

HTML5와 JavaScripts : Review(3)



Canvas and event handling

2

HTML5 Canvas API for 2D context(1)

▶ HTML5 CANVAS ELEMENT

▶ Html5 canvas element

```
<canvas id="myCanvas" width="500" height="300">
```

▶ Html5 canvas element with fallback content

```
<canvas id="myCanvas" width="500" height="300">  
  your browser doesn't support canvas!  
</canvas>
```

▶ 2D context

```
var context = canvas.getContext('2d');
```

▶ WebGL context(3D)

```
var context = canvas.getContext('webgl');
```

Sample code : text in drawing area

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      body {
        margin: 0px;
        padding: 0px;
      }
    </style>
  </head>
  <body>
    <canvas id="myCanvas" width="780" height="200"></canvas>
    <script>
      var canvas = document.getElementById('myCanvas');
      var context = canvas.getContext('2d');

      // do cool things with the context
      context.font = '30pt Calibri';
      context.fillStyle = 'blue';
      context.fillText('Pusan National Univ. CS.CSE@PNU', 50, 100);
    </script>
  </body>
</html>
```

HTML5 Canvas API for 2D context(2)

► Paths(Lines & Curves)

► Begin Path

```
context.beginPath();
```

► Line

```
context.lineTo(x, y);
```

► Arc

```
context.arc(x, y, radius, startAngle, endAngle,  
counterClockwise);
```

► Quadratic curve

```
context.quadraticCurveTo(cx, cy, x, y);
```

► Bezier curve

```
context.bezierCurveTo(cx1, cy1, cx2, cy2, x, y);
```

► Close path

```
context.closePath();
```

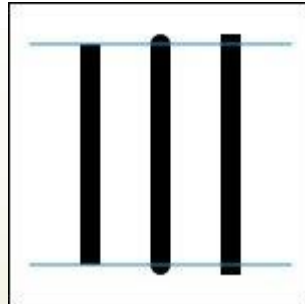
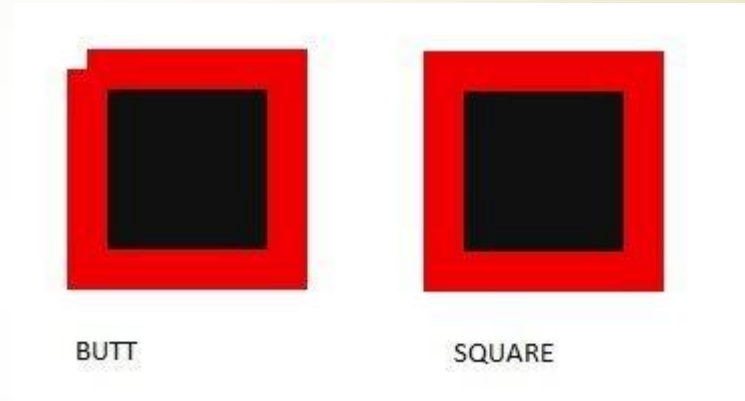
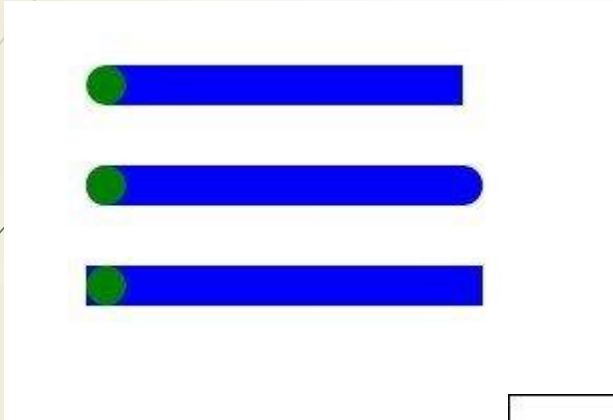
Sample code : Lines

```
<body>
  <canvas id="myCanvas" width="780" height="200"></canvas>
  <script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');

