Fast campus

JSON
(Javascript Object Notation)

Giftbot

JSON

- Javascript 언어로부터 파생 (JavaScript Object Notation)

- 프로그래밍 언어와 플랫폼 간 독립적이고 가벼워서 XML 방식을 대체하여
  현재 거의 표준으로 사용되고 있는 데이터 교환 형식

- 두 개의 구조를 기본으로 가짐
  - 'Name : Value' 형태의 쌍을 이루는 콜렉션 타입. 각 언어에서 Hash table, Dictionary 등으로 구현
  - 값들의 순서화된 리스트. 대부분의 언어들에서 Array, Vector, List 또는 Sequence 로 구현

- XML 에 비해 기능이 적고 구조가 단순하여 파싱이 쉽고 빠르며 적은 용량으로 저장 가능
  따라서 사람이 읽고 쓰는 것뿐 아니라 기계가 분석하고 생성하는 것에도 (상대적으로) 더 용이

- contents type 은 application/json 이며, 파일 확장자는 .json, 기본 인코딩은 UTF-8 을 사용

JSON Format Example


```
{
    "message": "success",
    "number": 3,
    "people": [
        {
            "craft": "ISS",
            "name": "Anton Shkaplerov"
        },
        {
            "craft": "ISS",
            "name": "Scott Tingle"
        },
        {
            "craft": "ISS",
            "name": "Norishige Kanai"
        }
    ]
}
```

JSON vs XML

```
{"widget": {
    "debug": "on",
    "window": {
        "title": "Sample Konfabulator Widget",
        "name": "main_window",
        "width": 500,
        "height": 500
    },
    "text": {
        "data": "Click Here",
        "size": 36,
        "style": "bold",
        "name": "text1",
        "hOffset": 250,
        "vOffset": 100,
        "alignment": "center",
        "onMouseUp": "sun1.opacity = (sun1.opacity / 100) * 90;"
    }
}}
```

JSON vs XML

```
<widget>
    <debug>on</debug>
    <window title="Sample Konfabulator Widget">
        <name>main_window</name>
        <width>500</width>
        <height>500</height>
    </window>
    <text data="Click Here" size="36" style="bold">
        <name>text1</name>
        <hOffset>250</hOffset>
        <vOffset>100</vOffset>
        <alignment>center</alignment>
        <onMouseUp>
            sun1.opacity = (sun1.opacity / 100) * 90;
        </onMouseUp>
    </text>
</widget>
```

json.org

JSON 개요

How JavaScript Works by Douglas Crockford

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)은 경량의 DATA-교환 형식이다. 이 형식은 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 형식이다. 이러한 속성들이 JSON을 이상적인 DATA-교환 언어로 만들고 있다.

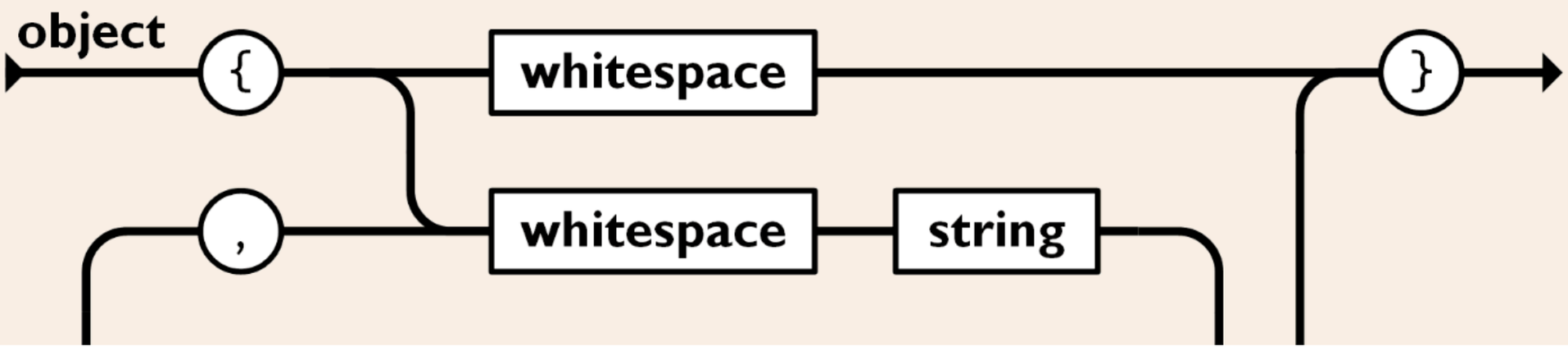
JSON은 두개의 구조를 기본으로 두고 있다:

name/value 형태의 쌍으로 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:

*object*는 name/value 쌍들의 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.




```
json
    element

value
    object
    array
    string
    number
    "true"
    "false"
    "null"

object
    '{' ws '}'
    '{' members '}'

members
    member
    member ',' members

member
    ws string ws ':' element

array
    '[' ws ']'
    '[' elements ']'

elements
    element
```

JSON in Swift

Fast campus

```
let jsonString = """
{
  "someNumber" : 1,
  "someString" : "Hello",
  "someArray"  : [1, 2, 3, 4],
  "someDict"   : {
    "someBool" : true
  }
}
"""
let jsonData = jsonString.data(using: .utf8)!
let jsonObject = try! JSONSerialization.jsonObject(with: jsonData)
print(jsonObject)
```

JSONSerialization

- JSON 과 이에 상응하는 Foundation 객체 간 변환하는 객체이며 iOS 7 이후로 Thread Safety

- Data 는 다음의 5가지 인코딩 지원 형식 중 하나여야 하며, 이 중 UTF-8 이 기본값으로 쓰이고 가장 효율적
  (UTF-8, UTF-16LE, UTF-16BE, UTF-32LE, UTF-32BE)

- JSON 으로 변환되기 위한 Foundation 객체는 다음 속성을 따라야 함
  - Top Level Object : NSArray, NSDictionary
  - 모든 객체는 NSString, NSNumber, NSArray, NSDictionary, NSNull 의 인스턴스
  - 모든 Dictionary 의 Key 는 NSString 인스턴스
  - 숫자는 NaN 이나 무한대 값이 아니어야 함

- JSON data 로 변환 가능 여부는 isValidJSONObject(_:) 메서드를 통해 확인 가능