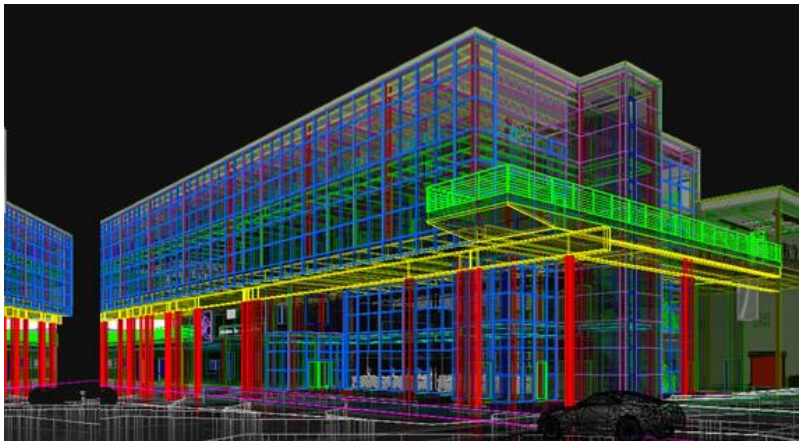
CHAPTER 11.
자바스크립트와 캔버스로 게임만들기

캔버스

- 캔버스는 <canvas> 요소로 생성
- 캔버스는 HTML 페이지 상에서 사각 형태의 영역
- 실제 그림은 자바스크립트를 통하여 코드로 그려야 한다.



컨텍스트 객체

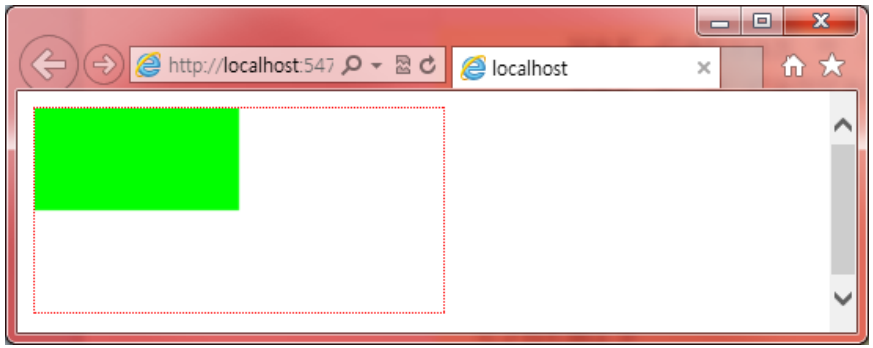
- **컨텍스트(context) 객체 :** 자바스크립트에서 물감과 붓의 역할을 한다.

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

예제

```
<!DOCTYPE html>
<html>
<body>
    <canvas id="myCanvas" width="200" height="100"
        style="border: 1px dotted red"></canvas>
    <script>
        var canvas = document.getElementById("myCanvas");
        var context = canvas.getContext("2d");
        context.fillStyle = "#00FF00";
        context.fillRect(0, 0, 100, 50);
    </script>
</body>
</html>
```

캔버스로 그림을 그릴 때는 해당 코드를 <body>요소의 맨 끝에 두어야 한다.

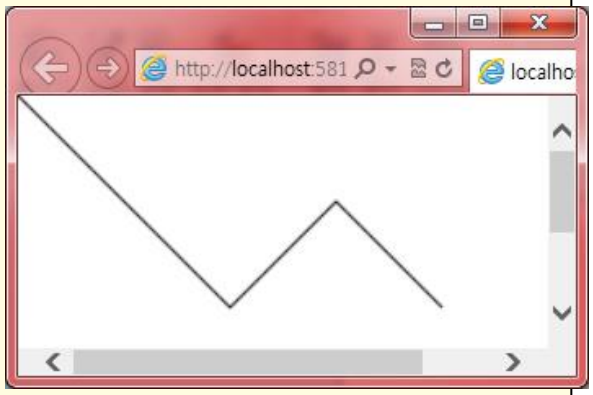


직선 그리기 예제

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      body {
        margin: 0px;
        padding: 0px;
      }
    </style>
  </head>
  <body>
    <canvas id="myCanvas" width="300" height="200"></canvas>
    <script>
        var canvas = document.getElementById('myCanvas');
        var context = canvas.getContext('2d');

