	280.000000	2.563400	119600.000000
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000000	409.000000	3.534800	179700.000000
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000000	605.000000	4.743250	264725.000000
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.000100	500001.000000

∴ 총 10개 column 중 위의 "ocean_proximity"는 범주형, 나머지 9개 column은 숫자형

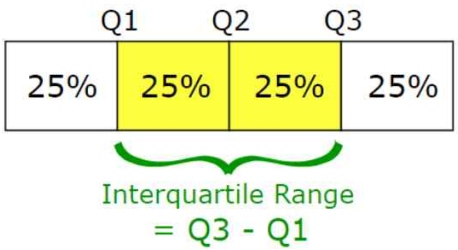
2.3 데이터 가져오기

2.3.3 데이터 구조 훑어보기

분위수 (Quantile)
: 자료 크기의 순서에 따른 위치값



사분위수 (Quartile)
: 크기 순서로 나열한 자료를 4등분하는 관측값



사분위수범위 (Interquartile range, IQR)
: 전체 자료의 50% 를 포함하는 범위
(Q3 – Q1)

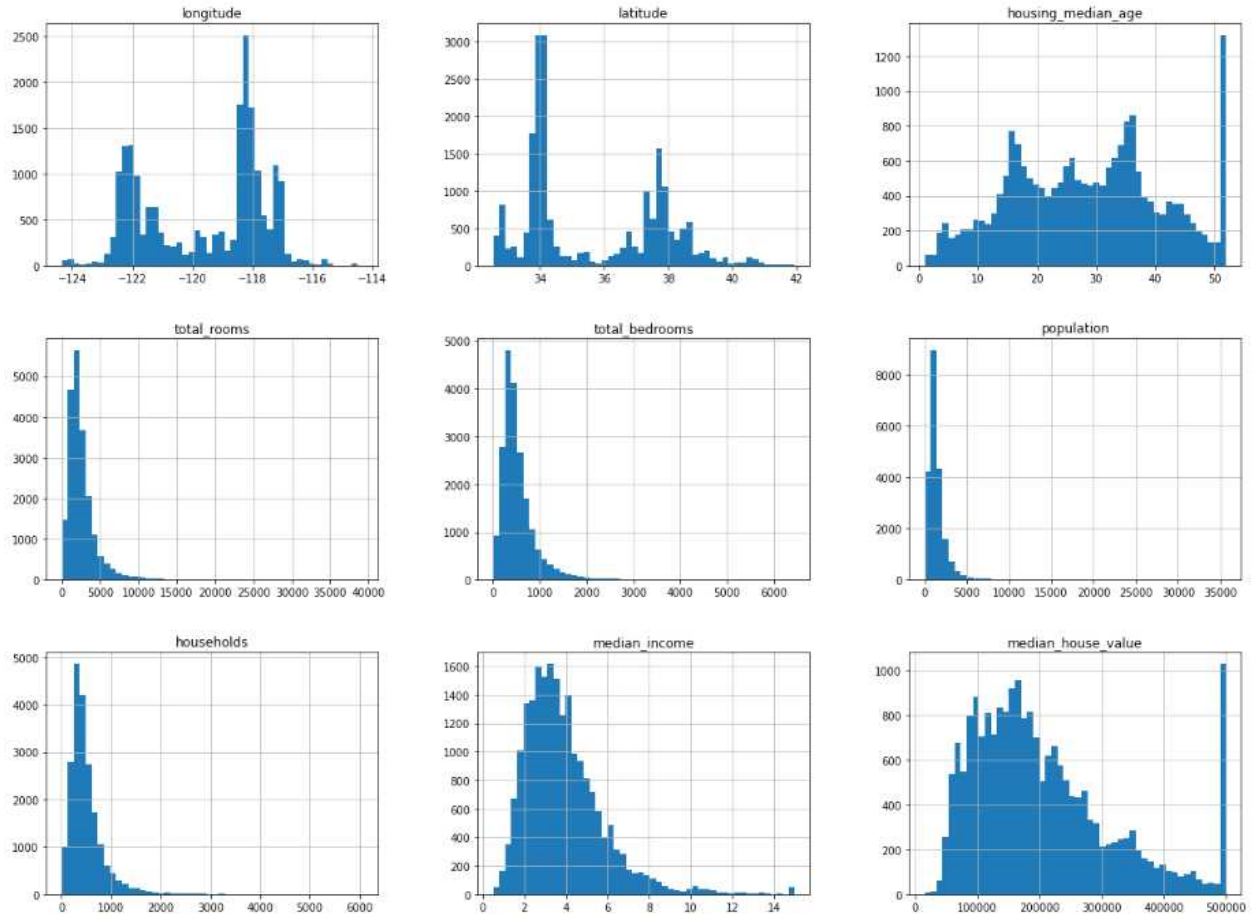
https://bioinformaticsandme.tistory.com/246

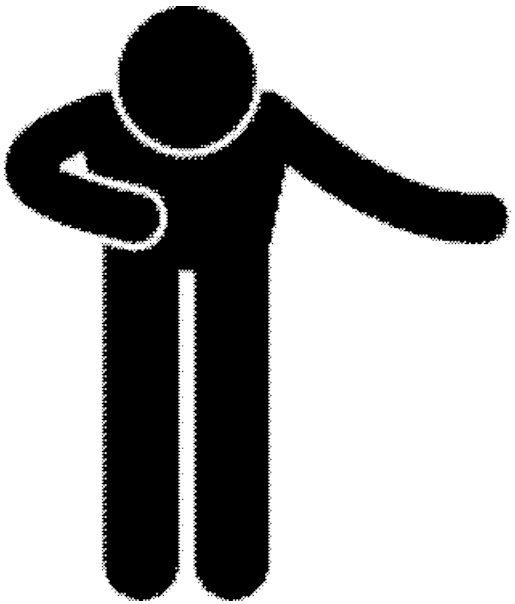
2.3 데이터 가져오기

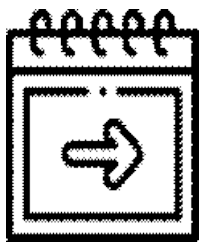
2.3.3 데이터 구조 훑어보기

히스토그램 (Histogram)

```
In [22]: %matplotlib inline
         import matplotlib.pyplot as plt
         housing.hist(bins=50, figsize=(20,15))
         plt.show()
```







2.3.4 테스트 세트 만들기