JSON

1. Lab

[js/member.js]

```
var Member = function(name, id, securityNo) {
    this.name = name;
    this.id = id;
    this.securityNo = securityNo;
}
Member.prototype.setValue = function(newName, newId, newSecurityNo) {
    this.name = newName;
    this.id = newId;
    this.securityNo = newSecurityNo;
}
Member.prototype.getAge = function() {
    var birthYear = parseInt(this.securityNo.substring(0, 2));
    var code = this.securityNo.substring(6,7);
    if (code == '1' || code == '2') {
        birthYear += 1900;
    } else if (code == '3' || code == '4') {
        birthYear += 2000;
    }
    var today = new Date();
    return today.getFullYear() - birthYear;
}
Member.prototype.toString = function() {
    return this.name + "[" + this.id + "]";
}
```

[useMember.html]

```
<!doctype html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Member 클래스 사용</title>
    <script src="js/member.js"></script>
    <script>
        function log(msg) {
            var console = document.getElementById("debugConsole");
            if (console != null) {
                console.innerHTML += msg + "<br/>";
            }
        }
        window.onload = function() {
            var mem = new Member("ajax", "복종순", "7700001000000");
            log('변경전');
            log('회원 : ' + mem.toString());
            log('나이 : ' + mem.getAge());
            mem.setValue("ajax1", "조성모", "8000001000000");
            log('변경후');
            log('회원 : ' + mem.toString());
            log('나이 : ' + mem.getAge());
        }
    </script>
</head>
<body>
<div id="debugConsole"></div>
</body>
</html>
```

2. What is JSON?

1)Stands for JavaScript Object Notation
2)Is lightweight data interchange format.
3)Is language independent.
4)Is "self-describing"(human readable) and easy to understand.
5)Is an easier to use alternative to XML.

6)It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999.
-http://www.ecma-international.org/publications/standards/Ecma-262.htm
7)http://www.json.org
8)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.
9) https://github.com/douglascrockford/JSON-js
-Downloads json2.js

3. JSON vs XML
1)XML Example

```
<employees>
   <employee>
      <firstName>John</firstName> <lastName>Doe</lastName>
   </employee>
   <employee>
      <firstName>Anna</firstName> <lastName>Smith</lastName>
   </employee>
   <employee>
      <firstName>Peter</firstName> <lastName>Jones</lastName>
   </employee>
</employees>
```

2)JSON Example

```
{"employees":[
            {"firstName":"John", "lastName":"Doe"},
            {"firstName":"Anna", "lastName":"Smith"},
            {"firstName":"Peter", "lastName":"Jones"}
   ]
}
```

3)Much Like XML
-Both JSON and XML is plain text.
-Both JSON and XML is "self-describing" (human readable).
-Both JSON and XML is hierarchical (values within values).
-Both JSON and XML can be fetched with an HttpRequest.

4) Much Unlike XML
-JSON doesn't use end tag.
-JSON is shorter.
-JSON is quicker to read and write.
-JSON can use arrays.
-The biggest difference is:
--XML has to be parsed with an XML parser, JSON can be parsed by a standard JavaScript function.

[jsondemo.html]

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<script>
   var member = {"name" : "한지민",
                 "favoriteColor" : ["파랑", "노랑", "빨강"]
                  };
   var message = member.name + "님이 좋아하시는 색깔은 " +
                               member.favoriteColor.length + "개 이고, \n" +
                               " 그 중에서 가장 좋아하는 색깔은 " +
                                member.favoriteColor[1] + "  입니다.";
      alert(message);
</script>
```

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

4. JSON Syntax Rules
   1)Data is in name/value pairs.
   2)Data is separated by commas.
   3)Curly braces hold objects.
   4)Square brackets hold arrays.

5. JSON Data - A Name and a Value
   1)JSON data is written as name/value pairs.
   2)A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value:

```
"firstName":"John"
```

6. JSON Values
   1)A number (integer or floating point)
   2)A string (in double quotes)
   3)A Boolean (true or false)
   4)An array (in square brackets)
   5)An object (in curly braces)
   6)null

7. JSON Objects
   1)JSON objects are written inside curly braces.
   2)Just like in JavaScript, objects can contain multiple name/values pairs:

```
{"firstName":"John", "lastName":"Doe"}
```

8. JSON Arrays
   1)JSON arrays are written inside square brackets.
   2)Just like in JavaScript, an array can contain multiple objects:

```
"employees":[
   {"firstName":"John", "lastName":"Doe"},
   {"firstName":"Anna", "lastName":"Smith"},
   {"firstName":"Peter", "lastName":"Jones"}
]
```

9. JSON Uses JavaScript Syntax
   1)Because JSON uses JavaScript syntax, no extra software is needed to work with JSON within JavaScript.
   2)With JavaScript you can create an array of objects and assign data to it like this:
   -Example

