

HTML5 CANVAS - CREATE GRADIENTS

http://www.tutorialspoint.com/html5/canvas_create_gradients.htm

Copyright © tutorialspoint.com

HTML5 canvas allows us to fill and stroke shapes using linear and radial gradients using the following methods –

Sr.No. Method and Description

- 1 addColorStop***offset, color*
This method adds a color stop with the given color to the gradient at the given offset. Here 0.0 is the offset at one end of the gradient, 1.0 is the offset at the other end.
- 2 createLinearGradient***x0, y0, x1, y1*
This method returns a CanvasGradient object that represents a linear gradient that paints along the line given by the coordinates represented by the arguments. The four arguments represent the starting point *x1, y1* and end point *x2, y2* of the gradient.
- 3 createRadialGradient***x0, y0, r0, x1, y1, r1*
This method returns a CanvasGradient object that represents a radial gradient that paints along the cone given by the circles represented by the arguments. The first three arguments define a circle with coordinates *x1, y1* and radius *r1* and the second a circle with coordinates *x2, y2* and radius *r2*.

Linear Gradient Example

Following is a simple example which makes use of above mentioned methods to create Linear gradient.

```
<!DOCTYPE HTML>
<html>
  <head>
    <style>
      #test {
        width:100px;
        height:100px;
        margin:0px auto;
      }
    </style>
    <script type="text/javascript">
      function drawShape(){
        // get the canvas element using the DOM
        var canvas = document.getElementById('mycanvas');

        // Make sure we don't execute when canvas isn't supported
        if (canvas.getContext){
          // use getContext to use the canvas for drawing
          var ctx = canvas.getContext('2d');

          // Create Linear Gradients
          var lingrad = ctx.createLinearGradient(0,0,0,150);

          lingrad.addColorStop(0, '#00ABEB');
          lingrad.addColorStop(0.5, '#fff');
```

```

    lingrad.addColorStop(0.5, '#66CC00');
    lingrad.addColorStop(1, '#fff');

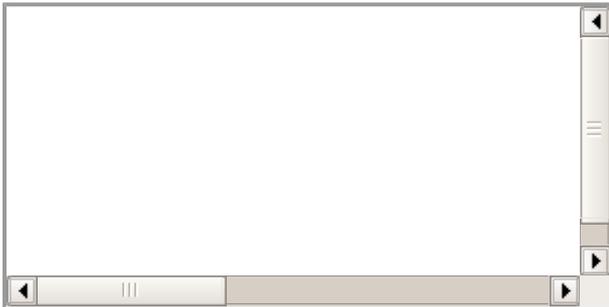
    var lingrad2 = ctx.createLinearGradient(0,50,0,95);
    lingrad2.addColorStop(0.5, '#000');
    lingrad2.addColorStop(1, 'rgba(0,0,0,0)');

    // assign gradients to fill and stroke styles
    ctx.fillStyle = lingrad;
    ctx.strokeStyle = lingrad2;

    // draw shapes
    ctx.fillRect(10,10,130,130);
    ctx.strokeRect(50,50,50,50);
}

else {
    alert('You need Safari or Firefox 1.5+ to see this demo.');
```

The above example would produce following result



Radial Gradient Example

Following is a simple example which makes use of above mentioned methods to create Radial gradient.

```

<!DOCTYPE HTML>
<html>
  <head>

    <style>
      #test {
        width:100px;
        height:100px;
        margin:0px auto;
      }
    </style>

    <script type="text/javascript">
      function drawShape(){

        // get the canvas element using the DOM
        var canvas = document.getElementById('mycanvas');

        // Make sure we don't execute when canvas isn't supported
        if (canvas.getContext){
```

```

// use getContext to use the canvas for drawing
var ctx = canvas.getContext('2d');

// Create gradients
var radgrad = ctx.createRadialGradient(45,45,10,52,50,30);
radgrad.addColorStop(0, '#A7D30C');
radgrad.addColorStop(0.9, '#019F62');
radgrad.addColorStop(1, 'rgba(1,159,98,0)');

var radgrad2 = ctx.createRadialGradient(105,105,20,112,120,50);
radgrad2.addColorStop(0, '#FF5F98');
radgrad2.addColorStop(0.75, '#FF0188');
radgrad2.addColorStop(1, 'rgba(255,1,136,0)');

var radgrad3 = ctx.createRadialGradient(95,15,15,102,20,40);
radgrad3.addColorStop(0, '#00C9FF');
radgrad3.addColorStop(0.8, '#00B5E2');
radgrad3.addColorStop(1, 'rgba(0,201,255,0)');

var radgrad4 = ctx.createRadialGradient(0,150,50,0,140,90);
radgrad4.addColorStop(0, '#F4F201');
radgrad4.addColorStop(0.8, '#E4C700');
radgrad4.addColorStop(1, 'rgba(228,199,0,0)');

// draw shapes
ctx.fillStyle = radgrad4;
ctx.fillRect(0,0,150,150);

ctx.fillStyle = radgrad3;
ctx.fillRect(0,0,150,150);

ctx.fillStyle = radgrad2;
ctx.fillRect(0,0,150,150);

ctx.fillStyle = radgrad;
ctx.fillRect(0,0,150,150);
}
else {
  alert('You need Safari or Firefox 1.5+ to see this demo.');
```

```

}
}
}
</script>
</head>

<body >
  <canvas ></canvas>
</body>
</html>
```

The above example would produce following result –

