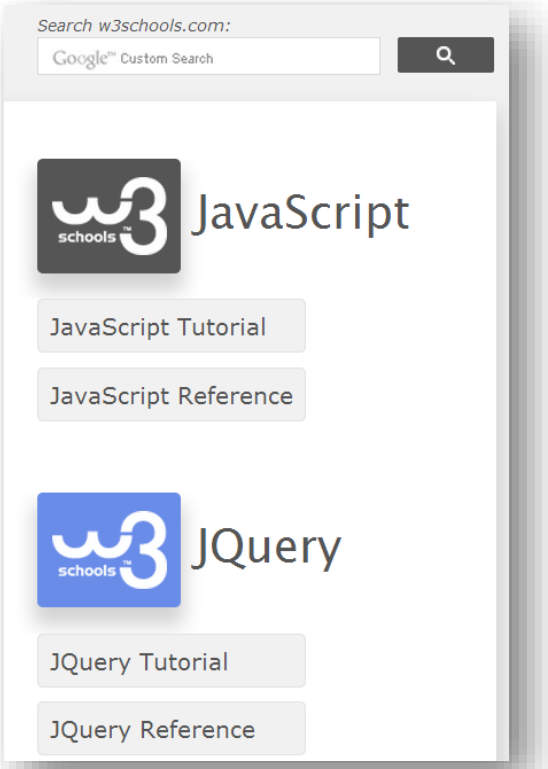
Mobile Simulation

2017-2

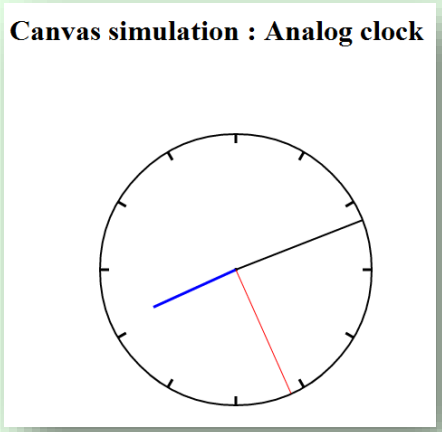


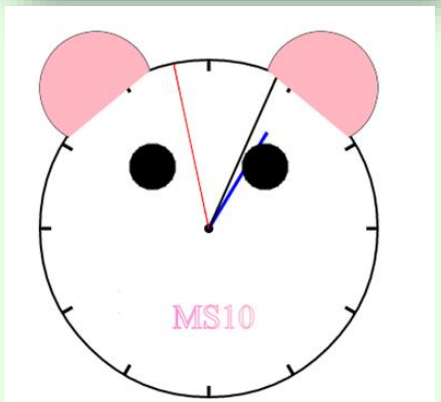
Weekly plan (HTML5, 1st semester 2017)

- **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)

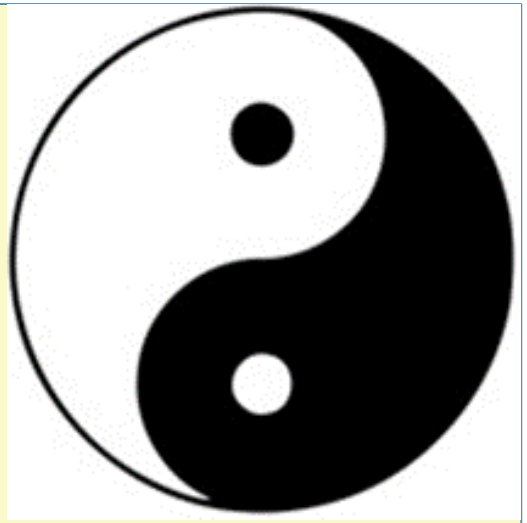
- **wk01 : Introduction to curriculum & current state**
- **wk02 : Browser Object Model (BOM), installing Brackets editor**
- **wk03 : Canvas graphics I. Basic**
- **wk04 : Canvas graphics II. Image & Transformation**
- **wk05 :**
- **wk06 :**
- **wk07 :**
- **wk08 : Mid-term Exam.**
- **wk09 :**
- **wk10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





과제02. msnn_rpt02.zip

[실습과제02] Canvas에 태극 문양 그리기



[1] 태극 문양에 색과 그라디언트 효과 추가.
[2] 캔버스의 하단에 본인 ID를 텍스트로 그려서 추가.
[3] 파일명: MSnn_rpt02.html

파일명: msnn_rpt02.html

가점: Javascript 프로그래밍 응용 능력.

