

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))
```

<class 'dict'>

```
{
  "request_id": "0",
  "return_type": "omAnalysis",
  "result": 0,
  "reason": "",
  "return_object": {
    "query": "\uc544\ub984\ub2f5\uc2b5\ub2c8\ub2e4.",
    "type": "\uac10\uc131\ubd84\uc1s"
  },
  "result_code": "success"
}
```

<class 'str'>

```
{
  "request_id": "0",
  "return_type": "omAnalysis",
  "result": 0,
  "reason": "",
  "return_object": {
    "query": "\ud55c\uad6d\uc758 \uac00\uc744\uc740 \ub9e4\u"
    "type": "\uac10\uc131\ubd84\uc11d",
    "score": 0.9999995231628418,
    "label": "\uae0d\uc815"
  },
  "result_code": "success"
}
```

<class 'str'>

```
{
  "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"
}
```

<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
```

