**[JSON - 소개]**

JSON 예제

[JSON 예제1](http://kinyang.kr/samples/JSON/json01.html)

**[XML과 유사한 점]**

평이한 텍스트

“self-describing" (사람이 읽을 수 있는)

계층 구조

Javascript로 파싱 가능

JSON 데이터는 AJAX로 전송 가능

**[XML과 다른 점]**

종료 태그가 없음

더 짧음

읽고 쓰는데 더 빠름

javascript의 내장 함수인 eval()를 사용해서 파싱

배열 사용

예약어 없음

**[왜 JSON인가?]**

AJAX 어플리케이션에서 JSON은 XML보다 빠르고 더 쉽습니다.

 (비동기 통신)

XML을 사용하면,

XML문서를 패치

XML DOM을 사용해서 문서를 루핑(loop)

값을 추출해서(extract) 변수에 저장

JSON을 사용하면,

JSON 문자열을 패치

JSON 문자열은 eval() 함수를 사용

**[JSON 문법의 규칙]**

JSON 문법은 javascript [오브젝트] 표기법의 부분집합입니다.

데이터는 name/value 쌍으로 되어 있습니다.

데이터는 ‘,’로 분리합니다.

중괄호로 오브젝트를 묶습니다.

대괄호로 배열을 묶습니다.

**[JSON Name/Value Pairs]**

JSON 데이터는 name/value 쌍으로 적습니다.

Name/Value 쌍은 필드명으로 이루어집니다.(큰따옴표 사용)

“firstName" : "태연"

이해하기 쉽게 javascript 문장으로 바꾸면 다음과 같습니다.

firstName = "김"

**[JSON Values]**

JSON 값은 다음이 될 수 있습니다.

숫자

문자열

Boolean

배열

오브젝트

null

**[JSON Objects]**

JSON 오브젝트는 중괄호로 둘러싸여집니다.

오브젝트는 여러 개의 name/value 쌍을 포함시킬 수 있습니다.

{ “firstName" : "태연" , "lastName" : "김" }

이해하기 쉽게 javascript 문장으로 바꾸면 다음과 같습니다.

firstName = "태연"

lastName = "김"

**[JSON Arrays]**

JSON 배열은 대괄호로 둘러싸여집니다.

배열은 다음의 예처럼 여러 개의 오브젝트를 포함할 수 있습니다.

{

“employees" : [

{ "firstName" : "지은" , "lastname" : "이" },

{ "firstName" : "윤아" , "lastname" : "임" },

{ "firstName" : "수연" , "lastname" : "정" }

]

}

위의 예처럼, 오브젝트 “employees"는 3개의 오브젝트를 포함하고 있는 배열입니다. 각 오브젝트는 성과 이름으로 이뤄진 사람의 기록입니다.

**[JSON은 Javascript 문법을 사용합니다]**

JSON은 Javascript 문법을 사용하기 때문에, JSON으로 작업할 때 별도의 소프트웨어가 필요없습니다.

Javascript를 사용해서 다음과 같이 오브젝트 배열을 생성하고 데이터를 할당할 수 있습니다.

var employees = [

{ "firstName" : "지은" , "lastName" : "이" },

{ "firstName" : "윤아" , "lastName" : "임" },

{ "firstName" : "수연" , "lastName" : "정" }

];

Javascript 오브젝트 배열에서 첫 번째 엔트리에 다음과 같이 접근할 수 있습니다.

employees[0].lastName;

반환된 내용은 다음과 같을 것입니다.

이 또한 데이터는 다음과 같이 수정할 수 있습니다.

employees[0].lastName = "수영";

다음 강좌에서는 JSON 텍스트를 Javascript 오브젝트로 바꾸는 방법을 알아보겠습니다.

**[JSON Files]**

JSON 파일의 형식는 “.json"입니다.

JSON 텍스트의 MIME 타입은 “application/json"입니다.

**[JSON 텍스트를 Javascript 오브젝트로 변환하기]**

JSON에서 가장 많이 사용하는 것 중에 하나가 JSON 데이터를 (파일형태나 HttpRequest를 사용하여) 웹서버에서 가져온 후, 이 JSON 데이터를 Javascript 오브젝트로 변환하고, 웹 페이지에서 이것을 사용하는 것입니다.

**[JSON Example - 문자열에서 오브젝트로]**

JSON 문법 형태를 하고 있는 Javascript 문자열을 만들어보면 다음과 같습니다.

var txt = '{ "employees" : ['

+'{ "firstName" : "지은" , "lastName" : "이" },'

+'{ "firstName" : "윤아" , "lastName" : "임" },'

+'{ "firstName" : "수연" , "lastName" : "정" } ]}';

JSON 문법은 Javascript 문법을 사용하기 때문에, Javascript 함수인 eval() 함수를 사용해서 JSON 텍스트를 Javascript 오브젝트로 변환하여 사용할 수 있습니다.

eval() 함수는 JSON 텍스트를 파싱하고 Javascript 오브젝트를 생성하는 Javascript 컴파일러를 사용합니다.

이때, 이 텍스트는 문법에러가 생기지 않게 하기 위해 괄호로 묶어줘야 합니다.

var obj = eval ( "(" + txt + ")" );

웹페이지에 Javascript 오브젝트를 다음과 같이 사용할 수 있습니다.

