

## 8. JSON 표기법

JSON(JavaScript Object Notation)은 경량의 데이터교환 형식으로 구문 형식은 Javascript를 따르지만 완전히 독립적이기 때문에 다른 프로그래밍 언어간에 데이터를 교환하기 위해서 널리 사용되는 표기법이다.

JSON은 첫 번째 Javascript의 문법으로 사용된다. 하나의 객체안에 함수와 변수를 함께 그룹화하여 포함시키기 위한 문법의 형태로 사용되며 jQuery에서 기본 문법 체계로 사용된다.

두 번째는 JSON 자체가 하나의 데이터로 사용되는 경우이다. Javascript 소스 코드 안에서 여러 개의 변수값들을 계층적으로 표현하기 위해서 사용된다. Ajax에서는 JSON을 통신 데이터 규격으로 사용하기도 한다.

### 1) JSON 표기법

```
var 객체명 = { 이름:값, 이름:값, ... , 이름:값};  
var 객체명 = { key:value, key:value, ... , key:value};  
  
var value = 객체명.이름;  
객체명.이름 = "새로운 값";
```

json1.html

```
<!DOCTYPE html>  
<html lang="ko">  
  <head>  
    .. 생략 ..  
    <script type="text/javascript">  
      var ju = {  
        "name" : "송준기",  
        "age" : 30,  
        "gender" : "남자",  
        "visual" : "잘생김"  
      };  
  
      document.write("<h1>" + ju.name + "</h1>");  
      document.write("<h1>" + ju.age + "</h1>");  
      document.write("<h1>" + ju.gender + "</h1>");  
      document.write("<h1>" + ju.visual + "</h1>");  
    </script>  
  </head>  
  <body> </body>  
</html>
```

## 2) 다양한 형태의 JSON 데이터 표현

### ① JSON 데이터에서 배열 사용

JSON에 배열 데이터 할당

```
var 객체명 = { 이름:배열, 이름:배열, ... , 이름:배열};

var json = { "nation": ["대한민국", "미국", "중국", "일본"] };
```

JSON에 포함된 배열 읽기

```
객체이름.이름[인덱스];

var json = { "nation": ["대한민국", "미국", "중국", "일본"] };

var nation_name = json.nation[0];
```

json2.html

```
<!DOCTYPE html>

<html lang="ko">

    <head>

        .. 생략 ..

        <script type="text/javascript">

            function initCompany() {

                // 회사 정보를 표현하는 데이터 구조를 JSON으로 정의

                // 하나의 JSON 데이터 안에는 일반 변수와 배열 데이터가 공존할 수 있다.

                var company = {

                    "name" : "(주)굿디자인",

                    "since" : 2015,

                    "department" : ["기획팀", "영업팀", "디자인팀", "개발팀"],
```

```

        "biz" : ["소프트웨어개발", "통신판매업", "위치정보서비스"]

    };

    // 회사이름 출력하기

    document.getElementById("name").innerHTML = company.name;

    // 설립년도 출력하기

    document.getElementById("since").innerHTML = company.since;


    // 주요 사업 부서를 출력하기 위하여 배열을 반복문으로 읽는다.
    var dept_li = "";

    for (var i = 0; i < company.department.length; i++) {

        dept_li += "<li>" + company.department[i] + "</li>";

    }

    document.getElementById("department").innerHTML = dept_li;


    // 주요 사업 분야를 출력하기 위하여 배열을 반복문으로 읽는다.
    var biz_li = "";

    for (var i = 0; i < company.biz.length; i++) {

        biz_li += "<li>" + company.biz[i] + "</li>";

    }

    document.getElementById("biz").innerHTML = biz_li;

    }

</script>

</head>

<body onload="initCompany()">

    <h1 id="name"> </h1>

    <p>

```

```

        <strong>설립년도: </strong> <span id="since"> </span>

    </p>

    <h2>주요 사업 부서</h2>

    <ul id="department"> </ul>

    <h2>주요 사업 분야</h2>

    <ul id="biz"> </ul>

</body>

</html>