        context.beginPath();
        context.moveTo(0, 0);
        context.lineTo(100, 100);
        context.lineTo(150, 50);
        context.lineTo(200, 100);
        context.stroke();
    </script>
  </body>
</html>
```

beginPath()를 호출해서 경로를 초기화

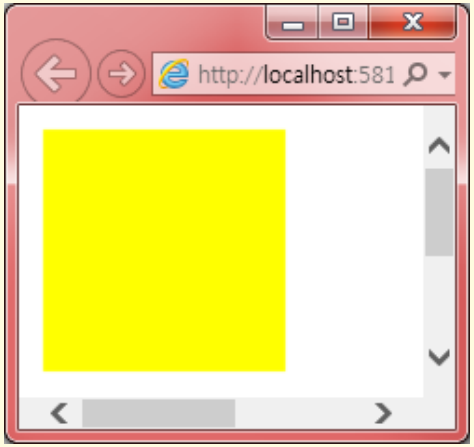
마지막 stroke 주의!



사각형 예제

```
<!DOCTYPE HTML>
<html>
<head>
    <style>
        body {
            margin: 0px;
            padding: 0px;
        }
    </style>
</head>
<body>
    <canvas id="myCanvas" width="300" height="200"></canvas>
    <script>
        var canvas = document.getElementById('myCanvas');
        var context = canvas.getContext('2d');

        context.beginPath();
        context.rect(10, 10, 100, 100);
        context.fillStyle = "yellow";

        context.fill();
    </script>
</body>
</html>
```

도형을 채울 때는 stroke()가 아니라 fill() 사용



원 예제

```
<!DOCTYPE HTML>
<html>
<head>
   <style>
      body {
         margin: 0px;
         padding: 0px;
      }
   </style>
</head>
```

원그리기

arc(x, y, radius, startAngle, endAngle, antiClockwise) :

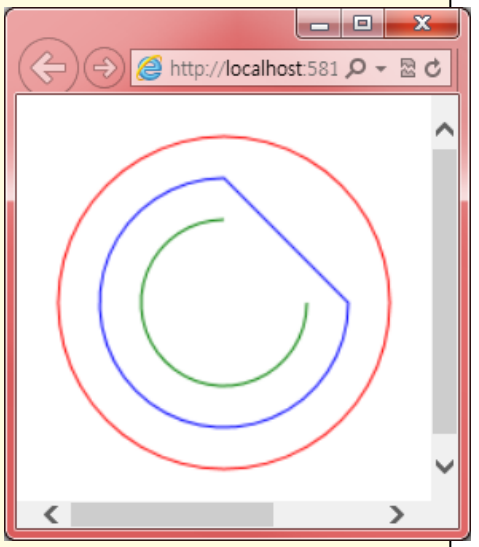
(x, y)를 중심으로 반지름이 radius인 원을 그리는데,

start는 시작 각도이고 end는 종료 각도이다.

antiClockwise가 true이면 반시계 방향, false이면 시계 방향

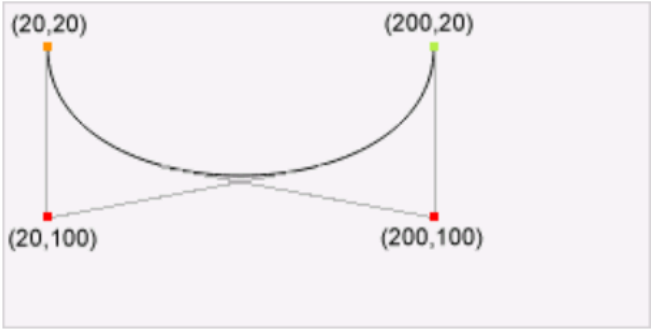
원 예제

```
<body>
   <canvas id="myCanvas" width="300" height="200"></canvas>
    <script>
       var canvas = document.getElementById('myCanvas');
       var context = canvas.getContext('2d');

       context.beginPath();
       context.arc(100, 100, 80, 0, 2.0 * Math.PI, false);
       context.strokeStyle = "red";
       context.stroke();

       context.beginPath();
       context.arc(100, 100, 60, 0, 1.5 * Math.PI, false);
       context.closePath(); // 시작점과 끝점을 연결
       context.strokeStyle = "blue";
       context.stroke();

       context.beginPath();
       context.arc(100, 100, 40, 0, 1.5 * Math.PI, false);
       context.strokeStyle = "green";
       context.stroke();

