

파이썬

33강. 특수 파일 처리(JAON)



1. JSON 파일

- JSON(JavaScript Object Notation)은 네트워크에서 표준으로 사용되는 파일이다. 즉 인터넷 상에서 자료를 주고받을 때 해당 자료를 표현하는 파일 형식이다. JSON 파일은 "키-값" 쌍으로 자료를 표현하는 자바스크립트의 구문 형식을 따른다. 하지만 프로그래밍 언어나 플랫폼에 독립적이므로 JSON 자료는 C, JAVA, Python 등 다양한 프로그래밍 언어에서 이용할 수 있다.
- 파이썬에서는 JSON 파일을 처리할 수 있도록 json 모듈을 제공한다. 다음은 json 모듈에서 제공하는 인코딩(encoding)과 디코딩(decoding) 기능에 대한 설명이다.
- 인코딩 : Python 객체(dict or list) -> json 문자열('키-값'쌍)
- 디코딩 : json 문자열('키-값'쌍) -> Python 객체(dict or list)

1. JSON 파일

- 인코딩은 파이썬의 객체를 json 문자열로 변환하는 과정이다. 다음은 json 모듈에서 지원하는 dumps() 함수를 이용하여 파이썬 객체를 json 문자열로 인코딩하는 형식이다.

```
json.dumps(객체) ↵
```

1. JSON 파일

- 디코딩은 json 문자열을 파이썬의 딕트나 리스트로 변환하는 과정이다. 다음은 json 모듈에서 지원 하는 loads() 함수를 이용하여 json 문자열을 파이썬 객체로 디코딩하는 형식이다.
- 만약 JSON 파일의 자료를 파이썬으로 가져오는 작업을 수행한다면 이는 디코딩 작업을 수행한다고 말할 수 있다.

```
json.loads(json문자열) ↵
```

1. JSON 파일

- 다음 예문은 json 모듈을 이용한 인코딩과 디코딩 예이다.

chapter08.lecture.step07_json_file.py ↵

Python Console ↵

```
# (1-1) json 인코딩 ↵  
user = { 'id': 1234, 'name': '홍길동' } # Python Dict ↵  
hon Dict ↵  
print(type(user)) # <class 'dict'> ↵  
print(user['name']) # 홍길동 ↵
```

1. JSON 파일

```
# (1-2) json 인코딩↵
jsonString = json.dumps(user, ensure_ascii↵
=False)↵
# ASCII 인코딩 방식 적용 안함↵

# 문자열 출력↵
print(jsonString) # {"id": 1234, "name": "홍길동"}↵
# {"id": 1234, "name": "홍길동"}↵
# <class 'str'>↵
print(type(jsonString)) # class str↵
#print(jsonString['name']) # Error 발생↵

# (2) json 디코딩↵
pyObj = json.loads(jsonString)↵
print(type(pyObj)) # <class 'dict'>↵
# <class 'dict'>↵

# Dict 자료 확인↵
print(pyObj['name']) # 홍길동↵
for key in pyObj :↵
    print(key, ':' , pyObj[key])↵
# 홍길동↵
# id : 1234↵
# name : 홍길동↵
```

1. JSON 파일

- json 모듈의 인코딩과 디코딩 예
- # (1-1) json 인코딩
- 파이썬의 딕트 객체를 대상으로 json의 문자열로 변경하는 과정이다.
user 객체는 'id' 또는 'name'
- 키를 이용하여 값을 참조하는 딕트 자료구조이다.
- # (1-2) json 인코딩
- json 모듈에서 제공하는 dumps() 메서드를 이용하여 딕트 객체를 json의 문자열로 변경한다. 명령어 형식은 다음과 같다.
- 형식) json.dumps(객체)
- dumps()함수의 ensure_ascii=False 파라미터는 한글 상태를 그대로 유지하기 위해서 ASCII 인코딩 방식을 적용하지 않겠다는 의미이다.

1. JSON 파일

- # (2) json 디코딩
- 인코딩된 json 문자열을 다시 파이썬의 딕트 객체로 변환하는 과정이다. json 모듈에서 제공하는
- loads() 메서드를 이용하여 json 문자열을 딕트 객체로 변경한다. 명령어 형식은 다음과 같다. 형식) json.loads(json문자열)
- json 모듈의 디코딩 기능은 json 파일에 포함된 자료 즉 json 문자열을 파이썬으로 읽어올 때 적용 할 수 있다. 다음 예문은 json 파일을 읽어서 딕트 객체로 변환하는 과정이다.
- 예문에서 사용되는 usagov_bitly.txt 파일은 3,560개의 행의 길이를 갖는 JSON 형식의 파일이다. 파일의 일부분의 내용은 다음과 같다.