    // do cool things with the context
    context.beginPath();
    context.moveTo(100, 150);
    context.lineTo(450, 50);
    //width
    context.lineWidth=20
    // set line color
    context.strokeStyle = '#ff0000';
    context.lineCap = 'round'; //butt, round,square
    context.stroke();
  </script>
</body>
```

Caps

► Butt, round, square

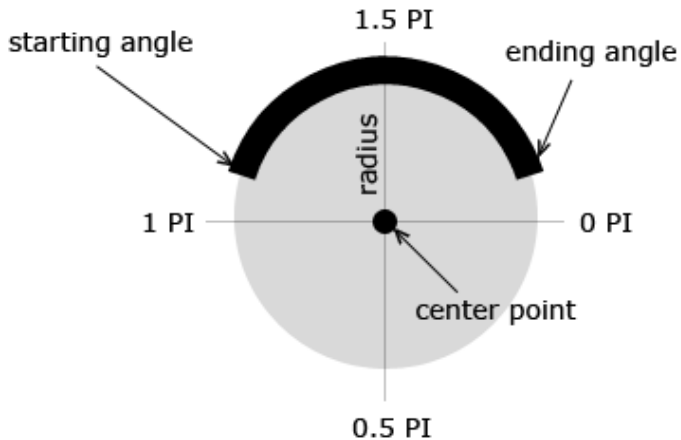


Sample code : Arc

```
<script>
  var canvas = document.getElementById('myCanvas');
  var context = canvas.getContext('2d');
  var x = canvas.width / 2;
  var y = canvas.height / 2;
  var radius = 75;
  var startAngle = 1.1 * Math.PI;
  var endAngle = 1.9 * Math.PI;
  var counterClockwise = false;

  context.beginPath();
  context.arc(x, y, radius, startAngle, endAngle, counterClockwise);
  context.lineWidth = 15;

  // line color
  context.strokeStyle = 'black';
  context.stroke();
</script>
```

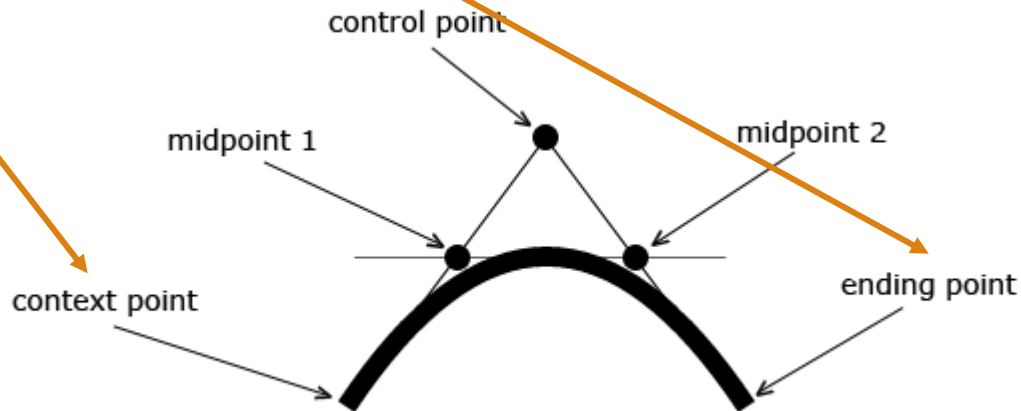


Sample code : Quadratic curve

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

  context.beginPath();
  context.moveTo(188, 150);
  context.quadraticCurveTo(288, 0, 388, 150);
  context.lineWidth = 10;

  // line color
  context.strokeStyle = 'black';
  context.stroke();
</script>
```

