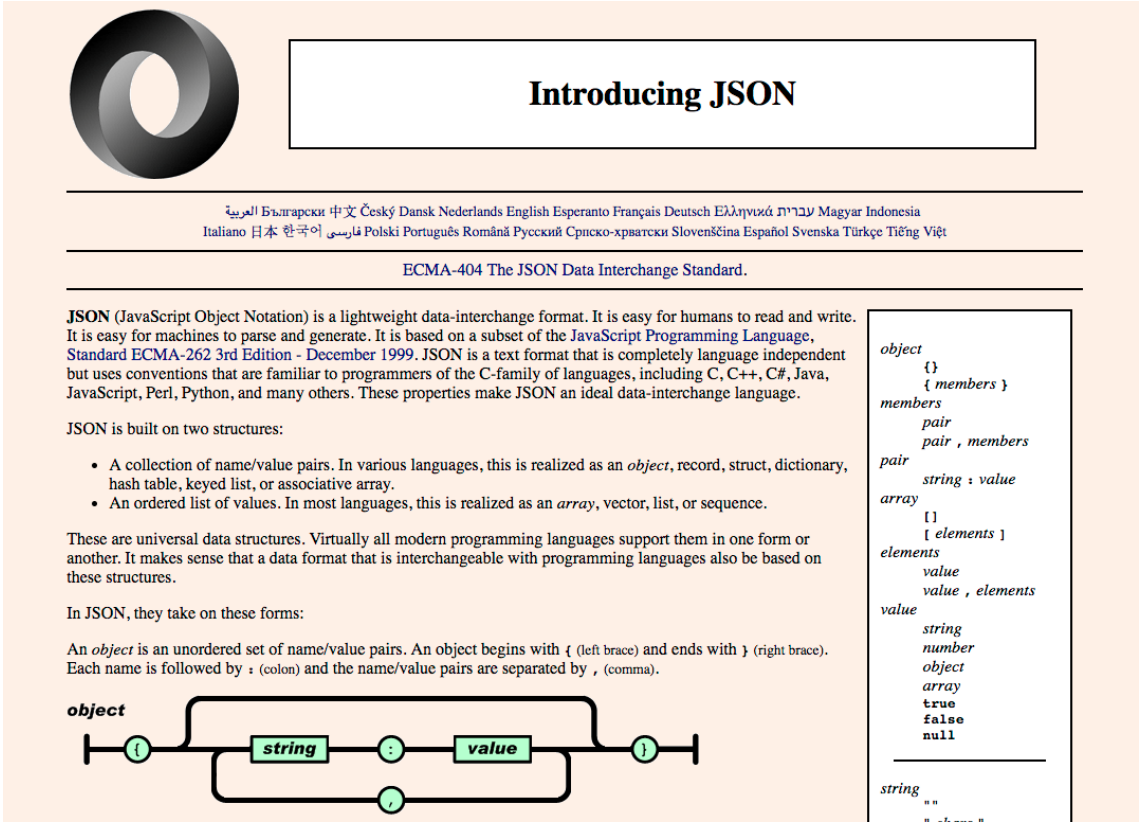
JSON

수업목표

- JSON 문법을 이해하고 처리 방법을 학습한다
- 오픈 API 의 자료 제공 형식으로 활용되는 JSON 을 이해한다



Introducing JSON

العربية Български 中文 Český Dansk Nederlands English Esperanto Français Deutsch Ελληνικά עברית Magyar Indonesia
Italiano 日本 한국어 فارسی Polski Português Română Русский Српско-хрватски Slovenščina Español Svenska Türkçe Tiếng Việt

ECMA-404 The JSON Data Interchange Standard.

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

In JSON, they take on these forms:

An *object* is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).

object
{}
{ *members* }
members
pair
pair , *members*
pair
string : *value*
array
[]
[*elements*]
elements
value
value , *elements*
value
string
number
object
array
true
false
null

string
""

www.json.org

JavaScript
문법에서 유래함
Python 사전문법
유사함

JSON 예

- JSON 은 JavaScript Object Notation 의 약어임
  - 정보는 객체들로 구성 됨
  - 객체는 name 과 value 의 쌍으로 표현함

예제 :

```
[
{
    "name":“홍길동",
    "city":＂서울",
    “books":[
        “태백산맥",
        ＂해리포터"
    ],
    "job":＂교사"
},
{
    "name":＂이순신＂,
    ＂city＂:＂천안",
    "books":[
        ＂라루토",
        “드레곤볼"
    ],
    "job":＂연구원"
}
]
```

JSON syntax, I

- 객체(*object*) 은 name/value 쌍의 집합
  - name/value 쌍은 { } 에 포함됨
  - name과 value 사이에는 : 을 사용함
  - name/value 쌍은 , 로 구분함
  - 예: {"name":"홍길동", "city":" 서울"}
- 배열(*array*) 값의 모임(collection)
  - 값은 [과] 사이에 작성함
  - 값은 , 로 분리됨
  - 예: "books":["태백산맥", " 해리포터"]

JSON syntax, I

```
who = {
    "name": 'Chuck',
    "age": 29,
    "college" : true,
    "offices" : [ '3350DMC', '3437NQ' ],
    "skills" : { "fortran": 10,
       "C": 10,
       "C++": 5,
       "python" : '7'
    }
};
```

String

Integer

Boolean

List/Array

Object

JSON – 자료형과 값

- JSON 의 자료형과 값
  - 문자열 (“ ” 또는 ‘ ’에 포함됨): "John"
  - 숫자: 30
  - 객체(JSON 객체):

    {

    "employee“: {"name":"John", "age":30, "city":"New York" }

    }
  - 배열: [" John ", " Anna ", " Peter "]
  - 불리언: true, false
  - 널: null