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



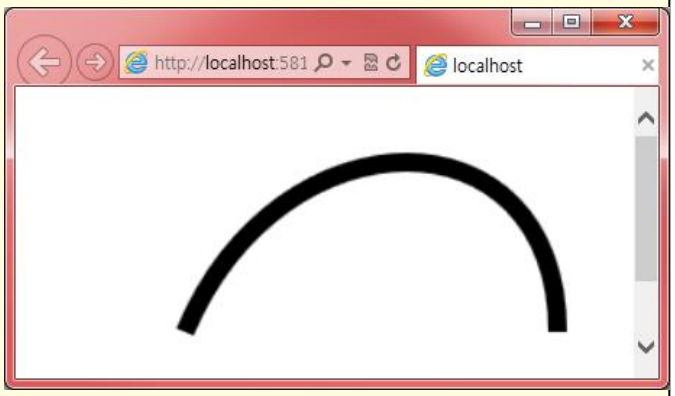
곡선 그리기

- *context*.bezierCurveTo(*cp1x,cp1y,cp2x,cp2y,x,y*);
  - the cubic Bézier calculation and the last point is the ending point for the curve. The starting point for the curve is the last point in the current path. If a path does not exist, use the beginPath() and moveTo() methods to define a starting point.
- Start point
  - moveTo(**20,20**)
- Control point 1
  - bezierCurveTo(**20,100**,200,100,200,20)
- Control point 2
  - bezierCurveTo(20,100,**200,100**,200,20)
- End point
  - bezierCurveTo(20,100,200,100,**200,20**)



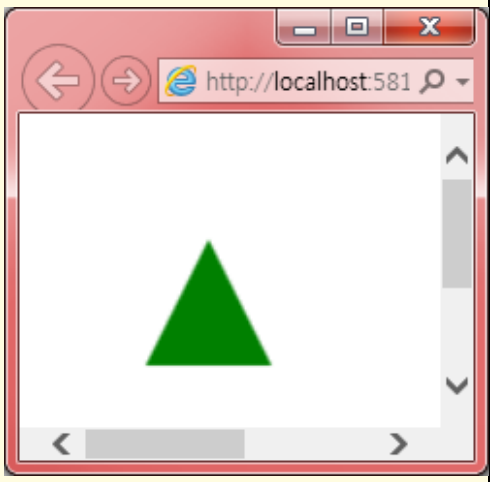
커브 예제

```
<!DOCTYPE HTML>
<html>
<head>
   <style>
      body {
         margin: 0px;
         padding: 0px;
      }
   </style>
</head>
<body>
   <canvas id="myCanvas" width="300" height="200"></canvas>
 <script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');

    context.beginPath();
    context.moveTo(90, 130);
    context.bezierCurveTo(140, 10, 288, 10, 288, 130);
    context.lineWidth = 10;

    context.strokeStyle = 'black';
    context.stroke();
</script>
</body>
</html>
```



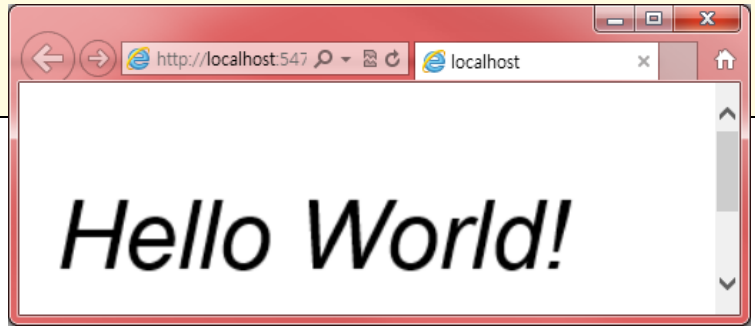
도형 예제

```
<!DOCTYPE HTML>
<html>
<head>
   <style>
      body {
         margin: 0px;
         padding: 0px;
      }
   </style>
</head>
<body>
   <canvas id="myCanvas" width="300" height="200"></canvas>
 <script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');

    context.beginPath();
    context.moveTo(50, 100);
    context.lineTo(75, 50);
    context.lineTo(100, 100);
    context.closePath();
    context.fillStyle = "green";
    context.fill();
</script>
</body>
</html>
```



