Module 2: Creating web pages with HTML5

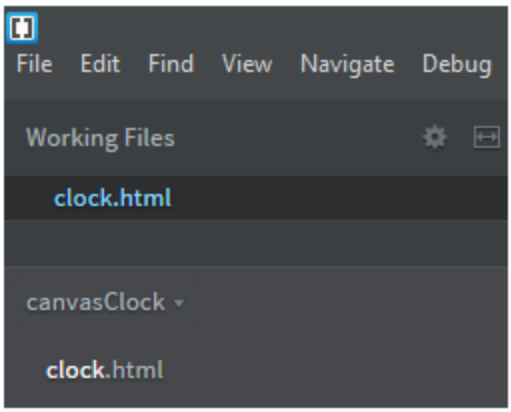
Example

Creating HTML5 Canvas Clock

HTML5 canvas gives a lot of capability to draw graphics on a web page. We will build a clock using HTML5 canvas.

HTML5 Canvas Clock

So lets get started, below is the project structure in Adobe Brackets IDE. We just have an HTML file(clock.html).




```
<html>

<head>
  <title>HTML5 Canvas Clock</title>
</head>

<body background="clock.jpg">
     <canvas id="canvas" width="500" height="500"></canvas>
</body>

</html>
```

clock.html

Below is the screenshot of what your clock.html will look like

We haven't drawn anything on the canvas yet, we just defined a canvas in our html page with width and height set to 500.

So lets start drawing, we will write some JavaScript code.

Note that hours handle completes full circle which is 360 degree in 12 hours. So every hour, hour handle should move 30 degree (360/12= 30).

Minutes handle completes full circle which is 360 degree in 1 hour. So every minute, minute handle should move 6 degree (360/60=6).

Seconds handle completes full circle which is 360 degree in 1 minute. So every second, second handle should move 6 degree (360/60=6).

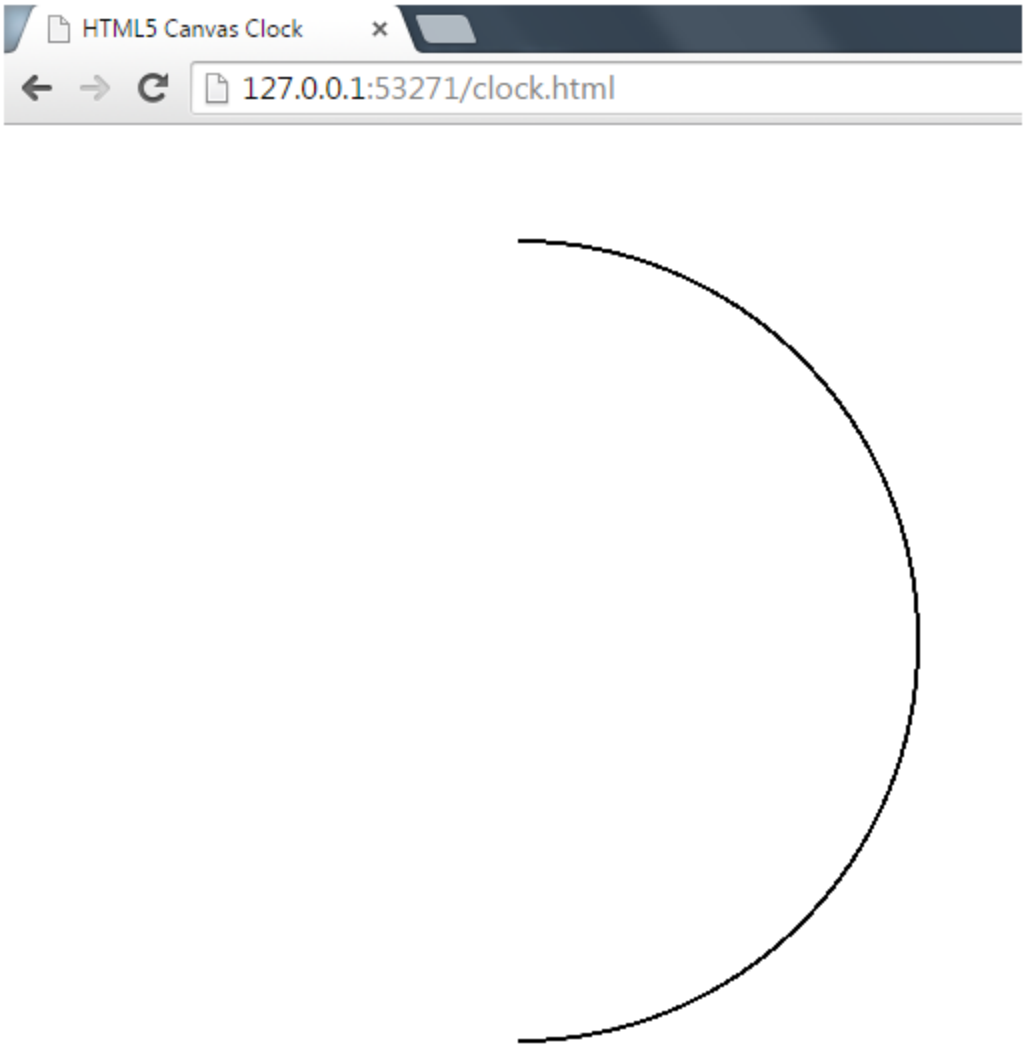
We have defined a JavaScript method degToRad() which takes degrees and converts it to radian. JavaScript will return 24 clock time so if time is 1 PM it will return 13 as the value of hour. To handle that we have subtracted 12 if hours values is greater than 12.

```
<body>
    <canvas id="canvas" width="500" height="500"></canvas>
     <script>
     var canvas = document.getElementById('canvas');
     var ctx = canvas.getContext('2d');
      //this function will convert degree to radian
    function degToRad(degree) {
      var factor = Math.PI/180;
      return degree*factor;
     }

    function showTime(){
        var now = new Date();
        var today = now.toDateString();
        var time = now.toLocaleTimeString();
        var hours = now.getHours();
        if(hours>12){
            hours=hours-12;
        }
        var minutes = now.getMinutes();
        var seconds = now.getSeconds();
        var milliseconds = now.getMilliseconds();
        //we will use newSeconds to give a smooth transition while seconds arc moves
        var newSeconds = seconds+ (milliseconds/1000);

       //Creating Hours Circle
       ctx.beginPath();
       ctx.arc(250, 250, 200, degToRad(270), degToRad((hours*30)-90));
       ctx.stroke();
    }
      //showTime method will be called after 40 milliseconds
     setInterval(showTime, 40);
    </script>
</body>
```