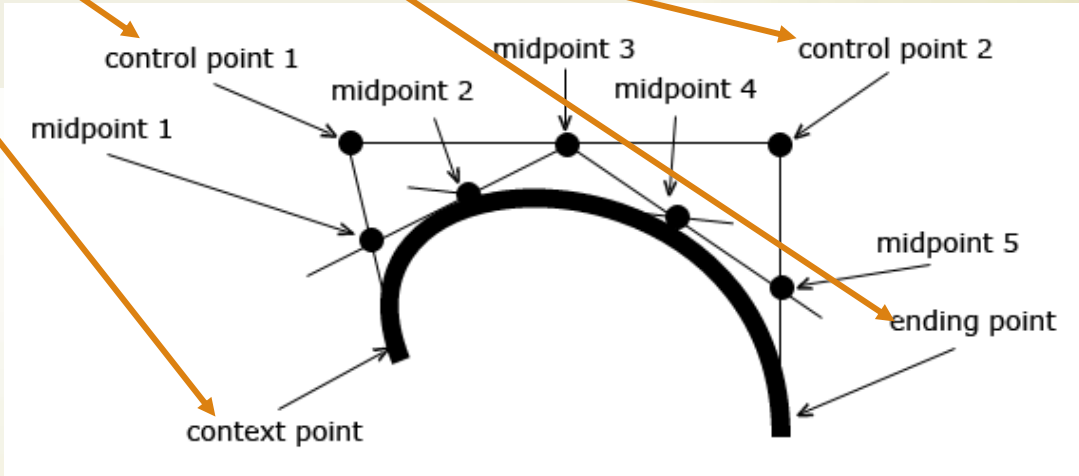


Sample code : Bezier curve

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

  context.beginPath();
  context.moveTo(188, 130);
  context.bezierCurveTo(140, 10, 388, 10, 388, 170);
  context.lineWidth = 10;

  // line color
  context.strokeStyle = 'black';
  context.stroke();
</script>
```



Sample code : together

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

  context.beginPath();
  context.moveTo(100, 20);

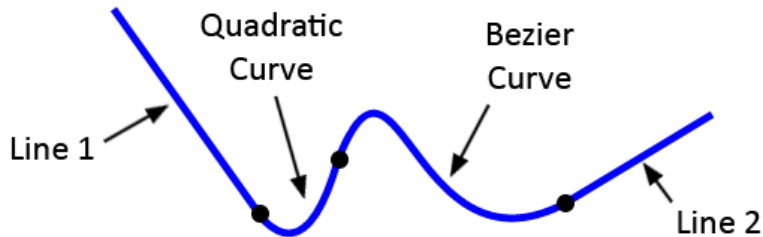
  // line 1
  context.lineTo(200, 160);

  // quadratic curve
  context.quadraticCurveTo(230, 200, 250, 120);

  // bezier curve
  context.bezierCurveTo(290, -40, 300, 200, 400, 150);

  // line 2
  context.lineTo(500, 90);

  context.lineWidth = 5;
  context.strokeStyle = 'blue';
  context.stroke();
</script>
```



Sample code : line joins

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

// set line width for all lines
context.lineWidth = 25;

// miter line join (left)
context.beginPath();
context.moveTo(99, 150);
context.lineTo(149, 50);
context.lineTo(199, 150);
context.lineJoin = 'miter';
context.stroke();

// round line join (middle)
context.beginPath();
context.moveTo(239, 150);
context.lineTo(289, 50);
context.lineTo(339, 150);
context.lineJoin = 'round';
context.stroke();

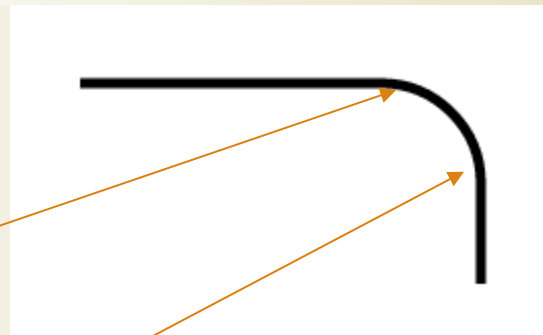
// bevel line join (right)
context.beginPath();
context.moveTo(379, 150);
context.lineTo(429, 50);
context.lineTo(479, 150);
context.lineJoin = 'bevel';
context.stroke();
</script>
```



Sample code : Rounded Corners

