

# JSON    JavaScript Object Notation    자바스크립트 객체 표기법 (2019-06-15)

## JSON object, JSON 객체, JSONP

Top > 통신/네트워킹 > 인터넷/데이터통신 > 웹기술 > 웹 서비스

Top > 정보기술(IT) > 프로그래밍 > 프로그래밍언어 종류 > 자바스크립트 > JS 웹문서 제어 > Ajax

### 1. JSON (JavaScript Object Notation)

- 자바스크립트에서의 객체 표기 방법을 기반으로 한 데이터 형식
  - 다양한 형식으로 구조화된 데이터를 직렬화시킨 텍스트 포맷
  - 각 데이터(토큰)를 `속성(Property)`과 `값(Value)`의 쌍으로 표현
- 비록 명칭에서 Javascript가 붙었지만,
  - 자바스크립트 이외에도, 다양한 응용 프로그램 또는 웹 서비스 또는 서버/클라이언트 간의 가벼운 데이터 교환에 이용됨
- 표준 : RFC 8259 (The JavaScript Object Notation (JSON) Data Interchange Format)

### 2. JSON 특징

- 문법이 직관적이고, 가벼움
- 원래는, 웹 서버와 웹 브라우저의 자바스크립트 간에 데이터 전송의 수단으로 쓰였음
- 현재는, 웹 상에서 구조화된 문서 교환을 위한 공통 포맷으로 정착되고 있음
  - 폭넓은 언어 및 라이브러리에서 지원됨

### 3. JSON 데이터 형식/형태

- ※ 객체 리터럴 표기법을 기반으로 함
  - 각 데이터(토큰)는, 콜론(:)으로 구분되는, `속성`과 `값`으로 구성 됨
- `속성`은, 콜론 좌측에 큰 따옴표(") 로 둘러싸인 문자열
  - 작은 따옴표(') 사용 불가
- `값`은, 다음과 같은 종류들이 가능
  - 문자열 : 큰따옴표(")로 둘러싸여 표현됨
  - 숫자 : 그냥 숫자 자체로 표현됨
  - 진리값 : true 또는 false
  - null
  - 배열 : 대괄호([ ])로 둘러싸이며, 각 요소는 쉼표(,)로 구분되어 표현됨
  - 객체 : 중괄호({ })로 둘러싸여 표현됨
- ※ JSON 데이터(토큰) 사이사이에,
  - 띄어쓰기, 탭으로 보기 좋게 표현 가능하나, 주석과 같은 설명문 요소는 불가

#### 4. JSON 데이터 처리를 위한 언어별 함수/메소드

##### ○ 자바스크립트 메소드

- objJS = parse(jsonString, function (key, value) { })
  - . jsonString : JSON 형식의 문자열 (객체 리터럴)
  - . key,value : 키,값의 쌍으로된 매개변수
  - . JSON 형식 데이터를 자바스크립트 객체 형식으로 반환
    - .. 즉, JSON 데이터를 자바스크립트 객체 그대로 사용이 가능
- jsonString = stringify(objJS)
  - . 자바스크립트 객체(objJS)를 JSON 형식의 문자열로 변환시켜 줌

##### ○ PHP 함수

- PHP 배열 = json\_decode(jsonString)
- jsonString = json\_encode(PHP 배열)

#### 5. JSON 사용 예

##### ○ PHP

- 임의 형식의 데이터
  - . 객체 형식 예) aString = { "name1" : "value1" , "name2" : "value2" , ... }
  - . 배열 형식 예) aString = [ "name1" : "value1" , "name2" : "value2" , ... ]

- 임의 형식의 데이터 -> PHP 배열
  - . \$arr = json\_decode ( \$aString );

- PHP 배열 -> JSON 형식의 데이터
  - . HTML 헤더 (server -> client)  
<meta http-equiv='content-type' content='application/json'>  
<meta charset='utf-8'>
    - . echo json\_encode ( \$arr, JSON\_UNESCAPED\_UNICODE );

##### ○ Javascript

- JSON 형식의 데이터(jsonString) -> 자바스크립트 객체(objJS)
  - . var objJS = JSON.parse ( jsonString );
- 자바스크립트 객체(objJS) -> JSON 형식의 데이터(jsonString)
  - . var jsonString = JSON.stringify ( objJS );