1. JSON 파일

```
{ "a": "Mozilla\5.0 (Windows NT 6.1; WOW64) AppleWebKit\535.11  
(KHTML, like Gecko) Chrome\17.0.963.78 Safari\535.11", "c": "US",  
"nk": 1, "tz": "America\New_York", "gr": "MA", "g": "A6qOVH", "h":  
"wfLQtf", "l": "orofrog", "al": "en-US,en;q=0.8", "hh":  
"1.usa.gov", "r":  
"http:\\\\www.facebook.com\1\7AQEFzjSi\1.usa.gov\wfLQtf", "u":  
"http:\\\\www.ncbi.nlm.nih.gov\pubmed\22415991", "t": 1331923247,  
"hc": 1331822918, "cy": "Danvers", "ll": [ 42.576698, -70.954903 ]  
}  
  
{ "a": "GoogleMaps\RochesterNY", "c": "US", "nk": 0, "tz":  
"America\Denver", "gr": "UT", "g": "mwszkS", "h": "mwszkS", "l":  
"bitly", "hh": "j.mp", "r": "http:\\\\www.AwareMap.com\1", "u":  
"http:\\\\www.monroecounty.gov\etc\911\rss.php", "t":  
1331923249, "hc": 1308262393, "cy": "Provo", "ll": [ 40.218102,  
-111.613297 ] }
```

1. JSON 파일

- JSON 파일에서 한 줄은 {} 기호로 표시되고, {} 기호 안에는 "키":"값" 형식의 원소들로 구성된다

```
{ "a": "Mozilla\5.0 (Windows NT 6.1; WOW64) AppleWebKit\535.11  
(KHTML, like Gecko) Chrome\17.0.963.79 Safari\535.11", "c": "US",  
"nk": 0, "tz": "America\New_York", "gr": "MA", "g": "9b6kNl", "h":  
"9b6kNl", "l": "bitly", "al": "en-US,en;q=0.8", "hh": "bit.ly",  
"r": "http:\\\\www.shrewsbury-ma.gov\\selco\\", "u":  
"http:\\\\www.shrewsbury-ma.gov\\egov\\gallery\\134127368672998.png  
", "t": 1331923251, "hc": 1273672411, "cy": "Shrewsbury", "ll":  
[42.286499, -71.714699 ] }  
{ "a": "Mozilla\5.0 (Windows NT 6.1; WOW64) AppleWebKit\535.11  
(KHTML, like Gecko) Chrome\17.0.963.79 Safari\535.11", "c": "US",  
"nk": 0, "tz": "America\New_York", "gr": "MA", "g": "axNK8c", "h":  
"axNK8c", "l": "bitly", "al": "en-US,en;q=0.8", "hh": "bit.ly",  
"r": "http:\\\\www.shrewsbury-ma.gov\\selco\\", "u":  
"http:\\\\www.shrewsbury-ma.gov\\egov\\gallery\\134127368675393.png  
", "t": 1331923252, "hc": 1273672506, "cy": "Shrewsbury", "ll":  
[42.286499, -71.714699 ] }  
생략 ...
```

2. JSON 파일 실습

chapter08.lecture.step07_json_file.py ↵

Python Console ↵

```
# (1) json file 읽기 ↵
file = open("chapter08/data/usagov_bitly.t
xt", ↵
mode='r', encoding='utf-8') ↵
lines = file.readlines() # 줄 단위 전체 읽기 ↵

# (2) json 디코딩 ↵
# file(json 문자열) -> python dict 객체 ↵
rows = [ json.loads(row) for row in lines ↵
] ↵
print('rows :', len(rows)) # rows : 3560 ↵ rows : 3560 ↵

# (3) 10개 원소 출력 ↵
for row in rows[:10] : ↵
    print(row) # 행 출력 ↵
    print(type(row)) # <class 'dict'> ↵
file.close() ↵

# (4) dict -> DataFrame 변환 ↵
import pandas as pd ↵
recode_df = pd.DataFrame(rows) ↵
print(recode_df.info()) ↵
```

<class
'pandas.core.frame.DataFrame'
>↵
RangeIndex: 3560 entries, 0↵
to 3559↵
Data columns (total 18 colum
ns):↵

2. JSON 파일 실습

- json 파일 디코딩 예
- # (1) json file 읽기
- usagov_bitly.txt 파일을 대상으로 줄 단위 전체 자료를 읽어온다. 즉 해당 파일을 줄 단위 문자열로 읽어온다.
- # (2) json 디코딩
- 줄 단위로 읽어온 문자열을 대상으로 loads() 메서드를 이용하여 파이썬의 딕트 객체로 변환한다. 리스트 내포에서 실행되는 과정은 다음과 같다.
- 줄 단위 읽기(json string) -> 딕트 객체 변환 -> 리스트 저장
- 리스트의 원소 길이는 JSON 파일의 전체 줄과 같다. 즉 리스트 원소 하나는 JSON 파일의 한 줄에 해당된다.

2. JSON 파일 실습

- # (3) 10개 원소 출력
- json 디코딩에 의해서 리스트의 원소들이 모두 'JSON 문자열'에서 딕트 객체로 변환된 것을 확인할 수 있다.
- # (4) DataFrame 변환
- json 디코딩의 결과를 저장한 rows를 인수로 pandas의 데이터프레임 형식으로 객체를 생성하면 행과 열의 2차원 자료구조로 변환된다. 데이터프레임 객체의 정보를 출력하면 JSON 파일의 전체 행수 3,560개와 각 행을 구성하는 열의 수 18개를 확인할 수 있다.
- ※ {'키:값', '키':'값', ...} 형식의 딕트 객체 보다 데이터프레임 형식으로 변환하면 자료의 가독성이 높아지고, 또한 데이터프레임 객체에서 제공하는 다양한 함수를 이용할 수 있기 때문에 여러 가지 측면에서 유용하다.

THANK YOU