

분류: 데이터 직렬화 형식 | 파일 형식

목차

- 1. 개요
- 2. 구조
- 3. 특징
- 4. 단점
- 5. JSONP
- 6. Python에서 Json처리

공식 홈페이지

JavaScript Object Notation<sup>[1]</sup>

1. 개요

일반적으로 서버에서 클라이언트로 데이터를 보낼 때 사용하는 양식. 클라이언트가 사용하는 언어에 관계 없이 통일된 데이터를 주고받을 수 있도록, 일정한 패턴을 지닌 문자열을 생성해 내 보내면 클라이언트는 그를 해석해 데이터를 자기만의 방식으로 온전히 저장, 표시할 수 있게 된다.

과거 웹 초기 시절부터 사용되어 온 **XML**은 헤더와 태그 등의 여러 요소로 가독성이 떨어지고, 쓸데없이 용량을 잡아먹는다는 단점이 항상 지적되어 왔다. 이에 대응해 간결하고 통일된 양식으로 각광을 받고 있는 것이 JSON이다.

2. 구조

```
{
  "이름공간(키)": "값",
  "값 구분자": "각각의 값들은 ', ' (coma)로 구분되어야 합니다.",
  "이스케이프": "기나 값에서 큰따옴표를 쓰고 싶으면-특정 문자를 이스케이프 하려면- \" 처럼 문자 앞에 역슬래시를 붙입니다.",
  "자료형": "표현 가능한 자료형은 문자열, 숫자, 불리언, 널, 객체, 배열 6개입니다.",
  "문자열 값": "나무위키, 여러분이 가꾸어 나가는 지식의 나무",
  "숫자 값": 19721121,
  "불리언 값": true,
  "널 값": null,
  "객체 값": {
    "값1": 3.141592653589793,
    "값2": false,
    "값3": {
      "객체 안에": "객체를 넣는것도 가능하지요",
      "구분자": "또한 키와 값은 ':' 로 구분됩니다"
    }
  },
  "배열 값": [
    "이것은 배열입니다.",
    {
      "현재 값의 인덱스": 1,
      "이런 식으로": "배열 안에 여러 값을 넣을 수 있습니다."
    },
    [ "배열", "안에", "배열을", "넣는것도", "가능하지요" ]
  ],
  "값의 개수가 적을때는": "다음과 같이 한 줄로도 객체와 배열 표현이 가능합니다.",
  "한 줄 객체": { "김두한": "개소리 집어쳐", "심영": "내가 고자라니", "의사양반": "전화는 없어요" },
  "한 줄 배열": [ "나무위키는", "누구나", "기여할", "수", "있는", "위키입니다." ]
}
```

이게 무슨말이냐면 { } 안의 것은 속성명<sup>[2]</sup>이 있는 객체를 의미하며, [ ] 안의 것은 순서가 있는 배열<sup>[3]</sup>, 객체 안에 객체를 넣을 수도 있어서 XML처럼 복잡한 구조 또한 표현이 가능하다. 표현할 수 있는 값의 자료형은 { }로 표기되는 객체, [ ]로 표기되는 **배열**, 문자열, 숫자, **불리언**, null의 6가지가 전부다. 또한 null의 존재에서 알 수 있듯, JSON은 JavaScript 이외의

언어에서 사용될 상황을 다분히 고려하고 있다.<sup>[4]</sup>

root 값은 객체 뿐만 아니라 배열로도 표현 가능하다<sup>[5]</sup>.

```
[
  { "나무위키": "여러분이 가꾸어 나가는 지식의 나무" },
  { "위키백과": "우리 모두의 백과사전" },
  { "백과사전": "너희 모두의 백과사전" },
  { "위키낱말사전": "말과 글의 누리" }
]
```

### 3. 특징

그래도 표면적으로 W3C의 표준은 XML인데, 그것과는 무관하게 JSON은 2009년 말 Ecma에 의해 ECMAScript5에서 스크립트 엔진의 기본 기능으로 내장되어 버렸기 때문에 새롭게 출시되는 브라우저들 기준에서는 DOM을 통해 XML 파싱하는 것보다 JavaScript 엔진에서 JSON을 메모리로 받는 쪽이 성능으로 보나 트래픽으로 보나 훨씬 더 나은 선택이 되어버렸다. 그런 관계로 웹에서 XML은 본격 기록화가 진행중...