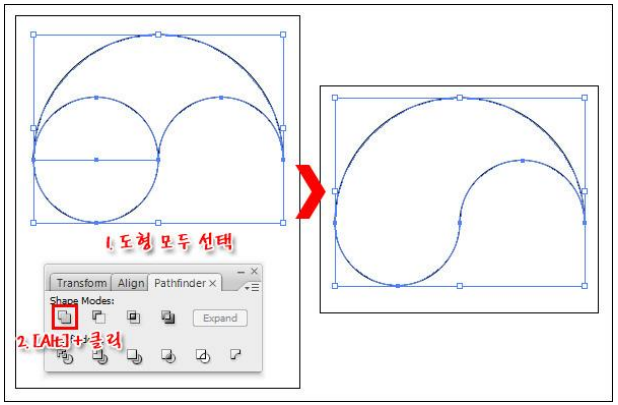
[제출파일] msnn_rpt02.html

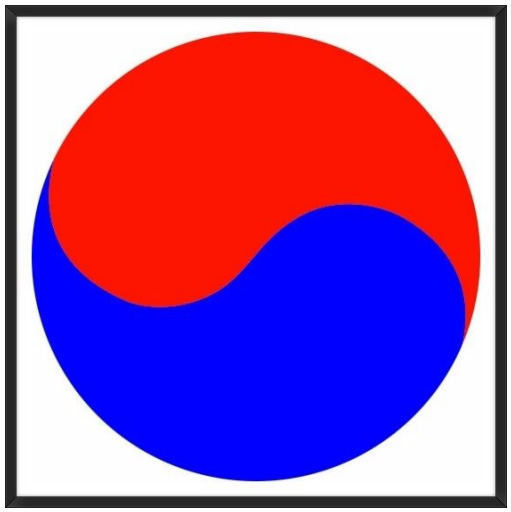
msnn_rpt02.html 을 이메일로 제출

Email : chaos21c@gmail.com

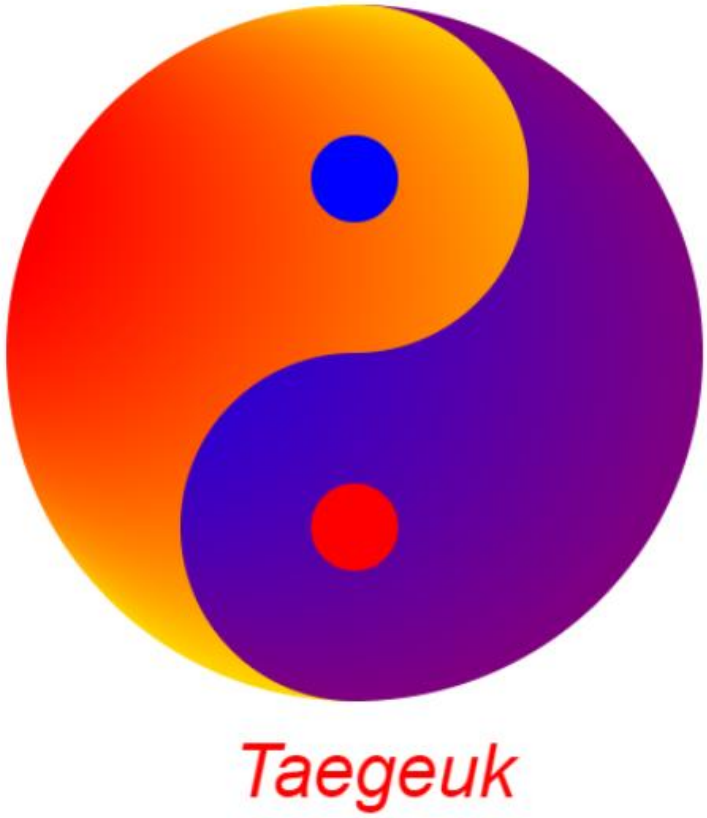
chaos21c@gmail.com

과제02. hint


1. 도형 모두 선택
Transform
Align
Pathfinder
Shape Modes:
Expand
2. [Alt]+클릭



Best Reports of wk03



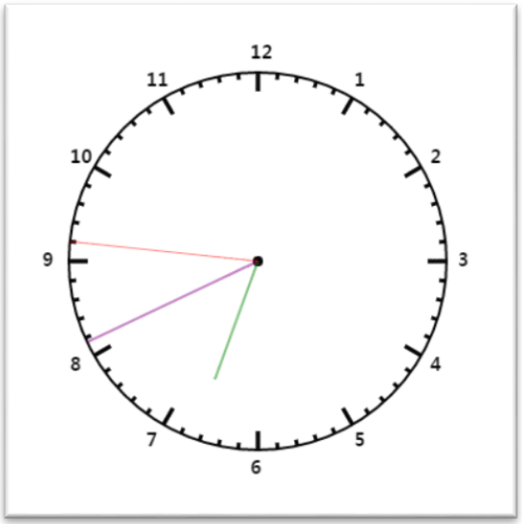

```
var gradient_1 = context.createRadialGradient(250, 250, 200, 0, 150, 80);
gradient_1.addColorStop(0, "gold");
gradient_1.addColorStop(1, "red");
var gradient_2 = context.createRadialGradient(250, 250, 200, 0, 150, 80);
gradient_2.addColorStop(0, "purple");
gradient_2.addColorStop(1, "blue");


context.fillStyle = gradient_1;
context.beginPath();
context.arc(250, 250, 200, Math.PI / 2, 3 / 2 * Math.PI, false);
context.fill();

context.fillStyle = gradient_2;
context.beginPath();
context.arc(250, 250, 200, Math.PI / 2, 3 / 2 * Math.PI, true);
context.fill();

context.fillStyle = gradient_1;
context.beginPath();
context.arc(250, 150, 100, 3 / 2 * Math.PI, Math.PI / 2, false);
context.fill();
context.fillStyle = gradient_2;
context.beginPath();
context.arc(250, 350, 100, 3 / 2 * Math.PI, Math.PI / 2, true);
context.fill();

context.fillStyle = "blue";
context.beginPath();
context.arc(250, 150, 25, 0, 2 * Math.PI, true);
context.fill();

context.fillStyle = "red";
context.beginPath();
context.arc(250, 350, 25, 0, 2 * Math.PI, true);
context.fill();

context.font = 'italic 32pt Arial'
context.fillText('Taegeuk', 180, 505);
```

11

HTML5 캔버스 그래픽

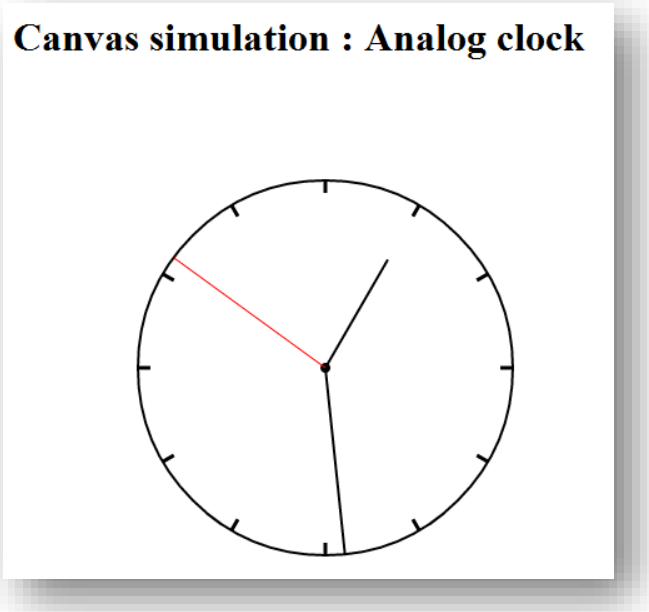
Canvas II. Image & Transformation

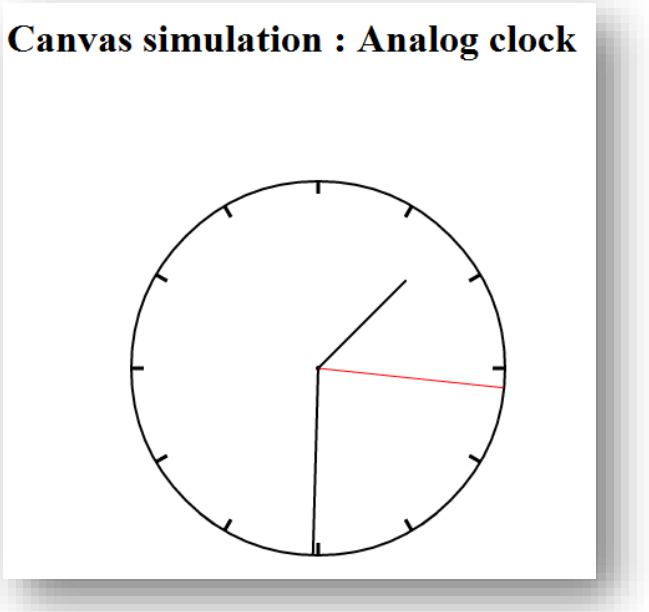
The <canvas> element is used to draw graphics, on the fly, on a web page.



