Canvas element

1. A Canvas is a rectangular area on an HTML page, and it is the new element introduced in HTML5 to draw graphics on the fly using script i.e., JavaScript.
2. Canvas acts like a container for graphics, we have to use script to draw graphics.
3. Canvas supports several methods for drawing paths, boxes, circles, characters, and adding images etc.
4. Canvas is supported by most of the browsers like Internet Explorer 9, Firefox, Opera, Chrome, and Safari.
5. Internet Explorer 8 and earlier versions, do not support the <canvas> element.
6. By default, the <canvas> element has no border and no content.

```
<canvas id="MyCanvas" width="300" height="100"></canvas>
```

ID attribute is used to refer canvas in script and a width and height attribute to define the size of the canvas. We can have multiple canvas elements with unique IDs in one HTML page.

Canvas Coordinates:

1. The canvas is a two-dimensional grid.
2. The upper-left corner of the canvas has coordinate (0,0)

Drawing Shapes

Example1: Creating a rectangle filed with Blue color.

```
<!DOCTYPE html>
<head>
<title>Demo Page</title>
</head>
<body>
<canvas id="MyCanvas" width="200" height="100" style="border:
1pxsolid#c3c3c3;"></canvas>
<script>
var c = document.getElementById("MyCanvas");
  var ctx = c.getContext("2d");
  ctx.fillStyle = "blue";
  ctx.fillRect(0, 0, 150, 75);
</script>
</body>
</html>
```

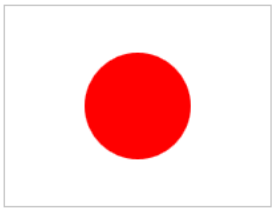
Understanding the Code:

- **getElementById("CanvasId"):** This JavaScript Method is used to get a reference to the canvas declared in the HTML page.
- **getContext("2d"):** This is a built-in HTML5 object, with many properties and methods for drawing paths, boxes, circles, text, images etc.
- **fillStyle:** This property can be a CSS color, a gradient, or a pattern. The default fillStyle is #000000 (black).
- **fillRect(x,y,width,height):** This method draws a rectangle filled with the current fill style.

Example: Creating a circle filed with Red color using arc() method.

arc(x,y,r,start,stop)

```
ctx.beginPath();
ctx.arc(100, 75, 40, 0, 2 * Math.PI);
ctx.fillStyle = "Red";
ctx.fill();
ctx.stroke(); //Draw an empty circle with black border
```



Using Pattern

```
var imageObj = new Image();
imageObj.onload = function () {
  var pattern = ctx.createPattern(imageObj, 'repeat');
  ctx.rect(0, 0, canvas.width, canvas.height);
  ctx.fillStyle = pattern;
  ctx.fill();
};
imageObj.src = 'Smiley.png';
```

Drawing Paths

Canvas paths are used to draw lines on a canvas. To draw the line, we must use one of the "ink" methods, like stroke().

1. `beginPath()`
2. `closePath()`
3. moveTo(*x,y*) defines the starting point of the line
4. lineTo(*x,y*) defines the ending point of the line
5. `arc(x, y, radius, startAngle, endAngle, counterClockwise)`
6. `quadraticCurveTo(cx, cy, x, y)`
7. `bezierCurveTo(cx1, cy1, cx2, cy2, x, y)`

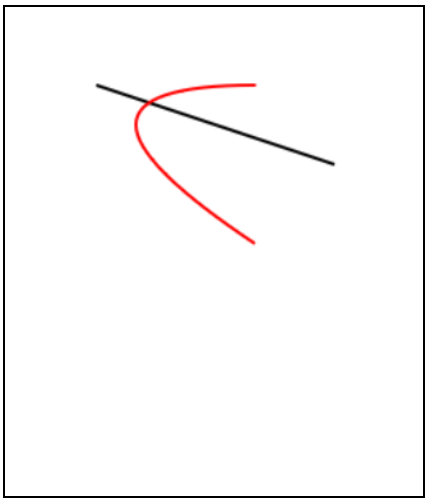
Example:

```
var canvas =
document.getElementById('MyCanvas');
var ctx = canvas.getContext('2d');
ctx.lineWidth = 20;
// miter line join (left)
ctx.beginPath();

ctx.moveTo(99, 150);
ctx.lineTo(149, 50);
ctx.lineTo(199, 150);
ctx.lineJoin = 'miter';
ctx.stroke();

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

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

To set the line join style of an HTML5 Canvas path, we can set the **lineJoin** context property. Paths can have one of three line joins: miter, round, or bevel. Unless otherwise specified, the HTML5 Canvas line join property is defaulted with the miter style.

Example:

```
ctx.moveTo(50, 50);
ctx.lineTo(200, 100);
ctx.lineWidth = 2;
ctx.strokeStyle = "black";
ctx.stroke();
ctx.strokeStyle = "red"
ctx.beginPath()
ctx.moveTo(150, 50);
ctx.quadraticCurveTo(0,
50, 150, 150);
ctx.stroke();
```



```
ctx.beginPath();
ctx.moveTo(170, 80);
ctx.bezierCurveTo(130, 100, 130, 150, 230, 150);
ctx.bezierCurveTo(250, 180, 320, 180, 340, 150);
ctx.bezierCurveTo(420, 150, 420, 120, 390, 100);
ctx.bezierCurveTo(430, 40, 370, 30, 340, 50);
ctx.bezierCurveTo(320, 5, 250, 20, 250, 50);
ctx.bezierCurveTo(200, 5, 150, 20, 170, 80);
ctx.strokeStyle = 'blue';
ctx.stroke();
```

Drawing Text

Following are the important properties and methods.

1. font - defines the font properties for text
2. fillText(text,x,y) - Draws "filled" text on the canvas
3. strokeText(text,x,y) - Draws text on the canvas (no fill)

Writing a 25px high filled text on the canvas, using "Arial" font:

```
ctx.font = "30px Arial";
ctx.textAlign = 'Center'
ctx.fillText("Deccansoft", 10, 50);
```

Deccansoft

strokeText()

```
ctx.font = "25px Arial";
ctx.strokeStyle = "black";
ctx.strokeText("Deccansoft", 35, 50);
```

Working with Gradients

Gradients are used to fill rectangles, circles, lines, text, etc. with multiple colors and shades.

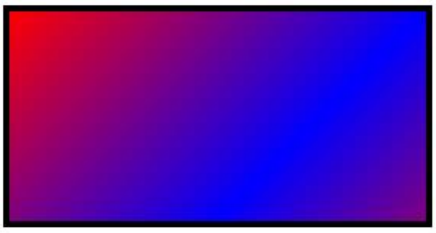
Two different types of gradients:

- **createLinearGradient**(*x,y,x1,y1*) - Creates a linear gradient
- **createRadialGradient**(*x,y,r,x1,y1,r1*) - Creates a radial/circular gradient

These methods returns gradient object. We can add two or more color stops using addColorStop() method to this gradient object. It takes gradient position and color as parameters. Gradient position can range from 0 to 1.

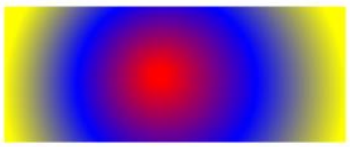
Example: Drawing a rectangle with three colors.

```
// Create gradient
var grd = ctx.createLinearGradient(0, 0, 200, 200);
grd.addColorStop(0, "red");
grd.addColorStop(0.5, "blue");
grd.addColorStop(1, "red");
// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(0, 0, 200, 100);
```



Example: createRadialGradient(*x,y,r,x1,y1,r1*):

```
// Create gradient
var grd = ctx.createRadialGradient(100, 50, 5, 110, 60, 100);
grd.addColorStop(0, "red");
grd.addColorStop(0.5, "blue");
grd.addColorStop(1, "white");
// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(10, 10, 200, 80);
```



Example: Text with Gradient.