기본적으로 JavaScript 객체 표기법의 부분집합이기 때문에 웹 브라우저 레벨에서 참 쉽게 해석할 수 있으며, 모양과 규칙 자체가 단순한 관계로 다른 언어에서 구현하기도 쉽다. 그래서 오늘날 사용되는 거의 모든 프로그래밍 언어에서 사용가능하다. 이러한 지원에 힘입어 AJAX 구현에도 상당히 빈번하게 쓰인다.

엄청나게 단순하지만 유연한 표기법이기 때문에 언어와 독립된 표준적인 데이터 표기법으로 확산되고 있다. 본래의 범위를 넘어서 바이너리 데이터 표기용으로 마개조한 BSON<sup>[6]</sup>도 있으며, NoSQL DB중 DocumentDB들에서는 그냥 표준 표기법으로 취급되고 있다. 최근에는 NoSQL뿐만 아니라 관계형DB인 PostgreSQL에서도 지원하기 시작. MySQL에서도 5.7에서 공식 지원한다.

그러나 확산되고 있다고 해도 어디까지나 웹 브라우저가 주가 되는 환경, 특히 사용자에게 보여주는 HTML 과 이를 실시간으로 가공하는 JavaScript가 연동하는 환경 위주로 퍼지고 있으며, 그렇지 않은 곳에서는 여전히 XML을 사용하는데 문제가없다. XML 파서도 다양한 플랫폼에 구현돼있고 XML이란 놈이 딱히 무슨 결함이 있는 것도 아니며 결정적으로 XML에는 스키마(Schema)가 있어서 데이터의 무결성을 검증할 수 있다. JSON은 데이터의 무결성을 검증하는 역할을 전적으로 그걸 사용하는 측에 떠맡긴다. 다만 JSON이 웹을 시작으로 편의성이 널리 알려지면서 최근 들어 다양한분야로 많이 쓰이는중이다.

그리고 JSON도 적용되는 곳이 늘어남에 따라 XML과 같이 valid한 검증이 필요해졌으며, Namespace 문제가 발생하는 경우가 늘고 있다. 이를 극복하기 위해 JSON-Schema, JSON-LD (JSON Linking Data) 같은 것이 등장하고 있다.

그러나 대응책으로 제시된 JSON 스키마는 현재 거의 쓰이지 않고 있다. 웹이라는 환경 특성상 데이터가 좀 잘못 들어간다고 해서 특별히 큰일이 나는 게 아니기 때문이다. 게다가 데이터가 잘못 들어가면 프로그램이 강제 종료한다거나 하는 크리티컬한 응용에서는 그냥 XML을 사용하고 있다. JSON은 콤마가 누락되거나 중괄호가 잘못 닫히는 등의 오염에 아주 취약하다. XML 은 SAX파서를 사용해서 이런 데이터 오류를 바로잡고 재동기화가 쉽다.

Unity3D 엔진에서 플러그인으로 시리얼라이저와 디시리얼라이저를 제공하기 시작하면서 모바일 플랫폼에서도 대거 사용되기 시작한다. CSV 등 기존 포맷에 비해 태그에 의한 데이터 확장 과 변형이 용이하고 과거에 비해 데이터 사용량이 자유롭기 때문에 이런 현상이 가속화 되고 있다. (단적인 예로 많은 게임 데이터들이 제이슨으로 되어있다.)

Sublime Text, Visual Studio Code에서는 환경 설정 파일에 JSON을 사용하고 있다.

ECMAScript는 5부터 Trailing comma(후행 쉼표)를 지원하지만 JSON은 그 이전에 규격이 정해졌기 때문에 Trailing comma를 지원하지 않는다.

⇒ <http://www.json.org>에서 표기법과 언어별 지원 라이브러리를 볼 수 있다.

⇒ [참고](#)

### 4. 단점

XML이나 다른 데이터 기술 형식에 비해 아래와 같은 단점이 있다.

- 문법 오류에 민감하다. 콤마가 누락되거나 중괄호가 잘못 닫히는 등 구두점에서 오타가 나면 전체 JSON파일이 망가진다. 찾기 기능이 없는 편집기에서 줄 길이가 길어지면 콤마가 누락 된 곳을 찾기 힘들다.
- 주석을 지원하지 않는다. 그래서 설정 파일을 JSON으로 작성하는 것은 어렵다. <sup>[7]</sup>
- 데이터 타입을 강제할 수 없다. JSON 스키마로 보완은 가능하지만 데이터 스스로 자신의 타입을 기술할 방법이 없다.

위와 같은 단점을 보완하고자 YAML 등의 대체 포맷을 쓰기도 하나 기계간 데이터 전송에 주로 쓰이는 JSON의 특성상 위의 단점은 큰 문제가 아니라서 JSON은 2017년 11월 현재도 XML 과 대등하게 널리 쓰이고 있다.