```
<script>
    var canvas = document.getElementById('myCanvas');
    var context = canvas.getContext('2d');
    var rectWidth = 200;
    var rectHeight = 100;
    var rectX = 189;
    var rectY = 50;
    var cornerRadius = 50;

    context.beginPath();
    context.moveTo(rectX, rectY);
    context.lineTo(rectX + rectWidth - cornerRadius, rectY);
    context.arcTo(rectX + rectWidth, rectY, rectX + rectWidth, rectY + cornerRadius, cornerRadius);
    context.lineTo(rectX + rectWidth, rectY + rectHeight);
    context.lineWidth = 5;
    context.stroke();
</script>
```



HTML5 Canvas API for 2D context(3)

▶ Shapes

▶ Draw rectangle

```
context.rect(x, y, width, height); context.fill(); context.stroke();
```

▶ Fill rectangle shorthand

```
context.fillRect(x, y, width, height);
```

▶ Stroke rectangle shorthand

```
context.strokeRect(x, y, width, height);
```

▶ Draw circle

```
context.arc(x, y, radius, 0, Math.PI * 2);  
context.fill();  
context.stroke();
```

Sample code : Rectangle

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

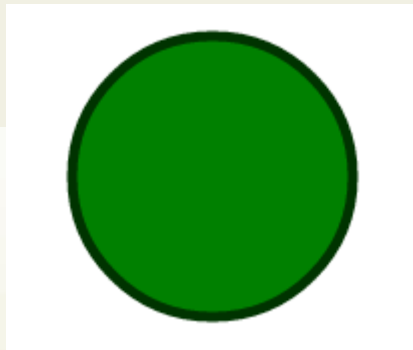
  context.beginPath();
  context.rect(188, 50, 200, 100);
  context.fillStyle = 'yellow';
  context.fill();
  context.lineWidth = 7;
  context.strokeStyle = 'black';
  context.stroke();
</script>
```



Sample code : Circle

```
<script>
  var canvas = document.getElementById('myCanvas');
  var context = canvas.getContext('2d');
  var centerX = canvas.width / 2;
  var centerY = canvas.height / 2;
  var radius = 70;

  context.beginPath();
  context.arc(centerX, centerY, radius, 0, 2 * Math.PI, false);
  context.fillStyle = 'green';
  context.fill();
  context.lineWidth = 5;
  context.strokeStyle = '#003300';
  context.stroke();
</script>
```



Sample code : semicircle

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

  context.beginPath();
  context.arc(288, 75, 70, 0, Math.PI, false);
  context.closePath();
  context.lineWidth = 5;
  context.fillStyle = 'red';
  context.fill();
  context.strokeStyle = '#550000';
  context.stroke();
</script>
```



HTML5 Canvas API for 2D context(4)

► Styles(1)

► Fill

```
context.fillStyle = 'red';  
context.fill();
```

► stroke

```
context.strokeStyle = 'red';  
context.stroke();
```

► Linear gradient

```
var grd = context.createLinearGradient(x1, y1, x2, y2);  
grd.addColorStop(0, 'red');  
grd.addColorStop(1, 'blue');  
context.fillStyle = grd;  
context.fill();
```

HTML5 Canvas API for 2D context(5)

➤ Styles(2)

➤ Radial gradient