Target:

Images & Transform on Canvas in HTML5





Reference

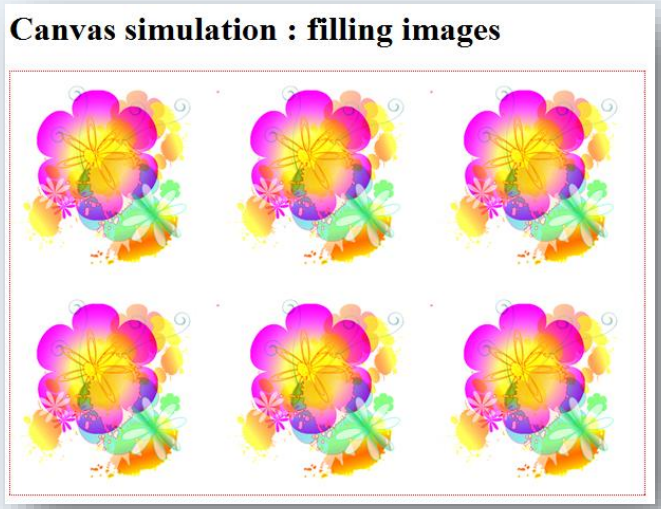
HTML DOM Canvas Object
http://www.w3schools.com/jsref/dom_obj_canvas.asp

HTML5 Canvas Image Tutorial
http://www.html5canvastutorials.com/tutorials/html5-canvas-images/

HTML Canvas Reference
http://www.w3schools.com/tags/ref_canvas.asp

wk04.1 패턴 채우기 (image file)

```
<!DOCTYPE HTML>
<html>
<head>
..
</head>
<body>
   <canvas id="myCanvas" width="600" height="400"
           style="border: 1px dotted red"></canvas>
  <script>
     var canvas = document.getElementById("myCanvas");
     var context = canvas.getContext("2d");

     var image = new Image();
     image.src = "media/pattern.png";
     image.onload = function () {
        var pattern = context.createPattern(image, "repeat");

        context.rect(0, 0, canvas.width, canvas.height);
        context.fillStyle = pattern;
        context.fill();
     };

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



패턴 채우기(image object)

Edit This Code: See Result »

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

<p>Image to use:</p>
<img src="img_lamp.jpg" id="lamp" width="32" height="32">
<p>Canvas:</p>

<button onclick="draw('repeat')">Repeat</button>
<button onclick="draw('repeat-x')">Repeat-x</button>
<button onclick="draw('repeat-y')">Repeat-y</button>
<button onclick="draw('no-repeat')">No-repeat</button>

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

<script>

function draw(direction) {
    var c = document.getElementById("myCanvas");
    var ctx = c.getContext("2d");
    ctx.clearRect(0, 0, c.width, c.height);
    var img = document.getElementById("lamp")
    var pat = ctx.createPattern(img, direction);
    ctx.rect(0, 0, 150, 100);
    ctx.fillStyle = pat;
    ctx.fill();
}

</script>

</body>
</html>
```

Result:

Image to use:

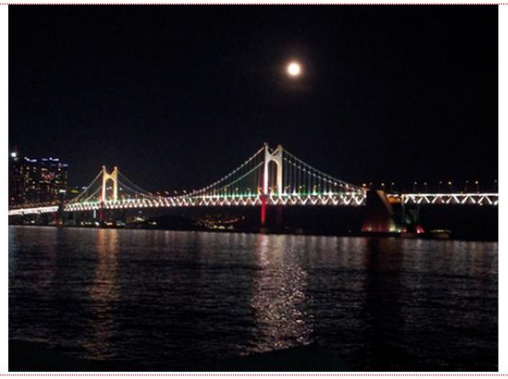
Canvas:

Repeat | Repeat-x | Repeat-y | No-repeat

wk04.2 이미지 그리기 (drawImage()

```
<body>
    <canvas id="myCanvas" width="600" height="400"></canvas>
    <script>
        var canvas = document.getElementById("myCanvas");
        var context = canvas.getContext("2d");

        var image = new Image();
        image.src = "media/html5_logo.png";

        image.onload = function () {
            context.drawImage(image, 0, 0);
        };

    </script>
</body>
```

Canvas simulation : draw image



Canvas simulation : draw image

이미지 그리기 API

Image Drawing

Method	Description
drawImage()	Draws an image, canvas, or video onto the canvas

Pixel Manipulation

Property	Description
width	Returns the width of an ImageData object
height	Returns the height of an ImageData object
data	Returns an object that contains image data of a specified ImageData object

Method	Description
createImageData()	Creates a new, blank ImageData object
getImageData()	Returns an ImageData object that copies the pixel data for the specified rectangle on a canvas
putImageData()	Puts the image data (from a specified ImageData object) back onto the canvas

wk04.3 이미지 그리기

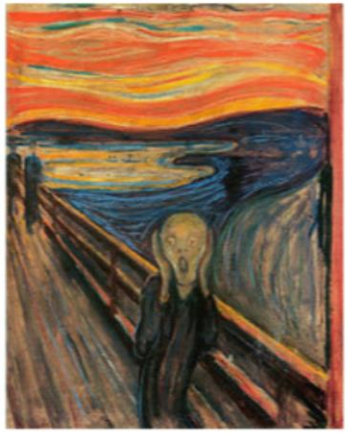
w3schools.com

Run »

Result Size: 524 x 715

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

<p>Image to use:</p>
<img id="scream" src="img_the_scream.jpg" alt="The Scream"
width="220" height="277">

<p>Canvas:</p>
<canvas id="myCanvas" width="250" height="300"
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");
var img = document.getElementById("scream");
ctx.drawImage(img, 10, 10);

</script>

</body>
</html>
```

Image to use:



Canvas:

Pixel 조작 API

Pixel Manipulation

Property	Description
width	Returns the width of an ImageData object
height	Returns the height of an ImageData object
data	Returns an object that contains image data of a specified ImageData object

Method	Description
createImageData()	Creates a new, blank ImageData object
getImageData()	Returns an ImageData object that copies the pixel data for the specified rectangle on a canvas
putImageData()	Puts the image data (from a specified ImageData object) back onto the canvas

Pixel 조작 예제 1.

w3schools.com


```
<!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");
var imgData = ctx.createImageData(100, 100);
alert("Height of imgData is: " + imgData.height);

var i;
for (i = 0; i < imgData.data.length; i += 4) {
  imgData.data[i+0] = 0;
  imgData.data[i+1] = 255;
  imgData.data[i+2] = 255;
  imgData.data[i+3] = 255;
  }

ctx.putImageData(imgData, 10, 10);