```

## ② 계층형 JSON 데이터

JSON을 포함한 JSON

```

var 객체명 = {

    하위객체명 : { 이름:값, 이름:값, ... , 이름:값 },

    하위객체명 : { 이름:값, 이름:값, ... , 이름:값 }

};

```

```

var korea = {

    "seoul" : { "people":54321, "desc":"대한민국 수도" },

    "jaeju" : { "people":12345, "desc":"아름다운 섬" }

};

```

하위 JSON 데이터에 접근

```
객체명.하위객체명.이름;
```

```
var description = korea.seoul.desc;
```

```

<!DOCTYPE html>

<html lang="ko">

    <head>

        .. 생략 ..

        <script type="text/javascript">

            // 하위 데이터 요소로서 JSON을 포함하고 있는 JSON 데이터

            var keyword_rank = {

                "r1" : { "k" : "HTML5", "s" : "+", "v" : 178 },

                "r2" : { "k" : "CSS3", "s" : "new", "v" : 150 },

                "r3" : { "k" : "jQuery", "s" : "+", "v" : 135 }

            };

            // JSON 데이터의 내용을 목록정의 태그로 구성

            var tag = "<h1>인기 검색어 순위</h1><ul>";

            tag += "<li>" + keyword_rank.r1.k + "(" + keyword_rank.r1.s +
keyword_rank.r1.v + ")";

            tag += "<li>" + keyword_rank.r2.k + "(" + keyword_rank.r2.s +
keyword_rank.r2.v + ")";

            tag += "<li>" + keyword_rank.r3.k + "(" + keyword_rank.r3.s +
keyword_rank.r3.v + ")";

            tag += "</ul>";

            // 구성한 내용을 화면에 출력

            document.write(tag);

        </script>

    </head>

```

```
<body></body>
</html>
```

### ③ 복잡한 데이터 구조의 표현

#### ▶ JSON에 포함된 배열 구조의 장단점

JSON 안에서 배열의 값으로 사용할 경우에는 하나의 이름안에 배열로 구성된 여러 개의 데이터를 포함시킬 수 있지만, 하나의 데이터는 오직 하나의 값으로만 구성되기 때문에 결국 단순한 키워드의 나열과 같은 경우 밖에 사용할 수 없다는 단점이 있다.

```
var rank = {
    "keyword": ["seoul", "jaeju"]
};
```

#### ▶ JSON에 포함된 JSON 구조의 장단점

JSON 데이터안에 새로운 JSON 데이터가 포함되어 있는 구조는 데이터가 계층적으로 구성될 수 있기 때문에 데이터 간의 포함관계를 표현하기 쉽다는 장점이 있다. 하지만 하위 객체로 포함된 JSON 이름들이 독립적이기 때문에 하나의 객체 안에 존재하는 하위 객체들간의 연관성이 적다는 단점을 가지고 있다.

```
var korea = {
    "seoul" : { "people":54321, "desc":"대한민국 수도" },
    "jaeju" : { "people":12345, "desc":"아름다운 섬" }
};
```

#### ▶ 두 방식의 장점만을 조합한 상태

JSON안에 새로운 JSON이 포함되는 계층형 구조에서 각각의 데이터가 독립적이라는 점을 배열로 보완하여 하나의 키 이름 안에서 여러 개의 데이터를 JSON으로 표현할 수 있다. 이렇게 구성된 데이터는 하나의 키에 대응되는 값들이 배열로 구성되어, 각 배열의 요소들이 동일한 구조의 JSON으로 구성된다.

```
var rank = {
    "keyword" : [
```

```

        { "people":54321, "desc": "대한민국 수도" },

        { "people":12345, "desc": "아름다운 섬" }

    ]

};

```

위와 같은 배열안에 새로운 JSON을 포함하는 복합 구조는 게시판과 같은 동일한 구조의 복합 데이터가 반복적으로 나타나는 형태에 적용할 수 있다.