```
var grd = context.createRadialGradient(x1, y1, radius1, x2, y2, radius2);  
grd.addColorStop(0, 'red');  
grd.addColorStop(1, 'blue');  
context.fillStyle = grd;  
context.fill();
```

➤ Pattern

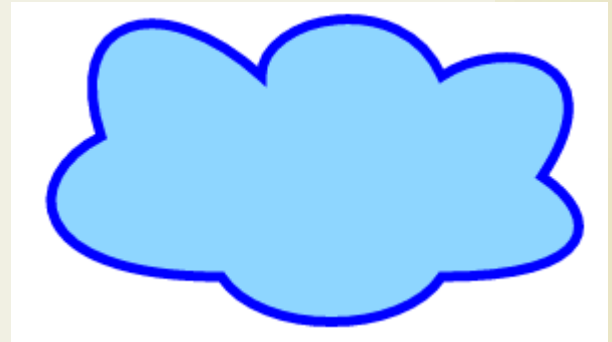
```
var imageObj = new Image();  
imageObj.onload = function() {  
    var pattern = context.createPattern(imageObj, 'repeat');  
    context.fillStyle = pattern;  
    context.fill();  
};  
imageObj.src = 'path/to/my/image.jpg';
```

Sample code : shape fill

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

  // begin custom shape
  context.beginPath();
  context.moveTo(170, 80);
  context.bezierCurveTo(130, 100, 130, 150, 230, 150);
  context.bezierCurveTo(250, 180, 320, 180, 340, 150);
  context.bezierCurveTo(420, 150, 420, 120, 390, 100);
  context.bezierCurveTo(430, 40, 370, 30, 340, 50);
  context.bezierCurveTo(320, 5, 250, 20, 250, 50);
  context.bezierCurveTo(200, 5, 150, 20, 170, 80);

  // complete custom shape
  context.closePath();
  context.lineWidth = 5;
  context.fillStyle = '#8ED6FF';
  context.fill();
  context.strokeStyle = 'blue';
  context.stroke();
</script>
```



Sample code : Linear Gradient

```
<script>
  var canvas = document.getElementById('myCanvas');
  var context = canvas.getContext('2d');
  context.rect(0, 0, canvas.width, canvas.height);

  // add linear gradient
  var grd = context.createLinearGradient(0, 0, canvas.width, canvas.height);
  // light blue
  grd.addColorStop(0, '#8ED6FF');
  // dark blue
  grd.addColorStop(1, '#004CB3');
  context.fillStyle = grd;
  context.fill();
</script>
```

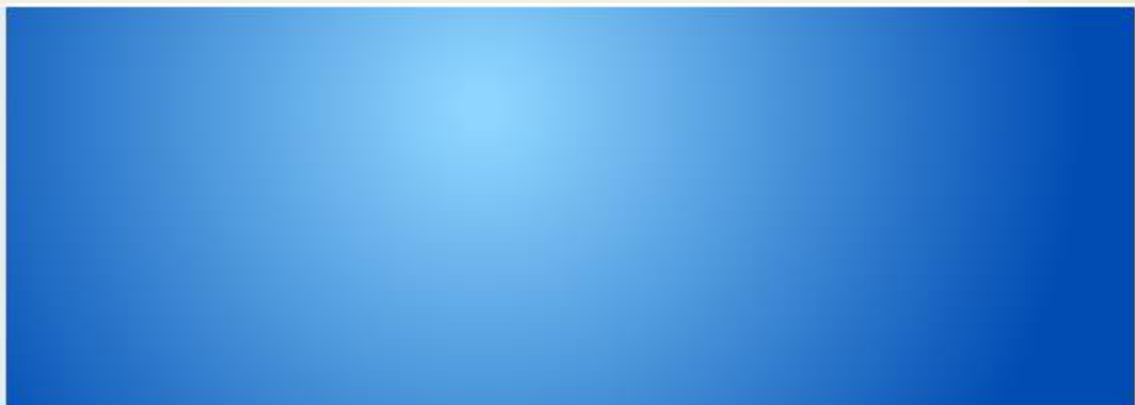


Sample code : Radial Gradient

```
<script>
  var canvas = document.getElementById('myCanvas');
  var context = canvas.getContext('2d');
  context.rect(0, 0, canvas.width, canvas.height);

  // create radial gradient
  var grd = context.createRadialGradient(238, 50, 10, 238, 50, 300);
  // light blue
  grd.addColorStop(0, '#8ED6FF');
  // dark blue
  grd.addColorStop(1, '#004CB3');

  context.fillStyle = grd;
  context.fill();
</script>
```

