

HTML5 CANVAS - TRANSFORMS

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

Copyright © tutorialspoint.com

HTML5 canvas provides methods which allow modifications directly to the transformation matrix. The transformation matrix must initially be the identity transform. It may then be adjusted using the transformation methods.

Sr.No. Method and Description

1 **transform***m11, m12, m21, m22, dx, dy*

This method changes the transformation matrix to apply the matrix given by the arguments.

2 **setTransform***m11, m12, m21, m22, dx, dy*

This method changes the transformation matrix to the matrix given by the arguments .

The *transformm11, m12, m21, m22, dx, dy* method must multiply the current transformation matrix with the matrix described by –

m11	m21	dx
m12	m22	dy
0	0	1

The *setTransformm11, m12, m21, m22, dx, dy* method must reset the current transform to the identity matrix, and then invoke the *transformm11, m12, m21, m22, dx, dy* method with the same arguments.

Example

Following is a simple example which makes use of transform and setTransform methods –

```
<!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');

          var sin = Math.sin(Math.PI/6);
          var cos = Math.cos(Math.PI/6);

          ctx.translate(200, 200);
          var c = 0;

          for (var i=0; i <= 12; i++) {
            c = Math.floor(255 / 12 * i);
            ctx.fillStyle = "rgb(" + c + "," + c + "," + c + ")";
            ctx.fillRect(0, 0, 100, 100);
            ctx.transform(cos, sin, -sin, cos, 0, 0);
          }
        }
      }
    </script>
  </head>
</html>
```

```

        ctx.setTransform(-1, 0, 0, 1, 200, 200);
        ctx.fillStyle = "rgba(100, 100, 255, 0.5)";
        ctx.fillRect(50, 50, 100, 100);
    }

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

```

    }
}
</script>
</head>

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

</html>
```

The above example would produce following result –

