Mobile Simulation

2017-2

명품

HTML5+ CSS3+ Javascript 웹 프로그래밍

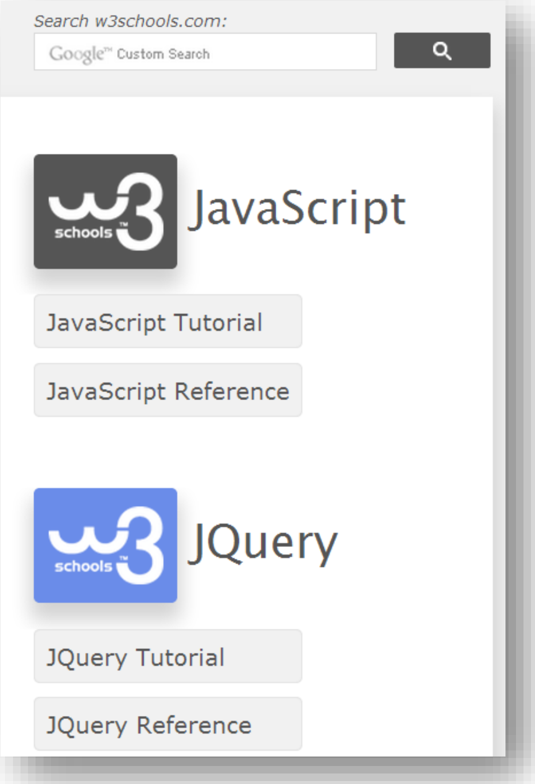
황기태 지음

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

- 어려운 내용을 그림과 실례로 설명해 깊이 있는 이해를 이끌어냅니다.
- 이론에서 실습까지 연결되는 예제로 손에 익히는 학습을 유도합니다.
- Open Challenge와 풍부한 연습 문제를 풀어 배운 내용을 정리합니다.
- 저자가 운영하는 명품 웹 프로그래밍 사이트를 통해 언제 어디서나 실습하고 질문할 수 있습니다.

명품 웹사이트에서 예제를 실행하고 직접 수정해 볼 수 있습니다

www.webprogramming.co.kr



Search w3schools.com:

w3schools JavaScript

- JavaScript Tutorial
- JavaScript Reference

w3schools JQuery

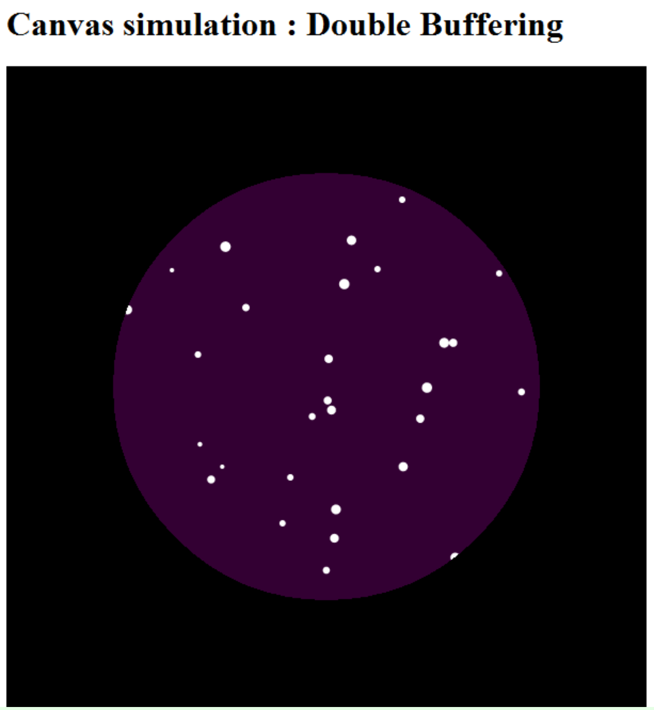
- JQuery Tutorial
- JQuery Reference

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 :**
- **wk07 :**
- **wk08 : Mid-term Exam.**
- **wk09 :**
- **wk10 :**
- **wk11 :**
- **wk12 :**
- **wk13 :**
- **wk14 :**
- **wk15 : Final exam.**



Canvas simulation : Double Buffering

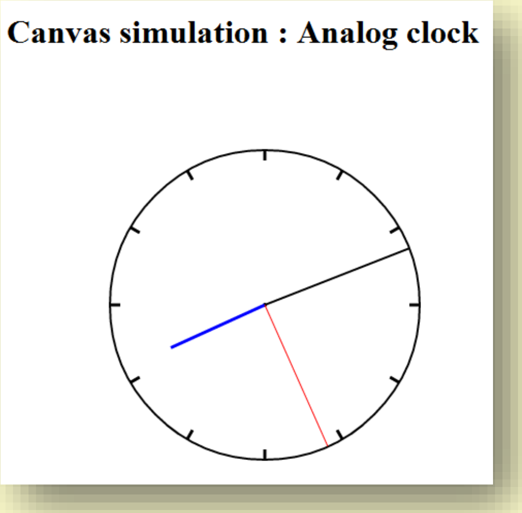
과제03. msnn_rpt03.zip

[실습과제03] Analog clock on canvas

[1] 캔버스에 자바스크립트만을 사용해서 아날로그 시계 만들기
[2] Date 객체로부터 현재 시간을 읽어서 시간을 최대한 정확하게 표시.
[3] 시계 주변에 ID, 이름 표시.

파일명: msnn_rpt03.html

가정: Javascript 프로그래밍 응용 능력.



Canvas simulation : Analog clock

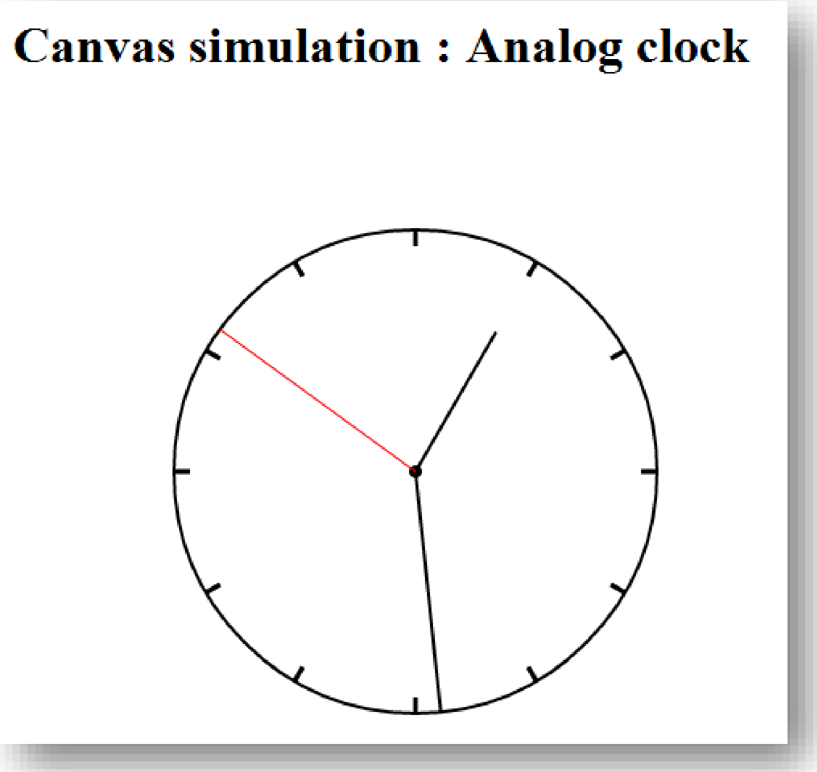
[제출파일] msnn_rpt03.zip

msnn_rpt03.html 과 사용된 그림을 이메일로 제출

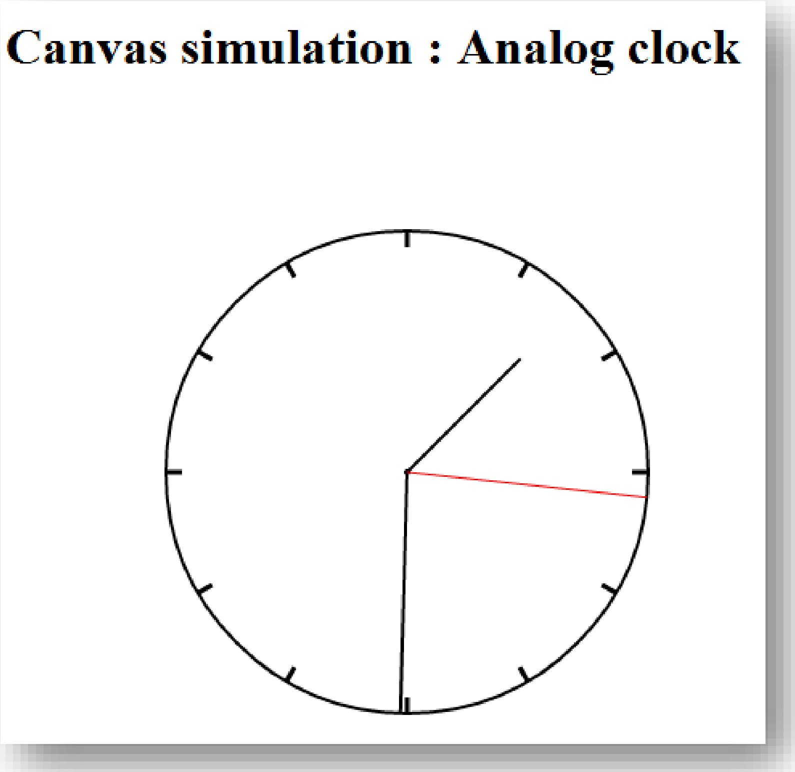
Email : chaos21c@gmail.com

chaos21c@gmail.com

과제03. hint



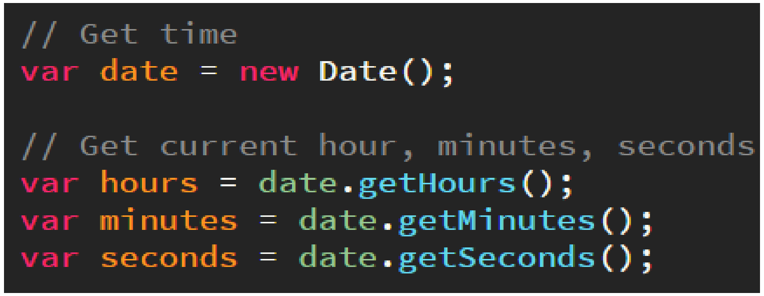
Canvas simulation : Analog clock



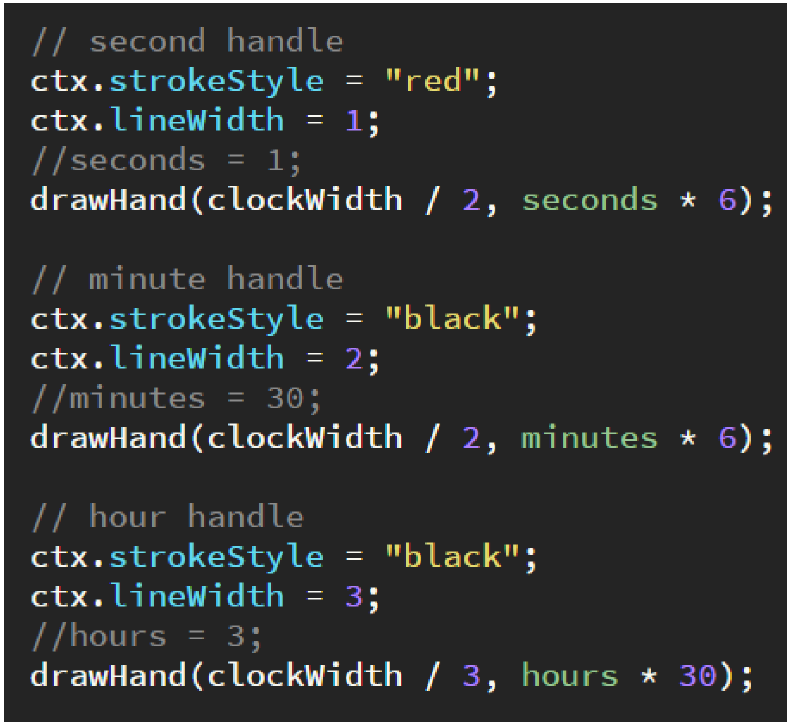
Canvas simulation : Analog clock

Which clock is running exactly in real time?