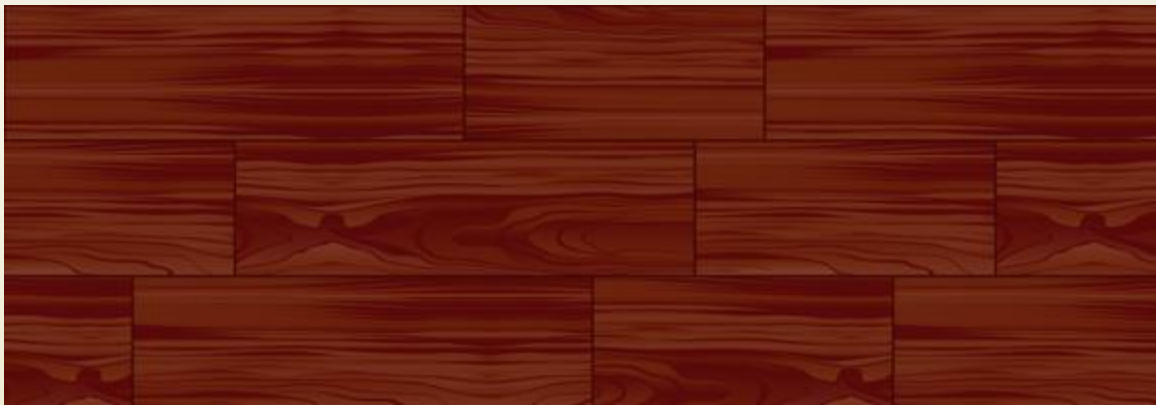


Sample code : pattern

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

  var imageObj = new Image();
  imageObj.onload = function() {
    var pattern = context.createPattern(imageObj, 'repeat');

    context.rect(0, 0, canvas.width, canvas.height);
    context.fillStyle = pattern;
    context.fill();
  };
  imageObj.src = 'http://www.html5canvastutorials.com/demos/assets/wood-pattern.png';
</script>
```



HTML5 Canvas API for 2D context(6)

Images

- Draw image with default size

```
var imageObj = new Image();
imageObj.onload = function() {
    context.drawImage(imageObj, x, y);
};
imageObj.src = 'path/to/my/image.jpg';
```

- Draw image and set size

```
var imageObj = new Image();
imageObj.onload = function() {
    context.drawImage(imageObj, x, y, width, height);
};
imageObj.src = 'path/to/my/image.jpg';
```

- Crop image

```
var imageObj = new Image();
imageObj.onload = function() {
    context.drawImage(imageObj, sx, sy, sw, sh, dx, dy, dw, dh);
};
imageObj.src = 'path/to/my/image.jpg';
```


Sample code : image with size

```
<script>
  var canvas = document.getElementById('myCanvas');
  var context = canvas.getContext('2d');
  var x = 188;
  var y = 30;
  var width = 200;
  var height = 137;
  var imageObj = new Image();

  imageObj.onload = function() {
    context.drawImage(imageObj, x, y, width, height);
  };
  imageObj.src = 'http://www.html5canvastutorials.com/demos/assets/darth-vader.jpg';
</script>
```



Sample code : image Crop

```
<script>
```

```
var canvas = document.getElementById('myCanvas');  
var context = canvas.getContext('2d');  
var imageObj = new Image();
```

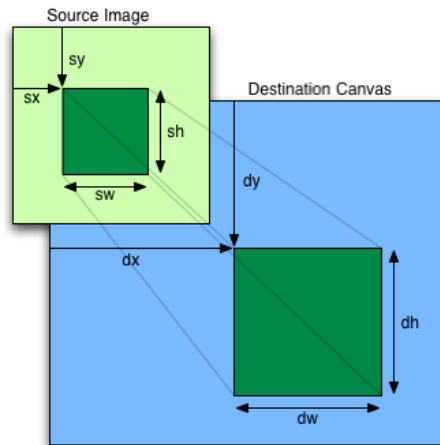
```
imageObj.onload = function() {  
    // draw cropped image  
    var sourceX = 150;  
    var sourceY = 0;  
    var sourceWidth = 150;  
    var sourceHeight = 150;  
    var destWidth = sourceWidth;  
    var destHeight = sourceHeight;  
    var destX = canvas.width / 2 - destWidth / 2;  
    var destY = canvas.height / 2 - destHeight / 2;
```