```
var employees = [
   {"firstName":"John", "lastName":"Doe"},
   {"firstName":"Anna", "lastName":"Smith"},
   {"firstName":"Peter", "lastName": "Jones"}
];
```

   -The first entry in the JavaScript object array can be accessed like this:

```
employees[0].firstName + " " + employees[0].lastName;
```

   -The returned content will be:

```
John Doe
```

-The data can be modified like this:

```
employees[0].firstName = "Gilbert";
```

10. JSON Files
1)The file type for JSON files is ".json"
2)The MIME type for JSON text is "application/json"

11. JSON Example - Object From String
1)Create a JavaScript string containing JSON syntax:

```
var text = '{ "employees" : [' +
        '{ "firstName":"John" , "lastName":"Doe" },' +
        '{ "firstName":"Anna" , "lastName":"Smith" },' +
        '{ "firstName":"Peter" , "lastName":"Jones" } ]}';
```

2)The JavaScript function JSON.parse(text) can be used to convert a JSON text into a JavaScript object:

```
var obj = JSON.parse(text);
```

[jsondemo1.html]

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JSON Demo</title>
<script>
   window.addEventListener("load", setup, false);
   function setup(){
      document.getElementById("mybtn").addEventListener("click", Click, false);
   }
   function Click(){
      var text = '{"irum" : "한지민", "gender" : "여성", "직업" : "탤런트 겸 영화배우", "phone" :
      "02-555-1234"}';
      var obj = JSON.parse(text);
      var str = obj.irum + "<br>" + obj.gender + "<br>" + obj.직업 + "<br>" + obj.phone;
      document.getElementById("demo").innerHTML = str;
   }
</script>
</head>
<body>
<h2>JSON Object Creation in JavaScript</h2>
<button id="mybtn">Show Data</button>
<p id="demo"></p>
</body>
</html>
```

12. Using eval()
1)Older browsers without the support for the JavaScript function JSON.parse() can use the eval() function to convert a JSON text into a JavaScript object:

```
var obj = JSON.parse(text);
```

2)The eval() function can compile and execute any JavaScript. This represents a potential security problem. Try to avoid it.
3)It is safer to use a JSON parser to convert a JSON text to a JavaScript object.
4)A JSON parser will recognize only JSON text and will not compile scripts.
5)In browsers that provide native JSON support, JSON parsers are also faster.
6)Native JSON support is included in all major browsers and in the latest ECMAScript (JavaScript) standard:

7)Web Browser Support
-Firefox 3.5
-Internet Explorer 8
-Chrome
-Opera 10
-Safari 4

8)For older browsers, a JavaScript library is available at
-https://github.com/douglascrockford/JSON-js
-json.js, json2.js
-The JSON format was originally specified by Douglas Crockford(http://www.crockford.com/)

12. Ajax and JSON

[member_json.jsp]

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
{
   "code" : "success",
   "data" : {
                  "member" : {
                                 "irum" : "이미자",
                                 "id" : "javaoracle",
                                 "juminnum" : 12345678
                              }
                  }

}
```

[ajaxdemo.html]

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="UTF-8">
      <title>Insert title here</title>
      <script>
         var xhrObj;
         window.addEventListener("load", setup, false);
         function setup(){
            xhrObj = new XMLHttpRequest();
            xhrObj.open("get", "member_json.jsp", true);
            xhrObj.onreadystatechange = callback;
            xhrObj.send(null);
         }
         function callback(){
            if(xhrObj.readyState == 4 && xhrObj.status == 200){
               //var jsonobj = eval("(" + xhrObj.responseText + ")");
               var jsonobj = JSON.parse(xhrObj.responseText);
               if(jsonobj.code == "success"){
                  var m = jsonobj.data.member;
                  alert(m.irum + " [ " + m.id + " ] " + m.juminnum);
               }
            }
         }
      </script>
   </head>
   <body>
   </body>
</html>
```