과제03. hint



```
// Get time
var date = new Date();

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

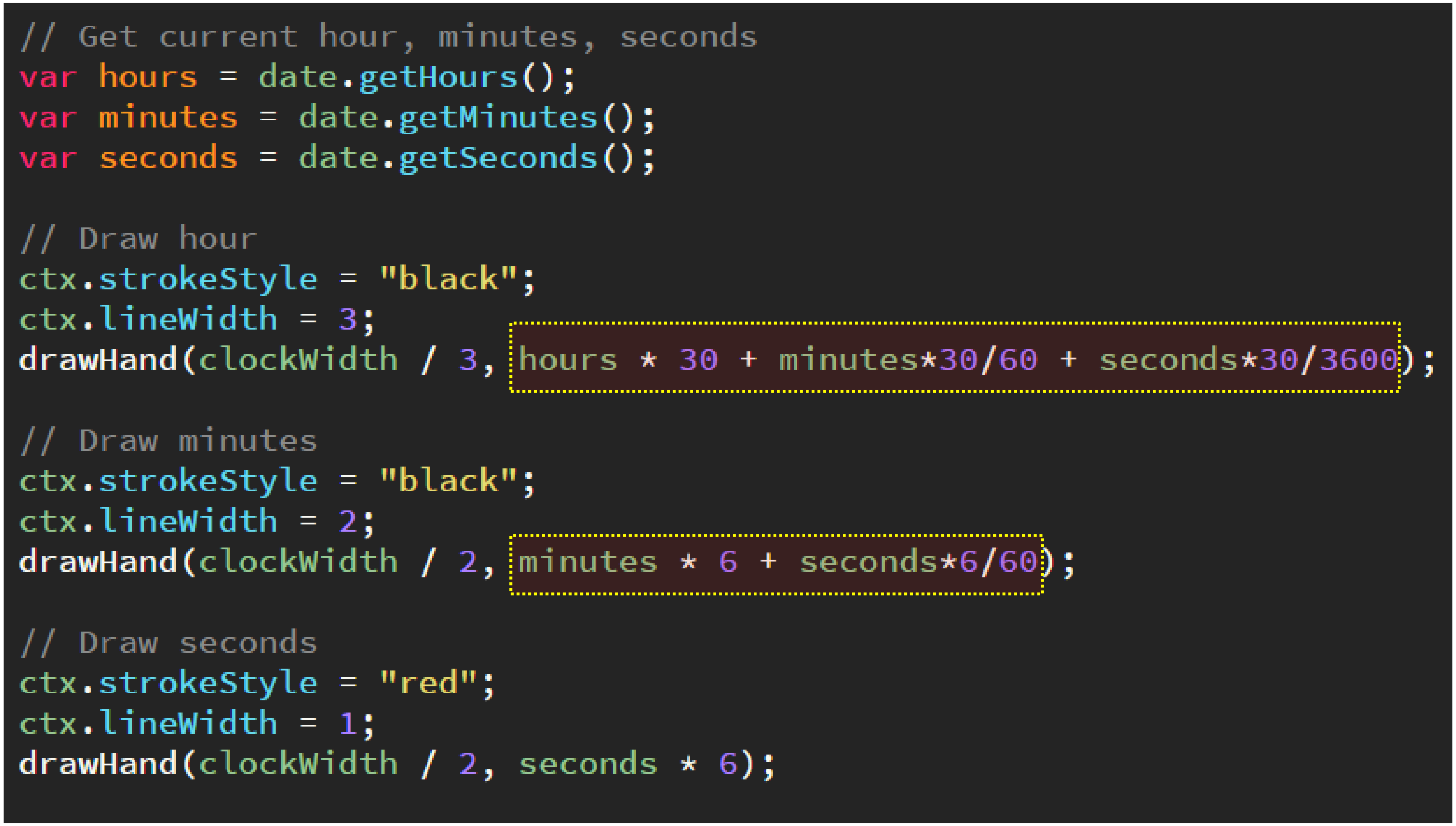


```
// second handle
ctx.strokeStyle = "red";
ctx.lineWidth = 1;
//seconds = 1;
drawHand(clockWidth / 2, seconds * 6);

// minute handle
ctx.strokeStyle = "black";
ctx.lineWidth = 2;
//minutes = 30;
drawHand(clockWidth / 2, minutes * 6);

// hour handle
ctx.strokeStyle = "black";
ctx.lineWidth = 3;
//hours = 3;
drawHand(clockWidth / 3, hours * 30);
```

?

How can you make every handle correctly move?

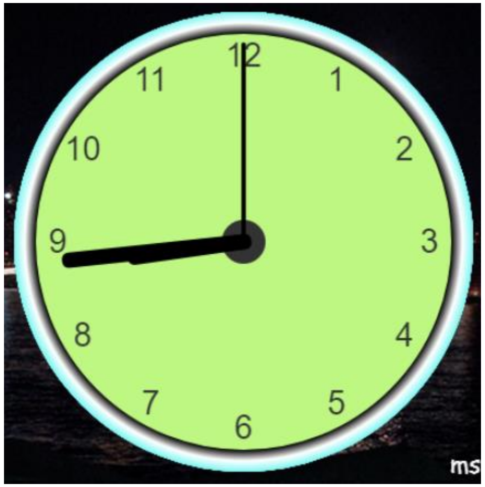
과제03. hint

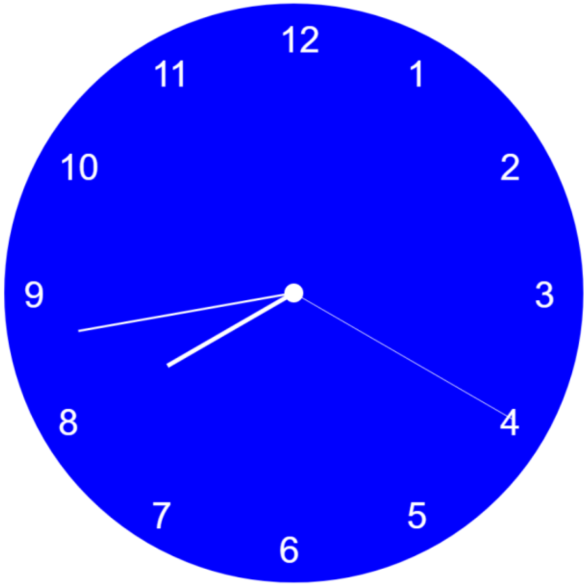


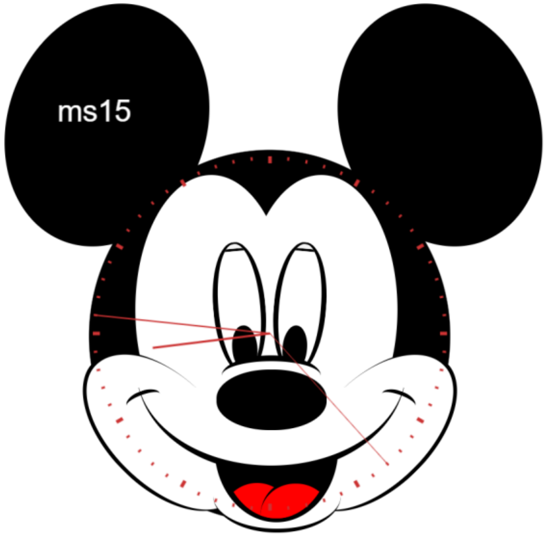
```
// Get current hour, minutes, seconds
var hours = date.getHours();
var minutes = date.getMinutes();
var seconds = date.getSeconds();

// Draw hour
ctx.strokeStyle = "black";
ctx.lineWidth = 3;
drawHand(clockWidth / 3, hours * 30 + minutes*30/60 + seconds*30/3600);

// Draw minutes
ctx.strokeStyle = "black";
ctx.lineWidth = 2;
drawHand(clockWidth / 2, minutes * 6 + seconds*6/60);

