

# 01. 파이썬 데이터 분석

## 학습 목표

- 파이썬 기본 문법을 활용한 데이터 분석 방법을 설명 할 수 있다.

## 파이썬 기본 문법을 활용한 데이터 분석 방법 의 이해

### - 데이터의 수집

- . 홈페이지, 데이터베이스, 외부 파일 등에서 수집한 데이터들은 수집 당시 형식은 모두 다르지만, 파이썬의 자료형으로 읽어와 대부분 숫자 또는 문자 형태 의 텍스트 데이터 로 이루어져 있음
  - > 파이썬의 기본 문법들을 활용하여 텍스트 데이터 내에서 불필요한 데이터 를 제거하거나 패턴을 찾아 필요한 데이터만 분리하는 등의 데이터 분석을 진행 할 수 있음

### - TXT,CSV 파일 데이터의 분석

- . **open()** 함수를 활용하여 txt 파일 형식의 데이터 를 읽어와 분석 할 수 있음.

```
f = open('2000_01_01_error_log.txt', 'r')
data = f.read()
print(data)
f.close()
```

```
[2000-01-01 12:00:00] Data Analysis Error - Message:
java.lang.IllegalStateException: Illegal access: this web application insta
s to attempt to terminate the thread which caused the illegal access.
    at org.apache.catalina.loader.WebappClassLoaderBase.checkState
```

- . CSV 형식을 읽을 경우 ';' 로 구분짓는 것이 아닌 ',' 자체를 문자로 인식하여 읽어 들임
  - > CSV : 데이터 와 데이터 를 , (콤마) 로 분리 하여 저장하는 파일 형태.  
간단한 데이터 등록 시 CSV 를 이용하여 데이터를 입/출력 할 수 있다.
  - > 파이썬 표준 모듈 중 **CSV 모듈** 을 활용할 경우 콤마로 구분된 상태 로 전처리 하여 읽어올 수 있다.

```
import csv
f = open('data.csv', 'r', encoding='utf-8-sig')
data = csv.reader(f)
print(type(data))
for i in data :
    print(i)
```

```
<class '_csv.reader'>
['name', 'phone', 'age', 'number']
['김범수', '111-1111', '38', '10']
['나얼', '222-2222', '37', '20']
['박효신', '333-3333', '36', '30']
['이수', '444-4444', '41', '40']
```

### - JSON 형식 데이터

- . 파이썬의 **JSON 모듈** 을 활용하여 JSON 형식의 데이터를 파이썬의 객체 로 변환하여 활용할 수 있음

```
import json
a = [1, 2, 3, {'4': 5, '6': 7}]
```

```
b = json.dumps(a)
print(b)
```

```
[1, 2, 3, {"4": 5, "6": 7}]
```

```
c = json.loads(b)
print(c)
```

```
[1, 2, 3, {'4': 5, '6': 7}]
```

→ dumps 함수:  
파이썬 → JSON 변환

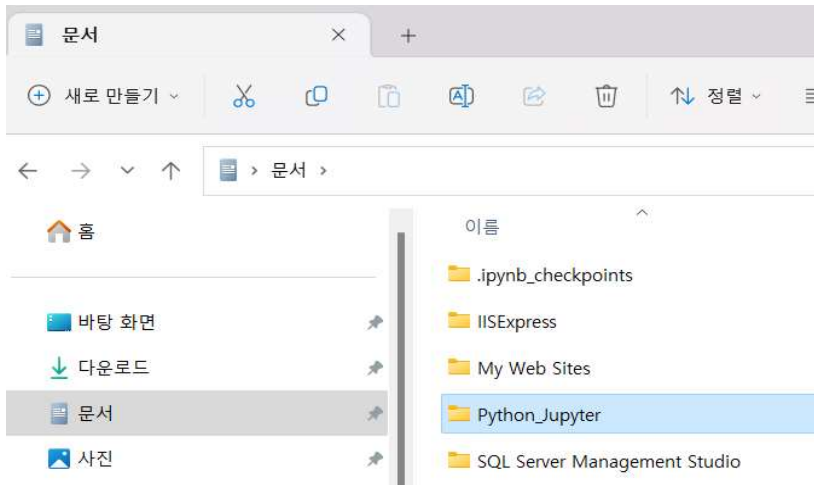
→ loads 함수:  
JSON → 파이썬 변환

\* 파일 형태로 된 데이터 분석을 위하여 처리 되는 유형은 txt, csv, json 파일 형식을 가장 많이 사용한다.