</script>

</body>
</html>
```

Pixel 조작 예제 2. (**mid-exam**)

w3schools.com

Edit This Code: See Result »

```
<!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(10, 10, 50, 50);

function copy() {
    var imgData = ctx.getImageData(10, 10, 50, 50);
    ctx.putImageData(imgData, 10, 70);
}

function copyBlue() {
    var imgData = ctx.getImageData(10, 10, 50, 50);
    for(var i=0; i<imgData.data.length; i+=4){
       imgData.data[i+0] = 0;
       imgData.data[i+1] = 0;
       imgData.data[i+2] = 255;
       imgData.data[i+3] = 255;
    }
    ctx.putImageData(imgData, 10, 70);
}
</script>

<br>
<button onclick="copy()">Copy</button>
<button onclick="copyBlue()">Copy in blue</button>
```

Result:

Image Compositing API

Compositing

Property	Description
globalAlpha	Sets or returns the current alpha or transparency value of the drawing
globalCompositeOperation	Sets or returns how a new image are drawn onto an existing image

Other

Method	Description
save()	Saves the state of the current context
restore()	Returns previously saved path state and attributes
createEvent()	
getContext()	
toDataURL()	

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

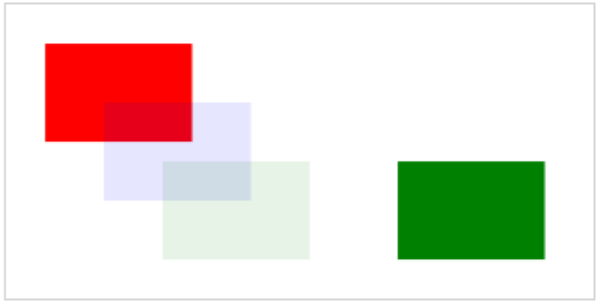
Context save() & restore().

w3schools.com

Edit This Code: See Result »

```
<!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>
```

Result:



globalAlpha : 0.1 1.0

ctx.save()는 alpha=1.0을 저장

그후에 alpha=0.1로 변경

ctx.restore()는 alpha=1.0을 복원

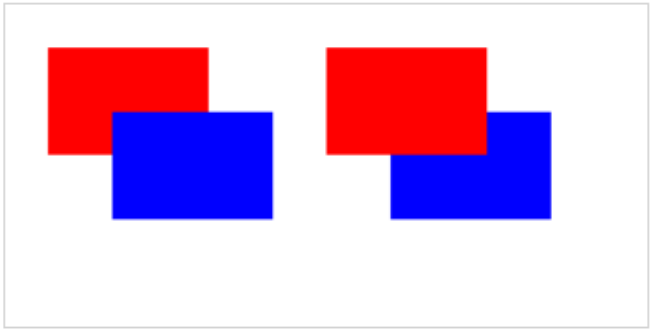
globalCompositeOperation.

w3schools.com

Code:

```
<!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.fillStyle="blue";
ctx.globalCompositeOperation="source-over";
ctx.fillRect(50,50,75,50);
ctx.fillStyle="red";
ctx.fillRect(150,20,75,50);
ctx.fillStyle="blue";
ctx.globalCompositeOperation="destination-over";
ctx.fillRect(180,50,75,50);

</script>

</body>
</html>
```

Result:

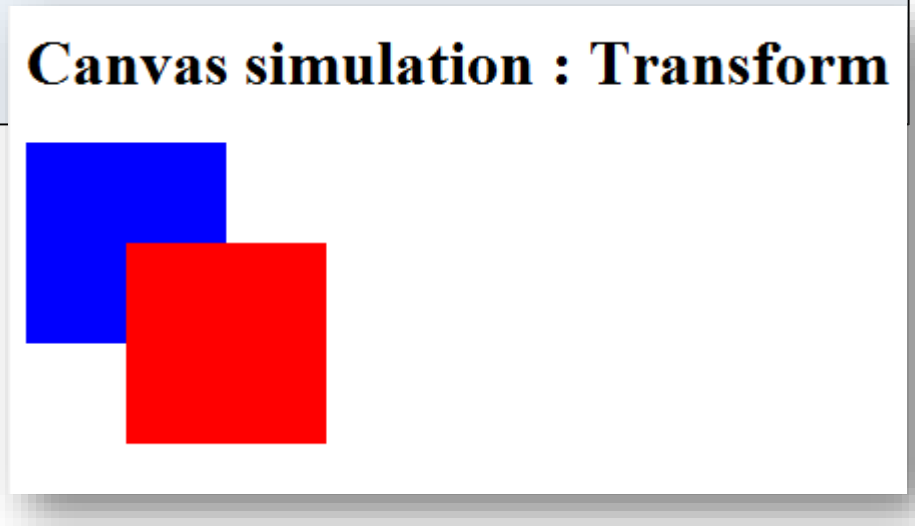


도형 변환 (Transformation)

- 평행이동(translation)
- 신축(scaling)
- 회전(rotation)
- 밀림(shear)
- 반사(mirror)
- **행렬**을 이용한 일반적인 변환 (Transform by **Matrix**)

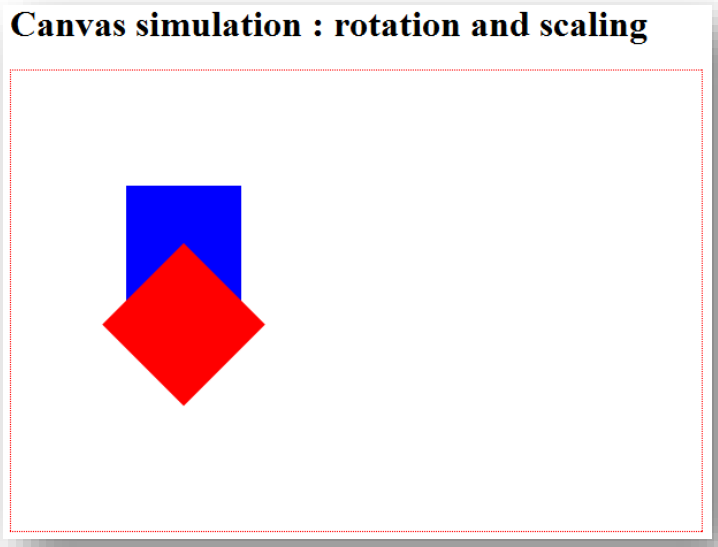
wk04.4 평행이동

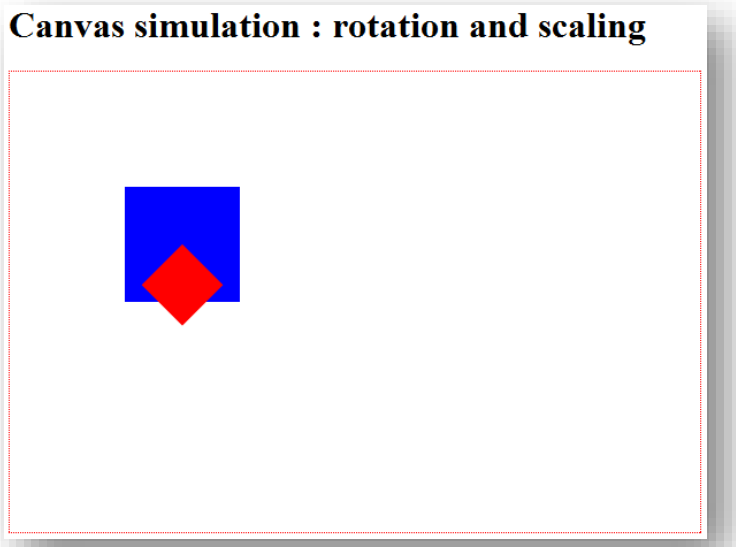
```
<body>
    <canvas id="myCanvas" width="600" height="400"></canvas>
    <script>
        var canvas = document.getElementById('myCanvas');
        var context = canvas.getContext('2d');

