HTML5 CANVAS - TRANSLATION

HTML5 canvas provides **translate***x, y* method which is used to move the canvas and its origin to a different point in the grid.

Here argument x is the amount the canvas is moved to the left or right, and y is the amount it's moved up or down

Example

Following is a simple example which makes use of above method to draw various Spirographs −

```
<!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.fillRect(0,0,300,300);

               for (i=0;i<3;i++) {
                  for (j=0;j<3;j++) {
                     ctx.save();
                     ctx.strokeStyle = "#FF0066";

                     ctx.translate(50+j*100,50+i*100);
                     drawSpirograph(ctx,10*(j+3)/(j+2),-2*(i+3)/(i+1),10);

                     ctx.restore();
                  }
               }
            }

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

         function drawSpirograph(ctx,R,r,O){
            var x1 = R-O;
            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+O)*Math.cos(((R+r)/r)*(i*Math.PI/72))
                var y2 = (R+r)*Math.sin(i*Math.PI/72) -
(r+O)*Math.sin(((R+r)/r)*(i*Math.PI/72))
                ctx.lineTo(x2,y2);

                x1 = x2;
                y1 = y2;
                i++;
            } while (x2 != R-O && y2 != 0 );
            ctx.stroke();
         }
      </script>
   </head>

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

</html>
```

The above example would produce following result −