

09. 파일 입/출력 프로그래밍

#src: 8_Python/ch09_파일입출력프로그래밍 참고

1) 파일 입출력; open()

f = open(file_name, mode, encoding='ASCII') #한글 파일 읽을 경우 encoding='UTF8'

(1) **mode** r: 읽기 모드, w: 쓰기 모드, a: 추가 모드, b: 바이너리 모드**2) 피클을 이용하여 객체 저장 및 불러오기: pickle.dump(), pickle.load()**

(1) 피클링; 파이썬 객체 계층 구조가 바이트 스트림으로 변환되는 것

피클링	언피클링: 피클링한 데이터 다시 불러옴
with open('파일명.data', 'wb') as f: import pickle L#바이너리 쓰기 pickle.dump(obj, file)	with open('파일명.data', 'rb') as f: ul = pickle.load(f) L#바이너리 읽기

3) csv형식 파일 읽기/쓰기

- 스프레드시트 또는 DB가져오거나 내보내기 할 때 가장 많이 사용하는 일반적인 형식

(1) reader() 와 writer() L#자동 형변환을 위한 딕셔너리 인수

① csv.reader(csvfile, [quotechar='"', quoting=csv.QUOTE_NONNUMERIC])

② csv.writer(csvfile, [quoting=csv.QUOTE_NONNUMERIC])

(2) DictReader(), DictWriter(); 딕셔너리 형식으로 데이터 읽고 쓰기

① csv.DictReader(csvfile, [fieldnames=None, restkey=None, restval=None]) #fieldnames: 열 이름

② csv.writer(csvfile, [restval=' ', extrasaction='raise']) #restval: 필드명보다 필드 수 적을 때 지정 값
writerow(); 한 개 행 저장, writerows(); 모든 딕셔너리 데이터 저장**4) JSON(JavaScript Object Notation) 데이터 저장하고 불러오기(dump, load)**

(1) json.dump(data, jsonfile, ensure_ascii=False, indent='Wt'); 파이썬 객체 → JSON형식으로 변환

L#ensure_ascii=False; 한글 깨질 경우, indent만큼 띄어쓰기

(2) json.load(jsonfile, [object_hook=리터럴 호출 함수]); JSON형식 데이터 → 파이썬 객체로 변환

L#디코딩 된 객체 리터럴의 결과로 호출되는 함수 지정

5) HDF5(Hierarchical Data Format) 파일 읽기/쓰기

; HDF그룹에 의해 관리되고 있는 이기종 데이터를 저장, 관리, 처리하기 위한 고성능 데이터 소프트웨어 라이브러리 및 파일 형식(HDF5는 빠른 입/출력 저장 및 처리를 위해 만들어짐)

(1) h5py 사용: File클래스를 이용해 파일 객체를 생성하고 읽음

(2) pandas 패키지 이용: read_hdf(), to_hdf() 함수를 사용해서 데이터 프레임으로 읽거나 씀

※데이터프레임 → 딕셔너리로 변경: 데이터프레임.to_dict()

※데이터(튜플, 리스트 등) → 데이터 프레임으로 변경: pandas.DataFrame("데이터")