        context.fillStyle = "blue";
        context.fillRect(0, 0, 100, 100);

        context.translate(50, 50); // translation
        context.fillStyle = "red";
        context.fillRect(0, 0, 100, 100);
    </script>
</body>
```



wk04.5 회전, 신축

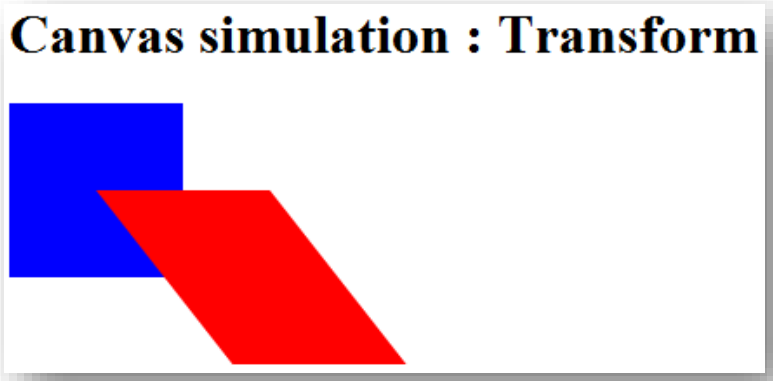
```
context.fillStyle = "blue";
context.fillRect(100, 100, 100, 100);

context.translate(150, 150);  // translation
context.rotate(Math.PI/4);  // rotation
//context.scale(0.5, 0.5);  // scaling

context.fillStyle = "red";
context.fillRect(0, 0, 100, 100);
```

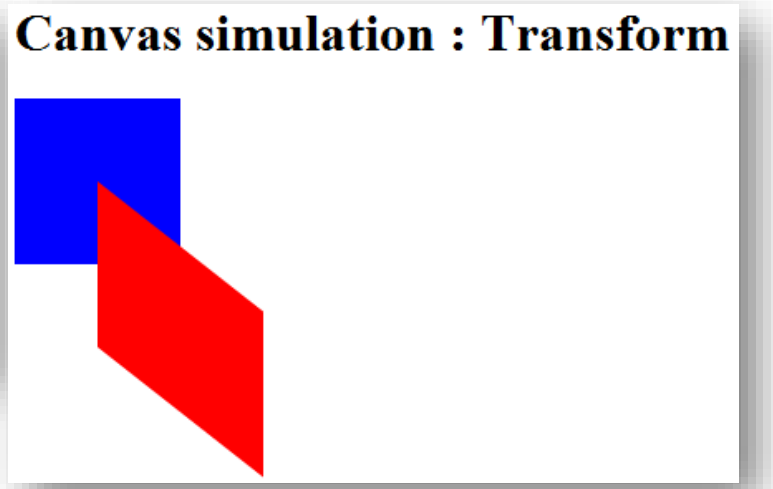




일반 변환 (general transformation)

```
// context.transform(a, b, c, d, e, f)
// context.transform(mx, sy, sx, my, tx, ty)
//context.transform(1, 0, 0, 1, 50, 50); // translation
//context.transform(0.5, 0, 0, 0.5, 50, 50); // scaling & translation
//context.transform(1, 0, Math.PI/4, 1, 50, 50); // skewing in x
//context.transform(1, Math.PI/4, 0, 1, 50, 50); // skewing in y
//context.transform(1, Math.PI/4, -Math.PI/4, 1, 50, 50); // rotation
```





Transformations

Transformations

Method	Description
scale()	Scales the current drawing bigger or smaller
rotate()	Rotates the current drawing
translate()	Remaps the (0,0) position on the canvas
transform()	Replaces the current transformation matrix for the drawing
setTransform()	Resets the current transform to the identity matrix. Then runs transform()

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Transformations

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = a * x + c * y + e,$$
$$y' = b * x + d * y + f$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} m_x & s_y & t_x \\ s_x & m_y & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = m_x * x + s_y * y + t_x,$$
$$y' = s_x * x + m_y * y + t_y$$

What are the meaning of a, b, c, d, e, f ?

Transform: translate(tx, ty)

w3schools.com

Edit This Code: See Result »

```
<!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.fillRect(10, 10, 100, 50);
ctx.translate(70, 70);
ctx.fillStyle="#0000ff";
ctx.fillRect(10, 10, 100, 50);

</script>

</body>
</html>
```

Result:

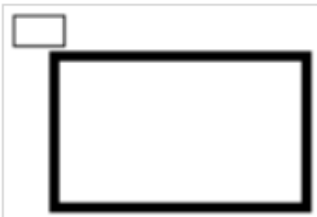
Transform: scale(mx, my)

w3schools.com

Edit This Code: See Result »

```
<!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.strokeRect(5, 5, 25, 15);
ctx.scale(5, 5);
ctx.strokeRect(5, 5, 25, 15);