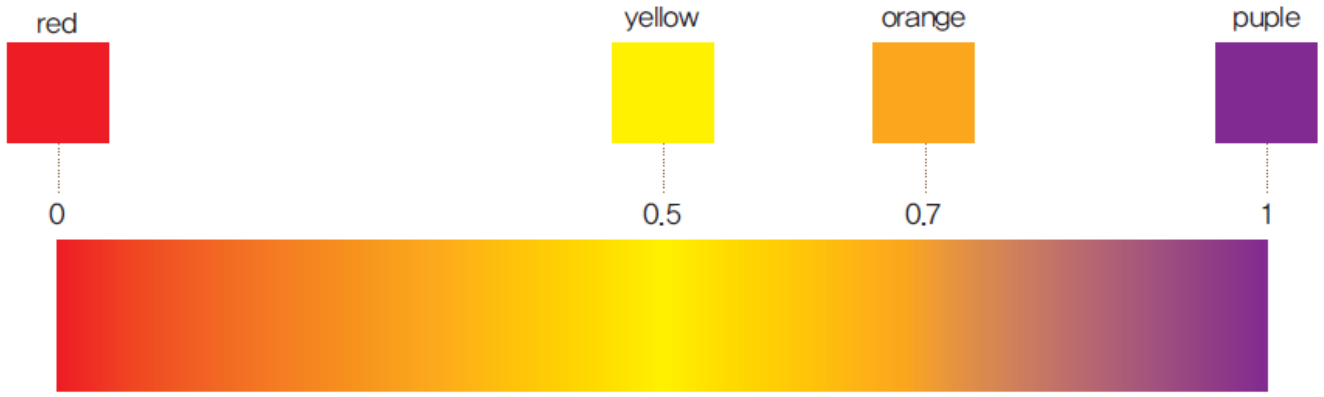
텍스트 예제

```
<!DOCTYPE HTML>
<html>
<head>
   <style>
      body {
         margin: 0px;
         padding: 0px;
      }
   </style>
</head>
<body>
   <canvas id="myCanvas" width="300" height="200"></canvas>
 <script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');

    context.font = 'italic 38pt Arial'
    context.fillText('Hello World!', 20, 100);
</script>
</body>
</html>
```



그라디언트

- createLinearGradient(x, y, x1, y1) - 선형 그라디언트를 생성한다.
- createRadialGradient(x, y, r, x1, y1, r1) - 원형 그라디언트를 생성한다.
- 그라디언트 객체가 생성되면 2개 이상의 종료 색상을 추가한다. 참고로 그라디언트 위치는 0과 1사이의 실수로 지정된다.

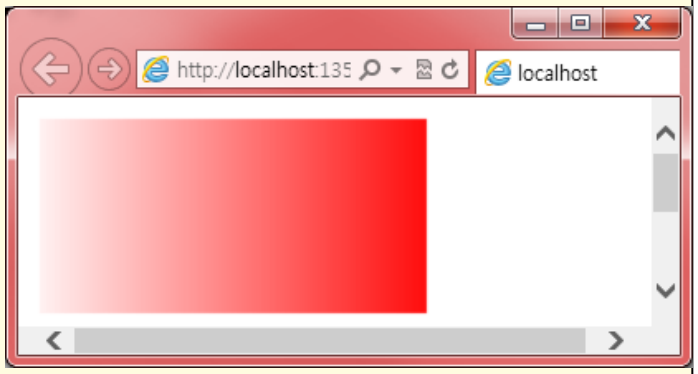


createLinearGradient(x,y,x1,y1)

- x: the x axis of the coordinate of the start point.
- y: the y axis of the coordinate of the start point.
- x1: the x axis of the coordinate of the end point.
- y1: the y axis of the coordinate of the end point.

선형 그라디언트 예제

```
<!DOCTYPE HTML>
<html>
<head>
   <style>
      body {
         margin: 0px;
         padding: 0px;
      }
   </style>
</head>
<body>
   <canvas id="myCanvas" width="300" height="200"></canvas>
  <script>
      var canvas = document.getElementById('myCanvas');
      var context = canvas.getContext('2d');

      var gradient = context.createLinearGradient(0, 0, 200, 0);
      gradient.addColorStop(0, "white");
      gradient.addColorStop(1, "red");

      context.fillStyle = gradient;
      context.fillRect(10, 10, 180, 90);
   </script>
</body>
</html>
```

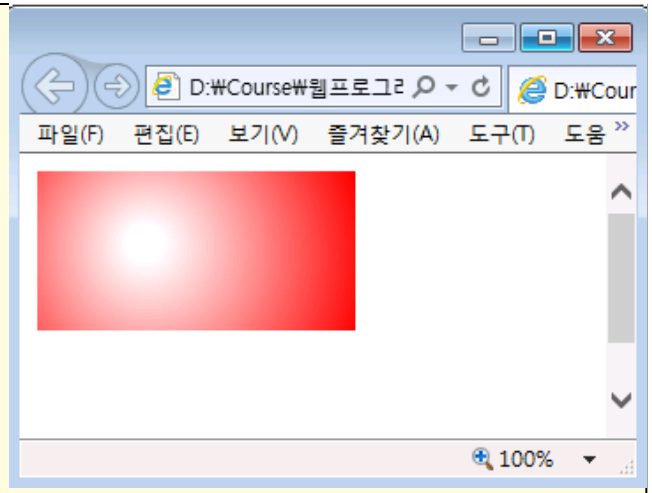


fillRect(x, y, width, height)

