Fractal Tree in HTML

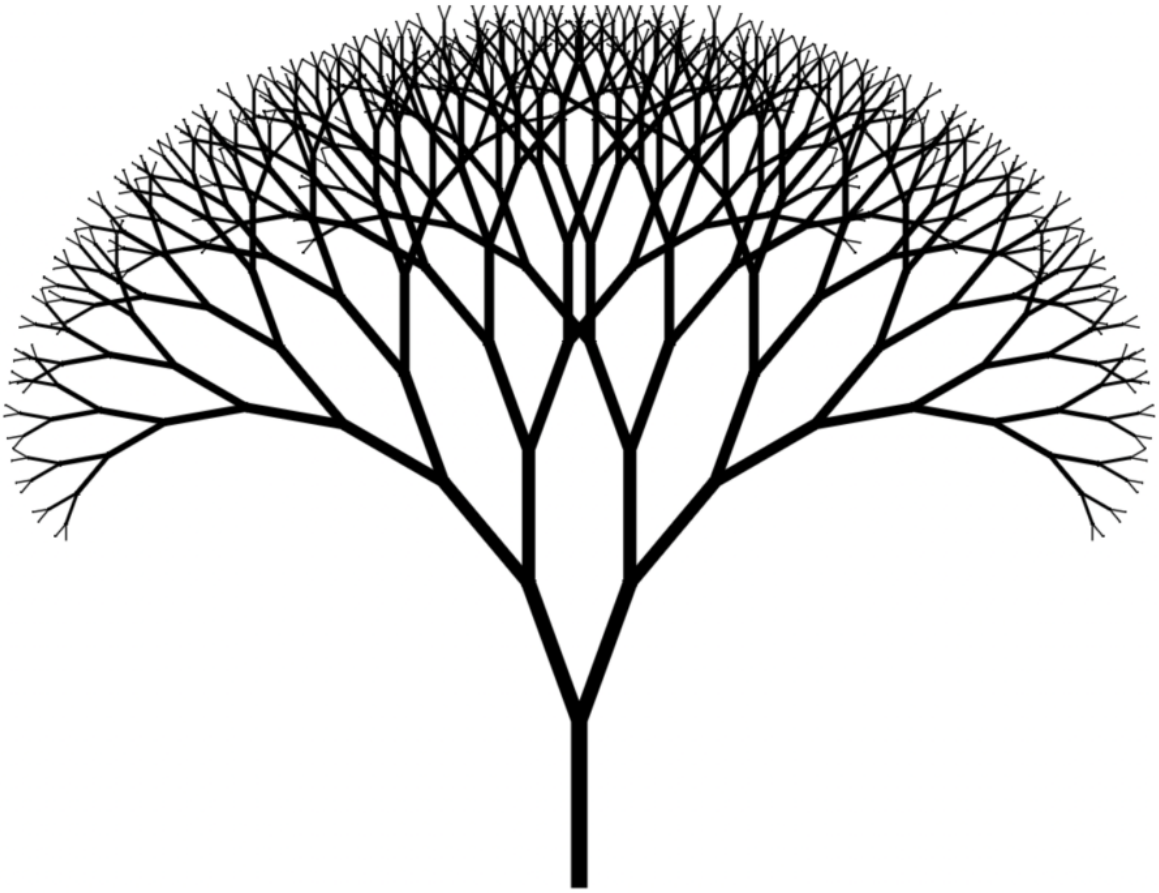
ddas.tech/fractal-tree-in-html/

September 13, 2022

Created By : Debasis Das (12-Sep-2022)

In this post we will create Fractal Trees in HTML using html canvas and Java Script.

Lets start with a **Simple Fractal Tree**



A simple fractal tree in HTML and Javascript

```
<html>
<head>
<script type="text/javascript">
    function draw(){
      var canvas = document.getElementById('canvas');
      var context = canvas.getContext('2d');
      context.beginPath();
      context.rect(0,0,1500,1000);
      context.fillStyle = "white";
      context.fill();
      drawTree(context);
    }

    function drawTree(context){
        branch(context, 750, 900, 270, 10);
    }
    function branch(context, x1, y1, angle, depth){
        var branchArmLength = 15;
        angle_deviation = 20
        if (depth != 0){
            var x2 = x1 + (Math.cos(angle*(Math.PI/180.0)) * depth *
branchArmLength);
            var y2 = y1 + (Math.sin(angle*(Math.PI/180.0)) * depth *
branchArmLength);

            line(context, x1, y1, x2, y2, depth);
            branch(context, x2, y2, angle - angle_deviation, depth - 1);
            branch(context, x2, y2, angle + angle_deviation, depth - 1);
        }
    }
    function line(context, x1, y1, x2, y2, thickness){
        context.strokeStyle = "black"
        context.lineWidth = thickness * 1.5;
        context.beginPath();
        context.moveTo(x1,y1);
        context.lineTo(x2, y2);
        context.closePath();
        context.stroke();
    }

</script>

<style type="text/css">
  canvas { border: 1px solid black; }
</style>

</head>

<body>
  <canvas id="canvas" width="1500" height="1000"></canvas>
  <script type="text/javascript">
  draw();
```

```
  </script>
</body>

</html>
```