Below is the screenshot of clock.html page after adding JavaScript code



Note that we have set starting angle to be 270 and ending angle to be (hours*15)-90 . Minus 90 is required as we are calculating the angle from the top (270 degree) not 0 degree.

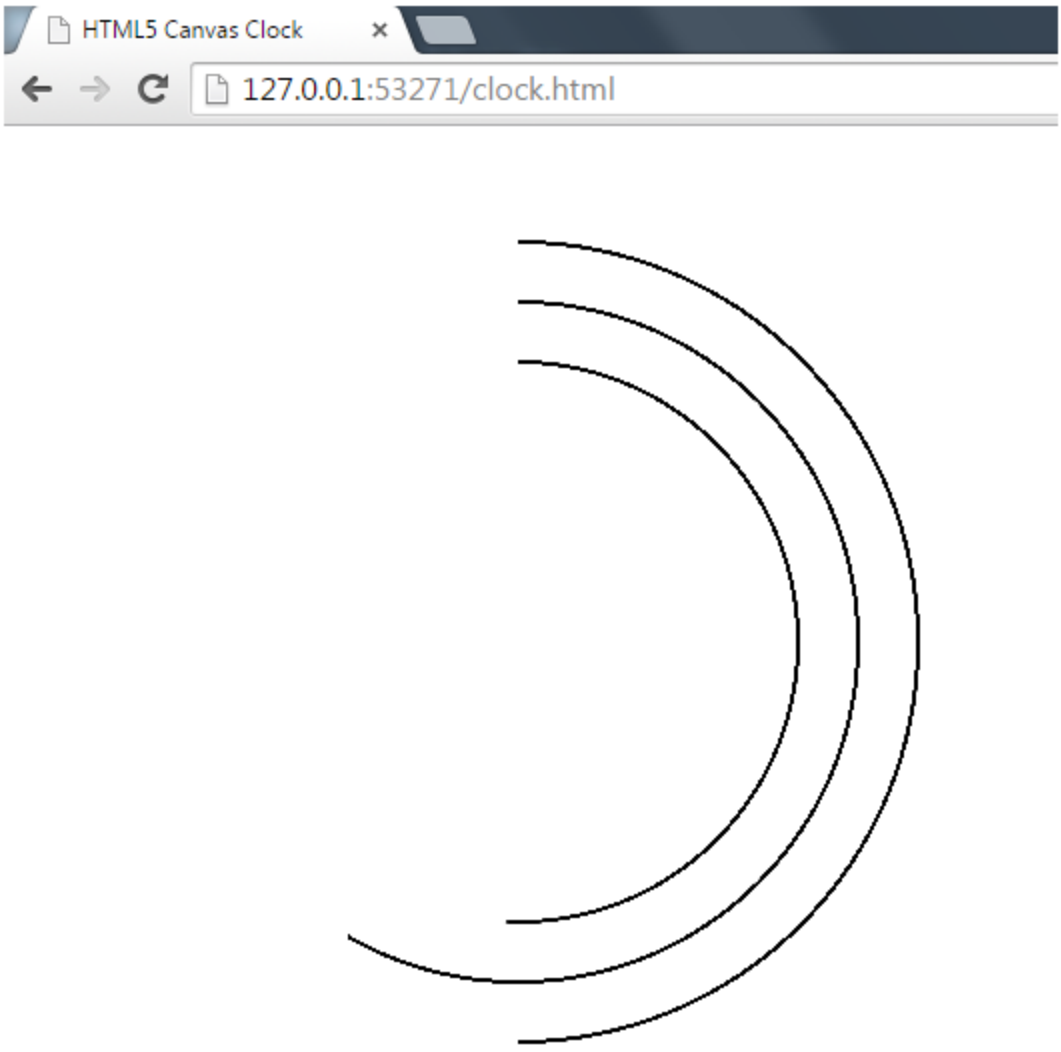
Lets write code for creating Minutes and Seconds circles

```
//Creating Hours Circle
ctx.beginPath();
ctx.arc(250, 250, 200, degToRad(270), degToRad((hours*30)-90));
ctx.stroke();

//Creating Minutes Circle
ctx.beginPath();
ctx.arc(250, 250, 170, degToRad(270), degToRad((minutes*6)-90));
ctx.stroke();

//Creating Seconds Circle
ctx.beginPath();
ctx.arc(250, 250, 140, degToRad(270), degToRad((newSeconds*6)-90));
ctx.stroke();
```