- x: 사각형 시작점의 x값
- y: 사각형 시작점의 y값
- width: 사각형 폭
- height: 사각형 높이

원형 그라디언트 예제

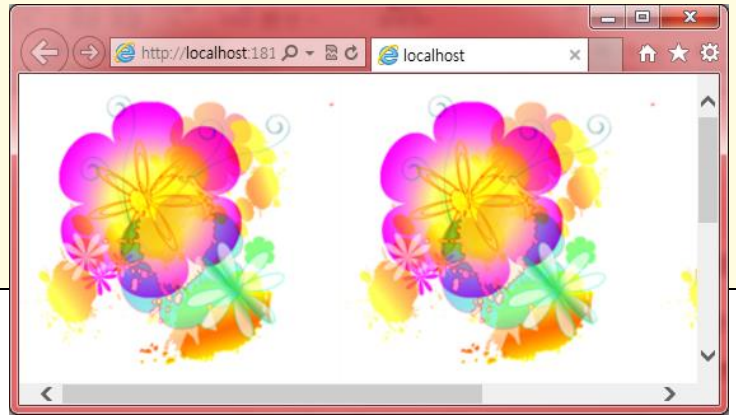
```
<!DOCTYPE HTML>
<html>
<head>
   <style>
      body {
         margin: 0px;
         padding: 0px;
      }
   </style>
</head>
<body>
   <canvas id="myCanvas" width="300" height="200"></canvas>
  <script>
      var canvas = document.getElementById('myCanvas');
      var context = canvas.getContext('2d');

      var gradient = context.createRadialGradient(70, 50, 10, 80, 60, 120);
      gradient.addColorStop(0, "white");
      gradient.addColorStop(1, "red");

      context.fillStyle = gradient;
      context.fillRect(10, 10, 180, 90);
   </script>
</body>
</html>
```



교재 **431**페이지

패턴 채우기

```
<!DOCTYPE HTML>
<html>
<head>
..
</head>
<body>
    <canvas id="myCanvas" width="600" height="200"></canvas>
   <script>
      var canvas = document.getElementById("myCanvas");
      var context = canvas.getContext("2d");

      var image = new Image();
      image.src = "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>
```



이미지 그리기

```
<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 = "html5_logo.png";


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

    </script>
</body>
```

Canvas에 그림을 그리려면

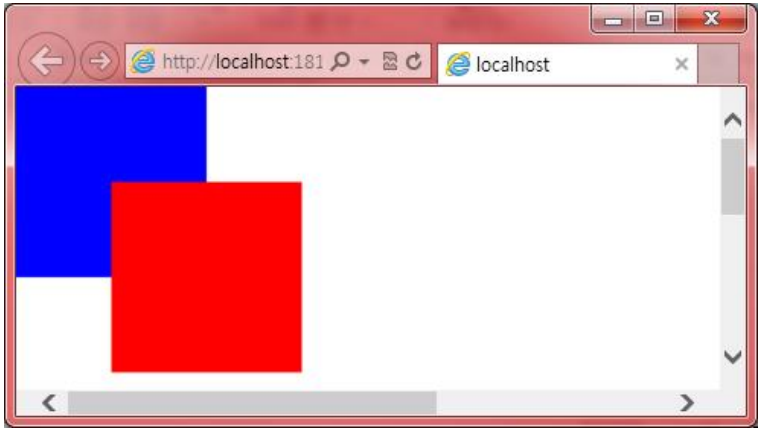
1. drawImage를 사용한다.
2. 그런데 drawImage를 사용하려면 그려질 Image 객체를 로드해야 한다.
3. 그래서 var image = new Image()와 image.src를 이용해서 그려질 이미지를 로드 → 그 후 이미지를 그림의 세 단계로 이미지가 그려지는 과정을 이해할 것

도형 변환

- 평행이동(translation)
- 신축(scaling)
- 회전(rotation)
- 밀림(shear)
- 반사(mirror)
- 행렬을 이용한 일반적인 변환

