


Mobile Simulation

2017-2



HTML5+ CSS3+ Javascript


웹 프로그래밍

황기태 지음

웹 프로그래밍의 완성에는 세 가지 기술이 필요합니다.
HTML5+CSS3+Javascript가 그것입니다.
이 책은 HTML5 웹 프로그래밍 작성에 필요한 3가지 기술을 균형 있게 구성하고 있습니다.

- 이 책은 내용을 그림과 삽화로 설명해 깊이 있는 이해를 도울 것입니다.
- 이론에서 실용까지 연결되는 단계로 손에 익히는 학습을 유도합니다.
- Open Challenge와 풍부한 연습 문제를 풀어 배운 내용을 정리합니다.
- 저자가 운영하는 영웅 웹 프로그래밍 사이트를 통해 언제 어디서든 실용하고
정확할 수 있습니다.

영웅 웹사이트에서 예제를 실행하고 직접 수정해 볼 수 있습니다
www.webprogramming.co.kr



Search w3schools.com:

Google™ Custom Search

 JavaScript


JavaScript Tutorial

JavaScript Reference

 JQuery

JQuery Tutorial

JQuery Reference



Weekly plan (HTML5, 1st semester 2017)

- **wk01 : Introduction to curriculum & current state of HTML5**
- **wk02 : Making HTML5 documents**
- **wk03 : Table, iframe and media**
- **wk04 : Semantic tag and Form**
- **wk05 : CSS3 I. Basic**
- **wk06 : CSS3 II. Advanced**
- **wk07 : CSS3 III. Animation**
- **wk08 : Mid-term Exam.**
- **wk09 : Javascript : Data types & operators**
- **wk10 : Javascript : Loop & functions**
- **wk11 : Javascript : Core objects**
- **wk12 : Javascript : DOM**
- **wk13 : Javascript : Event handling I**
- **wk14 : Javascript : Event handling II**
- **wk15 : Final exam.**

Weekly plan (Mobile Simulation, 2nd semester 2017)

- **wk01 : Introduction to curriculum & current state**
- **wk02 : Browser Object Model (BOM), installing Brackets editor**
- **wk03 : Canvas graphics I. Basic**
- **wk04 : Canvas graphics II. Image & Transformation**
- **wk05 : Canvas graphics III. Animation**
- **wk06 : Canvas graphics IV. Game**
- **wk07 : 보강 기간에 보강 실시**
- **wk08 : Mid-term Exam.**
- **wk09 : jQuery I. Basic**
- **wk10 : jQuery II. Application**
- **wk11 :**
- **wk12 :**
- **wk13 :**
- **wk14 :**
- **wk15 : Final exam.**

```
{  
  "name": "HongGilDong",  
  "age": 23,  
  "address": {  
    "nation": "Korea",  
    "city": "GimHae",  
    "postalCode": "12345"  
  },  
  "특기": ["jQuery", "Node.js",  
    "plotly.js", "Arduino", "R"],  
  "phone": "010-1234-5678"  
}
```

과제06. msnn_rpt06.zip

4

[실습과제06] jQuery Basic

[1] 실습 파일 정리.

[2] jQuery 실습 폴더 압축하여 제출.

**** 실습 파일 및 관련 파일(image 등..)을
MSnn_Rpt06.zip 으로 압축해서 제출하시오

[제출파일] [msnn_rpt06.zip](#) (10월24일 오후 6시 마감)

[html 파일과 사용된 그림을](#) 압축하여 이메일로 “msnn_rpt06” 제목으로 제출

Email : chaos21c@gmail.com

jQuery Quiz

W3Schools jQuery Quiz

jQuery QUIZ

1. Which of the following is correct?

- ☐ jQuery is a JavaScript Library
- ☐ jQuery is a JSON Library

Next

Total 20 questions

<http://www.w3schools.com/quiztest/quiztest.asp?qtest=jQuery>

CHAPTER 11-A2

jQuery II. application



Reference



jQuery Tutorial

<http://www.w3schools.com/jquery/default.asp>

jQuery References

http://www.w3schools.com/jquery/jquery_ref_selectors.asp

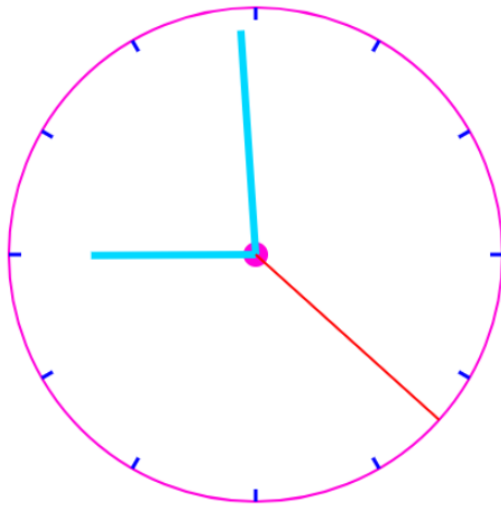
jQuery Quiz

http://www.w3schools.com/jquery/jquery_quiz.asp

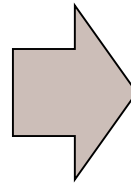
Target 1: Clock

Converting JS to jQuery

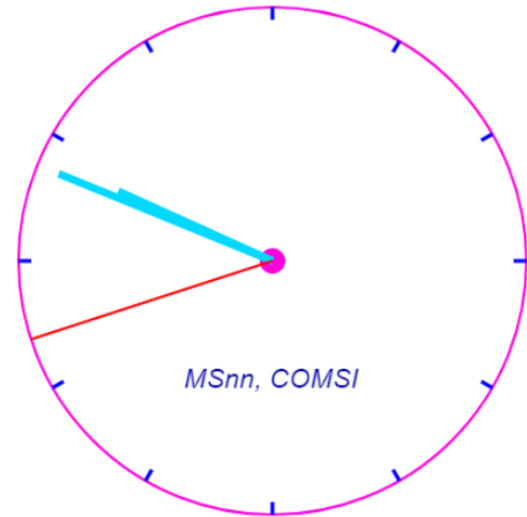
JavaScript simulation : Analog clock



MSnn Comsi



jQuery simulation : Analog clock



MSnn Comsi

Javascript Clock (script 1)



```
<html>
<head>
  <meta charset="utf-8" />
  <title> Mobile Simulation: Clock on Canvas </title>
  <style type="text/css">
    h1{
      text-align:center;
    }
    #container {
      margin: 0 auto;
      width : 800px;
      text-align: center;
    }
    .name{
      text-align:center;
    }
  </style>
</head>
<body>

<h1 style="color:#353535">JavaScript simulation : Analog clock</h1>

<div id="container">
<canvas id="analogClock" width="500" height="450">Clock</canvas>
</div>

<script>

</script>

<h2 style="color:#353535" class="name">MSnn Comsi</h2>
```

```
// run clock at each second
function tick() {
  var date = new Date();
  ctx.clearRect(0, 0, canvas.width, canvas.height);
  drawStatic();

  // Get current hour, minutes, seconds
  var hours = date.getHours();
  var minutes = date.getMinutes();
  var seconds = date.getSeconds();

  // Draw hour
  ctx.strokeStyle = "#00D8FF";
  ctx.lineWidth = 6;
  drawHand(clockWidth / 3, hours * 30+ minutes / 2 + seconds/120);

  // Draw minutes
  ctx.strokeStyle = "#00D8FF";
  ctx.lineWidth = 6;
  drawHand(5*clockWidth / 11, minutes * 6 + seconds/10);

  // Draw seconds
  ctx.strokeStyle = "red";
  ctx.lineWidth = 2;
  drawHand(clockWidth / 2, seconds * 6);
}
```