// Draw seconds
ctx.strokeStyle = "red";
ctx.lineWidth = 1;
drawHand(clockWidth / 2, seconds * 6);
```

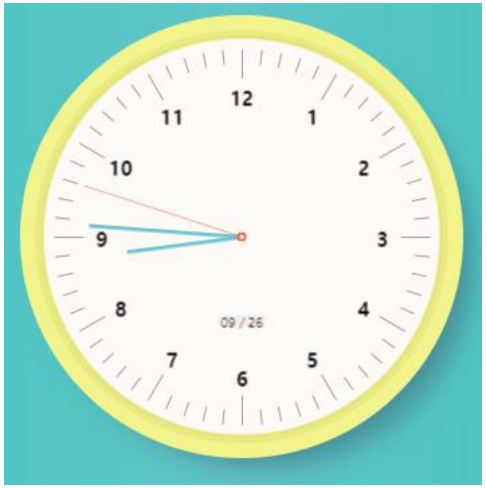
How can you make every handle correctly move?

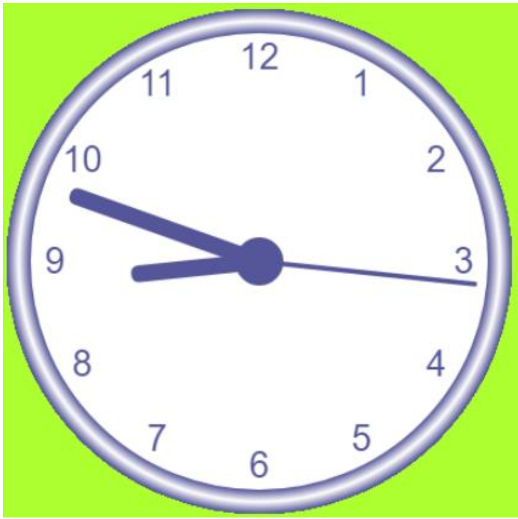
과제03. Good results







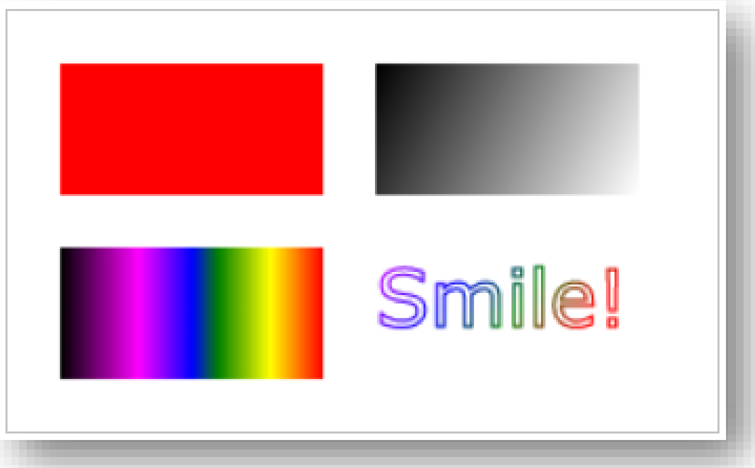




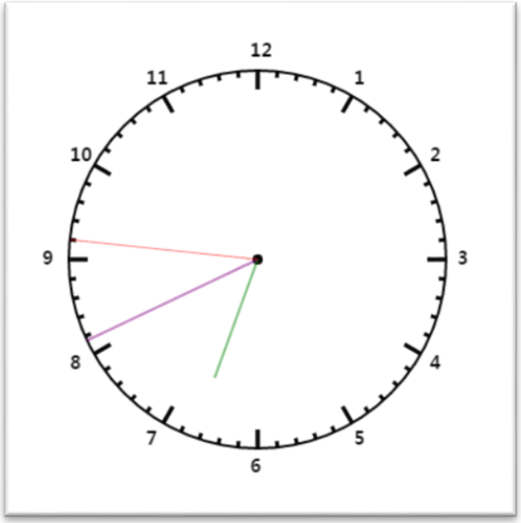
11

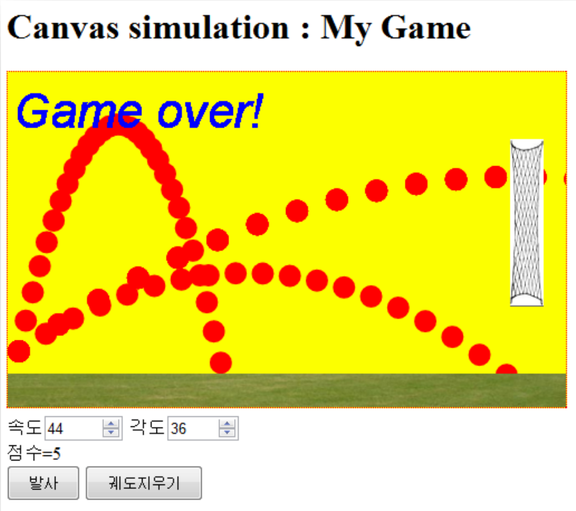
HTML5 캔버스 그래픽

Canvas III. Animation



The <canvas> element is used to draw graphics, on the fly, on a web page.



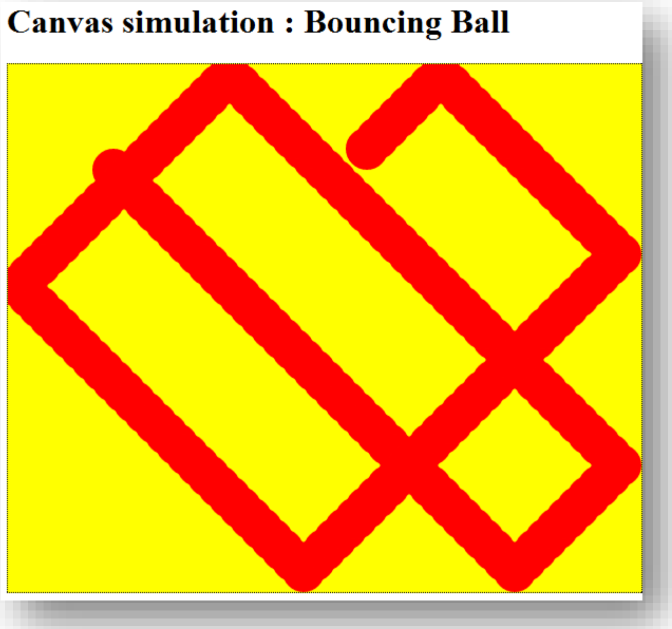

Canvas simulation : My Game

속도 44 각도 36

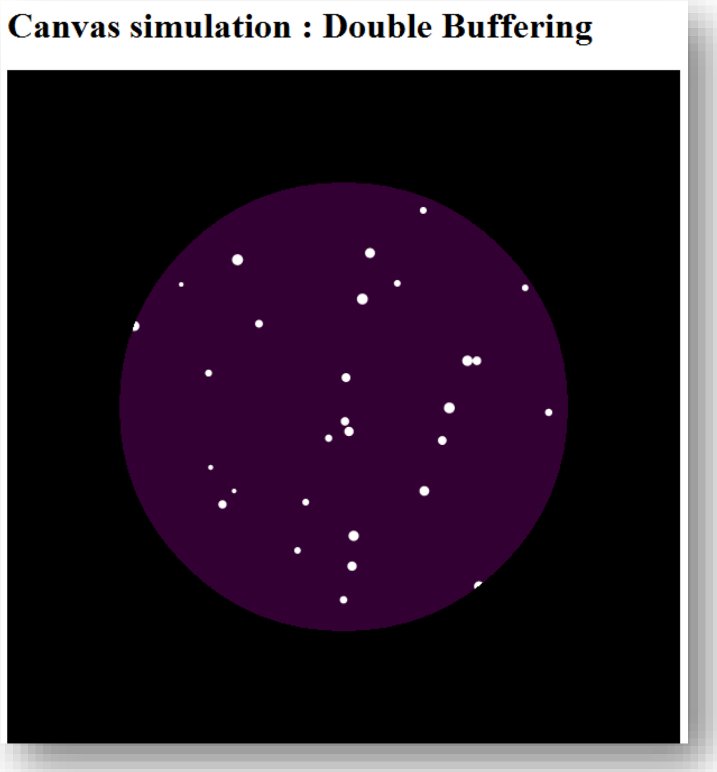
점수=5

Target:

Making HTML5 Animation using Canvas



Canvas simulation : Bouncing Ball



Canvas simulation : Double Buffering

Reference

HTML5 Canvas

http://www.w3schools.com/html/html5_canvas.asp

HTML DOM Canvas Object

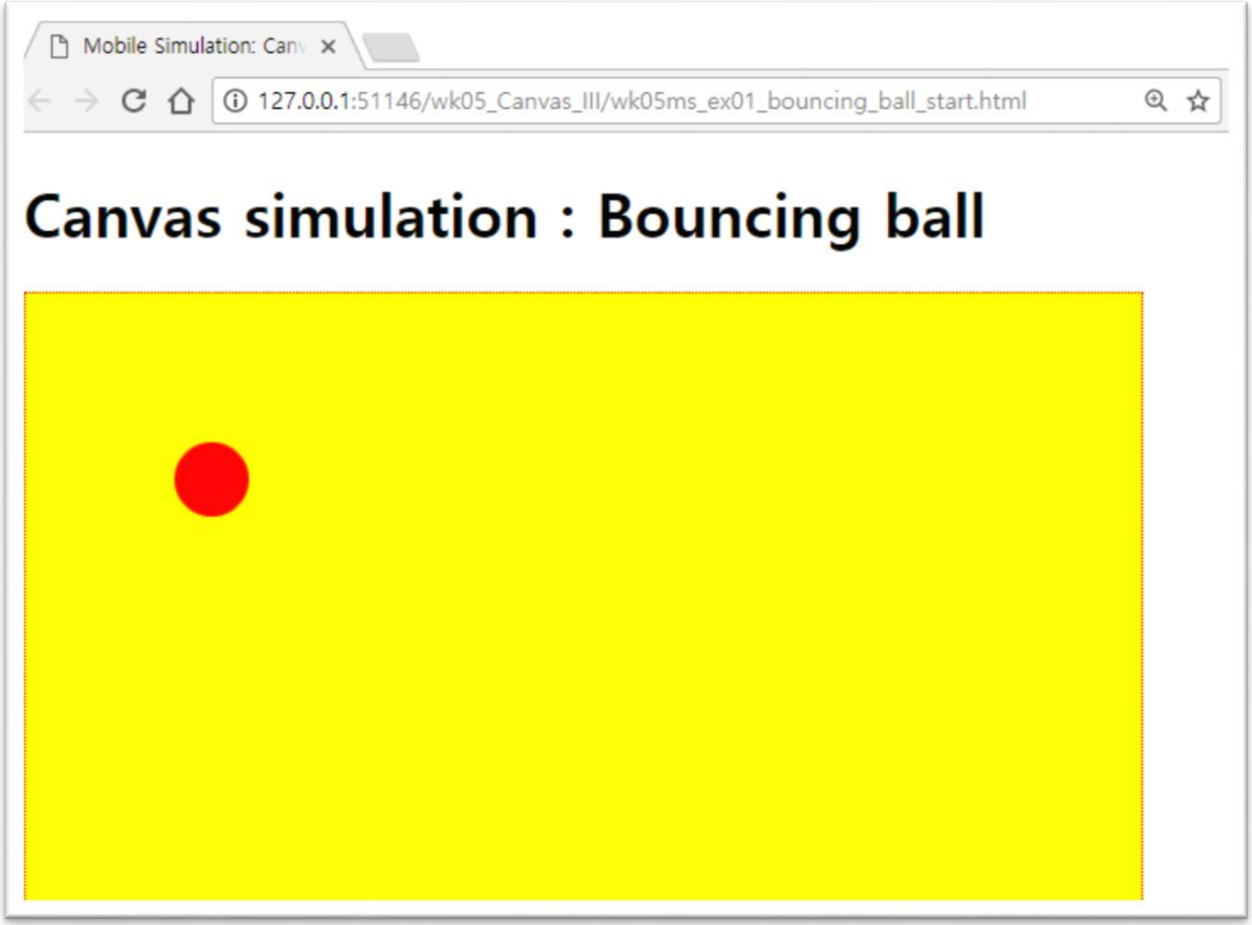
http://www.w3schools.com/jsref/dom_obj_canvas.asp

HTML Canvas Reference

http://www.w3schools.com/tags/ref_canvas.asp

애니메이션

- Bouncing Ball simulation



Mobile Simulation: Can
127.0.0.1:51146/wk05_Canvas_III/wk05ms_ex01_bouncing_ball_start.html

Canvas simulation : Bouncing ball

Bouncing Ball Simulation

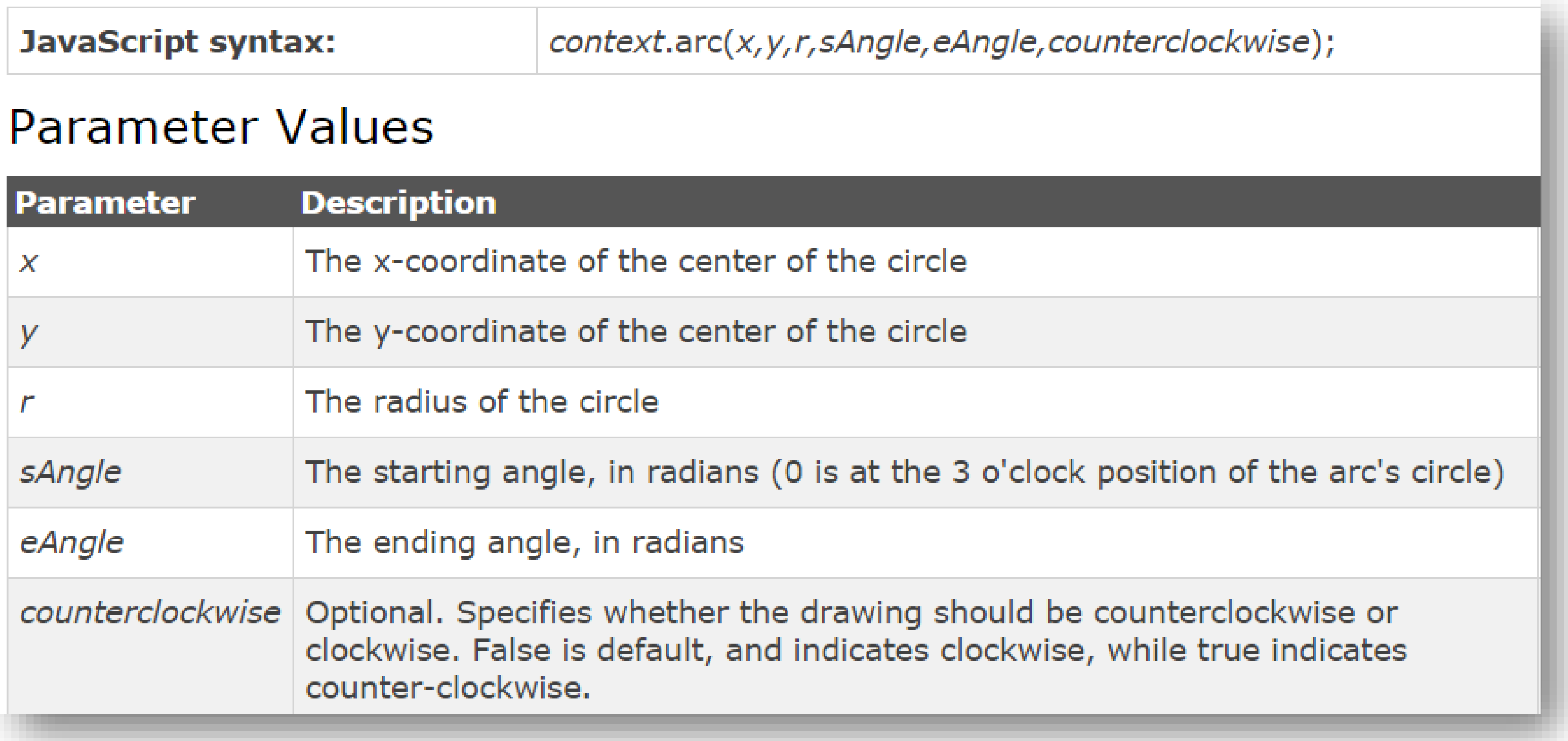
```
<!DOCTYPE html>
<html>
<head>
   <meta charset="utf-8" />
   <title> Mobile Simulation: Canvas </title>
   <style type="text/css">
          canvas {
                    background: yellow;
                    border: 1px dotted red;
          }
  </style>
```

```
<body>

   <h1>Canvas simulation : Bouncing Ball </h1>
   <canvas id="myCanvas" width="600" height="500">  </canvas>

</body>
```

Bouncing Ball code

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

  var dx = 5; // velocity in the x-direction
  var dy = 5; // velocity in the y-direction
  var x = 100;
  var y = 100;

  function draw() {
            context.clearRect(0, 0, 300, 200);
            context.beginPath();
            context.fillStyle = "red";
            context.arc(x, y, 20, 0, Math.PI * 2, true);
            context.closePath();
            context.fill();
            if (x < (0 + 20) || x > (300 - 20))
                        dx = -dx;
            if (y < (0 + 20) || y > (200 - 20))
                        dy = -dy;
            x += dx;
            y += dy;
  }
  setInterval(draw, 10);
</script>
```

Bouncing Ball API reference

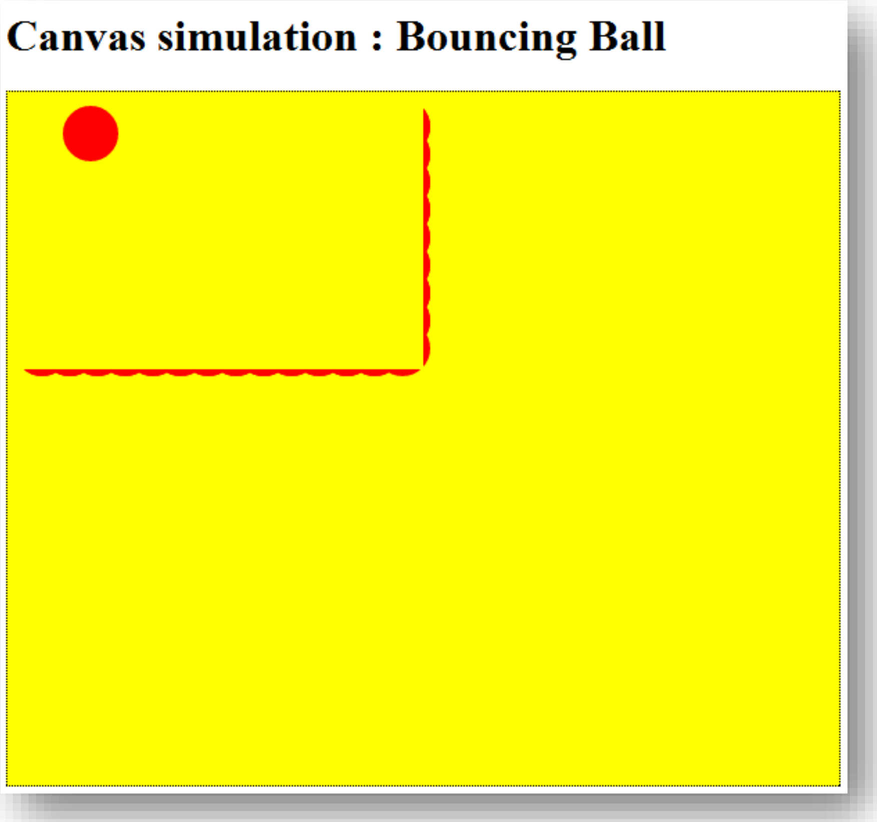


JavaScript syntax:	*context*.arc(*x*,*y*,*r*,*sAngle*,*eAngle*,*counterclockwise*);