</script>

</body>
</html>
```

Result:



Transform matrix

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

a, d : scaling
b, c : rotation and shearing
e, f : translation

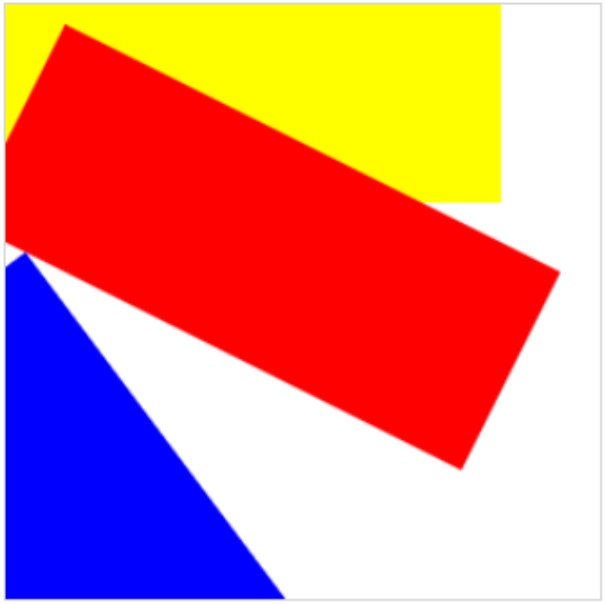
http://www.html5canvastutorials.com/advanced/html5-canvas-custom-transform/

transform(a, b, c, d, e, f)

Edit This Code: See Result »

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

<canvas id="myCanvas" width="300" height="300" 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 = "yellow";
ctx.fillRect(0, 0, 250, 100)

ctx.transform(1, 0.5, -0.5, 1, 30, 10);
ctx.fillStyle = "red";
ctx.fillRect(0, 0, 250, 100);

ctx.transform(1, 0.5, -0.5, 1, 30, 100);
ctx.fillStyle = "blue";
ctx.fillRect(0, 0, 250, 100);

</script>

</body>
</html>
```

Result:



setTransform(a, b, c, d, e, f)

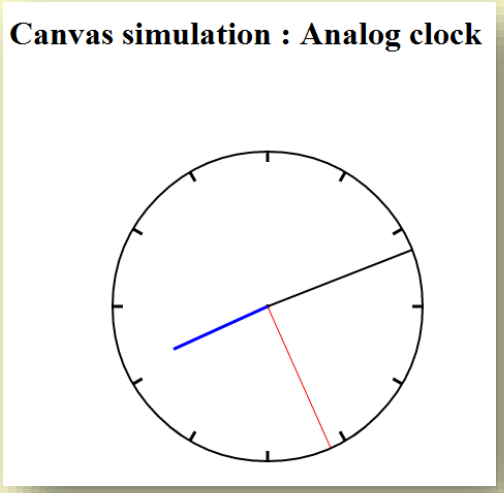
Edit This Code: See Result » Result:

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

<canvas id="myCanvas" width="300" height="300" 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 = "yellow";
ctx.fillRect(0, 0, 250, 100)

ctx.setTransform(1,0.5, -0.5, 1, 30, 10);
ctx.fillStyle = "red";
ctx.fillRect(0, 0, 250, 100);

ctx.setTransform(1,0.5, -0.5, 1, 30, 100);
ctx.fillStyle = "blue";
ctx.fillRect(0, 0, 250, 100);

</script>

</body>
</html>
```

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

가점: Javascript 프로그래밍 응용 능력.

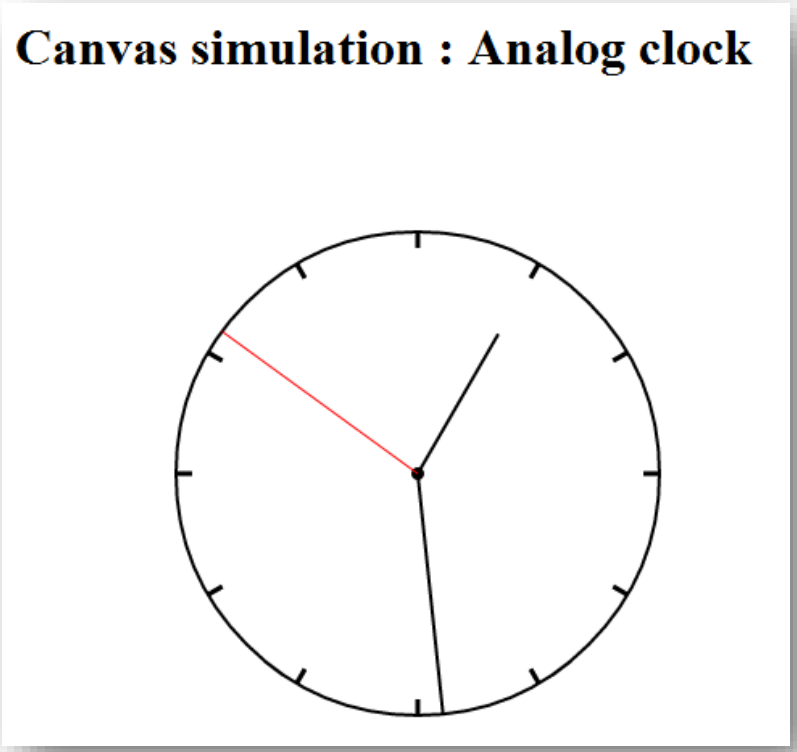
[제출파일] msnn_rpt03.zip

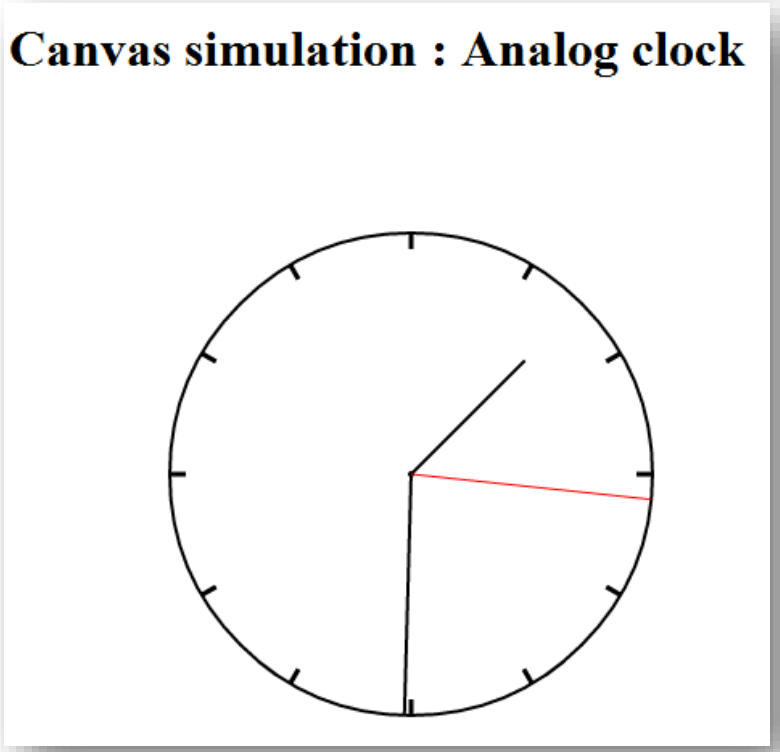
msnn_rpt03.html 과 사용된 그림을 이메일로 제출

Email : chaos21c@gmail.com

chaos21c@gmail.com

과제03. hint



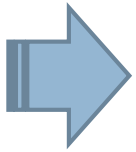


Which clock is running exactly in real time?

과제03. hint

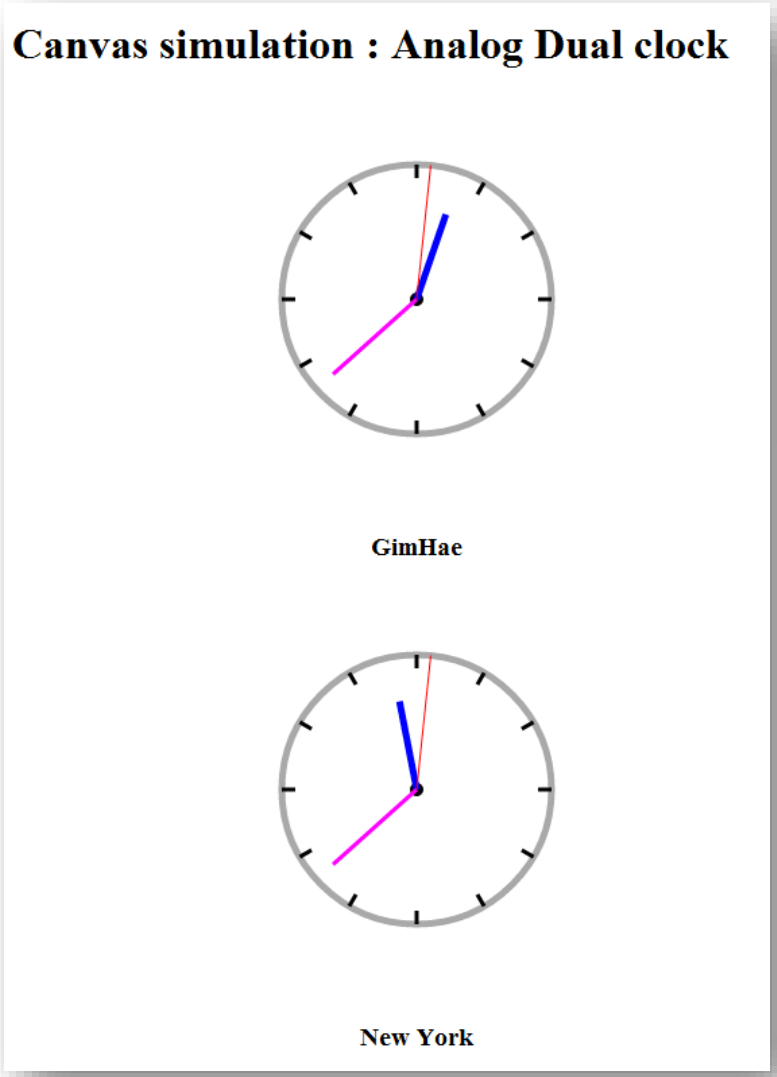