Javascript Clock (script 2)



```
// run clock at each second
function tick() {
    var date = new Date();
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    drawStatic();

    // Get current hour, minutes, seconds
    var hours = date.getHours();
    var minutes = date.getMinutes();
    var seconds = date.getSeconds();

    // Draw hour
    ctx.strokeStyle = "#00D8FF";
    ctx.lineWidth = 6;
    drawHand(clockWidth / 3, hours * 30 + minutes / 2 + seconds / 120);

    // Draw minutes
    ctx.strokeStyle = "#00D8FF";
    ctx.lineWidth = 6;
    drawHand(5 * clockWidth / 11, minutes * 6 + seconds / 10);

    // Draw seconds
    ctx.strokeStyle = "red";
    ctx.lineWidth = 2;
    drawHand(clockWidth / 2, seconds * 6);
}
```

```
// draw outline of clock.
function drawStatic() {
    ctx.beginPath();
    ctx.arc(centerX, centerY, clockWidth / 2, 0, 2 * Math.PI, false);
    ctx.strokeStyle = "#FF00DD";
    ctx.lineWidth = 2;
    ctx.stroke();
    ctx.closePath();

    ctx.beginPath();
    ctx.arc(centerX, centerY, 10, 0, 2 * Math.PI, false);
    ctx.fillStyle = "#FF00DD";
    ctx.fill();
    ctx.closePath();

    drawNumbers(); // Draw hour ticks
    // draw my id
    ctx.font = 'italic 16pt Arial';
    ctx.fillStyle = "#0000AA";
    ctx.fillText(myid, centerX - 70, centerY + 100);
}
```

Javascript Clock (script 3)



```
// draw ticks of time
function drawNumbers() {
    var i = 12;
    ctx.strokeStyle = "#0100FF";
    ctx.lineWidth = 3;
    while (i > 0) {
        ctx.save();
        ctx.beginPath();
        ctx.translate(centerX, centerY);
        var angle = (i * 30) * Math.PI / 180;
        ctx.rotate(angle);
        ctx.translate(0, -clockWidth / 2);

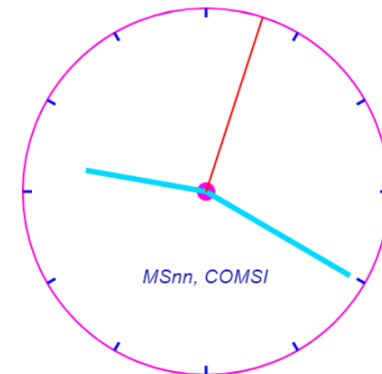
        //Drawing numbers doesn't look so good
        // because of the origin of the text
        // ctx.save();
        // ctx.translate(0, -10);
        // ctx.rotate(-angle);
        // ctx.fillText(i, -3, 0);
        // ctx.restore();

        ctx.moveTo(0, 0);
        ctx.lineTo(0, 10);
        ctx.stroke();
        ctx.closePath();
        ctx.restore();
        i--;
    }
}
```

```
// draw handle
function drawHand(length, angle) {
    ctx.save();
    ctx.beginPath();
    ctx.translate(centerX, centerY);
    // Correct for top left origin
    ctx.rotate(-180 * Math.PI / 180);
    ctx.rotate(angle * Math.PI / 180);
    ctx.moveTo(0, 0);
    ctx.lineTo(0, length);
    ctx.stroke();
    ctx.closePath();
    ctx.restore();
}

init();

</script>
```



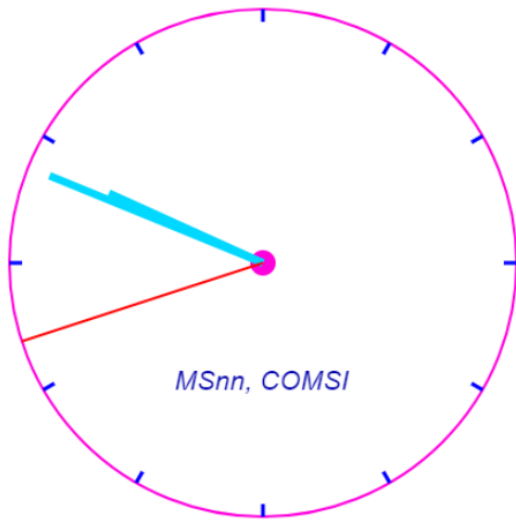
jQuery Clock (script 1)



```
<head>
<meta charset="utf-8" />
<title> Mobile Simulation: Clock on Canvas </title>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"> </script>
```

jQuery simulation : Analog clock



MSnn Comsi

```
<script>

// var canvas;
var ctx;

var centerX;
var centerY;

var cv_width;
var cv_height;

var clockWidth = 400;
var myid = "MSnn, COMSI";

// init clock.
function init().{
    //canvas = document.getElementById("analogClock");
    ctx = $('#analogClock')[0].getContext("2d");

    cv_width = $('#analogClock')[0].width;
    cv_height = $('#analogClock')[0].height;

    centerX = cv_width / 2;
    centerY = cv_height / 2;

    tick();
    window.setInterval(tick, 1000);
}
```

jQuery Clock (script 2)



```
// run clock at each second
function tick() {
    var date = new Date();
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    drawStatic();

    // Get current hour, minutes, seconds
    var hours = date.getHours();
    var minutes = date.getMinutes();
    var seconds = date.getSeconds();

    // Draw hour
    ctx.strokeStyle = "#00D8FF";
    ctx.lineWidth = 6;
    drawHand(clockWidth / 3, hours * 30 + minutes / 2 + seconds / 120);

    // Draw minutes
    ctx.strokeStyle = "#00D8FF";
    ctx.lineWidth = 6;
    drawHand(5 * clockWidth / 11, minutes * 6 + seconds / 10);

    // Draw seconds
    ctx.strokeStyle = "red";
    ctx.lineWidth = 2;
    drawHand(clockWidth / 2, seconds * 6);
}
```

```
// draw outline of clock.
function drawStatic() {
    ctx.beginPath();
    ctx.arc(centerX, centerY, clockWidth / 2, 0, 2 * Math.PI, false);
    ctx.strokeStyle = "#FF00DD";
    ctx.lineWidth = 2;
    ctx.stroke();
    ctx.closePath();

    ctx.beginPath();
    ctx.arc(centerX, centerY, 10, 0, 2 * Math.PI, false);
    ctx.fillStyle = "#FF00DD";
    ctx.fill();
    ctx.closePath();

    drawNumbers(); // Draw hour ticks
    // draw my id
    ctx.font = 'italic 16pt Arial';
    ctx.fillStyle = "#0000AA";
    ctx.fillText(myid, centerX - 70, centerY + 100);
}
```