```
    context.drawImage(imageObj, sourceX, sourceY, sourceWidth, sourceHeight, destX,  
destY, destWidth, destHeight);
```

```
};
```

```
imageObj.src = 'http://www.html5canvastutorials.com/demos/assets/darth-vader.jpg';
```

```
</script>
```



Sample code : image Loader

```
<script>
function loadImages(sources, callback) {
    var images = {};
    var loadedImages = 0;
    var numImages = 0;
    // get num of sources
    for(var src in sources) {
        numImages++;
    }
    for(var src in sources) {
        images[src] = new Image();
        images[src].onload = function() {
            if(++loadedImages >= numImages) {
                callback(images);
            }
        };
        images[src].src = sources[src];
    }
}

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

var sources = {
    darthVader: 'http://www.html5canvastutorials.com/demos/assets/darth-vader.jpg',
    yoda: 'http://www.html5canvastutorials.com/demos/assets/yoda.jpg'
};

loadImages(sources, function(images) {
    context.drawImage(images.darthVader, 100, 30, 200, 137);
    context.drawImage(images.yoda, 350, 55, 93, 104);
});
</script>
```



HTML5 Canvas API for 2D context(7)

► Transforms

► Traslate

```
context.translate(x, y);
```

► Scale

```
context.scale (x, y);
```

► Rotate

```
context.rotate(radians);
```

► Shear

```
context.transform(1,sy,sx,1,0,0);
```

Sample code : translate

```
<script>
  var canvas = document.getElementById('myCanvas');
  var context = canvas.getContext('2d');
  var rectWidth = 150;
  var rectHeight = 75;

  // translate context to center of canvas
  context.translate(canvas.width / 2, canvas.height / 2);

  context.fillStyle = 'blue';
  context.fillRect(rectWidth / -2, rectHeight / -2, rectWidth, rectHeight);
</script>
```



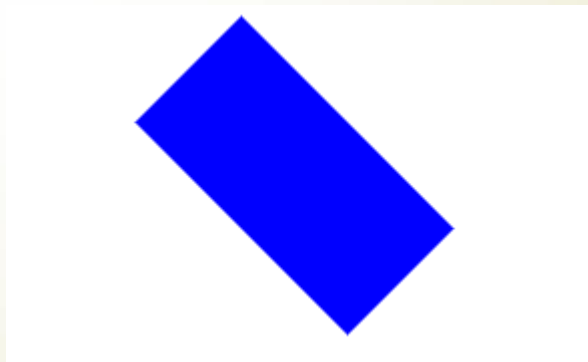
Sample code : translate

```
<script>
  var canvas = document.getElementById('myCanvas');
  var context = canvas.getContext('2d');
  var rectWidth = 150;
  var rectHeight = 75;

  // translate context to center of canvas
  context.translate(canvas.width / 2, canvas.height / 2);

  // rotate 45 degrees clockwise
  context.rotate(Math.PI / 4);

  context.fillStyle = 'blue';
  context.fillRect(rectWidth / -2, rectHeight / -2, rectWidth, rectHeight);
</script>
```



HTML5 Canvas API for 2D context

▶ HTML5 CANVAS CHEAT SHEET

<http://cheatsheetworld.com/programming/html5-canvas-cheat-sheet/>

Event handling

32

HTML5 events

- ▶ **Mouse events**
 - ▶ mouse(click, hover, etc) or touch
- ▶ **Media events**
 - ▶ Medias like videos, images and audio
- ▶ **Keyboard events**
 - ▶ Keyboard
- ▶ **Form events**
 - ▶ Action inside a HTML form
- ▶ **Windows events**
 - ▶ Window object

Mouse events

- **onclick**
- **onmouseover**
- **onmouseout**
- **ondrag**
- **ondragstart**
- **ondragend**
- **ondrop**
- **onscroll**
- **onmousewheel**

Sample code : mouse event