- ajax 호출 예 (client -> server)  
\$.ajax({

url: url,  
type: 'POST',

```
contentType:'application/json',  
dataType:'json',  
data: JSON.stringify(obj),  
success: function(result){  
    ...  
}  
});
```

## 6. JSONP (JSON with Padding)

- (동일 출처 정책에 제한 받지 않음)
  - 타 서버에 저장된 \*.js 파일 내에,
  - 자바스크립트 함수 내 인수로써 저장된 JSON 형식 데이터를 갖고 있으면서,
  - 클라이언트에 정의된 동일 함수를 실행하면,
  - 원격 타 서버에 있는 그 함수의 인수(파라미터)로써 데이터가 전달되는 구조
- 버튼 클릭 등 해당 이벤트가 발생하면,
  - 타 서버의 스크립트를 가져오기 위한 script 요소를 생성하고,
  - 원격 타 서버는 JSON 데이터를 포함한 함수 반환값을 클라이언트에 응답하면,
  - 클라이언트는 그 반환값을 콜백함수에서 처리하는 형태

파일 : ./data/data.json

```
{
  "item":[
    {"name":"홍길동","age":"SSG PAY 시스템 점검 안내","email":"moonseonjong@naver.com"},
    {"name":"김유신","age":"SSG PAY 시스템 점검 안내","email":"seonjong@naver.com"},
    {"name":"이순신","age":"SSG PAY 시스템 점검 안내","email":"moonjong@hanmail.net"}
  ]
}
```

```
<script>
  //JSON 테스트
  let obj =[
    {name:'홍길동', 'age':15,mail:'moonseonjong1@naver.com'},
    {name:'김유신', 'age':25,mail:'moonseonjong2@naver.com'},
    {name:'신윤복', 'age':33,mail:'moonseonjong3@naver.com'},
    {name:'강감찬', 'age':43,mail:'moonseonjong4@naver.com'},
    {name:'문선종', 'age':45,mail:'moonseonjong5@naver.com'}
  ];

  strJSON = JSON.stringify(obj);  //JSON 문자열로 변환
  parseJSON = JSON.parse(strJSON); //JSON 객체로 변환

  // AJAX 처리 비동기방식(Promise 사용)
  // fetch('https://jsonplaceholder.typicode.com/users')
  // fetch('https://jsonplaceholder.typicode.com/users.json')
  fetch('./data/data.json')
    .then( (resp) => {
      return resp.json()  //Promise 반환
    })
    .then( (data) => {      // 서버에서 주는 json데이터가 출력 됨

      //forEach() 반복문 사용하기
      data.item.forEach(function(el,index){
        console.log(data.item[index].name)
        console.log(data.item[index].age)
        console.log(data.item[index].email)
      })

    })

</script>
```

파일 : ./data/data.json

```
{
  "member":[
    {"userId":"moonjong1","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"김유신","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"이순신","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"류관순","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"박정희","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"}
  ]
}
```

```
<div id="app">
  <button v-on:click="fetchData">get data</button>
  <table>
    <tr v-for='item in arr'>
      <td>{{ item.userId }}</td>
      <td>{{ item.title }}</td>
      <td>{{ item.body }}</td>
    </tr>
  </table>
</div>

<script>
  new Vue({
    el: '#app',
    data:{
      arr:[]
    },
    methods: {
      fetchData: function() {
        axios.get('./data/mem.json')
          .then((obj) => { //data 가 붙는다
            //fetch 사용시는 item이 붙는다.
            console.log(obj.data.member[0].userId);
            console.log(obj.data.member[0].title);
            console.log(obj.data.member[0].body);
            console.log(obj.data.member[1].userId);
            console.log(obj.data.member[2].userId);
            this.arr = obj.data.member;

            this.arr.forEach(function(row,idx){
              console.log(idx,row.userId,row.title,row.body);
            });
          })
      }
    }
  })
}
```

```
        .catch(function(error) {  
            console.log(error);  
        });  
    }  
}  
})  
</script>
```

