HTML5 CANVAS - DRAWING BEZIER CURVES

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

There are following methods required to draw bezier on the canvas −

Sr.No.	Method and Description
1	**beginPath** This method resets the current path.
2	**moveTo***x, y* This method creates a new subpath with the given point.
3	**closePath** This method marks the current subpath as closed, and starts a new subpath with a point the same as the start and end of the newly closed subpath.
4	**fill** This method fills the subpaths with the current fill style.
5	**stroke** This method strokes the subpaths with the current stroke style.
6	**bezierCurveTo***cp1x, cp1y, cp2x, cp2y, x, y* This method adds the given point to the current path, connected to the previous one by a cubic Bezier curve with the given control points.

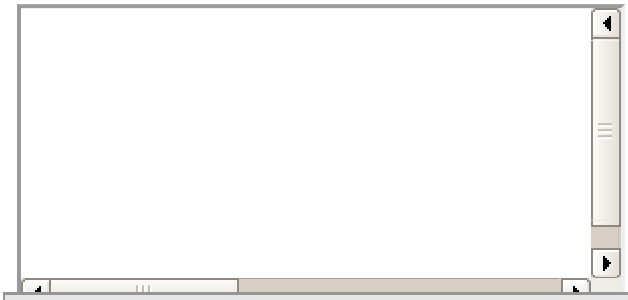
The x and y parameters in bezierCurveTo method are the coordinates of the end point. cp1x and cp1y are the coordinates of the first control point, and cp2x and cp2y are the coordinates of the second control point.

Example

Following is a simple example which makes use of above mentioned methods to draw a Bezier curves.

```
<!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');

               ctx.beginPath();
               ctx.moveTo(75,40);

               ctx.bezierCurveTo(75,37,70,25,50,25);
               ctx.bezierCurveTo(20,25,20,62.5,20,62.5);

               ctx.bezierCurveTo(20,80,40,102,75,120);
               ctx.bezierCurveTo(110,102,130,80,130,62.5);

               ctx.bezierCurveTo(130,62.5,130,25,100,25);
               ctx.bezierCurveTo(85,25,75,37,75,40);

               ctx.fill();
            }

            else {
               alert('You need Safari or Firefox 1.5+ to see this demo.');
            }
         }
      </script>
   </head>

   <body >
      <canvas ></canvas>
   </body>

</html>
```

The above example would draw following shape:



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