

4-10. 여러가지 파일 포맷 다루기 (3) JSON 파일

JSON

JSON은 JavaScript Object Notation의 약자로, 웹 언어인 JavaScript의 데이터 객체 표현 방식입니다. 웹 브라우저와 다른 애플리케이션 사이에서 HTTP 요청으로 데이터를 보낼 때 널리 사용하는 표준 파일 포맷중 하나로, XML과 더불어 웹 API나 config 데이터를 전송할 때 많이 쓰입니다.

대표적인 예로 트위터는 [개발자용 사이트](#)를 통해 여러가지 API를 JSON 형태로 제공합니다.

JSON 데이터의 예시를 함께 살펴 봅시다.

```
person = {
  "first name" : "Yuna",
  "last name" : "Jung",
  "age" : 33,
  "nationality" : "South Korea",
  "education" : [{"degree":"B.S degree", "university":"Daehan university", "major": "mechanical engineering", "graduated year":2010}]
}
```

얼핏 보아도 파이썬의 dictionary 타입과 매우 유사한 구조를 가지고 있는데요. CSV 파일에 비해 좀 더 유연하게 데이터를 표현할 수 있고 XML 파일보다 파일을 쉽게 읽고 쓸 수 있다는 장점이 있습니다. 뿐만 아니라 Javascript로 작성된 프로그램에서 쉽게 다룰 수 있다고 하는데요. 웹에서 JavaScript나 JavaScript 기반의 Framework가 많이 사용되고 있는 점을 미루어 봤을 때 이는 큰 강점이 될 수 있습니다.

JSON 파싱

(1) json 파일 저장

파이썬 dictionary 타입은 JSON으로 저장할 수 있습니다. 위 예시로 보인 person이라는 dict 객체를 JSON 파일로 저장해 보겠습니다.

[Input]

```
import json

person = {
    "first name" : "Yuna",
    "last name" : "Jung",
    "age" : 33,
    "nationality" : "South Korea",
    "education" : [{"degree": "B.S degree", "university": "Daehan university", "major": "mechanical engineering", "graduated year": 2010}]
}

with open("person.json", "w") as f:
    json.dump(person, f)
```



실행 ▶

[Output]

(2) json 파일 읽기

반대로, JSON 파일의 내용을 아래처럼 파이썬 dict 객체로 쉽게 읽어들이 수 있습니다.

[Input]

```
import json

with open("person.json", "r", encoding="utf-8") as f:
```



```
contents = json.load(f)
print(contents["first name"])
print(contents["education"])
```

실행 ▶

[Output]