jQuery Clock (script 3)



```
// draw ticks of time
function drawNumbers() {
    var i = 12;
    ctx.strokeStyle = "#0100FF";
    ctx.lineWidth = 3;
    while (i > 0) {
        ctx.save();
        ctx.beginPath();
        ctx.translate(centerX, centerY);
        var angle = (i * 30) * Math.PI / 180;
        ctx.rotate(angle);
        ctx.translate(0, -clockWidth / 2);

        //Drawing numbers doesn't look so good
        // because of the origin of the text
        // ctx.save();
        // ctx.translate(0, -10);
        // ctx.rotate(-angle);
        // ctx.fillText(i, -3, 0);
        // ctx.restore();

        ctx.moveTo(0, 0);
        ctx.lineTo(0, 10);
        ctx.stroke();
        ctx.closePath();
        ctx.restore();
        i--;
    }
}
```

```
// draw handle
function drawHand(length, angle) {
    ctx.save();
    ctx.beginPath();
    ctx.translate(centerX, centerY);
    // Correct for top left origin
    ctx.rotate(-180 * Math.PI / 180);
    ctx.rotate(angle * Math.PI / 180);
    ctx.moveTo(0, 0);
    ctx.lineTo(0, length);
    ctx.stroke();
    ctx.closePath();
    ctx.restore();
}
```

```
//init();
$(document).ready(function () {
    init();
});
```

jQuery로 변환된 파일을
[MSnn_jquery_clock.html](#) 로 저장하시오.

Convert Javascript to jQuery



Get Content - text(), html(), and val()

Three simple, but useful, jQuery methods for DOM manipulation are:

- `text()` - Sets or returns the text content of selected elements
- `html()` - Sets or returns the content of selected elements (including HTML markup)
- `val()` - Sets or returns the value of form fields

Definition and Usage

The `val()` method returns or sets the value attribute of the selected elements.

When used to return value:

This method returns the value of the value attribute of the FIRST matched element.

When used to set value:

This method sets the value of the value attribute for ALL matched elements.

Note: The `val()` method is mostly used with HTML form elements.

Syntax

Return the value attribute:

```
$(selector).val()
```

Set the value attribute:

```
$(selector).val(value)
```

Definition and Usage

The `html()` method sets or returns the content (innerHTML) of the selected elements.

When this method is used to **return** content, it returns the content of the FIRST matched element.

When this method is used to **set** content, it overwrites the content of ALL matched elements.

Tip: To set or return only the text content of the selected elements, use the `text()` method.

Syntax

Return content:

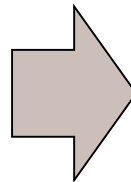
```
$(selector).html()
```

Set content:

```
$(selector).html(content)
```

Target 2: Game

Converting JS to jQuery



jQuery Game (script 0)



```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title> Mobile Simulation: Game </title>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"> </script>
  <style type="text/css">
    canvas {
      border: 1px dotted red; /* 캔버스에 경계선을 그려준다. */
      background-color: #fcff00; /* 캔버스의 배경색을 지정한다. */
    }
  </style>
</head>
<body onload="init();">
  <h1>Canvas simulation : jQuery CannonBall</h1>
  <canvas id="myCanvas" width="500" height="300"
    style="border: 1px dotted red"> </canvas>

  <div id="control">
    속도<input id="velocity" value="30" type="number" min="0" max="100" step="1" />
    각도<input id="angle" value="45" type="number" min="0" max="90" step="1" />
    넷높이<input id="netHeight" value="80" type="number" min="30" max="150" step="1" />
    <div id="score">점수 = 0</div>
    <div id="number">횟수 = 0</div>
    <button onclick="fire()" style="height:24pt;width:50pt">발사</button>
    <button onclick="reStart()" style="height:24pt;width:80pt">다시시작</button>
  </div>
```

jQuery Game (script 1)



<script>

```
/*
    Game upgrade
    - convert to jQuery
*/
```

```
var canvas = document.getElementById("myCanvas");
var context = canvas.getContext("2d");
```

```
/* 변수 초기화 */
```

```
var velocity; // 사용자가 입력한 공의 초기속도
var angle; // 사용자가 입력한 공의 초기각도
var net;
var ballv; // 공의 현재 속도 (2차원 속도)
var ballvx; // 공의 현재 x방향 속도
var ballvy; // 공의 현재 y방향 속도
var ballx = 10; // 공의 현재 x방향 위치
var bally = 260; // 공의 현재 y방향 위치 */
var ballRadius = 10; // 공의 반지름
var score = 0; // 점수 초기화
var netWidth = 30;
var net_h = 150;
var number = 0;
```

```
var image = new Image(); // 이미지 객체 생성
image.src = "media/lawn.png"; // 이미지 파일 이름 설정
var netimage = new Image();
netimage.src = "media/net.png";
var timer; // 타이머 객체 변수 to control cannonball
```

```
/* 공을 화면에 그린다. */
```

```
function drawBall() {
    context.beginPath();
    context.arc(ballX, ballY, ballRadius, 0, 2.0 * Math.PI, true);
    context.fillStyle = "red";
    context.fill();
}
```

```
function random(n1,n2) {
    return Math.floor(Math.random() * (n2-n1+1)+n1);
}
```

```
/* 배경을 화면에 그린다. */
```

```
function drawBackground() {
    context.drawImage(image, 0, 270);
    context.drawImage(netimage, 450, random(10,50),30,net_h);
}
```

```
/* 전체 화면을 그리는 함수 */
```

```
function draw() {
    context.clearRect(0, 0, 500, 300);
    /* 화면을 지운다. */
    drawBall();
    drawBackground();
    var msg = "Score : "+ score;
    context.font = "30px 'Gothic'";
    context.fillText(msg,15,50);
    gameOver();
    gameClear();
}
```

```
/* 초기화를 담당하는 함수 */
```

```
function init() {
    ballx = 10;
    bally = 260;
    ballRadius = 10;
    draw();
}
```

jQuery Game (script 2)



```
/* 사용자가 발사 버튼을 누르면 호출된다. */
function fire() {
    init();
    velocity = Number(document.getElementById("velocity").value);
    angle = Number(document.getElementById("angle").value);
    var angleR = angle * Math.PI / 180;

    ballVx = velocity * Math.cos(angleR);
    ballVy = -velocity * Math.sin(angleR); // negative y-direction

    net_h = Number(document.getElementById("netHeight").value);

    draw();
    number++;
    timer = setInterval(calculate, 100); // next position & velocity of cannonball
    document.getElementById("number").innerHTML = "횟수 = " + number;
    //return false;
}

/* 공의 현재 속도와 위치를 업데이트한다. */
function calculate() {
    ballVy = ballVy + 1.98; // y-방향 속도 계산 (Vy = Vy + g*dt)

    ballX = ballX + ballVx;
    ballY = ballY + ballVy;

    /* Hit test : 공이 목표물에 맞았으면 */
    if ((ballX >= 450) && (ballX <= 480) && (ballY >= 60) && (ballY <= 60+net_h)) {
        score++;

        document.getElementById("score").innerHTML = "점수 = " + score;
        clearInterval(timer);
        init();
    }

    /* 공이 경계를 벗어났으면 */
    if (ballY >= 300 || ballY < 0) {
        clearInterval(timer);
        init();
    }

    draw();
}
```