게시판의 글 목록은 번호, 제목, 조회수, 최근 수정일이라는 하위데이터를 갖는다. 이러한 구조를 가지는 데이터를 JSON으로 표현한다.

json4.html

```

<!DOCTYPE html>

<html lang="ko">

    <head>

        .. 생략 ..

        <script type="text/javascript">

            // 게시판의 데이터 표현

            var bbs = {

                "title": "Javascript",

                "description": "교육원에서 진행중인 Javascript 강의 자료 입니다. 각
게시판의 다운로드 권한은 매 강의 개강시마다 부여됩니다.",

                "total": 4,

                "item": [

                    {"no": "공지", "subject": "'Javascript' 강의 자료실 입니다.",
"hit": 3, "date": "2015.10.29"},

                    {"no": 3, "subject": "Javascript 3일차 강의자료 입니다.", "hit": 6,
"date": "2015.11.08"},

                    {"no": 2, "subject": "Javascript 2일차 강의자료 입니다.", "hit":

```

```

15, "date": "2015.11.07"},

                                {"no": 1, "subject": "Javascript 1일차 강의자료 입니다.", "hit":
19, "date": "2015.11.05"}

                                ]

                                };

                                // bbs안의 각 독립된 값들에 대한 출력 HTML 생성

                                var html = "<h1>" + bbs.title + "</h1>";

                                html += "<p>" + bbs.description + "</p>";

                                html += "<p>총 게시물 수: " + bbs.total + "개</p>";

                                html += "<ul>";

                                // 배열로 표현되는 게시물 정보에 대한 출력 HTML 생성

                                for (var i = 0; i < bbs.item.length; i++) {

                                    var item = bbs.item[i];

                                    html += "<li>(" + item.no + ") " + item.subject + "(조회수: " + item.hit
+ " / 날짜: " + item.date + ")</li>";

                                }

                                html += "</ul>";

                                // 화면 출력

                                document.write(html);

                                </script>

                                </head>

                                <body></body>

                                </html>

```



### 3) JSON 표기법을 적용한 객체지향 구현

JSON 표기법은 Javascript 안에서 데이터 표현을 위한 표현법으로 사용되기도 하지만, 함수와 변수를 하나의 그룹으로 묶기 위한 문법으로 사용되기도 한다.

#### ① 함수를 포함하는 JSON

##### ▶ 변수(값) = 함수

JSON내의 하나의 값에 함수가 대입될 수 있다.

“이름:값”의 형식으로 구성되는 JSON 데이터는 “이름:함수”의 형식으로 구성이 가능하다.

```
var hello = {  
    "sayKor" : function() { ... },  
    "sayEng" : function() { ... }  
};
```

JSON 내부의 함수를 호출

```
hello.sayKor();
```

#### ② 값과 함수가 공존하는 JSON

##### ▶ Javascript의 문법적 특성 활용

```
var hello = {  
    "msg" : "안녕하세요",  
    "sayHello" : function() {  
        alert(this.msg);  
    }  
};
```

```
<!DOCTYPE html>

<html lang="ko">

  <head>

    .. 생략 ..

    <script type="text/javascript">

      var sayHello = {

        "name": "Javascript",

        "eng" : function() {

          var console = document.getElementById("console");

          console.innerHTML = "Hello " + this.name;

          console.style.color = "#ff0000";

        },

        "kor" : function() {

          var console = document.getElementById("console");

          console.innerHTML = "안녕하세요. " + this.name;

          console.style.color = "#0000ff";

        }

      };

    </script>

  </head>

  <body>

    <h1>JSON으로 그룹화된 함수 호출</h1>

    <input type="button" value="sayEng" onclick="sayHello.eng()" />

    <input type="button" value="sayKor" onclick="sayHello.kor()" />

    <h1 id="console"> </h1>
```

```
</body>
```

```
</html>
```