

HTML5 CANVAS - SAVE AND RESTORE STATES

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

Copyright © tutorialspoint.com

HTML5 canvas provides two important methods to save and restore the canvas states. The canvas drawing state is basically a snapshot of all the styles and transformations that have been applied and consists of the followings –

- The transformations such as translate, rotate and scale etc.
- The current clipping region.
- The current values of the following attributes: *strokeStyle*, *fillStyle*, *globalAlpha*, *lineWidth*, *lineCap*, *lineJoin*, *miterLimit*, *shadowOffsetX*, *shadowOffsetY*, *shadowBlur*, *shadowColor*, *globalCompositeOperation*, *font*, *textAlign*, *textBaseline*.

Canvas states are stored on a stack every time the **save** method is called, and the last saved state is returned from the stack every time the **restore** method is called.

Sr.No.	Method and Description
--------	------------------------

- | | |
|---|---|
| 1 | save

This method pushes the current state onto the stack.. |
| 2 | restore

This method pops the top state on the stack, restoring the context to that state. |

Example

Following is a simple example which makes use of above mentioned methods to show how the *restore* is called, to restore the original state and the last rectangle is once again drawn in black.

```
<!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');

          // draw a rectangle with default settings
          ctx.fillRect(0,0,150,150);

          // Save the default state
```

```

        ctx.save();

        // Make changes to the settings
        ctx.fillStyle = '#66FFFF'
        ctx.fillRect( 15,15,120,120);

        // Save the current state
        ctx.save();

        // Make the new changes to the settings
        ctx.fillStyle = '#993333'
        ctx.globalAlpha = 0.5;
        ctx.fillRect(30,30,90,90);

        // Restore previous state
        ctx.restore();

        // Draw a rectangle with restored settings
        ctx.fillRect(45,45,60,60);

        // Restore original state
        ctx.restore();

        // Draw a rectangle with restored settings
        ctx.fillRect(40,40,90,90);
    }
    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 –



Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js