파일 : ./data/data.json

```
{
  "member":[
    {"userId":"moonjong1","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"김유신","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"이순신","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"류관순","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"박정희","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"}
  ]
}
```

```
<script>
  let $btn = document.querySelector('button');
  $btn.onclick = function(){
    fetch('./data.json')
    // fetch('https://jsonplaceholder.typicode.com/users')
    // .then(pResponse => pResponse.text())
    .then((data) => data.json())
    .then(function(arr){

      // console.log(x); item 꼭 사용할것
      arr.item.forEach(function(el,idx){
        console.log(idx,el);
      });
    })
  }

}

</script>
```

파일 : ./data/mem.json

```
{
  "member":[
    {"userId":"moonjong1","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"김유신","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"이순신","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"류관순","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"박정희","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"}
  ]
}
```

```
<div id="app">
  <button v-on:click="fetchData">get data</button>

  <div v-for='item in arr'>
    <h3>{{ item.userId }}</h3>
    <p>{{ item.title }}</p>
    <p>{{ item.body }}</p>
  </div>

</div>
<script src="./vue/lib/vue.js"></script>
<script src="./vue/lib/axios.min.js"></script>
<script>
  new Vue({
    el: '#app',
    data:{
      arr:[]
    },
    methods: {
      fetchData: function() {
        axios.get('./data/mem.json')
        // axios.post('./data/mem.json')
        .then((obj) => {
          console.log(obj.data.member[0].userId);
          console.log(obj.data.member[0].title);
          console.log(obj.data.member[0].body);
          console.log(obj.data.member[1].userId);
          console.log(obj.data.member[2].userId);
          this.arr = obj.data.member;
        })
      }
    }
  })
</script>
```



```
        this.arr.forEach(function(row,idx){
            console.log(idx,row.userId,row.title,row.body);
        });
    })
    .catch(function(error) {
        console.log(error);
    });
}

}

})
</script>
```

파일 : ./data/mem.json

```
{
  "member":[
    {"userId":"moonjong1","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"김유신","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"이순신","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"류관순","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"박정희","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"}
  ]
}
```

```
<div id="app">
  <button v-on:click="fetchData">get data</button>
  <table>
    <tr v-for='item in arr'>
      <td>{{ item.userId }}</td>
      <td>{{ item.title }}</td>
      <td>{{ item.body }}</td>
    </tr>
  </table>
</div>

<script>
  new Vue({
    el: '#app',
    data:{
      arr:[]
    },
    methods: {
      fetchData: function() {
        axios.get('./data/mem.json')
          .then((obj) => { //data 가 붙는다
            //fetch 사용시는 item이 붙는다.
            console.log(obj.data.member[0].userId);
            console.log(obj.data.member[0].title);
            console.log(obj.data.member[0].body);
            console.log(obj.data.member[1].userId);
            console.log(obj.data.member[2].userId);
            this.arr = obj.data.member;

            this.arr.forEach(function(row,idx){
              console.log(idx,row.userId,row.title,row.body);
            });
          });
      }
    }
  })
</script>
```

```
        });  
    })  
    .catch(function(error) {  
        console.log(error);  
    });  
}  
  
}  
  
})  
</script>
```

파일 : ./data/data.json

```
{
  "member":[
    {"userId":"moonjong1","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"김유신","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"이순신","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"류관순","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"},
    {"userId":"박정희","title":"SSG PAY 시스템 점검 안내","body":"2020-12-10"}
  ]
}
```

<button>JSON AJAX 로컬읽기</button>

<script>

```
let $btn = document.querySelector('button');
$btn.onclick = function(){
  fetch('./data/data.json')
    //.then( (res) => res.json())    // 리턴 생략 가능 한줄표현
    .then(function(res){
      console.log('res.type ', res.type );
      console.log('res.url ', res.url );
      console.log('res.redirected ', res.redirected );
      console.log('res.status ', res.status );
      console.log('res.statusText ', res.statusText );
      console.log('res.ok ', res.ok );
      console.log('responsive ', res );
      if(res.ok === true){
        console.log('responsive ', res.ok );
        return res.json();
      }
      else{
        console.log('responsive ', res.ok );
        console.log('프로미스(비동기 전송방식 AJAX) 실패');
      }
    })
    .then(function(data){
      data.item.forEach(function(el,idx){
        console.log(idx,el);
      });
    })
  }
}
```

</script>

--