### 5. JSONP

OpenAPI 서비스 중 CORS<sup>[8]</sup>가 설정되지 않은 구형 서비스에서 JSON 데이터를 돌려줄 때 사용하는 데이터 형식이다. P는 패딩(Padding)의 약어로, HTML script 태그를 이용하면 CORS를 우회하여 외부 스크립트를 실행할 수 있다는 점을 이용해 JSON 데이터 객체를 클라이언트가 지정한 callback 함수를 호출하는 유효한 JavaScript 스크립트 형식으로 패딩하여 돌려준다. callback 함수의 이름은 요청자가 정해서 서버에 넘겨준다. 예로 jQuery의 jsonp는 jQuery89034758903475같은 난수 이름을 자동으로 생성한다.

예를 들어 ⇨ [https://jsonplaceholder.typicode.com/posts/1?callback=\\_\\_cb12345](https://jsonplaceholder.typicode.com/posts/1?callback=__cb12345) 를 호출하면 아래와 같은 응답이 돌아온다.

```
/**/ typedef __cb12345 == 'function' && __cb12345({
  "userId": 1,
  "id": 1,
  "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
  "body": "quia et suscipitWnsuscipit recusandae consequuntur expedita et cumWnreprehenderit molestiae ut ut quas totamWnnostrum rerum est autem sunt rem
});
```

기본적으로 이 JSONP의 목적은 웹 브라우저 내에서의 CORS가 설정되지 않은 호스트와의 통신을 위한 것으로, 웹 브라우저 밖에서는 Same Origin 정책이 적용되지 않기 때문에 **JSONP**를 이용할 필요가 없다. 또한 API를 직접 건드릴 수 있다면 CORS 정책을 직접 설정하는 것이 좋다.

## ▼ 6. Python에서 Json처리

Python에서는 import json 모듈로 Json 파일의 로딩 및 저장을 할 수 있다. Python에서는 Json에 대응하는 데이터 타입을 dict로 사용하고 있으며, array는 list로, 문자열은 str로, 숫자는 int와 float으로 사용하고 있다.

- [1] 제이슨이라고 읽는다.
- [2] 키 또는 이름공간이라고도 불림
- [3] 내부적으로 0, 1, 2, ... 와 같은 숫자로 된 이름표(인덱스)가 자동으로 주어진다. 위 예제에 의하면 "0": "나무위키는", "1": "누구나", ... 의 순서로 이름표가 매겨진다.
- [4] JavaScript에서는 값이 없다는 뜻으로 undefined 키워드가 쓰이고 있다.
- [5] 사실 문자열이나 숫자, null등 모든 json값으로 쓰일 수 있는 자료형은 그 자체로도 유효한 json이다. 즉 "문자열" 또한 유효한 json형식이며, .json파일에 이 내용만 담겨져 있거나 이 상태로 요청을 보낼 수 있다는 뜻이다. 다만 실용성이 없기 때문에 쓰이지 않을 뿐이다.
- [6] MongoDB에서 쓰인다.
- [7] 단, 비공식 규격인 jsonc를 사용하면 된다. 또한 "\_comment"라는 키를 사용하는 경우도 있다.
- [8] 도메인이 서로 다른 곳끼리의 통신을 위한 규약.

파워링크 **광고**



**비교해봐도 커플링은 커플이야**  
www.couple2ya.com **N 로그인**  
33~45% 할인 실버 14K 18K 24K 모던커플링 평생AS,중  
로매장 방문할인

**커플링 싸곳은 SD DiA**  
www.sddiamond.co.kr  
오늘만 커플링 특가가 판매.최신디자인 많음.비교 방문 대환영.주차는 무  
료고 편함.



**효성주얼리시티 예뵌초대전**  
www.vipgold.co.kr  
디자인예뵌공방,명품커플링,소비자피해규정엄수,다이렉트  
결혼준비,GIAdia 직수입공급



이 저작물은 [CC BY-NC-SA 2.0 KR](#)에 따라 이용할 수 있습니다. (단, 라이선스가 명시된 일부 문서 및 삽화 제외)  
기여하신 문서의 저작권은 각 기여자에게 있으며, 각 기여자는 기여하신 부분의 저작권을 갖습니다.

나무위키는 백과사전이 아니며 검증되지 않았거나, 편향적이거나, 잘못된 서술이 있을 수 있습니다.  
나무위키는 위키위키입니다. 여러분이 직접 문서를 고칠 수 있으며, 다른 사람의 의견을 원할 경우 직접 토론을 발제할 수 있습니다.