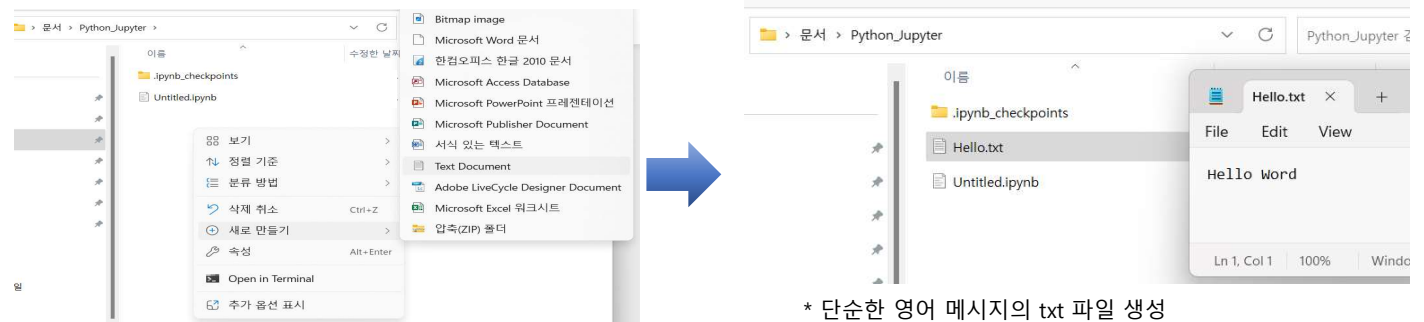
## JupyterNoteBook 을 이용한 Python 기본 문법 데이터 분석 활용

### - JupyterNotebook Python 프로젝트 만들기

. 내 문서 - Python\_Jupyter 폴더 생성

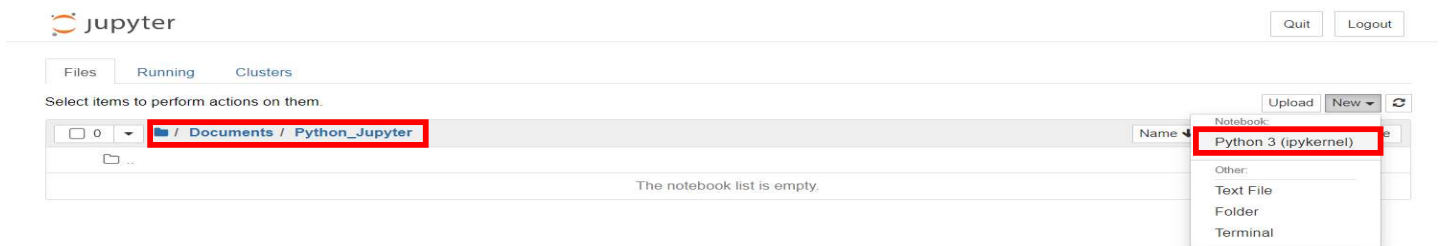


. 테스트 에 필요한 txt 파일 만들기



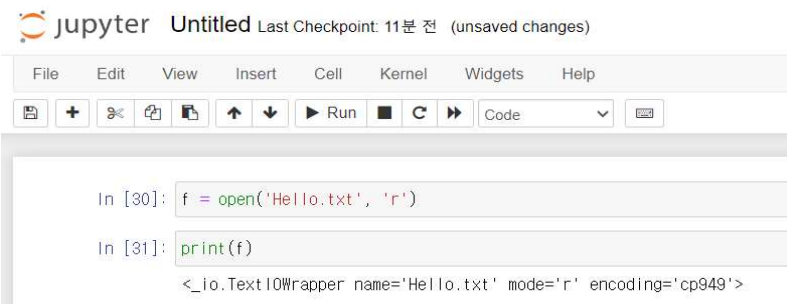
\* 단순한 영어 메시지의 txt 파일 생성

. Documents / Python\_Jupyter 폴더로 이동 후 New -> Python



### - Txt 파일 데이터 입/ 출력하기

. Shift + Enter 또는 ▶ Run



\* 데이터 를 정상적으로 읽어올 수 있을경우 메시지 출력

## . 실행 및 결과 확인

```
In [30]: f = open('Hello.txt', 'r')
```

```
In [31]: print(f)
<_io.TextIOWrapper name='Hello.txt' mode='r' encoding='cp949'>
```

```
In [32]: data = f.read()
```