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

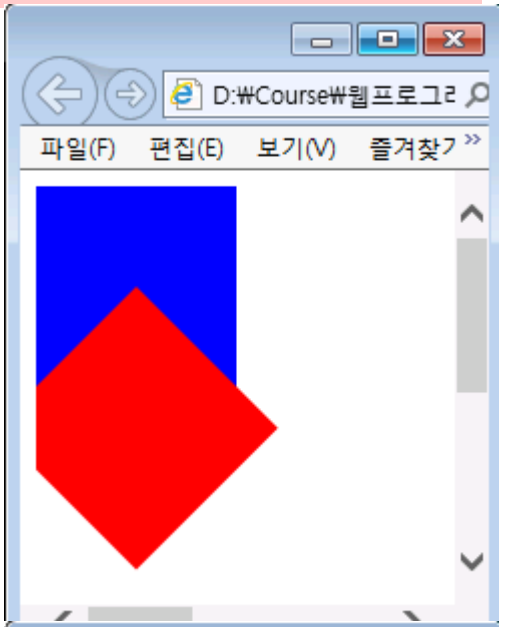


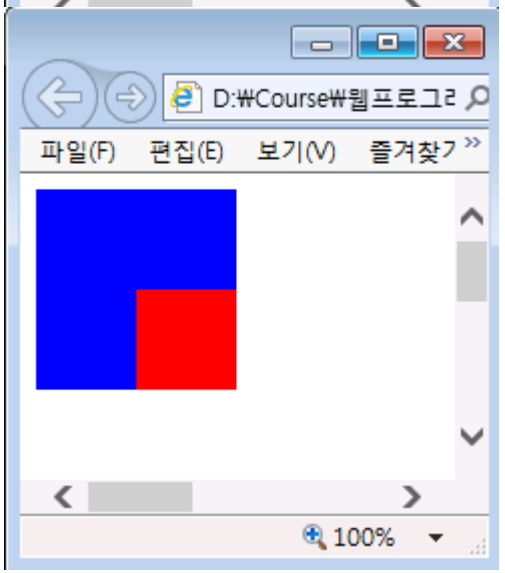
Degree와 radian

- 180 degree = π radian
- 1 degree = $\pi/180$ radian
- x degree = x $\times$ $\pi/180$ radian

회전, 신축, 일반 변환

- 회전
  - context.rotate(float degree) : (0, 0)을 기준으로 좌표 공간을 회전시킴
  - context.rotate(float degree, float px, float py) : (px, py)를 기준으로 좌표 공간을 회전시킴
  - 각도는 radian
    - 20도 회전 시 context.ratate(20*Math.PI/180);
- 신축
  - context.scale(scalewidth, scaleheight) : 현재 그림의 크기를 scalewidth 배 만큼 가로로 변화시키고, scaleheight 배 만큼 세로로 변화시킴
  - scalewidth와 scaleheight가 1이면 그대로, 0.5이면 절반, 2이면 2배





회전, 신축, 일반 변환

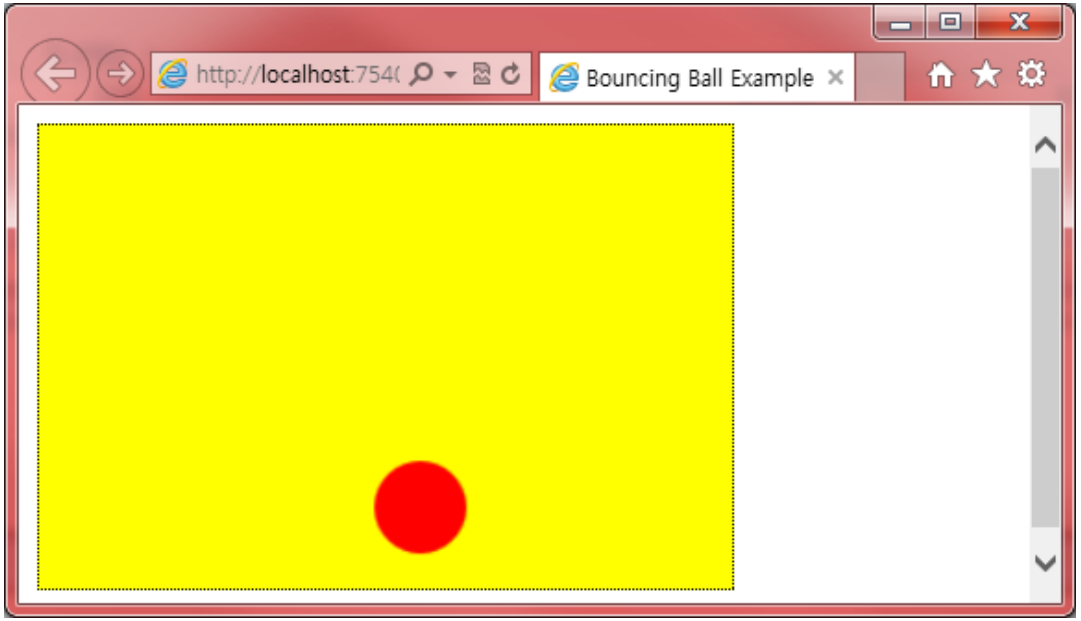
- 일반 변환
  - context.transform(a, b, c, d, e, f) : 변환 행렬의 값을 지정하여 변환

