

8월 28일 (토)

1. Codecademy - Async JavaScript and HTTP Requests - APIs and HTTP Requests

Article 3. Introduction to Web APIs

- What are APIs ?
- What can APIs do ?
- How do APIs work ?
- Summary

출처 : https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Introduction

Article 4. What Is JSON ?

- Introduction
- What is JSON ?
- Common Uses of JSON
- JSON Syntax
- JSON Data Types

- Introduction

데이터가 넘치는 세상에서 데이터가 어떻게 작동하는지 아는 것은 훨씬 더 중요해졌다. 프로그래머로서, 우리는 우리가 선택한 모든 언어의 채워진 데이터 구조를 다른 언어와 플랫폼에서 인식하고 읽을 수 있는 형식으로 전송할 수 있어야 한다. 다행히 그러한 데이터 교환 포맷이 존재한다.

- **What is JSON ?**

JSON (JavaScript Object Notation) 은 데이터를 저장하고 교환하기 위한 인기있는 언어 독립적 표준 형식이다. 정보 및 통신 시스템을 표준화하기 위해 1961년에 설립된 ECMA International 에서 채택한 JSON은 모든 프로그래밍 언어 간에 데이터를 저장하고 전송하는 것을 용이하게 하는 표준 형식이 되었다.

- **Common Uses of JSON**

JSON은 웹 브라우저와 같은 클라이언트와 서버 간의 웹 어플리케이션에서 데이터 전송을 용이하게 하는데 많이 사용한다. 데이터 전송이 발생하는 일반적인 예는 웹 form 을 입력할 때 발생한다. form 데이터는 HTML 에서 JavaScript 객체로, JSON 객체로 변환되고, 처리를 위해 원격 웹 서버에 보내진다. 일반적으로 데이터를 공유할 때 주고받는 정보는 JSON 형식이다. 즉, 객체형식으로 주고 받는다.

인기 있는 웹 API는 다음과 같다.

- [구글지도](#)
- [구글 인증 2.0 인증](#)
- [페이스북 소셜 그래프 API](#)
- [스포티파이 뮤직 웹 API](#)
- [LinkedIn 이력서 API](#)

- **JSON Syntax**

JSON은 JavaScript 프로그래밍 언어에서 파생되었기 때문에 모양이 JavaScript 객체와 유사하다.

아래 예시를 보자

```
{
  "student": {
    "name": "Rumaisa Mahoney",
    "age": 30,
    "fullTime": true,
    "languages": [ "JavaScript", "HTML", "CSS" ],
    "GPA": 3.9,
    "favoriteSubject": null
  }
}
```

- 중괄호로 객체를 구성한다.
- 대괄호로 배열을 구성한다.
- 데이터는 콜론(:) 으로 구분된 이름-값 쌍으로 저장된다.

- 모든 이름-값 쌍은 쉼표(,)로 다른 쌍과 구분한다.
- 후행쉼표는 금지한다. (마지막 이름-값 쌍 뒤에 쉼표를 붙이는 것)
- 프로퍼티 이름은 무조건 쌍따옴표(" ")로 써야한다. single - quoted(' ') 는 허용하지 않는다.

• JSON Data Types

데이터 유형은 다음 중 하나여야 한다.

- string (쌍따옴표 사용)
- number (정수 or 소수)
- object (이름- 값 쌍)
- array
- boolean
- null

JSON은 모든 데이터 유형을 다루지 않는다. 날짜와 같이 JSON 으로 표시되지 않는 유형은 문자열로 저장하고 언어별 데이터 구조로 변환할 수 있다. ISO 8601 에서 국제적으로 인정되는 날짜 형식은 다음과 같다.

```
"2014-01-01T23:28:56.782Z"
```

위 형식은 사실 문자열 그대로 읽기 힘들고 사용하기 힘든 형태이다. 편리하게도 모든 프로그래밍 언어는 위 스트링을 훨씬 더 읽기 쉽고, 사용하기 쉬운 형식으로 아래 예시와 같이 전환해주는 빌트인 JSON 기능을 가지고 있다.

```
Wed Jan 01 2014 13:28:56 GMT-1000 (Hawaiian Standard Time)
```

Article 5. Working with JSON in JavaScript

- Introduction
- JSON Object vs. JavaScript Object
- Reading a JSON String
- Exercise: Reading a JSON String
- Writing a JSON String
- Exercise: Writing a JSON String
- Review

- Introduction

JavaScript Object Notation의 약자 JSON은 업계 표준으로 받아들여진 언어 독립적인 데이터 형식이다. JavaScript 프로그래밍 언어를 기반으로 하기 때문에 JSON의 구문은 약간의 차이점이 있는 JavaScript 객체와 유사합니다. 우리는 그들 사이의 미묘한 차이점을 살펴볼 것입니다. 나중에 JSON을 구문 분석하고 콘텐츠를 JavaScript로 추출하는 방법을 배울 것이다. 마지막으로 JavaScript로 JSON 객체를 작성하는 방법을 배울 것이다.