jQuery Game (script 3)



```
function reStart(){
    //context.clearRect(0,0,500,300);
    if(timer != null)
        clearInterval(timer);

    ballX = 10;
    ballY = 260;
    ballRadius = 10;
    number = 0;
    score = 0;
    document.getElementById("number").innerHTML = "횟수 = " + number;
    document.getElementById("score").innerHTML = "점수 = " + score;
    document.getElementById("velocity").value = 30;
    document.getElementById("angle").value = 45;
    document.getElementById("netHeight").value = 80;
    draw();
    drawBall();
    drawBackground();
}
function gameOver(){
    if((score < 6)&&(number == 10)){
        clearInterval(timer);
        var k = "Game over";
        var k1 = "Press new start.";
        context.font = "30px 'Gothic'";
        context.fillStyle = "red";
        context.fillText(k,145,150);
        context.fillText(k1,145,180);
    }
}
function gameClear(){
    if(score == 10){
        clearInterval(timer);
        var k = "Game Clear!";
        var k1 = "Press new start.";
        context.font = "30px 'Gothic'";
        context.fillStyle = "red";
        context.fillText(k,145,150);
        context.fillText(k1,145,180);
    }
}
```

jQuery Game (script 4)



```
<body onload="init();">
```



```
$(document).ready(function () {  
    init();  
})
```

```
</script>
```

jQuery로 변환된 파일을
[MSnn_jquery_cannonball.html](#) 로 저장하시오.

jQuery 게임 update #2



[Target]

0. Javascript 를 jQuery로 변환 (기본점수 : 3).

1. canvas에 현재 점수 표시 및 갱신 (+1).

2. 게임 CLEAR : 10점 이상을 얻으면 게임 클리어. 다시 시작 안내. (+1)

3. 게임 OVER : 초기 10회 발사에서 6점 이상을 얻지 못하면 게임 끝.
게임 끝나면 점수 초기화 및 초기 화면 재구성. (+1)

4. 게임 난이도 조정 (+3)

게임 클리어 후 넷 높이를 현재 높이에서 30만큼 작게 조정.

높이가 60 미만으로 변경된 네트를 통과하면 점수가 2점
올라가게 코드를 변경.

5. JQUERY로 coding

CHAPTER 12-III

AJAX & JSON

AJAX = Asynchronous JavaScript and XML.

AJAX is the art of exchanging data with a server, and **updating parts of a web page - without reloading the whole page.**

JSON: JavaScript Object Notation.

JSON is a **syntax** for **storing and exchanging data.**

Target:

Learning AJAX & JSON

```
{  
  "name": "HongGilDong",  
  "age": 23,  
  "address": {  
    "nation": "Korea",  
    "city": "GimHae",  
    "postalCode": "12345"  
  },  
  "특기": ["jQuery", "Node.js",  
           "plotly.js", "Arduino", "R"],  
  "phone": "010-1234-5678"  
}
```

MSnn.json

Learning AJAX : load JSON

MS00, My Personal Information:

Name : HongGilDong

Nation : Korea

City : GimHae

Post Code : 12345

Phone : 010-1234-5678

주특기1 : jQuery

주특기2 : Node.js

주특기3 : plotly.js

Get my information

Reference (w3schools.com)

AJAX Tutorial

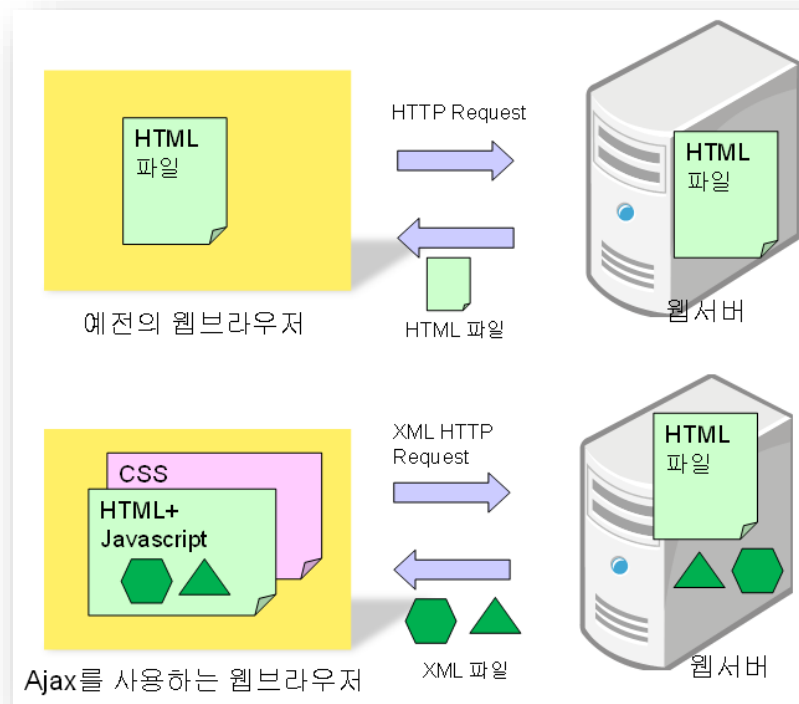
https://www.w3schools.com/xml/ajax_intro.asp

JSON Tutorial

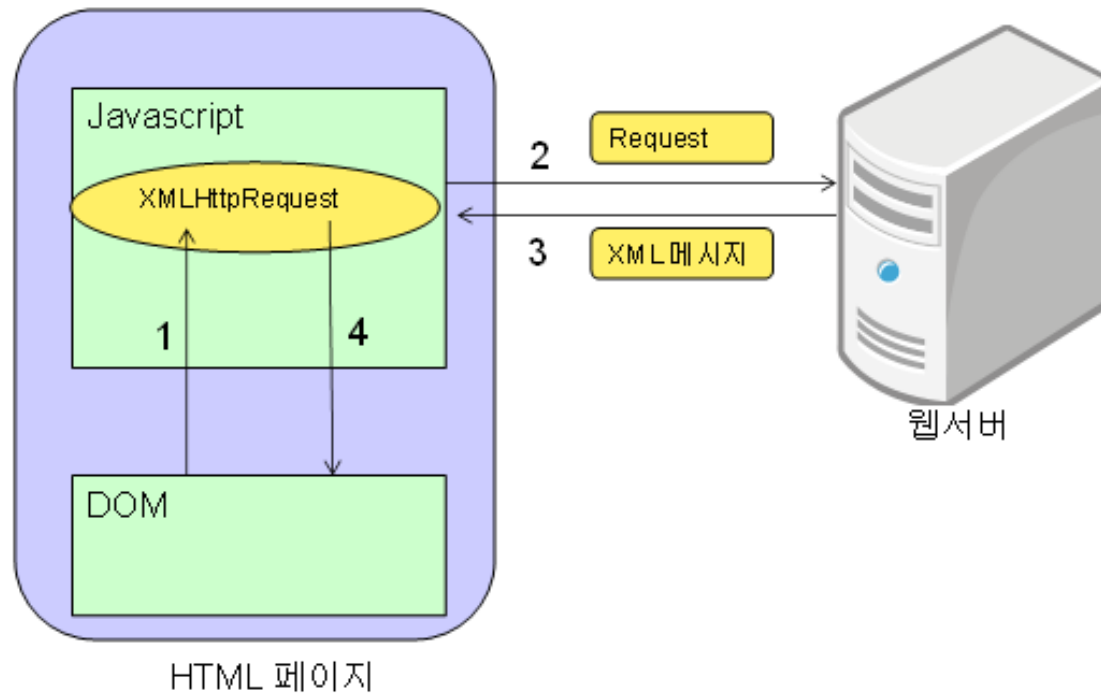
https://www.w3schools.com/js/js_json_intro.asp