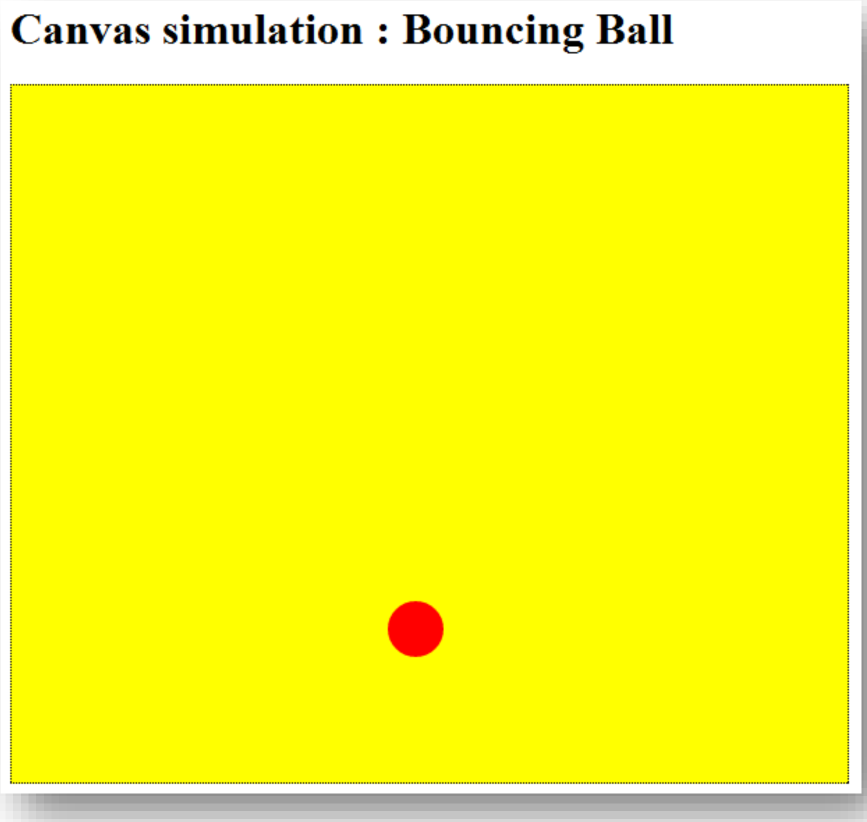
Parameter Values

Parameter	Description
x	The x-coordinate of the center of the circle
y	The y-coordinate of the center of the circle
r	The radius of the circle
sAngle	The starting angle, in radians (0 is at the 3 o'clock position of the arc's circle)
eAngle	The ending angle, in radians
counterclockwise	Optional. Specifies whether the drawing should be counterclockwise or clockwise. False is default, and indicates clockwise, while true indicates counter-clockwise.

Bouncing Ball simulation: 결과



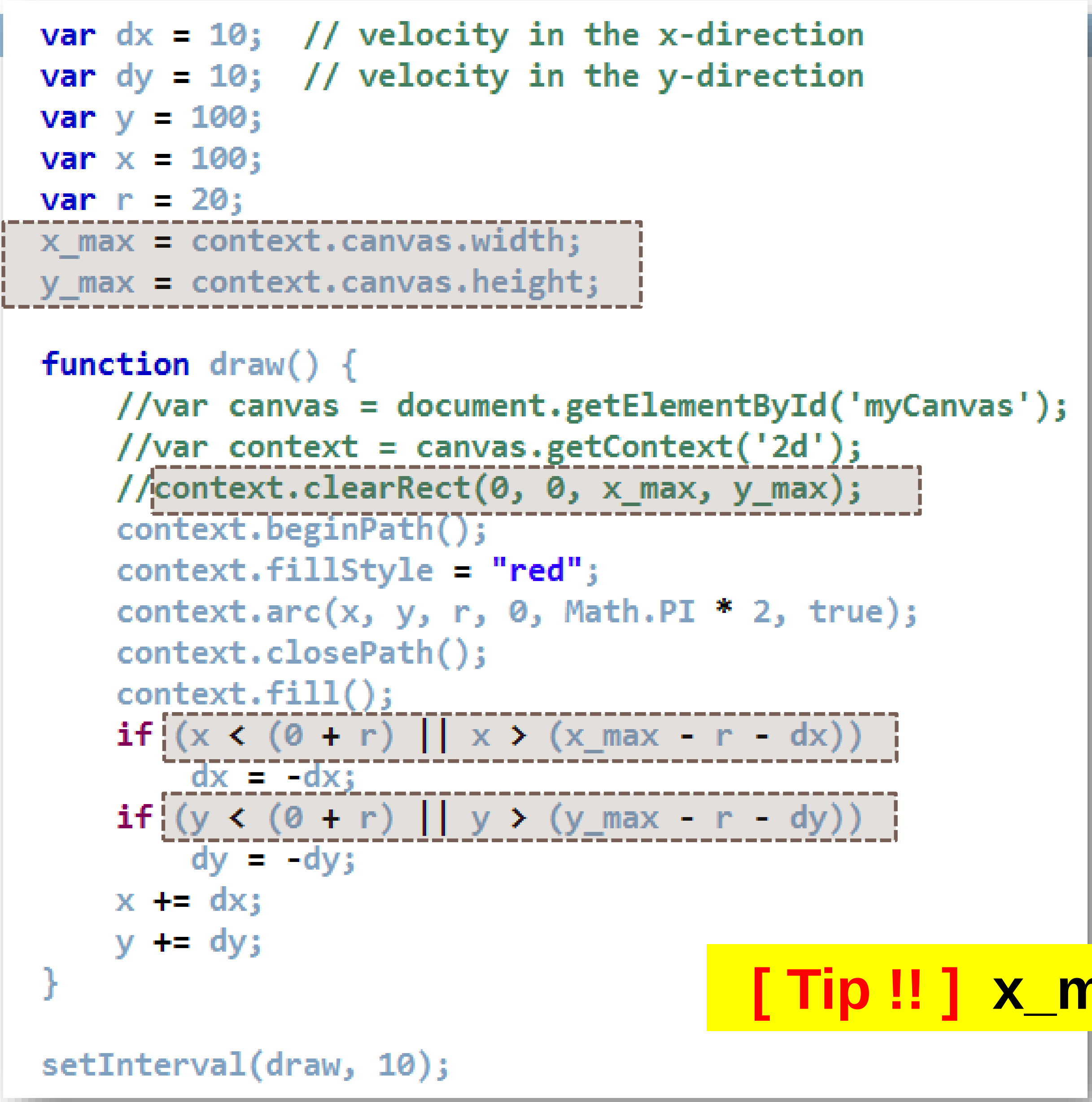
Canvas simulation : Bouncing Ball



Canvas simulation : Bouncing Ball

[DIY] 버그를 찾아서 버그 처리

Bouncing Ball update # 1



```
var dx = 10;  // velocity in the x-direction
var dy = 10;  // velocity in the y-direction
var y = 100;
var x = 100;
var r = 20;
x_max = context.canvas.width;
y_max = context.canvas.height;

function draw() {
    //var canvas = document.getElementById('myCanvas');
    //var context = canvas.getContext('2d');
    //context.clearRect(0, 0, x_max, y_max);
    context.beginPath();
    context.fillStyle = "red";
    context.arc(x, y, r, 0, Math.PI * 2, true);
    context.closePath();
    context.fill();
    if (x < (0 + r) || x > (x_max - r - dx))
        dx = -dx;
    if (y < (0 + r) || y > (y_max - r - dy))
        dy = -dy;
    x += dx;
    y += dy;
}

setInterval(draw, 10);
```

[Tip !!] x_max - r - dx

Bouncing Ball update # 2(모듈화-1)

```
<script>
var canvas = null;
var context = null;

var dx = 10; // velocity in the x-direction
var dy = 10; //Math.random()*20+10;  // velocity in the y-direction
var x = 100;
var y = 100;
var r = 20;
var x_max = 0;
var y_max = 0;

function init() {
  canvas = document.getElementById('myCanvas');
  context = canvas.getContext("2d");
  x_max = context.canvas.width;
  y_max = context.canvas.height;

  blank();

  context.beginPath();
  context.fillStyle = "red";
  context.arc(x, y, r, 0, Math.PI * 2, true);
  context.closePath();
  context.fill();
   // start animation
  setInterval(draw, 10);
}
```

Bouncing Ball update # 2(모듈화-2)

```
function blank() {
  context.fillStyle = "yellow";
  context.fillRect(0,0,context.canvas.width, context.canvas.height);
}

function draw() {

     //blank();

     if (x < (0 + r) || x > (x_max - r - dx))
     dx = -dx;
     if (y < (0 + r) || y > (y_max - r - dy))
     dy = -dy;
     x += dx;
     y += dy;

     context.beginPath();
     context.fillStyle = "red";
     context.arc(x, y, r, 0, Math.PI * 2, true);
     context.closePath();
     context.fill();
}
```

=== Project 1. ===

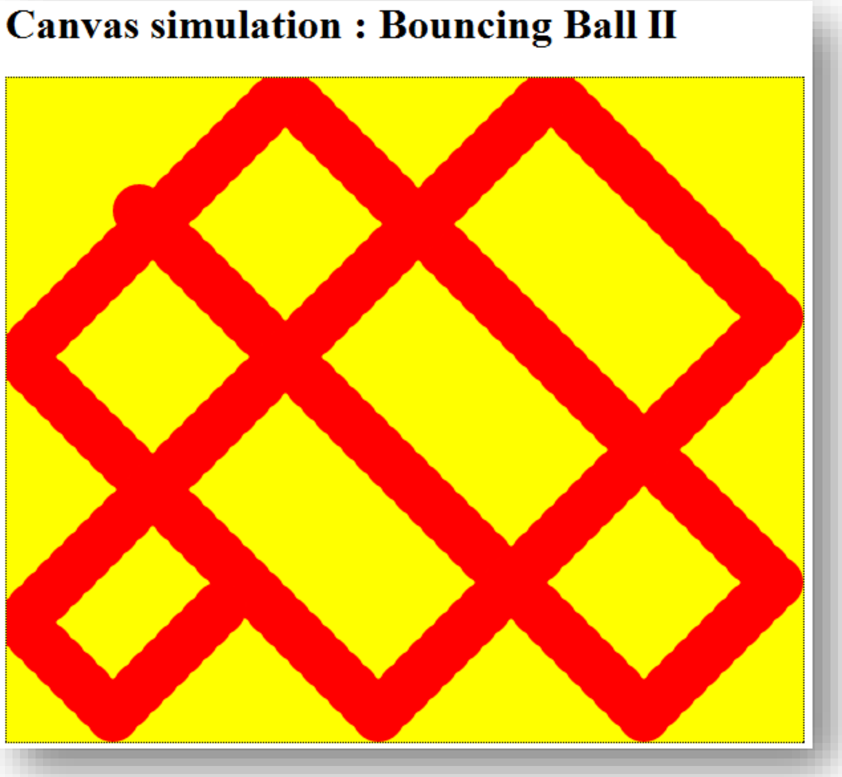
[1] 벽에 충돌 후 가로 및 세로 방향 속도가 무작위로 변하면서 상자 내에서 운동하도록 코드를 수정.
[2] Canvas 밖에 버튼을 만들어서 운동 종료 기능 추가.
[3] 각자 아이디어 추가.