Parameter	Description
a	Scales the drawing horizontally
b	Skew the the drawing horizontally
c	Skew the the drawing vertically
d	Scales the drawing vertically
e	Moves the the drawing horizontally
f	Moves the the drawing vertically

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

애니메이션

- Bouncing Ball 예제



실행(클릭)

- 애니메이션 작성 순서
  - 캔버스를 지운다.
    - context.clearRect(x, y, width, height);
  - (x, y) 위치에 그림을 그린다.
  - 위치를 업데이트한다.
    - x += dx;
    - y += dy;
  - 위의 절차를 반복한다.
    - setTimeout(doSomething, 500); 또는
    - setInterval(doSomething, 500); 사용

Bouncing Ball 예제

```
<!DOCTYPE html>
<html>
<head>
   <title>Bouncing Ball Example</title>
   <style>
      canvas {
         background: yellow;
         border: 1px dotted black;
      }
   </style>

<script>
      var context;
      var dx = 5;
      var dy = 5;
      var y = 100;
      var x = 100;
```

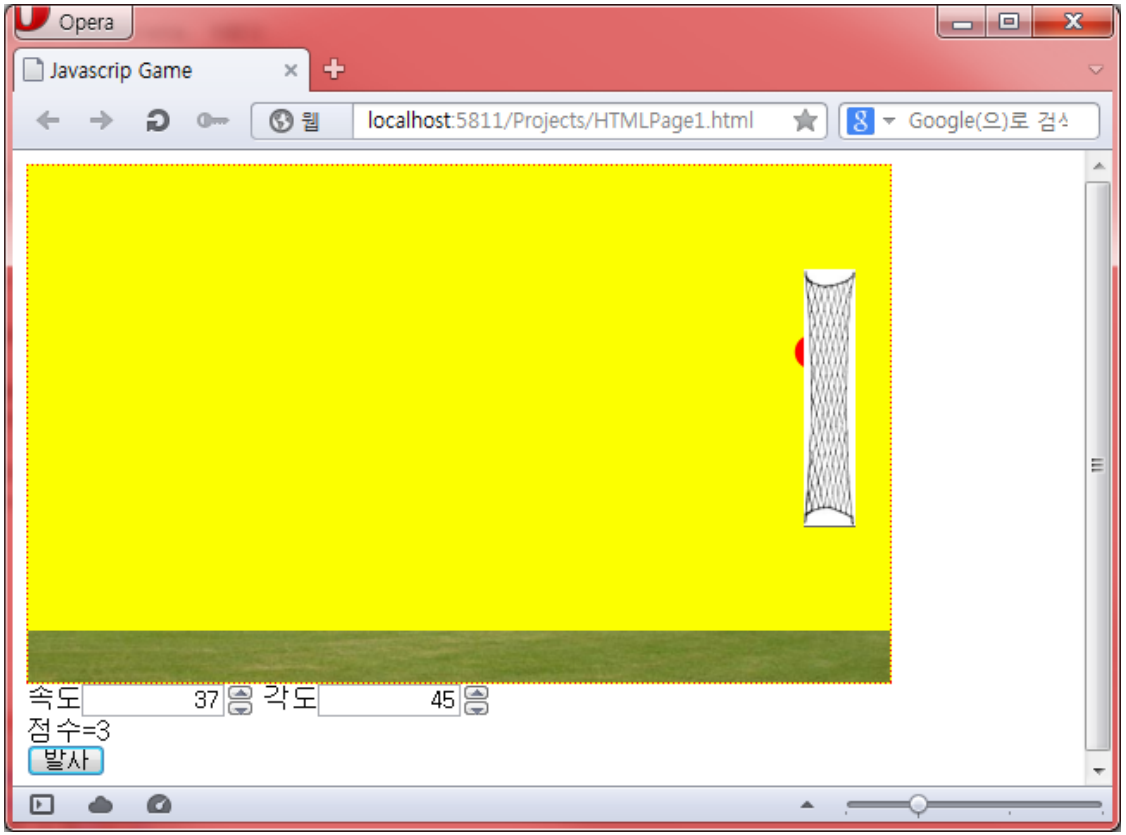
Bouncing Ball 예제

```
        function draw() {
            var canvas = document.getElementById('myCanvas');
            var context = canvas.getContext('2d');
            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>
</head>

<body>
    <canvas id="myCanvas" width="300" height="200"></canvas>
</body>
</html>
```