```
// Get time
var date = new Date();

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

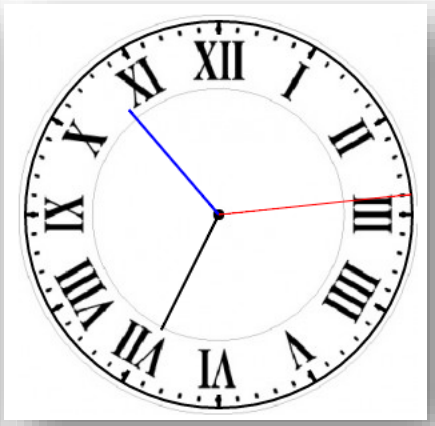
```
// second handle
ctx.strokeStyle = "red";
ctx.lineWidth = 1;
//seconds = 1;
drawHand(clockWidth / 2, seconds * 6);

// 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);
```

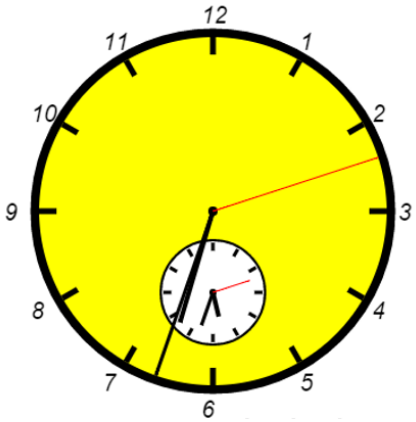


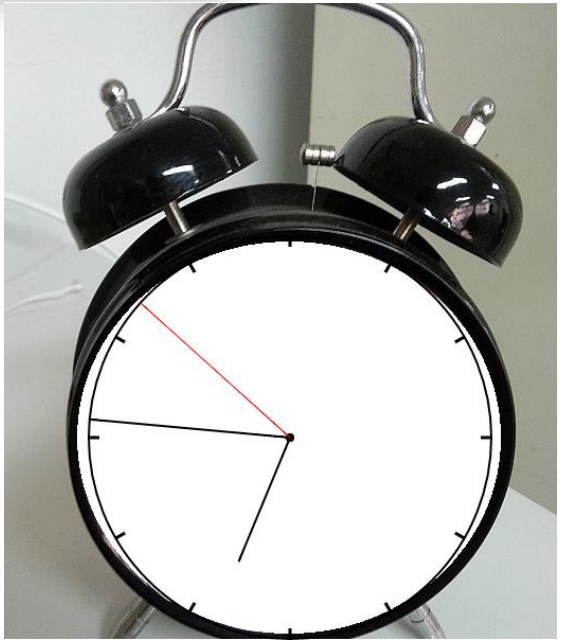
How can you make every handle correctly move?