```
<!DOCTYPE HTML>
<html>
  <head>
  </head>
  <body>
    <canvas id="sampleCanvas" width="300" height="100" style="background:black">
    </canvas>
    <script>
      var canvas = document.getElementById('sampleCanvas');
      var context = canvas.getContext('2d');
      canvas.addEventListener('mousemove', handleMoveEvent, false);

      function handleMoveEvent(evt) {
        var mousePos = getMousePos(canvas, evt);
        var message = 'Mouse position: ' + mousePos.x + ',' + mousePos.y;
        writeMessage(canvas, message);
      }

      function writeMessage(canvas, message) {
        var context = canvas.getContext('2d');
        context.clearRect(0, 0, canvas.width, canvas.height);
        context.fillStyle = 'red';
        context.font = '18px Arial';
        context.fillText(message, 10, 25);
      }

      function getMousePos(canvas, evt) {
        var rect = canvas.getBoundingClientRect();
        return {
          x: evt.clientX - rect.left,
          y: evt.clientY - rect.top
        };
      }
    </script>
  </body>
</html>
```



Mouse position: 279,76

Sample code : mouse event

```

<!DOCTYPE HTML>
<html>
  <head>
    <style type="text/css">
      #divMain {width:500px;height:170px;border:1px solid #545454;}
    </style>
    <script>
      function allowDrop(event)
      {
        event.preventDefault();
      }

      function drag(event)
      {
        event.dataTransfer.setData("Text",event.target.id);
      }

      function drop(event)
      {
        event.preventDefault();
        var data=event.dataTransfer.getData("Text");
        event.target.appendChild(document.getElementById(data));
      }
    </script>
  </head>
  <body>
    <p>Drag the image into the rectangle:</p>
    <div id="divMain" ondrop="drop(event)" ondragover="allowDrop(event)"></div>
    <br>
    
  </body>
</html>

```

Drag the image into the rectangle:



Button in HTML(1)

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<button onclick="document.getElementById('demo').innerHTML=Date()">The  
time is?</button>
```

```
<p id="demo"></p>
```

```
</body>
```

```
</html>
```

Button in HTML(2)

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

<p>Click the button to trigger a function that will output "Hello
World" in a p element with id="demo".</p>

<button onclick="myFunction()">Click me</button>

<p id="demo"></p>

<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Hello World";
}
</script>

</body>
</html>
```

Sample code : mouse drawing(1)

```
<script>
  var canvas;
  var context;
  var buttonClicked;
  var x,y;

  function load(){
    canvas = document.getElementById('drawCanvas');
    var context = canvas.getContext('2d');
    buttonClicked=0;

    canvas.addEventListener('click',handleMouseEvent, false);
  }

  function handleMouseEvent(event){
    var context = canvas.getContext('2d');
    buttonClicked += 1;

    if(buttonClicked == 1){
      var mousePos = getMousePos(canvas, event);
      x = mousePos.x;
      y = mousePos.y;
    }else if(buttonClicked == 2){
      mousePos = getMousePos(canvas, event);
      context.beginPath();
      context.moveTo(x, y);
      context.lineTo(mousePos.x, mousePos.y);
      context.stroke();
      context.closePath();
      buttonClicked=0;
    }
  }
}
```

Sample code : mouse drawing(2)

```
function getMousePos(canvas, evt) {  
    var rect = canvas.getBoundingClientRect();  
    return {  
        x: evt.clientX - rect.left,  
        y: evt.clientY - rect.top  
    };  
}
```

```
</script>
```

```
<body onload="load()">
```

```
    <canvas id="drawCanvas" width="500" height="300" style="background:yellow">
```

```
    </canvas>
```

```
</body>
```


JavaScript Events examples

► http://www.w3schools.com/js/js_events_examples.asp



Just try ... in Lab

42