> 파일 스트림의 내용을 읽어 data 변수에 담는다

```
In [33]: print(data)
```

> data 변수에 담긴 내용을 출력한다.

```
Hello Word
```

```
In [34]: f.close()
```

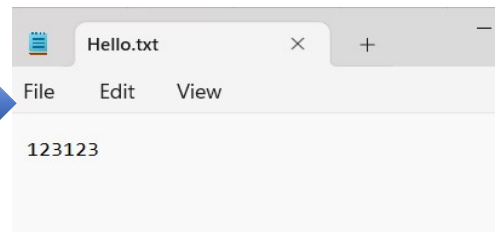
## . W 모드를 통한 데이터 쓰기

```
In [35]: f = open('Hello.txt', 'w')
```

```
In [36]: f.write('123123')
```

```
Out[36]: 6
```

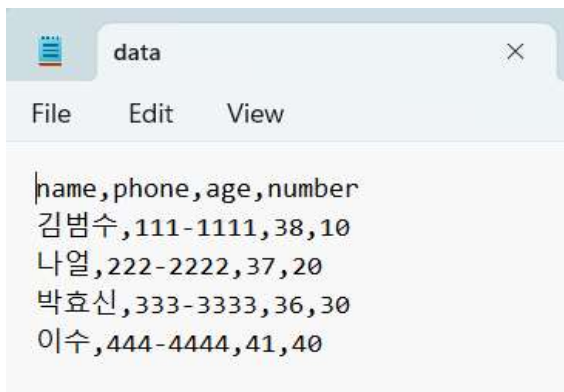
```
In [37]: f.close()
```



## - CSV 파일 데이터 입 / 출력 하기

### . CSV 파일 확인 (Data.csv)

\* 엑셀로 읽을수 있지만 메모장으로 확인하면 필드(컬럼) 간 데이터 가 ',' 로 구분 되어 있는것을 확인 할 수 있다.



\* 구분자는 컴퓨터 환경 마다 다를 수 있다.

## . CSV 파일 불러오기

> encoding='utf-8' : 한글 / 영문 자리수 차이의 문자 의 유형을 맞추기 위한 설정

```
f = open('data.csv', 'r', encoding='utf-8')
data = f.read()
f.close()
print(data)
```

>파일 스트림을 통한 Data.csv 를 읽기

```
name,phone,age,number
김범수,111-1111,38,10
나얼,222-2222,37,20
박효신,333-3333,36,30
이수,444-4444,41,40
```

> 결과는 모두 문자열 인것을 확인 할 수 있다.  
\*, 를 포함한 모든 문자열

```
# 데이터 의 타입은 문자열
print(type(data))
```

```
<class 'str'>
```

```
# , 를 기준으로 데이터 를 분할 할 수 있다.
data.split(',')
```

> 데이터를 활용하기 위해서는 split 를 통해 전처리 과정을 거쳐야 사용할 수 있는 상태 가 된다.

```
['#ufeffname',
 'phone',
 'age',
 'number\n김범수',
 '111-1111',
 '38',
 '10\n나얼',
 '222-2222',
 '37',
 '20\n박효신',
 '333-3333',
 '36',
 '30\n이수',
 '444-4444',
 '41',
 '40\n']
```

## . Csv 모듈을 import 하여 CSV 파일 불러오기

> csv.reader(f) 함수 를 활용하여 기본적으로 전처리 된 csv 데이터 를 추출 할 수 있다.

```
In [58]: import csv
```

```
In [59]: f = open('ItemMaster.csv', 'r', encoding='utf-8')
data = csv.reader(f)
```

```
In [60]: for i in data:
          print(i)
          f.close()
```

```
['#ufeffplantcode;Itemcode;ItemName;UseFlag']
['1000;111-2222;배어링 케이스;Y']
```

## - json 파일 데이터 입 / 출력 하기

. Json 모듈을 import 하여 dumps() 와 loads() 함수 를 통하여 json 파일 데이터 를 입 / 출력 할 수 있다.

```
In [1]: import json
a = [1,2,3,{ '4' : 5, '6' : 7}]
```

```
In [3]: b = json.dumps(a)
print(b)

[1, 2, 3, {"4": 5, "6": 7}]
```

```
In [4]: c = json.loads(b)
print(c)

[1, 2, 3, {'4': 5, '6': 7}]
```