[sungjuk_utf8.json]

```
{
   "students" : [
      {"hakbun" : "1101", "irum" : "한송이", "kor" : 78, "eng" : 87, "mat" : 83, "edp" : 78},
      {"hakbun" : "1102", "irum" : "정다워", "kor" : 88, "eng" : 83, "mat" : 57, "edp" : 98},
      {"hakbun" : "1103", "irum" : "그리운", "kor" : 76, "eng" : 56, "mat" : 87, "edp" : 78},
      {"hakbun" : "1104", "irum" : "고아라", "kor" : 83, "eng" : 57, "mat" : 88, "edp" : 73},
      {"hakbun" : "1105", "irum" : "사랑해", "kor" : 87, "eng" : 87, "mat" : 53, "edp" : 55},
```

```
            {"hakbun" : "1106", "irum" : "튼튼이", "kor" : 98, "eng" : 97, "mat" : 93, "edp" : 88},
            {"hakbun" : "1107", "irum" : "한아름", "kor" : 68, "eng" : 67, "mat" : 83, "edp" : 89},
            {"hakbun" : "1108", "irum" : "더크게", "kor" : 98, "eng" : 67, "mat" : 93, "edp" : 78},
            {"hakbun" : "1109", "irum" : "더높이", "kor" : 88, "eng" : 99, "mat" : 53, "edp" : 88},
            {"hakbun" : "1110", "irum" : "아리랑", "kor" : 68, "eng" : 79, "mat" : 63, "edp" : 66},
            {"hakbun" : "1111", "irum" : "한산섬", "kor" : 98, "eng" : 89, "mat" : 73, "edp" : 78},
            {"hakbun" : "1112", "irum" : "하나로", "kor" : 89, "eng" : 97, "mat" : 78, "edp" : 88}
        ]
    }

[ajaxdemo1.html]
    <!DOCTYPE html>
    <html>
        <head>
            <meta charset="UTF-8">
            <title>Insert title here</title>
            <script src="js/jquery-3.2.1.min.js"></script>
            <script>
                var xmlObj;
                $(document).ready(function(){
                    $('#btn').click(function(){
                        xmlObj = new XMLHttpRequest();
                        xmlObj.open("GET", "sungjuk_utf8.json");
                        xmlObj.onreadystatechange = callback;
                        xmlObj.send(null);
                    });
                });
                function displayHeader(){
                    var str = "<h2>성적 데이터(JSON)</h2>";
                    str += "<table><thead><tr>";
                    str +=
                    "<th>학번</th><th>이름</th><th>국어</th><th>영어</th><th>수학</th><th>전산</th>";
                    str += "<th>총점</th><th>평균</th><th>학점</th></tr></thead>";
                    str += "<tbody></tbody>";
                    str += "</table>";
                    return str;
                }
                function callback(){
                    if(xmlObj.readyState ==4 && xmlObj.status == 200){
                        $('#rcvMsg').html(displayHeader());
                        myparsor(xmlObj.responseText);   //매우 주의할 것(xml형식이 아닌 모든 텍스트형식 읽어올 때)
                    }
                }
                function myparsor(data){
                    var str = "";
                    var jsondata = JSON.parse(data);   //string --> JSON 객체로
                    var array = jsondata.students;
                    for(var i = 0 ; i < array.length ; i++){
                        str += "<tr>";
                        str += "<td>" + array[i].hakbun + "</td>";
                        str += "<td>" + array[i].irum + "</td>";
                        str += "<td>" + array[i].kor + "</td>";
                        str += "<td>" + array[i].eng + "</td>";
                        str += "<td>" + array[i].mat + "</td>";
                        str += "<td>" + array[i].edp + "</td>";
                        var sum = calcSum(array[i].kor, array[i].eng, array[i].mat, array[i].edp);
                        var avg = calcAvg(sum);
                        var grade = calcGrade(avg);
                        str += "<td>" + sum + "</td>";
                        str += "<td>" + avg + "</td>";
                        str += "<td>" + grade + "</td>";
                        str += "</tr>";
                    }
                    $('tbody').html(str);
                }
                function calcSum(kor, eng, mat, edp){
```

```
                var sum = parseInt(kor) + parseInt(eng) + parseInt(mat) + parseInt(edp);
                return sum;
            }
            function calcAvg(sum){
                var avg = sum / 4.;
                return avg;
            }
            function calcGrade(avg){
                avg = avg.toFixed(2);
                var grade = (avg <= 100 && avg >= 90) ? 'A' :
                                        (avg < 90 && avg >= 80) ? 'B' :
                                            (avg < 80 && avg >= 70) ? 'C' :
                                                (avg < 70 && avg >= 60) ? 'D' : 'F';
                return grade;
            }
        </script>
        <link rel="stylesheet" type="text/css" href="css/style.css" />
    </head>
    <body>
        <input type="button" id="btn" value="성적데이터수신(JSON)" /><br />
        <div id="rcvMsg"></div>
    </body>
</html>

[css/style.css]
    @CHARSET "UTF-8";
    h2{
        text-align:center
    }
    table{
        margin:auto;
        width:600px;
        border-collapse : collapse;
    }
    th, td{
        border : 1px solid black;
    }
    th{
        background-color:yellow;color:darkblue;
    }
    td{
        text-align:center;
    }
```