

[파일 I / O 프로그래밍]

1. 파일에 데이터 저장 후 불러오기

(1) **open('파일경로/파일명', '모드')** : 반드시 실행 후에 파일을 닫아야 한다.

```
ex) - sample.txt 파일 출력하기
f = open('data/sample.txt', 'w')      # 'w': 쓰기 모드
f.write('HelloWnWorldWnHelloWnPythonWn') # .write(): 파일에 쓸 내용 입력
f.close()                             # 파일 닫기(반드시 실행할 것)
```

(2) **with open('파일경로/파일명', '모드')**: 파일을 닫지 않아도 실행 가능하다.

```
ex) - numbers.txt 파일 출력하기
with open('data/ch09_member.txt', 'r', encoding = 'UTF8') as f: # 'r': 읽기 모드 / encoding = : 인코딩 체계
    lines = f.readlines() # .readlines(): 내용 한꺼번에 불러오기
```

2. 피클링: 파이썬 객체 계층 구조가 바이트 스트림으로 변환되는 것

```
ex) f = open('member.data', 'wb') # 'wb': 피클링할 때의 쓰기 모드
                                     (피클링 할 때는 바이너리(b)모드로 열어야 한다.)

import pickle                       # 피클링을 위한 import
pickle.dump(user_list, f)           # member.data 파일에 저장할 내용(user_list) 피클링하기
f.close()
```

3. csv파일의 자료를 딕셔너리 데이터로 읽기 / 쓰기

(1) 함수: 읽어오기 - **csv.DictReader()** / 쓰기 - **csv.Dict.Writer()**

```
ex) - f라는 csv파일 딕셔너리로 읽어오기
csv.DictReader(f, fieldnames = ['name', 'age', 'address'], # fieldnames: 객체가 헤더가 없는 경우, 필드명 지정
                restkey = 'job',                          # restkey: 필드 수보다 더 많은 행이 있는 경우, 그 남은 필드명을 지정
                restval = None) # restval: 필드명보다 필드 수가 적을 경우, 자리를 채울 값 지정
```

4. JSON데이터 저장하고 불러오기

- json 불러오기: **import json**
- json 저장하기: **json.dump()**
- json파일 불러오기: **json.load()**

5. 판다스 라이브러리로 HDF5(Hierarchical Data Format) 입출력

- 판다스 불러오기: **import pandas as pd** # 판다스는 보통 *pd*라는 별칭으로 많이 사용한다.
- hdf5 불러오기: **import h5py**
- hdf5 출력: **with h5py.File('파일명', 'w')**
- 판다스로 hdf5파일 읽어오기 : **pd.read_hdf("파일명")**