

HTML5 CANVAS - SCALING

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

Copyright © tutorialspoint.com

HTML5 canvas provides **scale***x,y* method which is used to increase or decrease the units in our canvas grid. This can be used to draw scaled down or enlarged shapes and bitmaps.

This method takes two parameters where x is the scale factor in the horizontal direction and y is the scale factor in the vertical direction. Both parameters must be positive numbers.

Values smaller than 1.0 reduce the unit size and values larger than 1.0 increase the unit size. Setting the scaling factor to precisely 1.0 doesn't affect the unit size.

Example

Following is a simple example which uses spirograph function to draw nine shapes with different scaling factors.

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

    <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');

          ctx.strokeStyle = "#fc0";
          ctx.lineWidth = 1.5;
          ctx.fillRect(0,0,300,300);

          // Uniform scaling
          ctx.save()
          ctx.translate(50,50);
          drawSpirograph(ctx,22,6,5);

          ctx.translate(100,0);
          ctx.scale(0.75,0.75);
          drawSpirograph(ctx,22,6,5);

          ctx.translate(133.333,0);
          ctx.scale(0.75,0.75);
          drawSpirograph(ctx,22,6,5);

          ctx.restore();

          // Non uniform scaling (y direction)
          ctx.strokeStyle = "#0cf";
          ctx.save()
          ctx.translate(50,150);
          ctx.scale(1,0.75);
          drawSpirograph(ctx,22,6,5);

          ctx.translate(100,0);
          ctx.scale(1,0.75);
          drawSpirograph(ctx,22,6,5);

          ctx.translate(100,0);
          ctx.scale(1,0.75);
```

```

        drawSpirograph(ctx,22,6,5);
        ctx.restore();

        // Non uniform scaling (x direction)
        ctx.strokeStyle = "#cf0";
        ctx.save()
        ctx.translate(50,250);
        ctx.scale(0.75,1);
        drawSpirograph(ctx,22,6,5);

        ctx.translate(133.333,0);
        ctx.scale(0.75,1);
        drawSpirograph(ctx,22,6,5);

        ctx.translate(177.777,0);
        ctx.scale(0.75,1);
        drawSpirograph(ctx,22,6,5);

        ctx.restore();
    }

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

```

    }

    function drawSpirograph(ctx,R,r,0){
        var x1 = R-0;
        var y1 = 0;
        var i = 1;

        ctx.beginPath();
        ctx.moveTo(x1,y1);

        do {
            if (i>20000) break;
            var x2 = (R+r)*Math.cos(i*Math.PI/72) -
(r+0)*Math.cos(((R+r)/r)*(i*Math.PI/72))
            var y2 = (R+r)*Math.sin(i*Math.PI/72) -
(r+0)*Math.sin(((R+r)/r)*(i*Math.PI/72))
            ctx.lineTo(x2,y2);

            x1 = x2;
            y1 = y2;
            i++;
        }

        while (x2 != R-0 && y2 != 0 );
        ctx.stroke();
    }
}
</script>
</head>

<body onload="drawShape();">
    <canvas ></canvas>
</body>

</html>

```

The above example would produce following result –