- JSON Object vs. JavaScript Object

```
{
  "person": {
    "name": "Kate",
    "age": 30,
    "hobbies": [ "reading", "writing", "cooking", "tennis" ]
  }
}
```

```
{
  person: {
    name: 'Kate',
    age: 30,
    hobbies: [ 'reading', 'writing', 'cooking', 'tennis' ]
  }
}
```

- Reading a JSON String

JSON 파일을 자바스크립트로 읽는 법을 알아보자.

자바스크립트에서는 JSON 클래스안에 있는 .parse() 메서드로 JSON 객체를 javascript 객체로 변환해준다. 이를 변환하면 JSON 객체를 자바스크립트 객체처럼 사용할 수 있는 것이다.

```
const jsonData = '{ "book": { "name": "JSON Primer", "price": 29.99, "inStock": true, "rating": null } }';  
  
const jsObject = JSON.parse(jsonData);  
  
console.log(jsObject);  
  
//-----This will print out jsObject as follows:  
  
{  
  book: { name: 'JSON Primer', price: 29.99, inStock: true, rating: null }  
}
```

자바스크립트 객체로 변환한 JSON 객체는 다음과 같이 접근하여 사용할 수 있다. 자바스크립트 객체에 접근하는 방법과 같다.

```
// Using the dot notation  
const book = jsObject.book;  
console.log(book);  
console.log(book.name, book.price, book.inStock);  
  
// Using the bracket notation  
const book2 = jsObject['book'];  
console.log(book2);  
console.log(book2["name"], book2["price"], book2["inStock"]);  
  
//-----Both ways of accessing the book property return the same output:  
  
{ name: 'JSON Primer', price: 29.99, inStock: true, rating: null }  
JSON Primer 29.99 true
```

- **Exercise: Reading a JSON String**

```

1  const jsonData = '{ "parent": {
    "name": "Sally", "age": 45,
    "children" : [ { "name": "Kim",
    "age": 3 }, { "name": "Lee", "age": 1
    } ] } }';

2
3  const jsObject = JSON.parse(jsonData);
4
5  console.log(jsObject.parent.children);

```

```

[ { name: 'Kim', age: 3 }, {
  name: 'Lee', age: 1 } ]

```

- **Writing a JSON String**

우리는 웹을 통해 데이터를 어딘가로 보내기 전에, 자바스크립트 객체에 담긴 데이터를 JSON 스트링으로 바꿔야 한다. 자바스크립트에서 우리는 빌트인 JSON 클래스 메서드인 `JSON.stringify()` 메서드를 자바스크립트 객체를 JSON 스트링으로 변환하기 위해 사용할 것이다.

```

const jsObject = { book: 'JSON Primer', price: 29.99, inStock: true, rating: null };
const jsonData = JSON.stringify(jsObject);
console.log(jsonData);

//-----This will display the following output:

{ "book": "JSON Primer", "price": 29.99, "inStock": true, "rating": null }

```

- **Exercise: Writing a JSON String**

개발자는 변수에 JSON 문자열 형식의 일부 데이터를 받습니다 `jsonData`. 그러나 이 내용은 `jsonData` 완전히 정확하지는 않습니다. `age` 상위 속성의 값은 35 대신이어야 합니다 45. 이 내용을 `jsonData` 직접 변경하지 않고 `age` 값을 업데이트한 다음 콘솔에 올바른 값으로 새 JSON 문자열을 기록합니다.

이 문제를 해결하기 위한 단계별 가이드는 다음과 같습니다.

1. `jsonData`를 사용하여 JavaScript 객체로 변환 `JSON.parse()`하고 `const` 변수로 저장합니다(예: `jsObject`).
2. 점, `.key` 또는 대괄호, `['key']` 표기법을 사용하여 `parent` 속성 `jsObject` 다음에 속성에 액세스하고 `age` 해당 값을에서 45로 변경 합니다 35.
3. 를 `jsObject` 사용하여 JSON 문자열로 다시 변환 `JSON.stringify()`하고 다른 `const` 변수로 저장합니다(예: `jsObjectToJson`).
4. `jsObjectToJson` 콘솔에 문자열을 기록합니다 .

```
1  const jsonData = '{"parent":
   {"name":"Sally","age":45,"children":
   [{"name":"Kim","age":3}, {"name":"Lee",
   "age":1}]}';
2
3
4  const jsObject = JSON.parse(jsonData);
5
6  jsObject.parent.age = 35;
7
8  const jsObjectToJson = JSON.stringify
   (jsObject);
9
10 console.log(jsObjectToJson);
```

• Review

JSON 과 JavaScript 객체 문법의 차이점을 비교하였다.

JSON string 을 JavaScript 객체로 전환시키는 법 - `JSON.parse(jsonData)`

JavaScript 객체를 JSON string 으로 전환시키는 법 - `JSON.stringify(jsObject)`

오늘의 단어

- inundate : 넘치다, 범람하다 ; 쇄도하다, 듬뿍받다.