

JSON

🕒 생성일	@2023년 8월 9일 오후 8:47
📅 날짜	@2023년 8월 9일
# 폴더명(DayOO)	21
⋮ 태그	

JSON(JavaScript Object Notation)

- Javascript 객체 문법으로 구조화된 데이터를 표현하기 위한 문자 기반의 표준 포맷
 - javascript 객체 리터럴 문법을 따르는 문자열
- 웹 어플리케이션에서 데이터를 전송할 때 일반적으로 사용
 - 서버에서 클라이언트로 데이터를 전송하여 표현하려거나 반대의 경우
- 네트워크를 통해 전송할 때 유용
 - 비동기 브라우저/서버 통신(AJAX) 을 위해, 넓게는 XML을 대체하는 주요 데이터 포맷
- 데이터를 저장하거나 전송할 때 많이 사용되는 경량의 데이터 교환 형식
- JSON 표현식은 사람과 기계 모두 이해하기 쉬우며 용량이 작음
- 데이터에 접근하기 위해서는 네이티브 JSON 객체로 변환될 필요가 있음
 - 파싱(Parsing)
 - 문자열에서 네이티브 객체로 변환하는 것
- JSON은 데이터 포맷일 뿐이며 어떠한 통신 방법도, 프로그래밍 문법도 아닌 단순히 데이터를 표시하는 표현 방법일 뿐
- 특징
 - 서버와 클라이언트 간의 교류에서 일반적으로 많이 사용
 - 자바스크립트 객체 표기법과 아주 유사
 - 자바스크립트를 이용하여 JSON 형식의 문서를 쉽게 자바스크립트 객체로 변환할 수 있음
 - JSON 문서 형식은 자바스크립트 객체의 형식을 기반으로 만들어짐
 - 자바스크립트의 문법과 굉장히 유사하지만 텍스트 형식일 뿐
 - 다른 프로그래밍 언어를 이용해서도 쉽게 만들 수 있음

- 특정 언어에 종속되지 않으며, 대부분의 프로그래밍 언어에서 JSON 포맷의 데이터를 핸들링 할 수 있는 라이브러리를 제공
- 데이터 유형

- 문자열

- JSON의 문자열은 유니코드 문자로 구성되며, 백슬래시(\) 이스케이프 문자를 사용
- example

```
{ "name" : "Jones" }
```

- 숫자

- JSON 숫자는 JavaScript의 배정도수 부동소수점 형식을 따름
- example

```
{  
  "number_1" : 210,  
  "number_2" : 215,  
  "number_3" : 21.05,  
  "number_4" : 10.05  
}
```

- 부울

- 부울 값은 `true` 또는 `false` 둘 중 하나로 지정
 - 따옴표로 묶이지 않으며 문자열 값으로 취급
- example

```
{ "AllowPartialShipment" : false }
```

- Null

- 빈 값
 - 키에 어떤 값도 할당되어 있지 않은 경우 Null로 취급
- example

```
{ "Special Instructions" : null }
```

- 객체

- JSON 객체 데이터는 {}(중괄호) 사이에 삽입된 한 쌍의 이름 또는 값
- 키는 문자열이어야 하며 쉼표로 구분되어야 함
- 중괄호({})로 둘러싸야 표현
- example

```
{
  "Influencer" :  { "name" : "Jaxon" ,  "age" : "42" ,  "city" ,  "New York" }
}
```

- 배열

- 배열 데이터 유형은 순서가 지정된 값의 모음
- JSON에서 배열 값은 문자열, 숫자, 객체, 배열, 부울 또는 **Null** 유형이어야 함
- 대괄호([])로 둘러싸야 표현
- example

```
{
  "Influencers" :  [
    {
      "name" : "Jaxon",
      "age" : 42,
      "Works At" : "Tech News"
    }
    {
      "name" : "Miller",
      "age" : 35
      "Works At" : "IT Day"
    }
  ]
}
```

- 구조

- Javascript의 기본 데이터 타입인 문자열, 숫자, 배열, 불리언 그리고 다른 객체를 포함할 수 있음
- name - value 형식의 쌍
 - { String key : String value }
- 값들의 순서화된 리스트 형식

- [value1, value2, ...]
- • 일반 자바스크립트의 객체처럼 원하는 만큼 중첩시켜서 사용할 수도 있음
- example

```
{
  "squadName": "Super hero squad",
  "homeTown": "Metro City",
  "formed": 2016,
  "secretBase": "Super tower",
  "active": true,
  "members": [
    {
      "name": "Molecule Man",
      "age": 29,
      "secretIdentity": "Dan Jukes",
      "powers": ["Radiation resistance", "Turning tiny", "Radiation blast"]
    },
    {
      "name": "Madame Uppercut",
      "age": 39,
      "secretIdentity": "Jane Wilson",
      "powers": [
        "Million tonne punch",
        "Damage resistance",
        "Superhuman reflexes"
      ]
    },
    {
      "name": "Eternal Flame",
      "age": 1000000,
      "secretIdentity": "Unknown",
      "powers": [
        "Immortality",
        "Heat Immunity",
        "Inferno",
        "Teleportation",
        "Interdimensional travel"
      ]
    }
  ]
}
```

XML 과의 비교

- XML
 - 데이터 값 양쪽으로 태그가 존재
- JSON
 - 중괄호({}) 같은 형식으로 하고, 값을 ','로 나열

- XML에 비해서 표현이 간단

▼ 참고

<https://developer.mozilla.org/ko/docs/Learn/JavaScript/Objects/JSON>

<https://www.oracle.com/kr/database/what-is-json/>

<https://ko.wikipedia.org/wiki/JSON>

<https://www.json.org/json-ko.html>

<https://velog.io/@surim014/JSON이란-무엇인가>

http://www.tcpschool.com/json/json_basic_structure

<https://codingazua.tistory.com/4>

<https://usefultoknow.tistory.com/15>

<https://docs.fileformat.com/ko/web/json/>

http://www.tcpschool.com/json/json_intro_basic