AJAX 개요

- AJAX (**Asynchronous JavaScript and XML**)는 서버와 데이터를 교환하는 기술의 하나
- 웹 문서 전체를 갱신하지 않고 **일부분만 갱신**
- HTML, CSS, Javascript, XML, XMLHttpRequest 표준 기술 연동
- **XML 파일 조각 단위로 처리** 가능



AJAX의 동작 원리



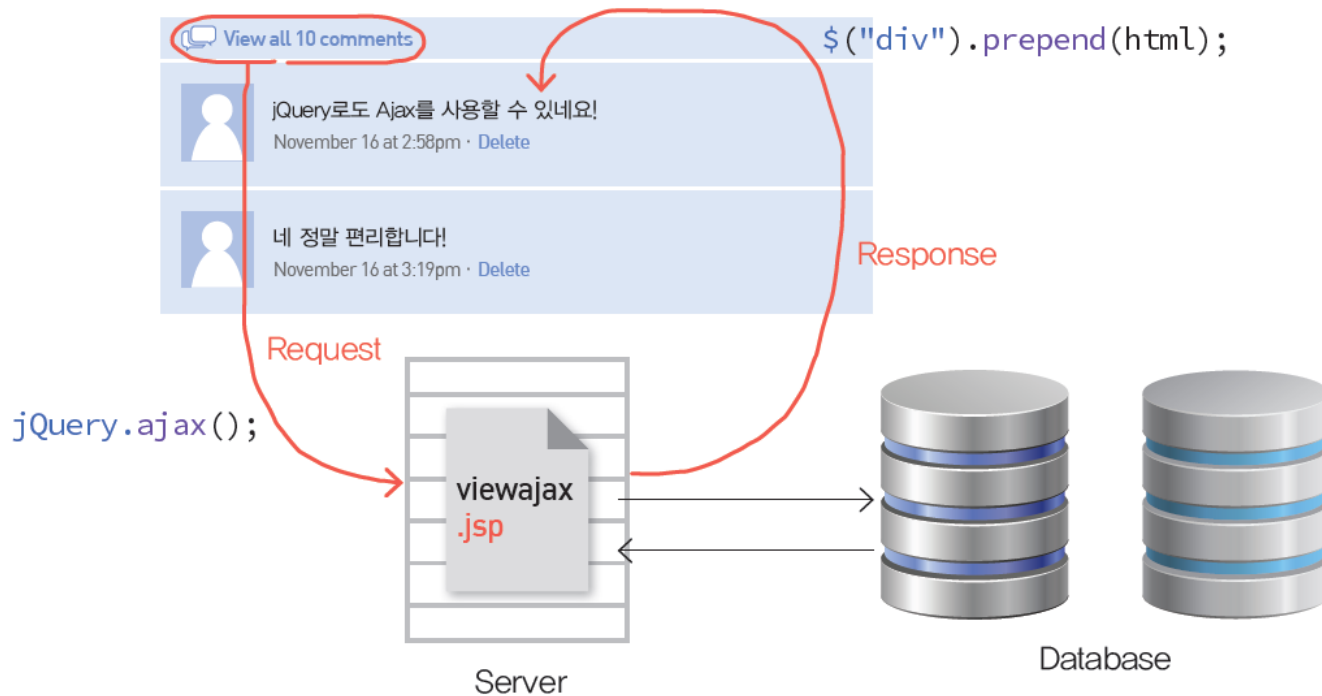
전체 페이지의 갱신 없이 특정 **DOM** 객체를 업데이트
XMLHttpRequest 객체로 통신

AJAX는 웹서버가 필요함

- AJAX는 필수적으로 웹서버가 필요
- Brackets, Aptana Studio, VS Express for Web은 자체적으로 웹서버를 운용한다.
- testfile1.txt 파일도 정해진 디렉토리에 있어야 한다.

jQuery를 이용한 AJAX

- jQuery를 사용하면 좀 더 쉽게 Ajax를 사용할 수 있다.
- jQuery 라이브러리 내에서 일관된 처리



Get data file by pure jQuery

Ajax는 **Asynchronous JavaScript and XML**의 약자입니다.

testfile1.txt

Ajax는 동적인 대화형 웹페이지를 만듭니다.

```
<body>
<h1>Learning AJAX : <br>read date by jQuery</h1>

<div id="target" style="width: 300px; height: 60px; border: solid 1px black">
  </div>
  <button>Get Data by jQuery</button>

  <script>
    $(document).ready(function () {
      $("button").click(function () {
        $("#target").load("data/testfile1.txt");
      });
    });
  </script>
</body>
```

**Learning AJAX :
read date by jQuery**

Ajax는 Asynchronous JavaScript and XML의 약자입니다. Ajax는 동적인 대화형 웹페이지를 만듭니다.

Get Data by jQuery

AJAX 예제 1 (XMLHttp)

Ajax는 **Asynchronous JavaScript and XML**의 약자입니다.

testfile1.txt

Ajax는 동적인 대화형 웹페이지를 만듭니다.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
  <title> Mobile Simulation: AJAX </title>

  <style type="text/css">

</style>

</head>

<body>
<h1>Learning AJAX : XMLHttpRequest</h1>

<div id="target" style="width: 300px; height: 60px; border: solid; 1px black;">
  </div>
  <button type="button" onclick="getFromServer()">GET DATA</button>
```

AJAX 예제 (XMLHttpRequest)

```
<script>
function getFromServer() {
    var req;
    if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari
        req = new XMLHttpRequest();
    }
    else { // code for IE6, IE5
        req = new ActiveXObject("Microsoft.XMLHTTP");
    }
    req.onreadystatechange = function () {
        if (req.readyState == 4 && req.status == 200) {
            document.getElementById("target").innerHTML = req.responseText;
        }
    };
    req.open("GET", "data/testfile1.txt", true); // async mode -> onreadystatechange
    req.send();
}
</script>
```

Learning AJAX : XMLHttpRequest

Ajax는 Asynchronous JavaScript and XML의 약자입니다. Ajax는 동적인 대화형 웹페이지를 만듭니다.

GET DATA

AJAX 예제 (XMLHttpRequest)

Send a Request To a Server

To send a request to a server, we use the `open()` and `send()` methods of the `XMLHttpRequest` object:

```
xmlhttp.open("GET","ajax_info.txt",true);  
xmlhttp.send();
```

Method	Description
<code>open(<i>method</i>,<i>url</i>,<i>async</i>)</code>	<p>Specifies the type of request, the URL, and if the request should be handled asynchronously or not.</p> <p><i>method</i>: the type of request: GET or POST <i>url</i>: the location of the file on the server <i>async</i>: true (asynchronous) or false (synchronous)</p>
<code>send(<i>string</i>)</code>	<p>Sends the request off to the server.</p> <p><i>string</i>: Only used for POST requests</p>

GET or POST?

GET is simpler and faster than POST, and can be used in most cases.

However, always use POST requests when:

- A cached file is not an option (update a file or database on the server)
- Sending a large amount of data to the server (POST has no size limitations)
- Sending user input (which can contain unknown characters), POST is more robust and secure than GET

AJAX 예제 (XMLHttpRequest)

XMLHttpRequest 객체의 세 가지 중요한 속성은 다음과 같다.

속성	설명
onreadystatechange	readyState 상태가 변경될 때마다 자동으로 호출되는 함수를 저장한다.
readyState	XMLHttpRequest의 4가지 상태를 저장한다. 0에서 4로 순차적으로 변화된다. 0: 요청이 초기화되지 않았다. 1: 서버 연결이 이루어졌다. 2: 요청이 수신되었다. 3: 요청을 처리 중이다. 4: 요청 처리가 종료되고 응답이 준비되었다.
status	200: "OK" 404: 페이지가 발견되지 않았다.

따라서 ReadyState가 4가 되고 status가 200이면 응답이 준비되었고 이상이 없는 것이다.

```
req.onreadystatechange = function () {  
    if (req.readyState == 4 && req.status == 200) {  
        document.getElementById("target").innerHTML = req.  
responseText;  
    }  
}
```

여기서 주의할 사항은 실제로 Onreadystatechange 이벤트는 5번 발생한다는 점이다. 즉 readyState가 0에서 4까지 변경될 때마다 한 번씩 발생하는 것이다. 따라서 반드시 readyState 값을 체크해야 한다.

AJAX 예제 (XMLHttpRequest)

The onreadystatechange event

When a request to a server is sent, we want to perform some actions based on the response.

The onreadystatechange event is triggered every time the readyState changes.

The readyState property holds the status of the XMLHttpRequest.

Three important properties of the XMLHttpRequest object:

Property	Description
onreadystatechange	Stores a function (or the name of a function) to be called automatically each time the readyState property changes
readyState	Holds the status of the XMLHttpRequest. Changes from 0 to 4: 0: request not initialized 1: server connection established 2: request received 3: processing request 4: request finished and response is ready
status	200: "OK" 404: Page not found

In the onreadystatechange event, we specify what will happen when the server response is ready to be processed.

When readyState is 4 and status is 200, the response is ready:

AJAX 예제 (XMLHttpRequest)

Server Response

To get the response from a server, use the `responseText` or `responseXML` property of the `XMLHttpRequest` object.

Property	Description
<code>responseText</code>	get the response data as a string
<code>responseXML</code>	get the response data as XML data

The `responseText` Property

If the response from the server is not XML, use the `responseText` property.

The `responseText` property returns the response as a string, and you can use it accordingly:

AJAX 예제 (실행 결과)



반드시 VS for Web을 사용한다! --- No!

-> Brackets, Aptana Studio 3 도 Web server를 제공한다.

Learning AJAX : XMLHttpRequest

Ajax는 Asynchronous JavaScript and XML의 약자입니다. Ajax는 동적인 대화형 웹페이지를 만듭니다.

GET DATA

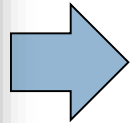
도전과제 #1

Javascript로 동작되는 아래 결과를 jQuery로 동작되게 변경하고 파일을 [MSnn_AJAX_jQuery.html](#) 로 저장하시오.

Learning AJAX : XMLHttpRequest

Ajax는 Asynchronous JavaScript and XML의 약자입니다. Ajax는 동적인 대화형 웹페이지를 만듭니다.

GET DATA



Learning AJAX : XMLHttpRequest by jQuery

Ajax는 Asynchronous JavaScript and XML의 약자입니다. Ajax는 동적인 대화형 웹페이지를 만듭니다.

GET DATA

AJAX 예제 2 (XML 파일 처리)

cd_catalog.xml

[illegible]

Learning AJAX : load XML

My CD Collection:

Get my CD collection

Learning AJAX : load XML

My CD Collection:

Bob Dylan
Bonnie Tyler
Dolly Parton
Gary Moore
Eros Ramazzotti
Bee Gees
Dr. Hook
Rod Stewart
Andrea Bocelli
Percy Sledge
Savage Rose
Many
Kenny Rogers
Will Smith
Van Morrison
Jorn Hoel
Cat Stevens
Sam Brown
T'Pau
Tina Turner
Kim Larsen
Luciano Pavarotti
Otis Redding
Simply Red
The Communards
Joe Cocker

Get my CD collection

AJAX 예제 (XML 파일 처리)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
  <title> Mobile Simulation: AJAX </title>

  <style type="text/css">

</style>

</head>

<body>
<h1>Learning AJAX : load XML</h1>

<h2>My CD Collection:</h2>
<div id="myDiv"> </div>
<button type="button" onclick="loadXMLDoc()">Get my CD collection
</button>
```

Learning AJAX : load XML

My CD Collection:

Get my CD collection

AJAX 예제 (XML파일 처리)

```
<script>
function loadXMLDoc() {
    var xmlreq;
    var txt, x, i;
    if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlreq = new XMLHttpRequest();
    } else { // code for IE6, IE5
        xmlreq = new ActiveXObject("Microsoft.xmlreq");
    }
    xmlreq.onreadystatechange = function() {
        if (xmlreq.readyState == 4 && xmlreq.status == 200) {
            xmlDoc = xmlreq.responseXML; // XML data
            txt = "";
            x = xmlDoc.getElementsByTagName("ARTIST"); // parsing data in xml
            for ( i = 0; i < x.length; i++) {
                txt = txt + x[i].childNodes[0].nodeValue + "<br>";
            }
            document.getElementById("myDiv").innerHTML = txt;
        }
    };
    xmlreq.open("GET", "data/cd_catalog.xml", true); // async mode
    xmlreq.send();
}
</script>
```

도전과제 #2

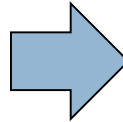
Javascript로 동작되는 아래 결과를 jQuery로 동작되게 변경하고
파일을 [MSnn_AJAX_jQuery_XML.html](#) 로 저장하시오.