<p>

First Name: <span id="fname"></span><br />

Last Name: <span id="lname"></span><br />

</p>

<script type="text/javascript">

document.getElementById("fname").innerHTML = obj.employees[1].firstName

document.getElementById("lname").innerHTML = obj.employees[1].lastName

</script>

**[JSON Parser]**

eval() 함수는 어떠한 Javascript라도 컴파일하고 실행할 수 있습니다. 이것은 잠재적인 보안문제를 일으킬 수 있다는 말로도 표현할 수 있습니다.

그렇기 때문에 JSON 텍스트를 Javascript 오브젝트로 변환시킬 때는 JSON 파서를 사용하는 것이 더 안전합니다.

JSON 파서는 JSON 텍스트만 인식하며, 스크립트는 컴파일하지 않습니다.

Native JSON support를 지원하는 브라우저에서도 JSON 파서를 사용하는 것이 안전합니다.

Native JSON support는 상위 브라우저와 최신의 ECMAScript standard에 포함되어 있습니다.

|  |  |
| --- | --- |
| **Web Browsers Support** | **Web Software Support** |
| Firefox (Mozilla) 3.5  Internet Explorer 8  Chrome  Opera 10  Safari 4 | jQuery  Yahoo UI  Prototype  Dojo  ECMAScript 1.5 |

JSON이란 ?

JavaScript Object Notation의 약자로 서로 다른 프로그래밍 언어간에 데이터를 교환하기 위한 표기법으로 읽고 쓰기 쉽도록 고안해낸 데이터 표기법이다.

- Key와 Value로 구성

- []를 사용하여 배열형 Value사용가능

- JSON공식 사이트 - [http://json.org](http://json.org/)

표현 방법

key:value 유형 :

{ key : value, ... }

배열 유형 :

[배열]

혼합형

key:value 유형 + 배열 유형

**key:value 유형**

var 객체이름 = { key : value, ... }

객체이름.key 형태로 value값을 불러올수있으며 key를 배열의 요소값으로도 사용할수 있음.

예시 )

var kosta = { title:'백문이불여일타', blog:'[http://kosta.co.kr](http://javaking75.blog.me/)' }

**객체.이름 형태로 값 불러오기**

kosta.blog  //[http://kosta.co.kr](http://javaking75.blog.me/)

**객체['이름'] 형태로 값 불러오기**

kosta ['blog']  //[http://kosta.co.kr](http://javaking75.blog.me/)

**배열 유형**

[ value , value, value ]

[

 {

    "key1" : "value",

    "key2" : "value"

},

{

   "key1" : "value",

   "key2" : "value"

}

]

**혼합형**

{

    "key1" : "value",

    "key2" : ["item1", "item2"]

}

SON의 공식홈페이지 : <http://www.json.org/json-ko.html>

JSON(JavaScript Object Notation)의 약자로**JavaScript에서 객체를 만들때 사용하는 표현식**을 의미한다.

JSON (JavaScript Object Notation)은 **경량의 DATA-교환 형식**이다.

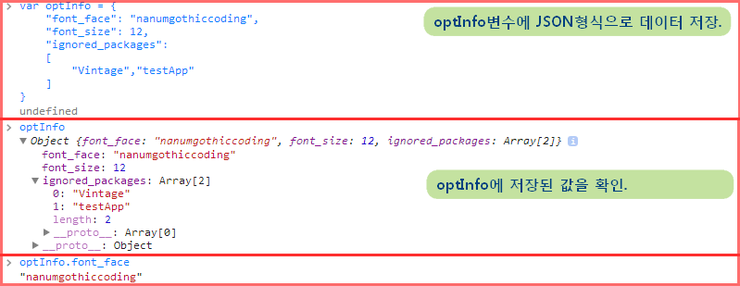
사람이 읽고 쓰기에 용이하며, 기계가 분석하고 생성함에도 용이하다. JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999의 일부에 토대를 두고 있다. JSON은 완벽하게 언어로 부터 독립적이지만 C-family 언어 - C, C++, C#, Java, JavaScript, Perl, Python 그외 다수 - 의 프로그래머들에게 친숙한 관습을 사용하는 **텍스트 형식**이다. 이러한 속성들이 JSON을 이상적인 DATA-교환 언어로 만들고 있다.

JSON은 두개의 구조를 기본으로 두고 있다:

* **name/value 형태의 쌍**으로 collection 타입. 다양한 언어들에서, 이는 object, record, struct(구조체), dictionary, hash table, 키가 있는 list, 또는 연상배열로서 실현 되었다.
* 값들의 순서화된 리스트. 대부분의 언어들에서, 이는 array, vector, list, 또는 sequence로서 실현 되었다.
* 이러한 것들은 보편적인 DATA 구조이다. 사실상 모든 현대의 프로그래밍 언어들은 어떠한 형태로든 이것들을 지원한다. 프로그래밍 언어들을 이용하여 호환성 있는 DATA 형식이 이러한 구조들을 근간에 두고 있는 것은 당연하다.

JSON 에서, 이러한 형식들을 가져간다:

* object는 **name/value 쌍**들의 비순서화된 SET이다. object는 { (좌 중괄호)로 시작하고 } (우 중괄호)로 끝내어 표현한다. 각 name 뒤에 : (colon)을 붙이고 , (comma)로 name/value 쌍들 간을 구분한다.


ECMAscript 5에는 JSON을 공식적으로 지원하는 API가 포함되었다.

JSON.parse()

인자로 전달된 문사열을 자바스크립트의 데이터(객체)로 변환한다.

### JSON.stringify()

인자로 전달된 자바스크립트의 데이터를 문자열로 변환한다.