

# 파이썬 JSON처리



# JSON(JavaScript Object Notation)

- JSON(JavaScript Object Notation)

- 데이터를 교환하는 포맷으로 키-값의 쌍으로 이루어진 컬렉션
- 데이터 포맷이 단순하고 유연함
- 웹 상에서 데이터를 교환하는 데 많이 사용되고 있음

- Python JSON 표준 라이브러리

- import json
- JSON 인코딩
  - Python 타입의 데이터를 JSON 문자열로 변경
- JSON 디코딩
  - JSON 문자열을 Python 타입으로 변환



# JSON 인코딩

- `json.dumps()`

- 딕셔너리, 리스트, 튜플을 JSON 문자열로 변경
- 반환값은 JSON 표현을 갖는 문자열(str 타입)
- "indent" 옵션
  - `json.dumps(파이썬자료, indent=4)`
  - JSON 문자열을 읽기 편하게 Indentation이 적용된 문자열 반환

# JSON 인코딩

```
import json
jdata = {
    "request_id": "0",
    "return_type": "omAnalysis",
    "result": 0,
    "reason": "",
    "return_object": {
        "query": "한국의 가을은 매우 아름답습니다.",
        "type": "감성분석",
        "score": 0.9999995231628418,
        "label": "긍정"
    },
    "result_code": "success"
}
print(type(jdata))

jsonStr = json.dumps(jdata)
print(jsonStr)

jsonStr1 = json.dumps(jdata, indent='\t')
print(jsonStr1)

jsonStr2 = json.dumps(jdata, indent='\t', ensure_ascii=False)
print(jsonStr2)

print(type(jsonStr2))
```

Diagram illustrating the JSON encoding process:

- `print(type(jdata))` outputs `<class 'dict'>`.
- `print(jsonStr)` outputs a compact JSON string: `{"request_id": "0", "return_type": "omAnalysis", "result": 0, "reason": "", "return_object": {"query": "\uc544\u2044\u204984\u2049b2f5\u2049uc2b5\u2049ub2c8\u2049ub2e4.", "type": "\uac10\u2049uc131\u2049ubd84\u2049uc1s"}, "result_code": "success"}`.
- `print(jsonStr1)` outputs a pretty-printed JSON string with tabs for indentation.
- `print(jsonStr2)` outputs a pretty-printed JSON string with tabs for indentation and UTF-8 encoding (using Korean characters).
- `print(type(jsonStr2))` outputs `<class 'str'>`.

# JSON 디코딩

- **json.loads()**

- JSON 문자열을 딕셔너리, 리스트, 튜플과 같은 Python 타입으로 변경

```
dt = json.loads(jsonStr2)
print(dt)
print(type(dt))

for key, value in dt.items() :
    print(key, value)

for key in dt :
    print(key, dt[key])

for key in dt :
    print(key, dt.get(key))
```

```
{'request_id': '0', 'return_type': 'omAnalysis', 'result': 0, 'reason': '성', 'score': 0.9999995231628418, 'label': '긍정'}, 'result_code': 'su'
<class 'dict'>
request_id 0
return_type omAnalysis
result 0
reason
return_object {'query': '한국의 가을은 매우 아름답습니다.', 'type': 'i
result_code success
```



# 파이썬 with...as

- with open(파일 경로, 모드) as 파일 객체:
  - 파일을 open한 후 with...as 구문이 끝나면 자동으로 close

```
import json

with open('data.json', 'r', encoding='utf-8') as f:
    data = f.read()

jdata = json.loads(data)
print(jdata)
print(type(jdata))

sdata = json.dumps(jdata, indent='\t', ensure_ascii=False)
print(type(sdata))

with open('data2.json', 'w', encoding='utf-8') as f:
    f.write(sdata)
```



# 웹 동작 방식



# 파이썬 urllib

- URL과 웹 요청에 관련된 모듈들 패키지로 묶어 제공
  - urllib.parse:
    - URL 해석·조작 기능을 담은 모듈
    - 예)url의 한글 처리  
parse.quote("한국의 가을은 매우 아름답습니다")
  - urllib.request:
    - HTTP 요청 기능을 담은 모듈





# urllib.request 모듈

- **urllib.request.urlopen() 함수**
  - 웹 서버에 정보를 요청한 후, 돌려받은 응답을 저장하여 '응답 객체(HTTPResponse)'를 반환
- **반환된 응답 객체의 read() 메서드를 실행**
  - 웹 서버가 응답한 데이터를 바이트 배열로 읽어들이м
  - 읽어들이인 바이트 배열
    - 이진수로 이루어진 수열이어서 텍스트 형식의 데이터를 decode() 메서드를 실행하여 문자열로 변환

# Byte 데이터

- 1바이트(0~255사이 코드)로 표현되는 문자 표현
- 문자열에서 사용하는 연산을 거의 제공
  - 인덱싱/슬라이싱, In, Upper(), Split()
- 지원 메소드
  - 바이트->문자열로 변환 : decode(인코딩값)
  - 문자열->바이트로 변환 : encode()

```
1  cstr = "안녕하세요"
2  bstr = cstr.encode()
3
4  print("-"*20)
5  print(type(cstr))
6  print(cstr)
7
8  print("-"*20)
9  print(type(bstr))
10 print(bstr[0])
11 print(str(bstr)[0])
12 print(bstr.decode("ascii", errors="replace"))
13 print(bstr.decode("utf-8"))
```

-----  
<class 'str'>

안녕하세요

-----  
<class 'bytes'>

236

b

????????????????

안녕하세요



# 해결문제

- 인공지능 오픈API를 이용하여 감성&감정 분석을 해보시오.

— <https://www.adams.ai/apiList>

감성분석문장을 입력하시오.=>인간을 비롯해 지상에 사는 생물에게 어떤 변화가 있을지는 아직 명확하지 않다.

부정

계속하시겠습니까?(y/n)y

감성분석문장을 입력하시오.=>코로나로 바뀐 언택트 사회, 일상에 맞게 디지털 라이프 패러다임을 혁신할 새로운 사회적 가치 활동을 추진하겠다

긍정

계속하시겠습니까?(y/n)n



# 해결문제2

- 인공지능 오픈API를 이용하여 두 문장을 입력 받아서 유사도 검사를 실시 하시오.

— <https://www.adams.ai/apiList>

첫번째 입력 문장=>한국의 가을은 매우 아름답습니다.

두번째 입력 문장=>한국의 가을은 매우 좋습니다.

0.7795010355900855

계속하시겠습니까?(y/n)y

첫번째 입력 문장=>마침내 도킹까지 성공

두번째 입력 문장=>작년 3월 인형 태우고 첫 도킹 시험 성공

0.40176480550409405

계속하시겠습니까?(y/n)n