Learning AJAX : load XML

My CD Collection:

Bob Dylan
Bonnie Tyler
Dolly Parton
Gary Moore
Eros Ramazzotti
Bee Gees
Dr.Hook
Rod Stewart
Andrea Bocelli
Percy Sledge
Savage Rose
Many
Kenny Rogers
Will Smith
Van Morrison
Jorn Hoel
Cat Stevens
Sam Brown
T'Pau
Tina Turner
Kim Larsen
Luciano Pavarotti
Otis Redding
Simply Red
The Communards
Joe Cocker

Get my CD collection



Learning AJAX : load XML by jQuery

My CD Collection:

Bob Dylan
Bonnie Tyler
Dolly Parton
Gary Moore
Eros Ramazzotti
Bee Gees
Dr.Hook
Rod Stewart
Andrea Bocelli
Percy Sledge
Savage Rose
Many
Kenny Rogers
Will Smith
Van Morrison
Jorn Hoel
Cat Stevens
Sam Brown
T'Pau
Tina Turner
Kim Larsen
Luciano Pavarotti
Otis Redding
Simply Red
The Communards
Joe Cocker

Get my CD collection

JSON

JSON Tutorial

« W3Schools Home



JSON: **J**ava**S**cript **O**bject **N**otation.

JSON is a **syntax** for storing and exchanging data.

JSON is an **easier to use** alternative to XML.

https://www.w3schools.com/js/js_json_intro.asp

JSON vs. XML

This JSON syntax defines an employees object, with an array of 3 employee records (objects):

JSON Example

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

JSON

This XML syntax also defines an employees object with 3 employee records:

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>
```

JSON vs. XML

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

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.

Why JSON?

For AJAX applications, JSON is faster and easier than XML:

Using XML

- Fetch an XML document
- Use the XML DOM to loop through the document
- Extract values and store in variables

Using JSON

- Fetch a JSON string
- JSON.Parse the JSON string

JSON

- **JSON (JavaScript Object Notation)**은 텍스트-기반의 데이터 교환 형식
- JSON은 자바 스크립트 언어에서 유래
- JSON 형식은 **Douglas Crockford**에 의하여 처음으로 지정되었으며, RFC 4627에 기술
- 공식적인 인터넷 미디어 타입은 **application/json**이며 파일 이름 확장자는 **.json**
- 특정 언어에 독립적인 데이터 교환 형식
- 태그를 사용하지 않으므로 xml 보다 간결하며 parsing도 쉬움.
- Array 를 구성 요소로 포함 가능.

JSON 구문

- JSON data

- ▣ { "name" : "Hong" }

- JSON object

- ▣ { "name" : "Hong", "age" : "21" }

- JSON array

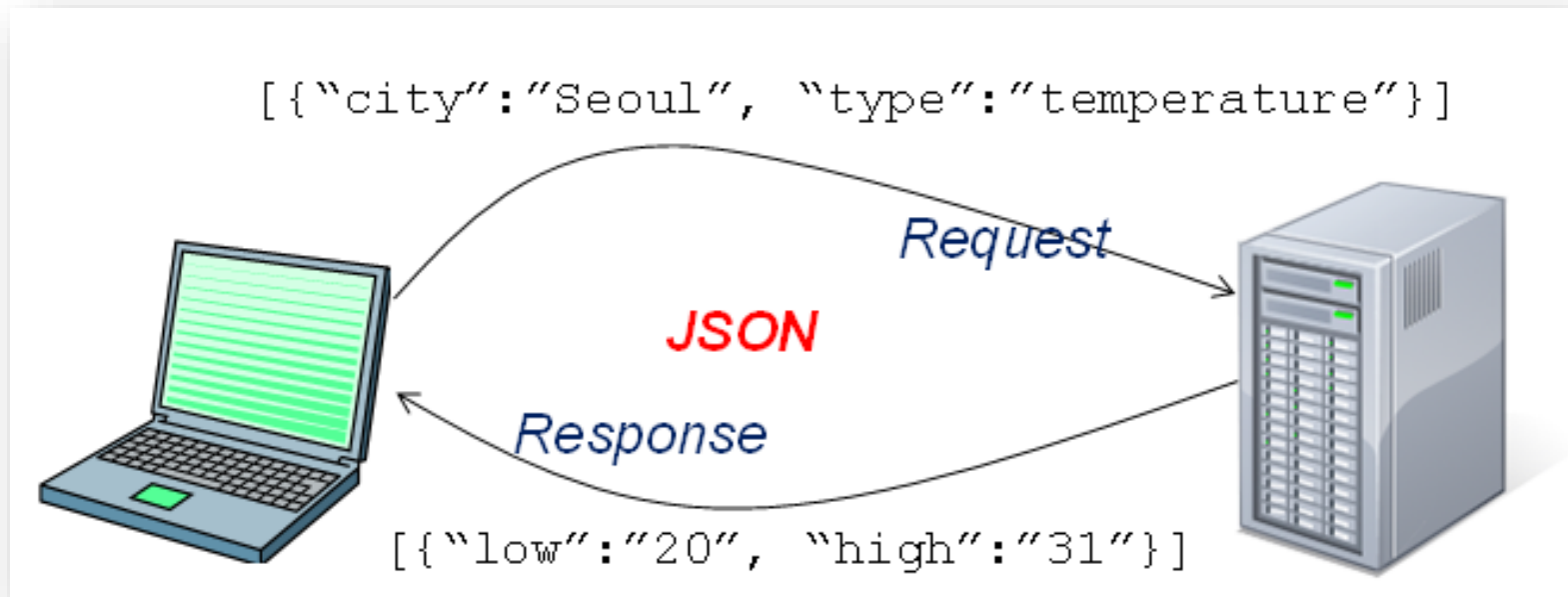
- ▣ [{ "name" : "Hong", "age" : "21" },
 { "name" : "Kim", "age" : "22" },
 { "name" : "Park", "age" : "23" }]

JSON으로 표현된 객체

```
{
  "name": "HongGilDong",
  "age": 23,
  "address": {
    "nation": "Korea",
    "city": "GimHae",
    "postalCode": "12345"
  },
  "특기": ["jQuery", "Node.js", "plotly.js"],
  "phone": "010-1234-5678"
}
```

MSnn.json

JSON의 사용



JSON 예제 (*eval()* 처리)

```
<!DOCTYPE html>
<html>
<body>
  <h4>학생 명단</h4>
  <p style="background-color: yellow">
    이름: <span id="name"></span>&nbsp;
    나이: <span id="age"></span>
  </p>

  <script>
    var s = '[' +
      '{"name":"Hong","age":"21" },' +

      '{"name":"Kim","age":"22" },' +
      '{"name":"Park","age":"23" }'];

    var students = eval("(" + s + ")");

    //students[1].name = "Lee";
    document.getElementById("name").innerHTML = students[1].name;
    document.getElementById("age").innerHTML = students[1].age;
  </script>

</body>
</html>
```

Learning JSON : eval()

학생 명단

이름: Kim 나이: 22

JSON 예제 (**parse()** 처리)

```
<!DOCTYPE html>
<html>
<body>
  <h4>학생 명단</h4>
  <p style="background-color: yellow">
    이름: <span id="name"></span>&nbsp;
    나이: <span id="age"></span>
  </p>

  <script>
    var s = '[' +
      '{"name":"Hong","age":"21" },' +

      '{"name":"Kim","age":"22" },' +
      '{"name":"Park","age":"23" }]];

    var students = JSON.parse(s);

    //students[1].name = "Lee";
    document.getElementById("name").innerHTML = students[1].name;
    document.getElementById("age").innerHTML = students[1].age;
  </script>