Save as

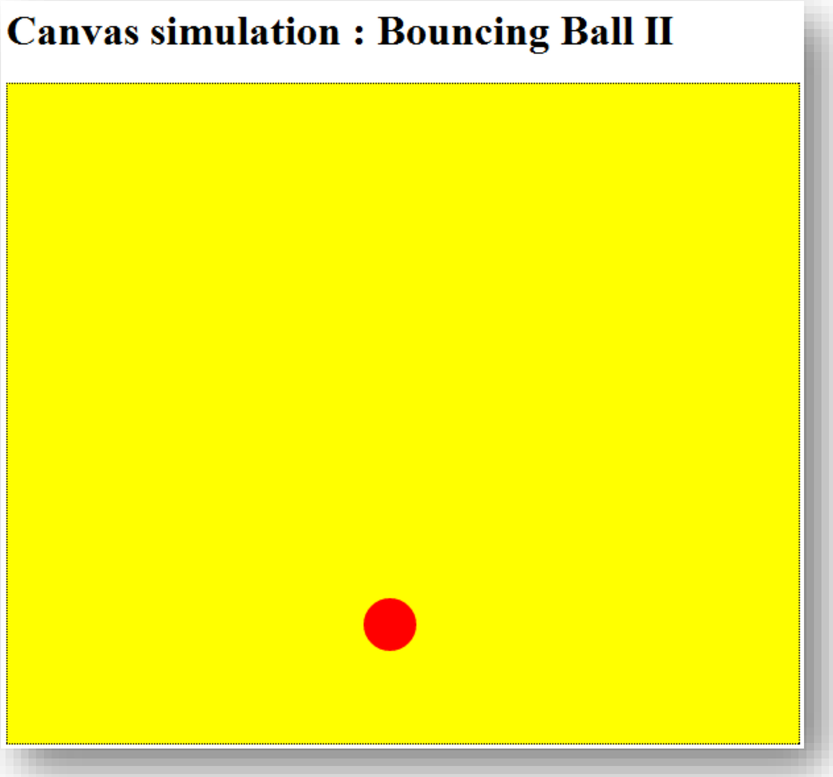
msnn_bouncing.html

Bouncing Ball update # 2(모듈화-3)

```
<body onload="init()">
  <h1>Canvas simulation : Bouncing Ball II</h1>
  <canvas id="myCanvas" width="600" height="500"> </canvas>
```



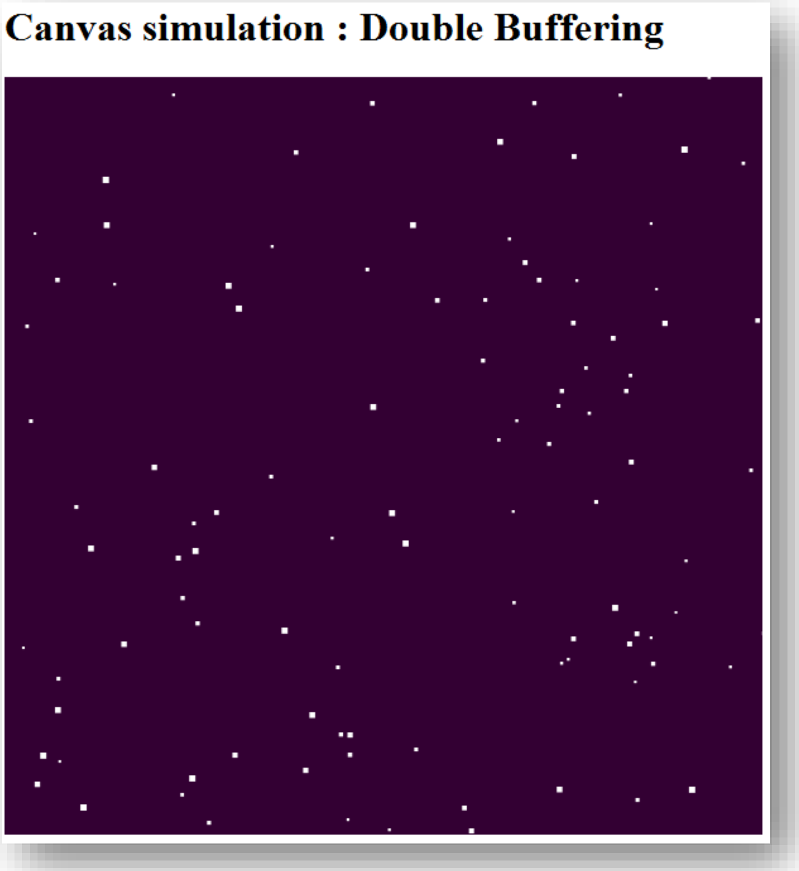
Canvas simulation : Bouncing Ball II



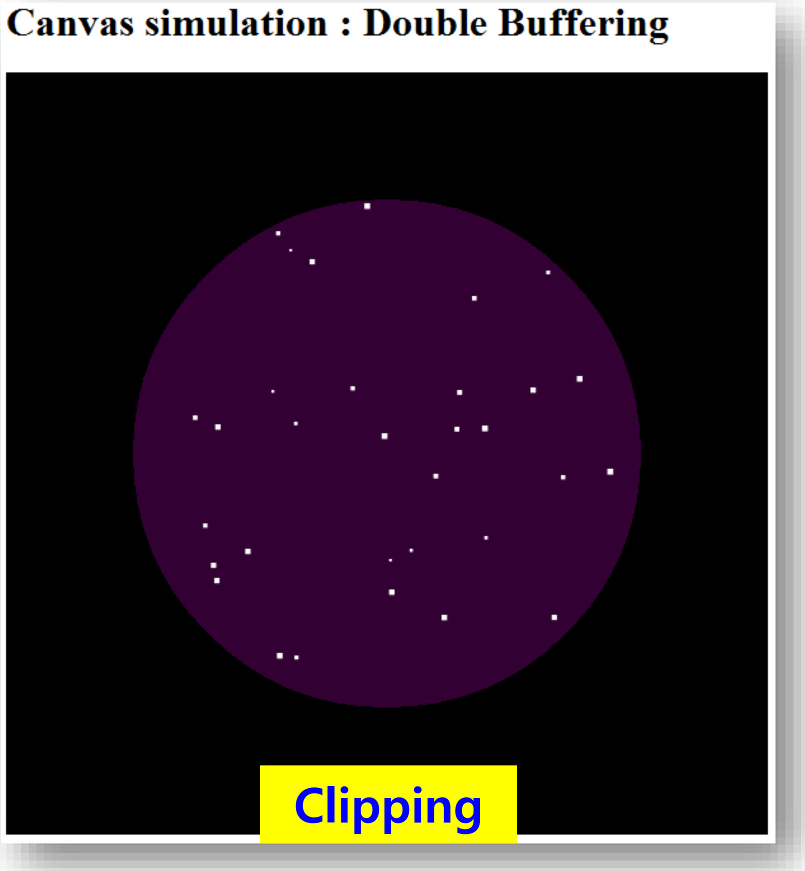
Canvas simulation : Bouncing Ball II

Animation Practice

Animation using Double Buffering (DB)