```
ctx.font = "30px Verdana";
var grd3 = ctx.createLinearGradient(50, 20, 100, 90);
grd3.addColorStop(0, "black");
grd3.addColorStop("0.6", "blue");   grd3.addColorStop("0.8",
"green");
grd3.addColorStop(1, "red");
ctx.strokeStyle = grd3;
ctx.strokeText("Deccansoft", 35, 50);
```

Creating Shadow Effect

```
ctx.shadowColor = 'black';
ctx.shadowBlur = 2;
ctx.shadowOffsetX = 3;
ctx.shadowOffsetY = 5;

ctx.font = "25px Arial";
ctx.strokeStyle = 'red'
ctx.strokeText("Deccansoft", 35, 50);
```

Translation:

```
ctx.translate(50, 50)
ctx.scale(2, 1);
ctx.rotate(Math.PI / 4);

ctx.font = "25px Arial";
ctx.strokeStyle = 'red'
ctx.strokeText("Deccansoft", 35, 50);
//ctx.scale(-1, 1) – To Flip Horizontal
```

Working with Images

Canvas – Image is used to draw an image on a canvas.

```
context.drawImage(imageObj, sx, sy, sw, sh, dx, dy, dw, dh);
```

```
<!DOCTYPE html>
<html>
<head>
```

```
    <title>An example to draw an polygon</title>

</head>
<body>
    <canvas id="MyCanvas" width="250" height="250" style="border: 1px solid
#c3c3c3;"></canvas>
    <script>
        var imageObj = new Image();
        imageObj.onload = function () {
            var c = document.getElementById("MyCanvas");
            ctx = c.getContext("2d");
            ctx.drawImage(imageObj, 0, 0, 100, 100, 20, 20, 50, 50);
        };
        imageObj.src = 'SharePoint.jpg';
    </script>
</body>
</html>
```

Example: Creating Smiley

```
<!DOCTYPE html>
<html>
<head>
  <title>An example to draw an polygon</title>
</head>
<body>
  <canvas id="MyCanvas" width="250" height="250" style="border: 1px solid #c3c3c3;"></canvas>
  <script>
    var c = document.getElementById("MyCanvas");
    var ctx = c.getContext("2d");
    ctx.arc(50, 50, 20, 0, 2 * Math.PI);
    var grd = ctx.createLinearGradient(40, 40, 50, 75);
    grd.addColorStop(0, '#F9FF00');
    grd.addColorStop(1, '#E0C000');
    // Set the fill style
    ctx.fillStyle = grd;
    ctx.fill();
```

```
// Create others circles (1st Eye)
ctx.beginPath();
ctx.arc(44, 45, 4, 0, 2 * Math.PI);
ctx.fillStyle = "#ffffff";
ctx.fill();

ctx.beginPath();
ctx.arc(44, 45, 2, 0, 2 * Math.PI);
ctx.fillStyle = "#000000";
ctx.fill();

//2nd Eye
ctx.beginPath();
ctx.arc(58, 45, 4, 0, 2 * Math.PI);
ctx.fillStyle = "#ffffff";
ctx.fill();

ctx.beginPath();
ctx.arc(58, 45, 2, 0, 2 * Math.PI);
ctx.fillStyle = "#000000";
ctx.fill();

ctx.beginPath();
ctx.arc(50, 55, 10, 0, Math.PI);
ctx.stroke();

//——second smiley————
ctx.beginPath();
ctx.arc(100, 50, 20, 0, 2 * Math.PI, true);
var grd = ctx.createLinearGradient(40, 40, 50, 75);
grd.addColorStop(0, '#F9FF00');
grd.addColorStop(1, '#E0C000');
ctx.fillStyle = grd;
ctx.fill();
```

```
    ctx.beginPath();
    ctx.arc(94, 45, 4, 0, 2 * Math.PI, true);
    ctx.fillStyle = "#000000";
    ctx.fill();

    ctx.beginPath();
    ctx.arc(106, 45, 4, 0, 2*Math.PI, true);
    ctx.fillStyle = "#000000";
    ctx.fill();

    ctx.beginPath();
    ctx.arc(100, 62, 10, 0, Math.PI, true);
    ctx.stroke();
  </script>
</body>
</html>
```

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