과제03. possible results

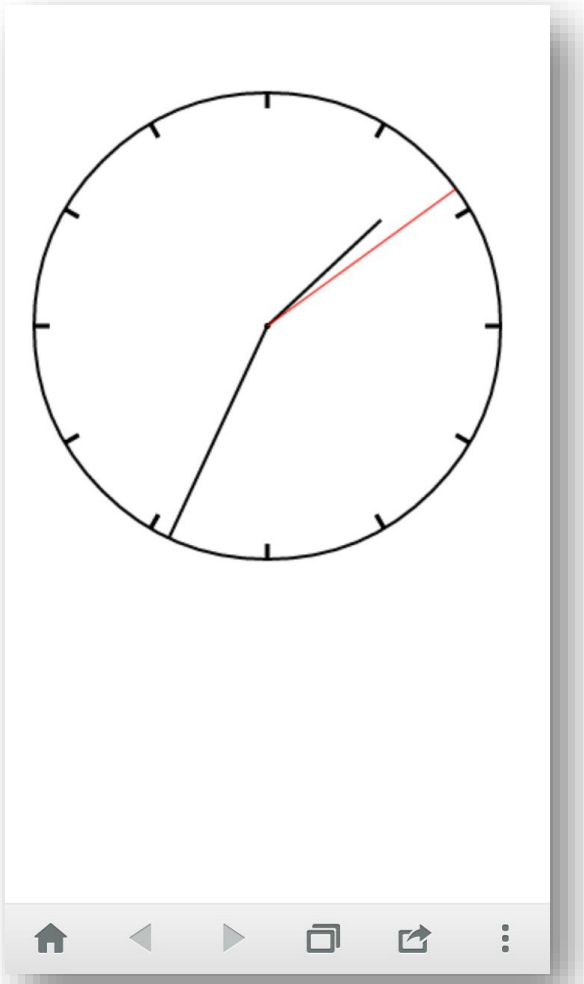


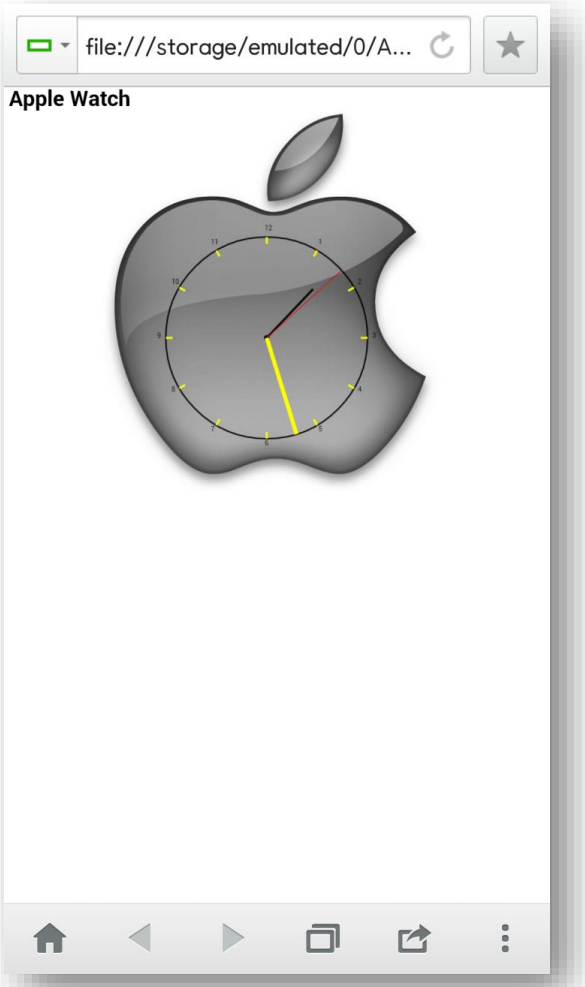






How to publish my clock on mobile?

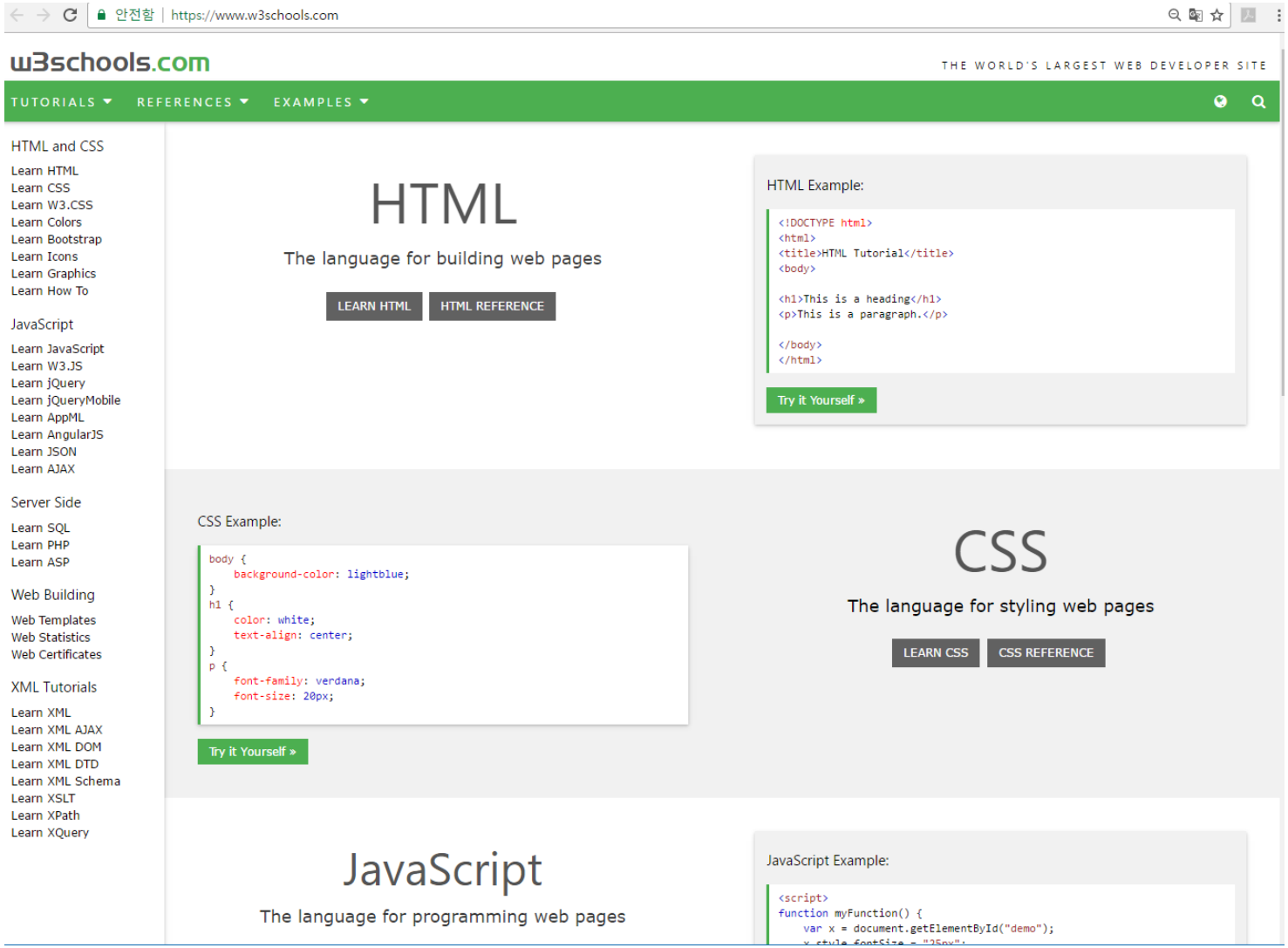




교재 WEB 강의 소개

http://webprogramming.co.kr

관련 WEB 강의 소개 – w3schools.com



http://www.w3schools.com