간단한 게임 제작

- 앵그리 버드와 유사한 다음과 같은 게임을 제작

실행(클릭)



간단한 게임 만들기

```
<html>
<head>
   <title>Javascript Game</title>
   <style>
      canvas {
         border: 1px dotted red;      /* 캔버스에 경계선을 그려준다. */
         background-color: #fcff00; /* 캔버스의 배경색을 지정한다. */
      }
   </style>
   <script>
      var context;                  /* 컨텍스트 객체*/
      var velocity;                 /* 사용자가 입력한 공의 초기속도 */
      var angle;                    /* 사용자가 입력한 공의 초기각도 */
      var ballV;                    /* 공의 현재 속도 */
      var ballVx;                   /* 공의 현재 x방향 속도 */
      var ballVy;                   /* 공의 현재 y방향 속도 */
      var ballX = 10;               /* 공의 현재 x방향 위치 */
      var ballY = 250;              /* 공의 현재 y방향 위치 */
      var ballRadius = 10;          /* 공의 반지름 */
      var score = 0;                /* 점수 */
```

간단한 게임 만들기

```
var image = new Image();                    /* 이미지 객체 생성 */
image.src = "lawn.png";                     /* 이미지 파일 이름 설정 */
var backimage = new Image();
backimage.src = "net.png";
var timer;                                          /* 타이머 객체 변수 */

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

/* 배경을 화면에 그린다. */
function drawBackground() {
    context.drawImage(image, 0, 270);
    context.drawImage(backimage, 450, 60);
}
/* 전체 화면을 그리는 함수 */
function draw() {
    context.clearRect(0, 0, 500, 300);    /* 화면을 지운다. */
    drawBall();
    drawBackground();
}
```

간단한 게임 만들기

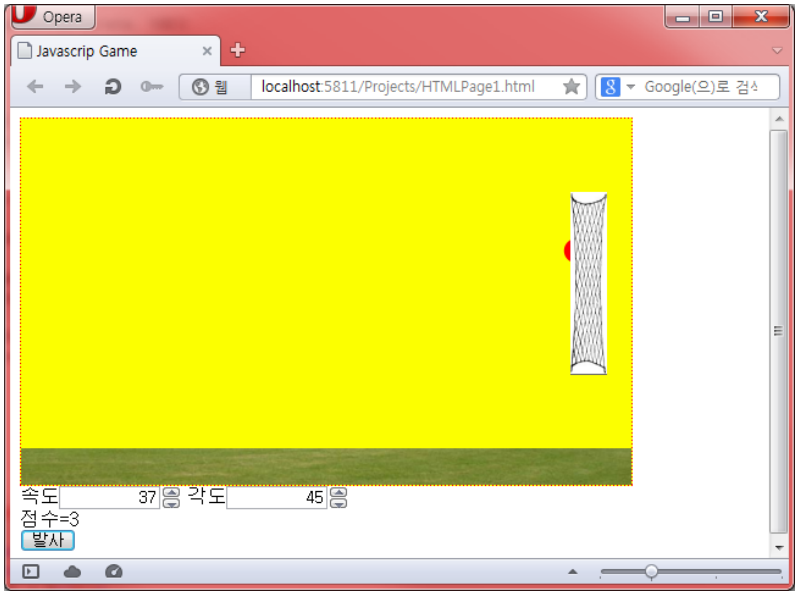
```
/* 초기화를 담당하는 함수 */
function init() {
    ballX = 10;
    ballY = 250;
    ballRadius = 10;
    context = document.getElementById('canvas').getContext('2d');
    draw();
}

/* 사용자가 발사 버튼을 누르면 호출된다. */
function start() {
    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);

    draw();
    timer = setInterval(calculate, 100);
    return false;
}
```

간단한 게임 만들기

```
/* 공의 현재 속도와 위치를 업데이트한다. */
        function calculate() {
            ballVy = ballVy + 1.98;

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

            /* 공이 목표물에 맞았으면 */
            if ((ballX >= 450) && (ballX <= 480) && (ballY >= 60) && (ballY <=
210)) {
                score++;
                document.getElementById("score").innerHTML = "점수=" +
score;
                clearInterval(timer);
            }
            /* 공이 경계를 벗어났으면 */
            if (ballY >= 300 || ballY < 0) {
                clearInterval(timer);
            }
            draw();
        }
    </script>
</head>
```

간단한 게임 만들기

```
<body onload="init();">
    <canvas id="canvas" width="500" height="300"></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" />
        <div id="score">점수 = 0</div>
        <button onclick="start()">발사</button>
    </div>
</body>
</html>
```



Q & A