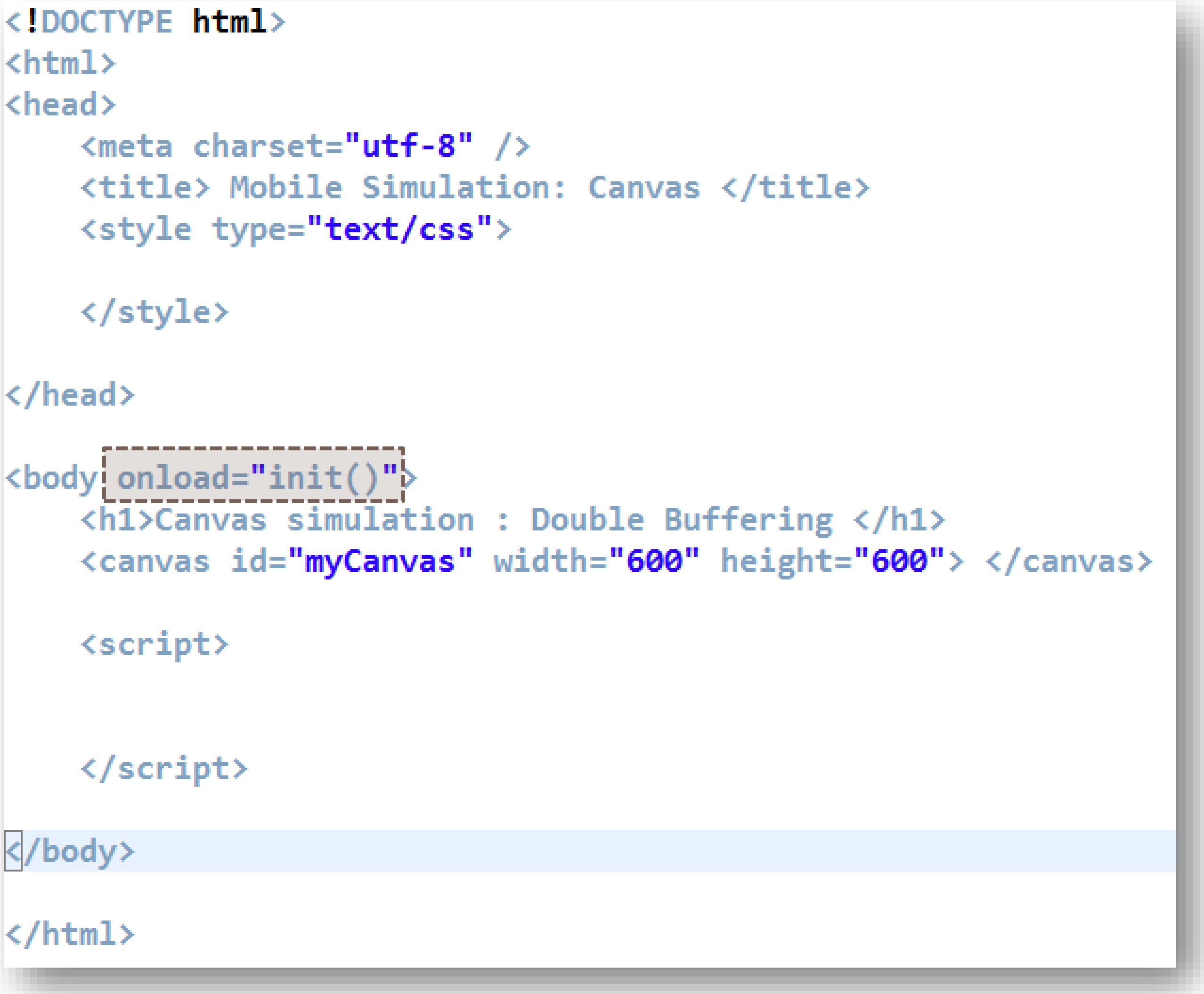
Canvas simulation : Double Buffering



Canvas simulation : Double Buffering

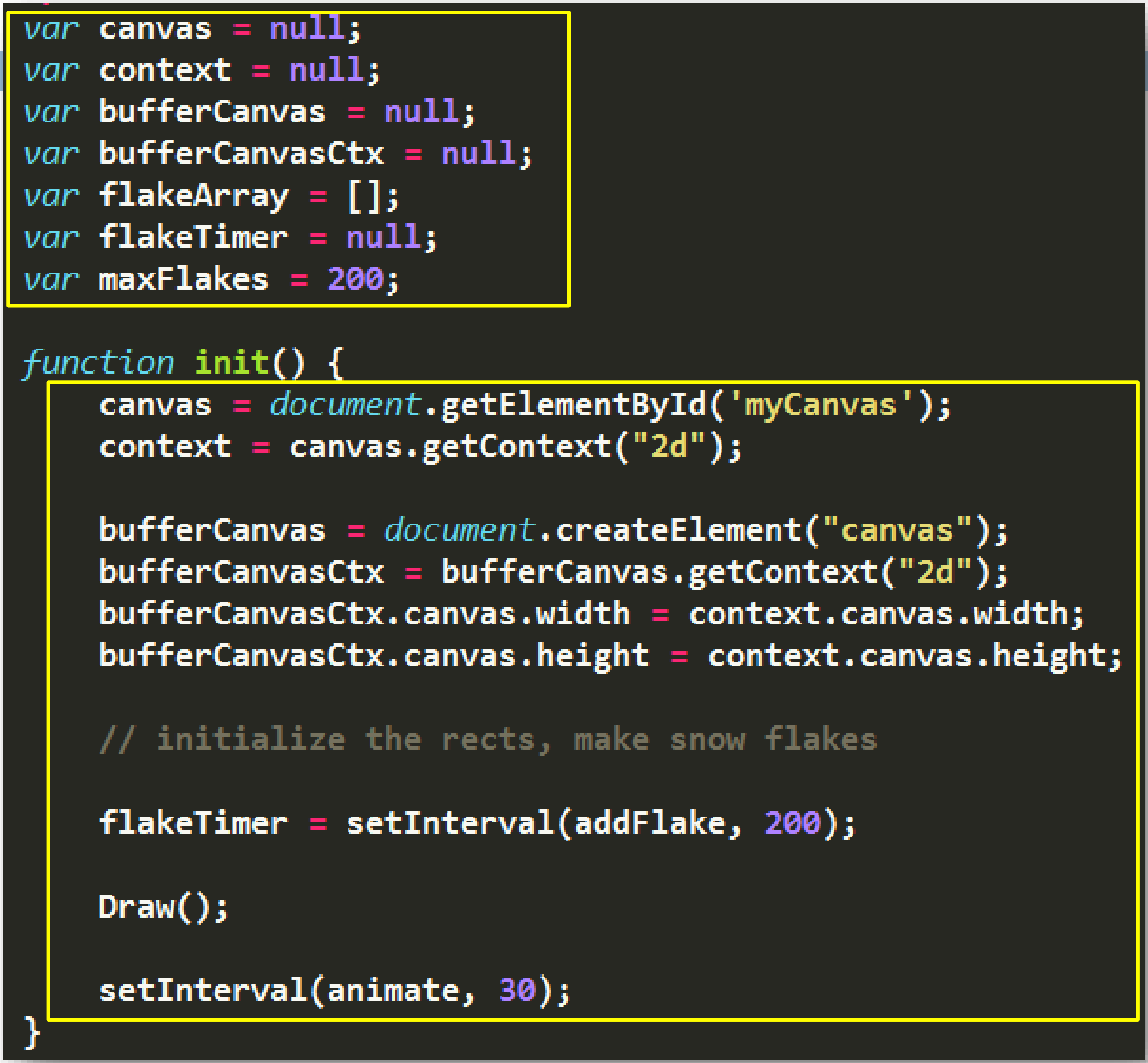
Clipping

Snowy night (DB) #0



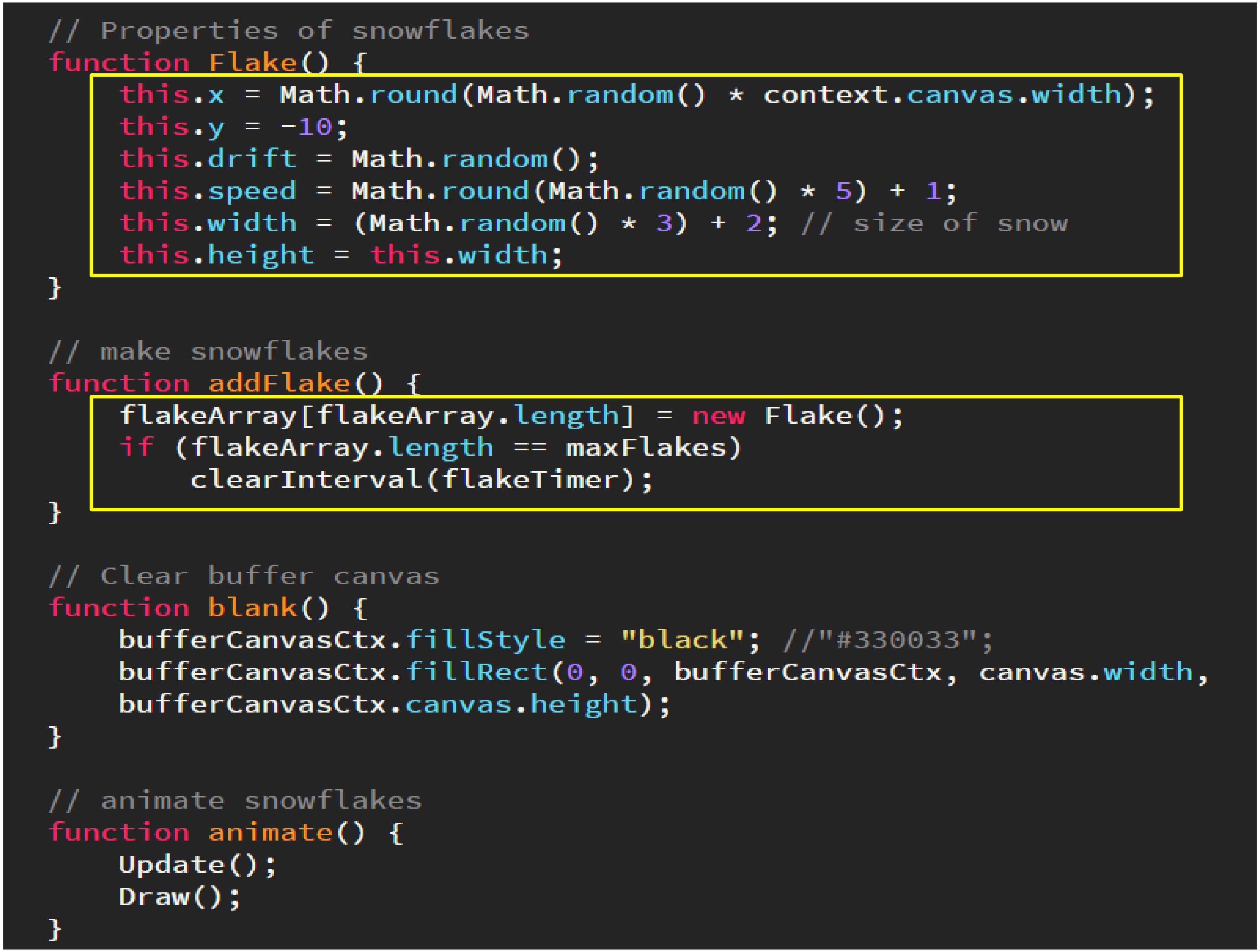
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title> Mobile Simulation: Canvas </title>
    <style type="text/css">

    </style>

</head>

<body onload="init()">
    <h1>Canvas simulation : Double Buffering </h1>
    <canvas id="myCanvas" width="600" height="600"> </canvas>

    <script>


    </script>

</body>

</html>
```

Snowy night (DB) #1



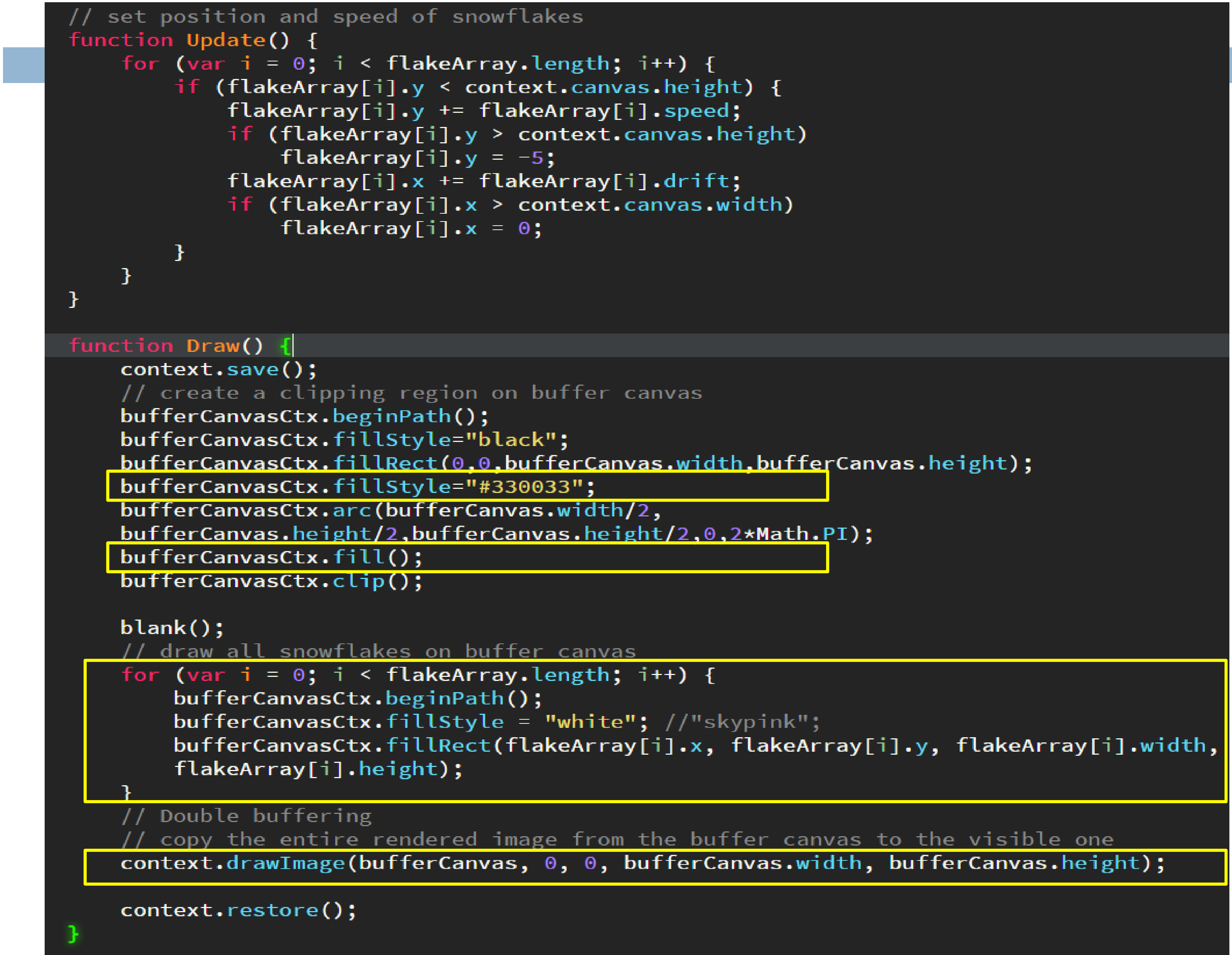
```
var canvas = null;
var context = null;
var bufferCanvas = null;
var bufferCanvasCtx = null;
var flakeArray = [];
var flakeTimer = null;
var maxFlakes = 200;

function init() {
    canvas = document.getElementById('myCanvas');
    context = canvas.getContext("2d");

    bufferCanvas = document.createElement("canvas");
    bufferCanvasCtx = bufferCanvas.getContext("2d");
    bufferCanvasCtx.canvas.width = context.canvas.width;
    bufferCanvasCtx.canvas.height = context.canvas.height;

    // initialize the rects, make snow flakes

    flakeTimer = setInterval(addFlake, 200);

    Draw();

    setInterval(animate, 30);
}
```

Snowy night (DB) #2



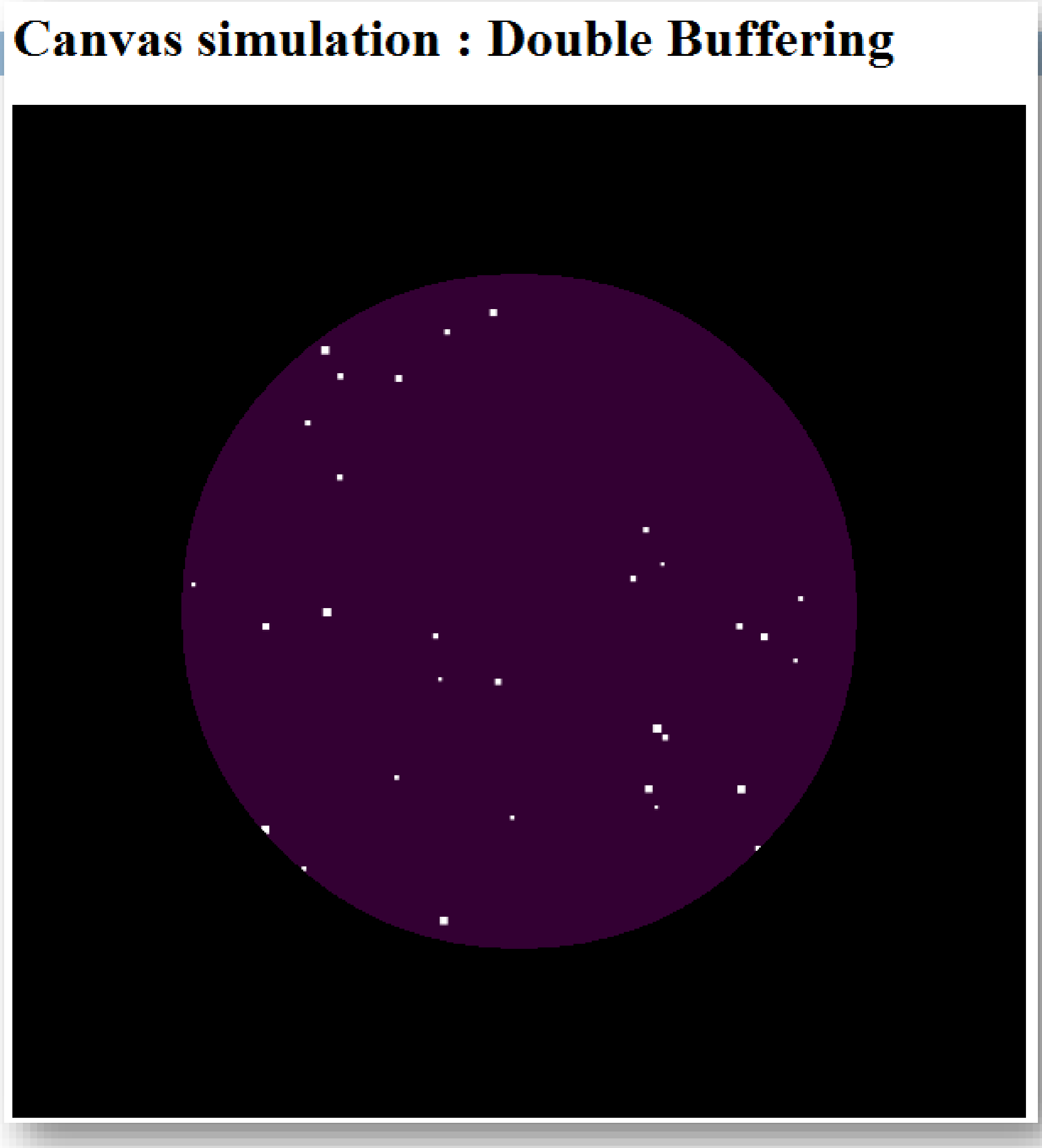
```
// Properties of snowflakes
function Flake() {
    this.x = Math.round(Math.random() * context.canvas.width);
    this.y = -10;
    this.drift = Math.random();
    this.speed = Math.round(Math.random() * 5) + 1;
    this.width = (Math.random() * 3) + 2; // size of snow
    this.height = this.width;
}