</body>
</html>
```

Learning JSON : **parse()**

학생 명단

이름: Kim 나이: 22

도전과제 #3

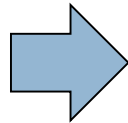
1. 본인의 다양한 정보를 담고 있는 json 파일을 만드시오.
파일명 : **MSnn.json** (nn is your id#)
2. Json 파일은 text 파일이므로 외부 txt 파일을 load 하는 **loadJSONDoc()** 함수를 만드시오. (XMLHttpRequest 사용)
3. 읽어들이는 json 파일에 담긴 내용을 parsing 하여 개인 정보를 출력하시오.

Html 파일을 **MSnn_AJAX_JSON.html** 로 저장하시오.

Learning AJAX : load JSON

MSnn, My personal information:

Get my Info.



Learning AJAX : load JSON

MSnn, My personal information:

Name : HongGilDong

Nation : Korea

Get my Info.

도전과제 #3

1. 본인의 다양한 정보를 담고 있는 json 파일을 만드시오.
파일명 : MSnn.json (nn is your id#)

```
{
  "name": "HongGilDong",
  "age": 23,
  "address": {
    "nation": "Korea",
    "city": "GimHae",
    "postalCode": "12345"
  },
  "특기": ["jQuery", "Node.js",
    "plotly.js", "Arduino", "R"],
  "phone": "010-1234-5678"
}
```

도전과제 #3 - hint !

```
<h1>Learning AJAX : load JSON</h1>

<h2>MSnn, My personal information: </h2>
<div id="myDiv"> </div>
<button type="button" onclick="loadJSONDoc()">Get my Info. </button>

<script>
    function loadJSONDoc() {
        var jsonreq;
        var txt;
        var jsonDoc;

        if (window.XMLHttpRequest) { // code for IE7+, Firefox, Chrome, Opera, Safari
            jsonreq = new XMLHttpRequest();
        } else { // code for IE6, IE5
            jsonreq = new ActiveXObject("Microsoft.xmlreq");
        }
        jsonreq.onreadystatechange = function() {
            if (jsonreq.readyState == 4 && jsonreq.status == 200) {
                jsonDoc = JSON.parse(jsonreq.responseText); // json data
                txt = "";
                txt = txt + "Name : " + jsonDoc.name + "<br>";
                txt = txt + "Nation : " + jsonDoc.address.nation + "<br>";

                $("#myDiv").html(txt);
            }
        };
        jsonreq.open("GET", "data/msnn.json", true); // async mode
        jsonreq.send();
    }
</script>
```

과제07. msnn_rpt07.zip

55

[실습과제07] Convert Javascript to jQuery & AJAX/JSON

- [1] MSnn_jquery_clock.html
- [2] MSnn_jquery_cannonball.html
- [3] MSnn_AJAX_JSON.html, MSnn.json

**** **html** 파일 및 관련 파일(image, json 등..)을
MSnn_Rpt07.zip 으로 압축해서 제출하시오

가점: Javascript, jQuery 프로그래밍 응용 능력.

[제출파일] **msnn_rpt07.zip** (10월31일 오후 6시 마감)

html 파일과 사용된 그림 및 데이터파일을 압축하여
이메일로 "msnn_rpt07" 제목으로 제출

Email : chaos21c@gmail.com

Result



Learning AJAX : load JSON

MS00, My Personal Information:

Get my information

```
{
  "name": "HongGilDong",
  "age": 23,
  "address": {
    "nation": "Korea",
    "city": "GimHae",
    "postalCode": "12345"
  },
  "특기": ["jQuery", "Node.js",
    "plotly.js", "Arduino", "R"],
  "phone": "010-1234-5678"
}
```

Learning AJAX : load JSON

MS00, My Personal Information:

Name : HongGilDong

Nation : Korea

City : GimHae

Post Code : 12345

Phone : 010-1234-5678

주특기1 : jQuery

주특기2 : Node.js

주특기3 : plotly.js

Get my information

교재 WEB 강의 소개

← → ↻ ⓘ webprogramming.co.kr ☆

명품 **HTML5+
CSS3+
Javascript** 웹 프로그래밍

Home Introduction Notice Board Support Code

명품 **HTML5 + CSS3 + Javascript** 웹 프로그래밍

HTML5로
여러분의 무한한
상상력을
표현해 보세요!



Sir Tim Berners-Lee
(1955.6.8 ~)

명품 웹 프로그래밍 소개
“웹 프로그래밍을
가장 쉽게 익힐 수 있는 책”

처음 웹 프로그래밍을 공부하는 입문자들도
모든 주제를 직관적으로 이해하고
빠르게 파악할 수 있습니다.

자세히보기 →



강력한 Q&A 피드백 제공
“빠르고, 간결하고, 정확한
저자의 직접적인 답변”

“이거 이해가 잘 안되는데.. 물어볼 사람도 없고..
더이상 고민하지 마세요.
명품 웹 프로그래밍 홈페이지에서는
누구나 저자가 직접 답변해주는
Q&A 게시판을 이용할 수 있습니다.

자세히보기 →



즉석 실행 가능한 예제 프로그램
“백문이 불여일견, 백견이 불여일타(打)!”

코드로만 설명되어 있는 예제들,
결과 화면이 있어도 이해가 잘 안되시죠?
예제 소스를 바탕으로, 내맘대로 수정한
코드를 즉석으로 웹 페이지로
변환해주는 예제 프로그램을 통해
모든 코드를 빠르고 쉽게
이해할 수 있습니다.

자세히보기 →



Notice

Test

2017-01-16 15:32

Know-How

Test

2017-01-17 14:04 관리자

관련 WEB 강의 소개 – w3schools.com

← → ↻ 🔒 안전함 | https://www.w3schools.com

w3schools.com THE WORLD'S LARGEST WEB DEVELOPER SITE

TUTORIALS ▾ REFERENCES ▾ EXAMPLES ▾

HTML and CSS
Learn HTML
Learn CSS
Learn W3.CSS
Learn Colors
Learn Bootstrap
Learn Icons
Learn Graphics
Learn How To

JavaScript
Learn JavaScript
Learn W3.JS
Learn jQuery
Learn jQueryMobile
Learn AppML
Learn AngularJS
Learn JSON
Learn AJAX

Server Side
Learn SQL
Learn PHP
Learn ASP

Web Building
Web Templates
Web Statistics
Web Certificates

XML Tutorials
Learn XML
Learn XML AJAX
Learn XML DOM
Learn XML DTD
Learn XML Schema
Learn XSLT
Learn XPath
Learn XQuery

HTML

The language for building web pages

LEARN HTML HTML REFERENCE

HTML Example:

```
<!DOCTYPE html>
<html>
<title>HTML Tutorial</title>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Try it Yourself »

CSS

The language for styling web pages

LEARN CSS CSS REFERENCE

CSS Example:

```
body {
  background-color: lightblue;
}
h1 {
  color: white;
  text-align: center;
}
p {
  font-family: verdana;
  font-size: 20px;
}
```

Try it Yourself »

JavaScript

The language for programming web pages

JavaScript Example:

```
<script>
function myFunction() {
  var x = document.getElementById("demo");
  x.style.fontSize = "35px";
}
```