Now lets add some randomness in branch angle and lets add some color to the leaves

HTML Fractal Tree – with leave color and different branch angles

```
<html>
<head>
<script type="text/javascript">
    function draw(){
      var canvas = document.getElementById('canvas');
      var context = canvas.getContext('2d');
      context.beginPath();
      context.rect(0,0,1500,1000);
      context.fillStyle = "white";
      context.fill();
      drawTree(context);
    }

    function drawTree(context){
        branch(context, 100, 900, -90, 5);
        branch(context, 200, 900, -90, 8);
        branch(context, 300, 900, -90, 4);
        branch(context, 350, 900, -90, 7);
        branch(context, 500, 900, -90, 10);
        branch(context, 900, 900, -90, 11);
        branch(context, 1200, 900, -90, 13);
    }
    function branch(context, x1, y1, angle, depth){
        var branchArmLength = random(0, 20);
        if (depth != 0){
            var x2 = x1 + (Math.cos(angle*(Math.PI/180.0)) * depth *
branchArmLength);
            var y2 = y1 + (Math.sin(angle*(Math.PI/180.0)) * depth *
branchArmLength);

            line(context, x1, y1, x2, y2, depth);
            branch(context, x2, y2, angle - random(15,20), depth - 1);
            branch(context, x2, y2, angle + random(15,20), depth - 1);
        }
        else{

        var x2 = x1 + (Math.cos(angle*(Math.PI/180.0)) * depth * branchArmLength);
        var y2 = y1 + (Math.sin(angle*(Math.PI/180.0)) * depth * branchArmLength);
        context.fillStyle = 'rgb(0,200,0)';
        context.arc(x2, y2, random(0,10), 0, 2 * Math.PI, false);
        context.fill();
        }
    }
    function line(context, x1, y1, x2, y2, thickness){
        context.strokeStyle = "black"
        context.lineWidth = thickness * 1.5;
        context.beginPath();
        context.moveTo(x1,y1);
        context.lineTo(x2, y2);
        context.closePath();
        context.stroke();
    }
```

```
    function random(min, max){
        return min + Math.floor(Math.random()*(max+1-min));
    }

</script>

<style type="text/css">
  canvas { border: 1px solid black; }
</style>

</head>

<body>
  <canvas id="canvas" width="1500" height="1000"></canvas>
  <script type="text/javascript">
  draw();
  </script>
</body>

</html>
```

Lets add a little more randomness to the leaves colors

More randomness added to the leaves color

```
function branch(context, x1, y1, angle, depth){
        var branchArmLength = random(0, 20);
                var leafColorArray =
['rgb(204,255,204)','rgb(0,153,0)','rgb(255,255,102)'];
                var randomLeafColor = leafColorArray[random(0,leafColorArray.length-
1)];
        if (depth != 0){
            var x2 = x1 + (Math.cos(angle*(Math.PI/180.0)) * depth *
branchArmLength);
            var y2 = y1 + (Math.sin(angle*(Math.PI/180.0)) * depth *
branchArmLength);

            line(context, x1, y1, x2, y2, depth);
            branch(context, x2, y2, angle - random(15,20), depth - 1);
            branch(context, x2, y2, angle + random(15,20), depth - 1);
        }
        else{

        var x2 = x1 + (Math.cos(angle*(Math.PI/180.0)) * depth * branchArmLength);
        var y2 = y1 + (Math.sin(angle*(Math.PI/180.0)) * depth * branchArmLength);
                context.fillStyle = randomLeafColor;
        context.arc(x2, y2, random(0,10), 0, 2 * Math.PI, false);
        context.fill();
        }
    }
```

You can also check the Fractal Tree Implementation in Swift here