// make snowflakes
function addFlake() {
    flakeArray[flakeArray.length] = new Flake();
    if (flakeArray.length == maxFlakes)
        clearInterval(flakeTimer);
}

// Clear buffer canvas
function blank() {
    bufferCanvasCtx.fillStyle = "black"; //"#330033";
    bufferCanvasCtx.fillRect(0, 0, bufferCanvasCtx, canvas.width,
    bufferCanvasCtx.canvas.height);
}

// animate snowflakes
function animate() {
    Update();
    Draw();
}
```

Snowy night (DB) #3



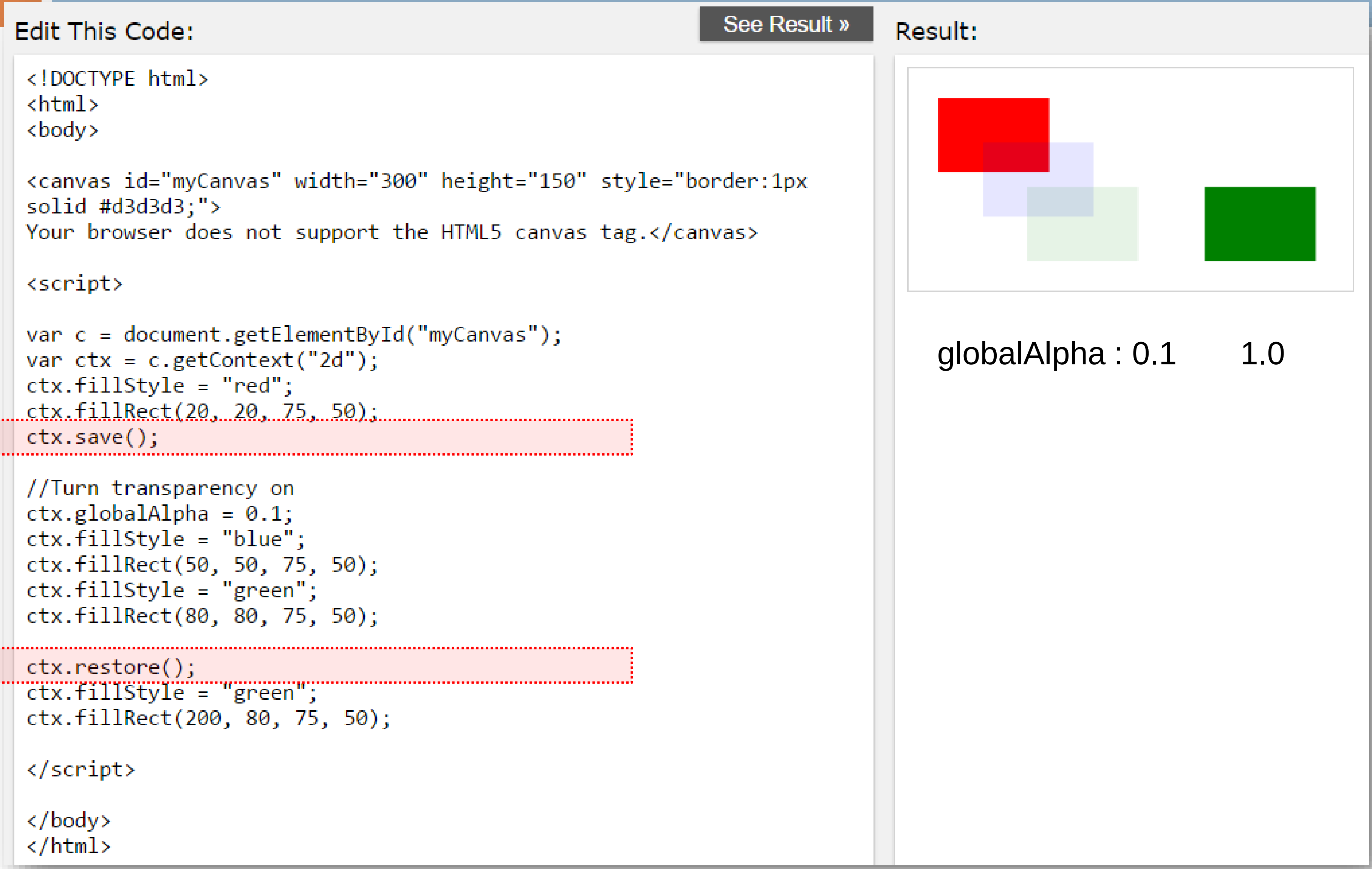
```
// set position and speed of snowflakes
function Update() {
    for (var i = 0; i < flakeArray.length; i++) {
        if (flakeArray[i].y < context.canvas.height) {
            flakeArray[i].y += flakeArray[i].speed;
            if (flakeArray[i].y > context.canvas.height)
                flakeArray[i].y = -5;
            flakeArray[i].x += flakeArray[i].drift;
            if (flakeArray[i].x > context.canvas.width)
                flakeArray[i].x = 0;
        }
    }
}

function Draw() {
    context.save();
    // create a clipping region on buffer canvas
    bufferCanvasCtx.beginPath();
    bufferCanvasCtx.fillStyle="black";
    bufferCanvasCtx.fillRect(0,0,bufferCanvas.width,bufferCanvas.height);
    bufferCanvasCtx.fillStyle="#330033";
    bufferCanvasCtx.arc(bufferCanvas.width/2,
    bufferCanvas.height/2,bufferCanvas.height/2,0,2*Math.PI);
    bufferCanvasCtx.fill();
    bufferCanvasCtx.clip();

    blank();
    // draw all snowflakes on buffer canvas
    for (var i = 0; i < flakeArray.length; i++) {
        bufferCanvasCtx.beginPath();
        bufferCanvasCtx.fillStyle = "white"; //"skypink";
        bufferCanvasCtx.fillRect(flakeArray[i].x, flakeArray[i].y, flakeArray[i].width,
        flakeArray[i].height);
    }
    // Double buffering
    // copy the entire rendered image from the buffer canvas to the visible one
    context.drawImage(bufferCanvas, 0, 0, bufferCanvas.width, bufferCanvas.height);

    context.restore();
}
```

Snowy night : final result



Canvas simulation : Double Buffering

[복습] Context save() & restore().



```
<!DOCTYPE html>
<html>
<body>

<canvas id="myCanvas" width="300" height="150" style="border:1px
solid #d3d3d3;">
Your browser does not support the HTML5 canvas tag.</canvas>

<script>

var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.fillStyle = "red";
ctx.fillRect(20, 20, 75, 50);
ctx.save();

//Turn transparency on
ctx.globalAlpha = 0.1;
ctx.fillStyle = "blue";
ctx.fillRect(50, 50, 75, 50);
ctx.fillStyle = "green";
ctx.fillRect(80, 80, 75, 50);

ctx.restore();
ctx.fillStyle = "green";
ctx.fillRect(200, 80, 75, 50);

</script>

</body>
</html>
```

globalAlpha : 0.1 1.0

w3schools.com

과제04. msnn_rpt04.zip

[실습과제04] Christmas card

[1] Bouncing ball 완성. – msnn_bouncing.html

[2] 캔버스와 자바스크립트만을 사용해서 크리스마스 카드 만들기

- 사각형 눈송이의 모양을 원형으로 변경
- 크리스마스 관련 그림을 활용 (배경 또는 애니메이션 추가)
- msnn_rpt04,html

**** html 파일 및 관련 파일(image 등..)을
MSnn_Rpt04.zip 으로 압축해서 제출하시오.

파일명: msnn_bouncing.html, msnn_rpt04.html

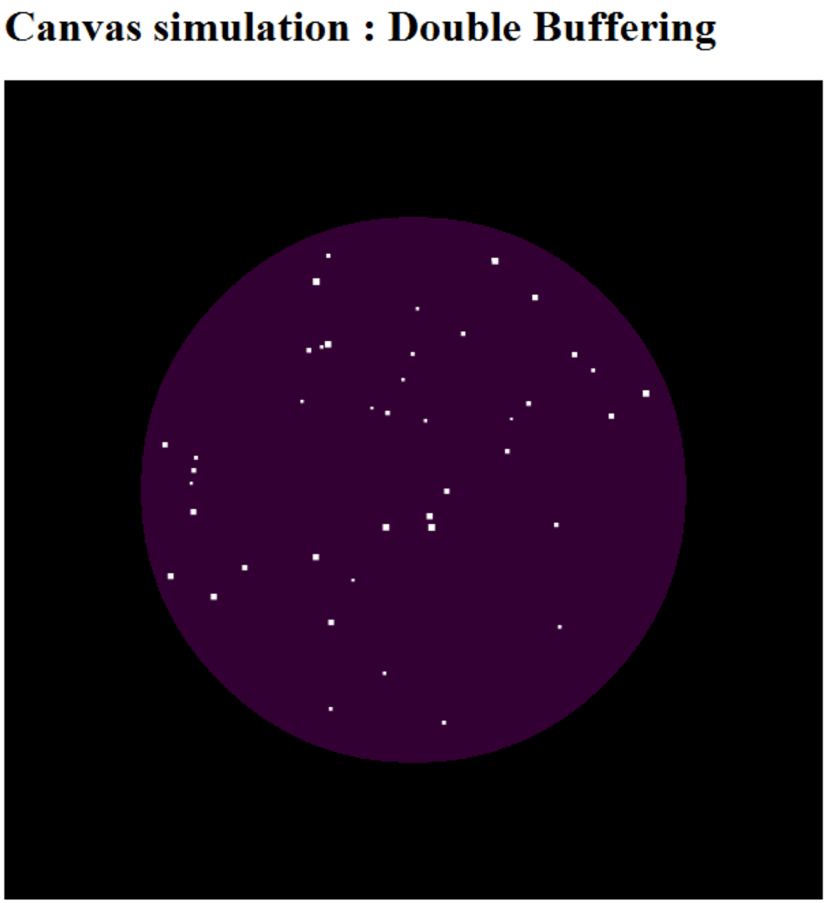
가점: Javascript 프로그래밍 응용 능력.

[제출파일] msnn_rpt04.zip

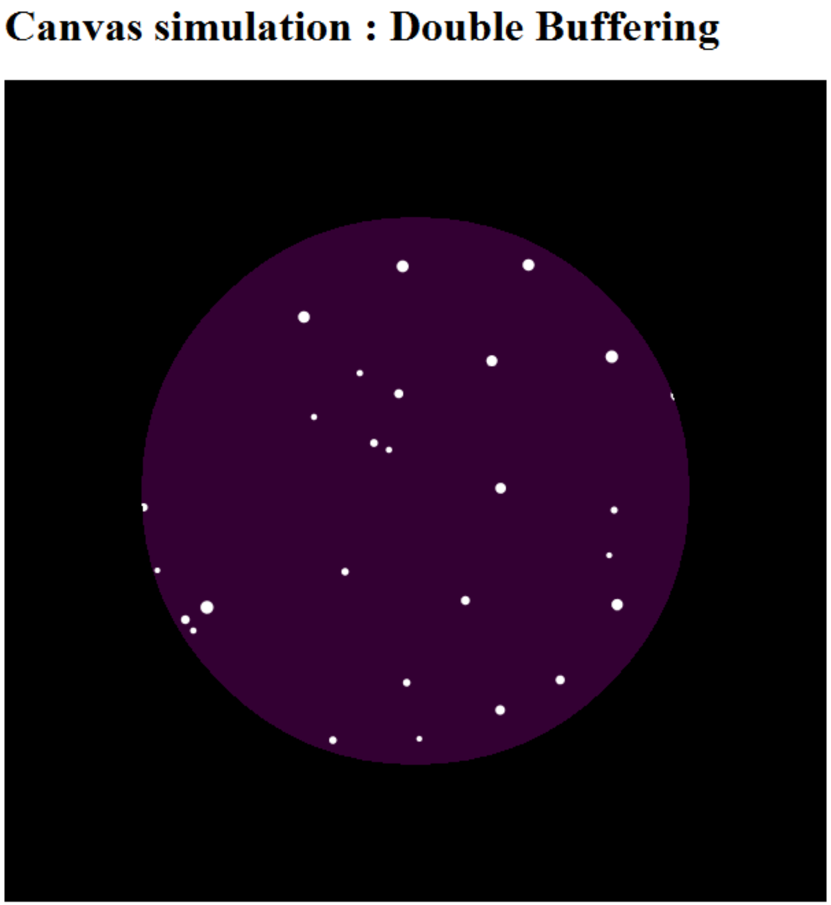
두 개의 html 파일과 사용된 그림을 이메일로 제출

Email : chaos21c@gmail.com

과제04. result



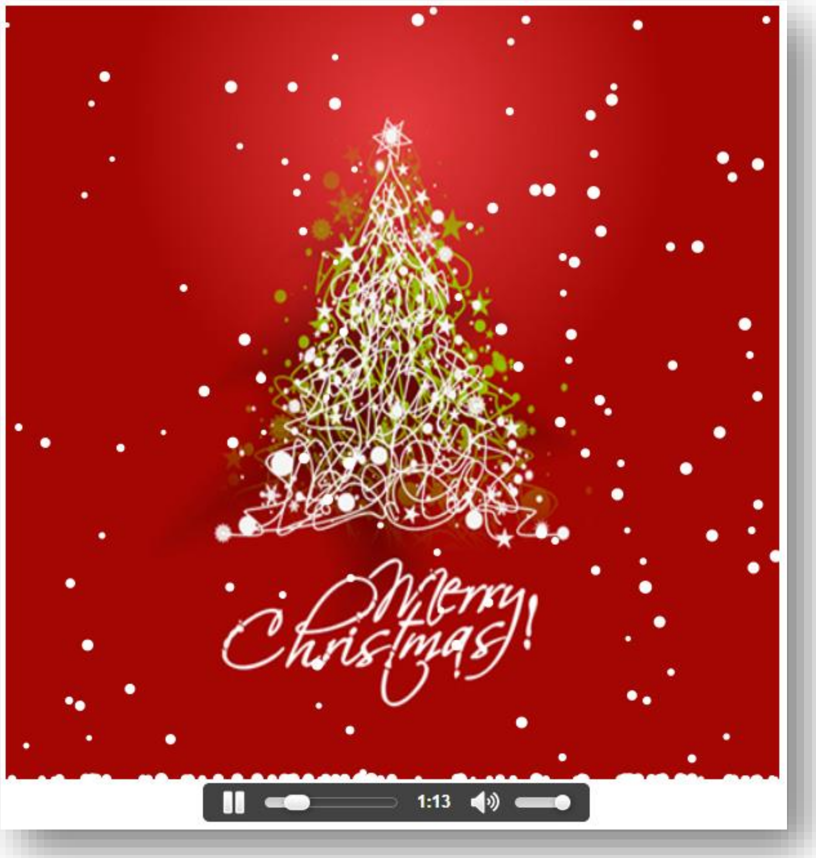
Canvas simulation : Double Buffering

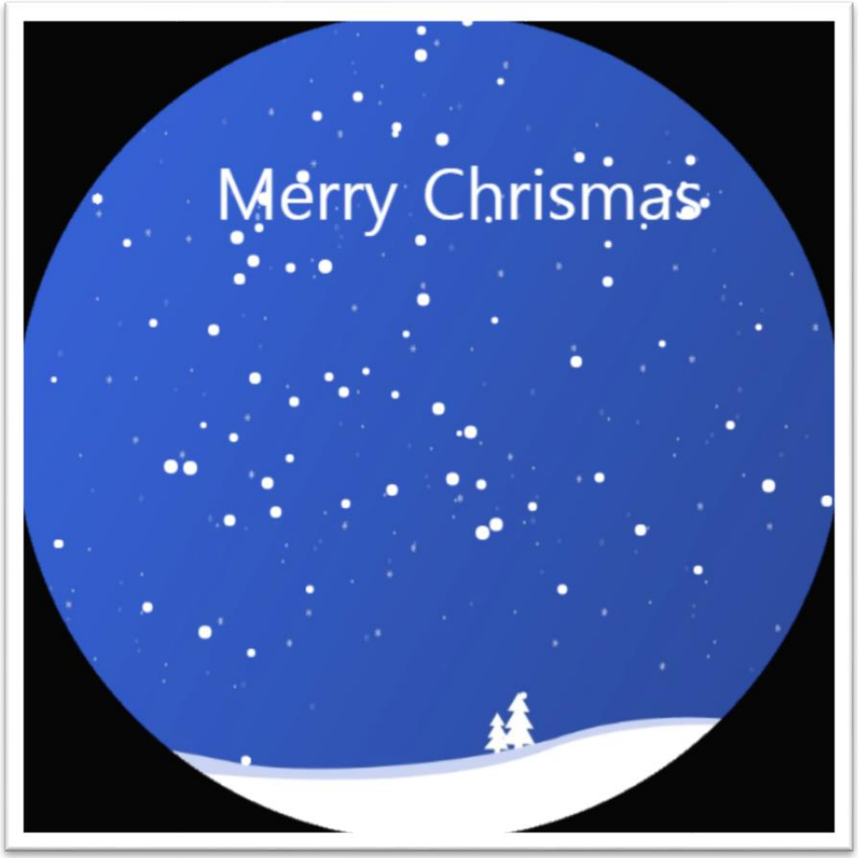


Canvas simulation : Double Buffering

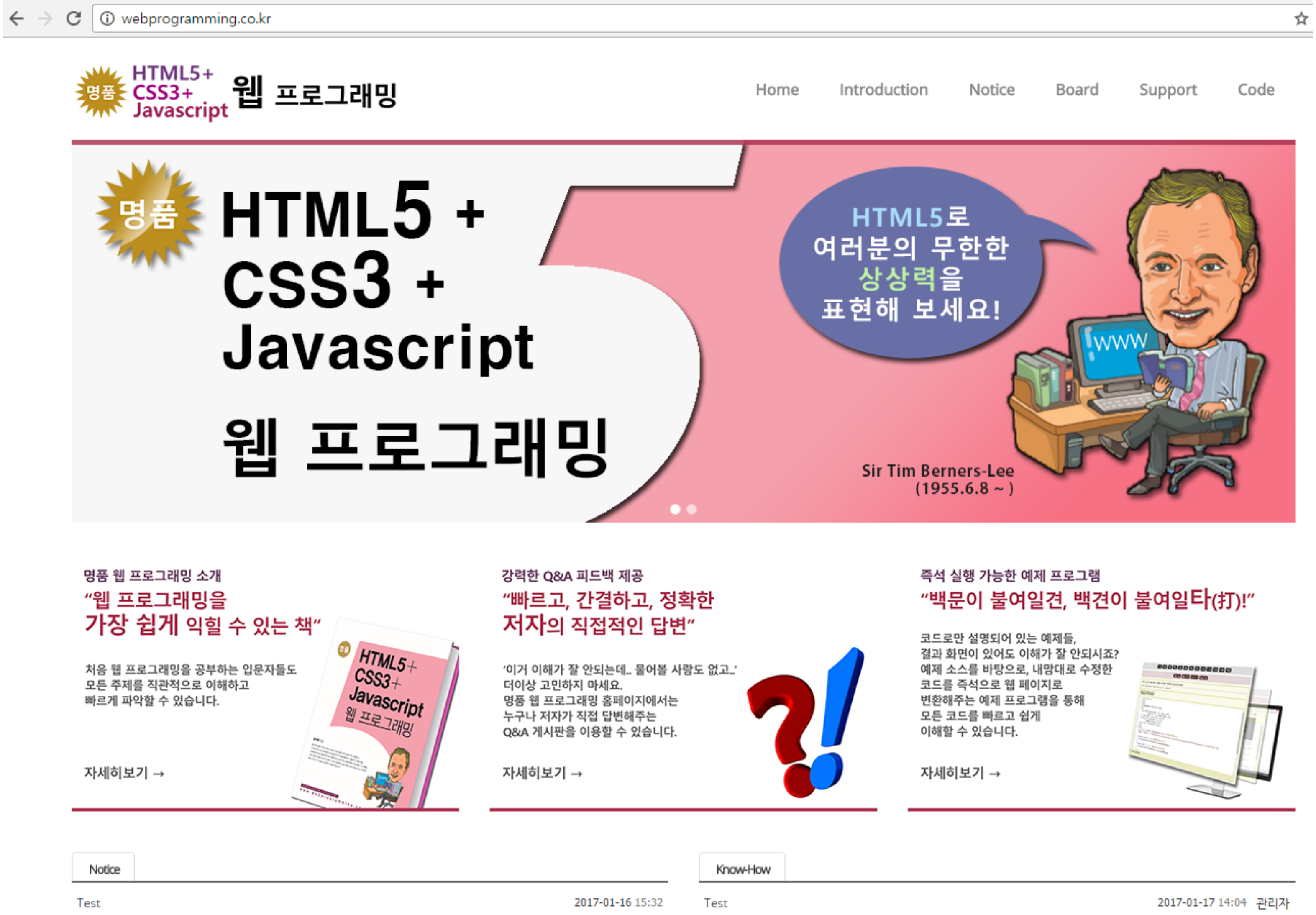
Round snowflakes and clipping effect

과제04. possible results





교재 WEB 강의 소개



webprogramming.co.kr

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.

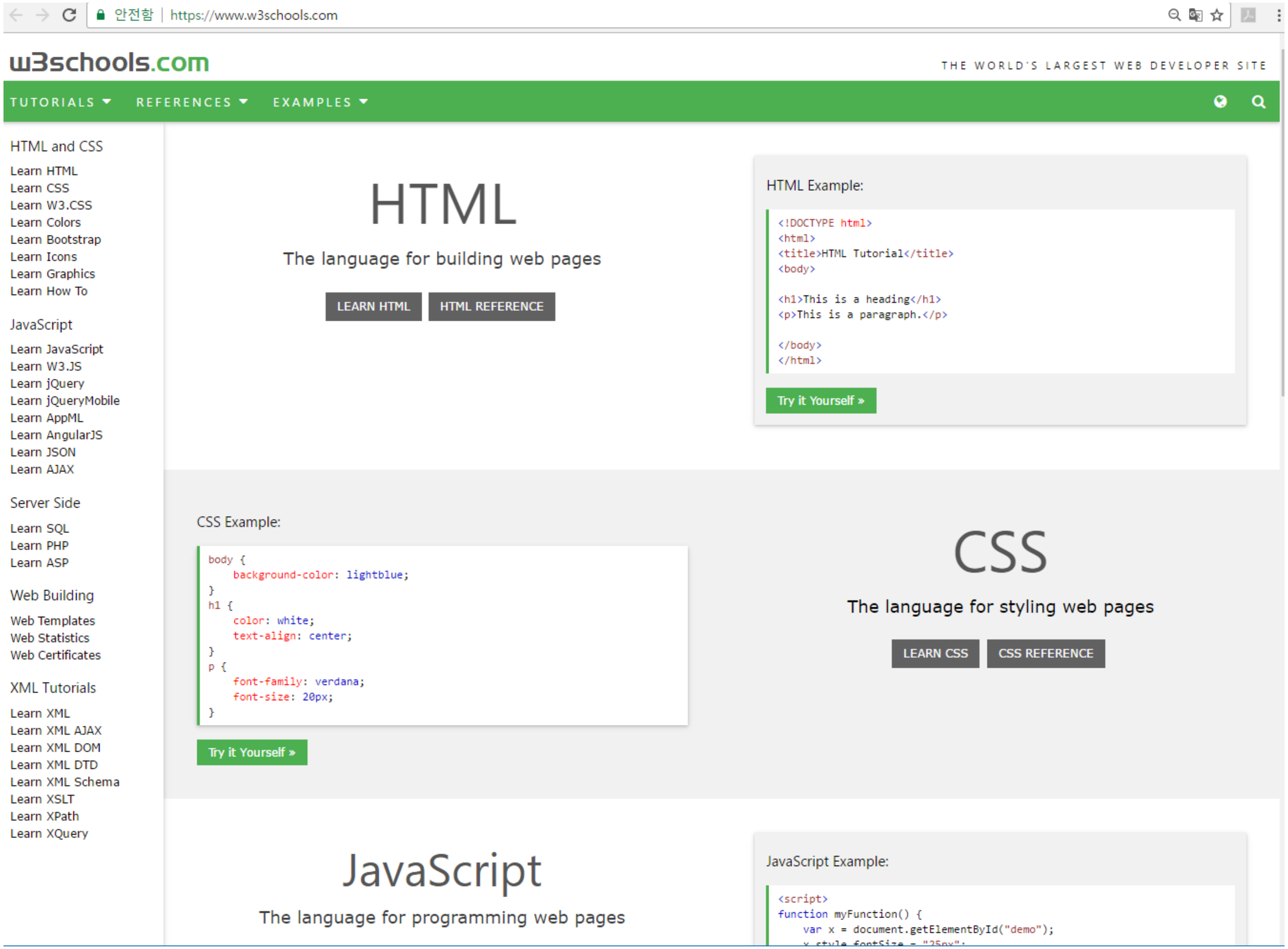
자세히보기 →

Notice	
Test	2017-01-16 15:32

Know-How		
Test	2017-01-17 14:04	관리자

http://webprogramming.co.kr

관련 WEB 강의 소개 – w3schools.com



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

HTML Example:

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

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

</body>
</html>
```

CSS Example:

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

CSS

The language for styling web pages

JavaScript

The language for programming web pages

JavaScript Example:

```
<script>
function myFunction() {
    var x = document.getElementById("demo");
    x.style.fontSize = "25px";
```

http://www.w3schools.com