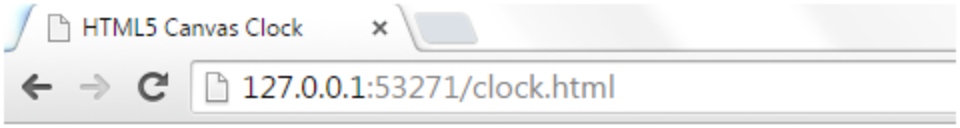
Below is the screenshot of the clock.html after minutes and seconds circle

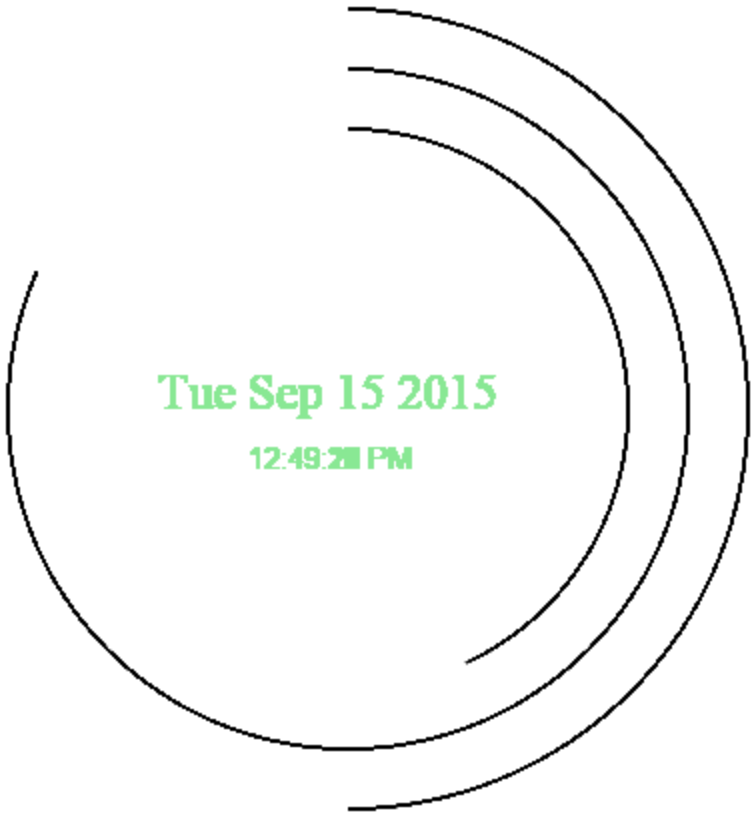


Next we are going to show current date and time at the center of the clock

```
 // Date
ctx.font = "25px Arial bold";
ctx.fillStyle = '#89E894';
ctx.fillText(today, 155, 250);

// Time
ctx.font = "15px Arial";
ctx.fillStyle = '#89E894';
ctx.fillText(time, 200, 280);
```

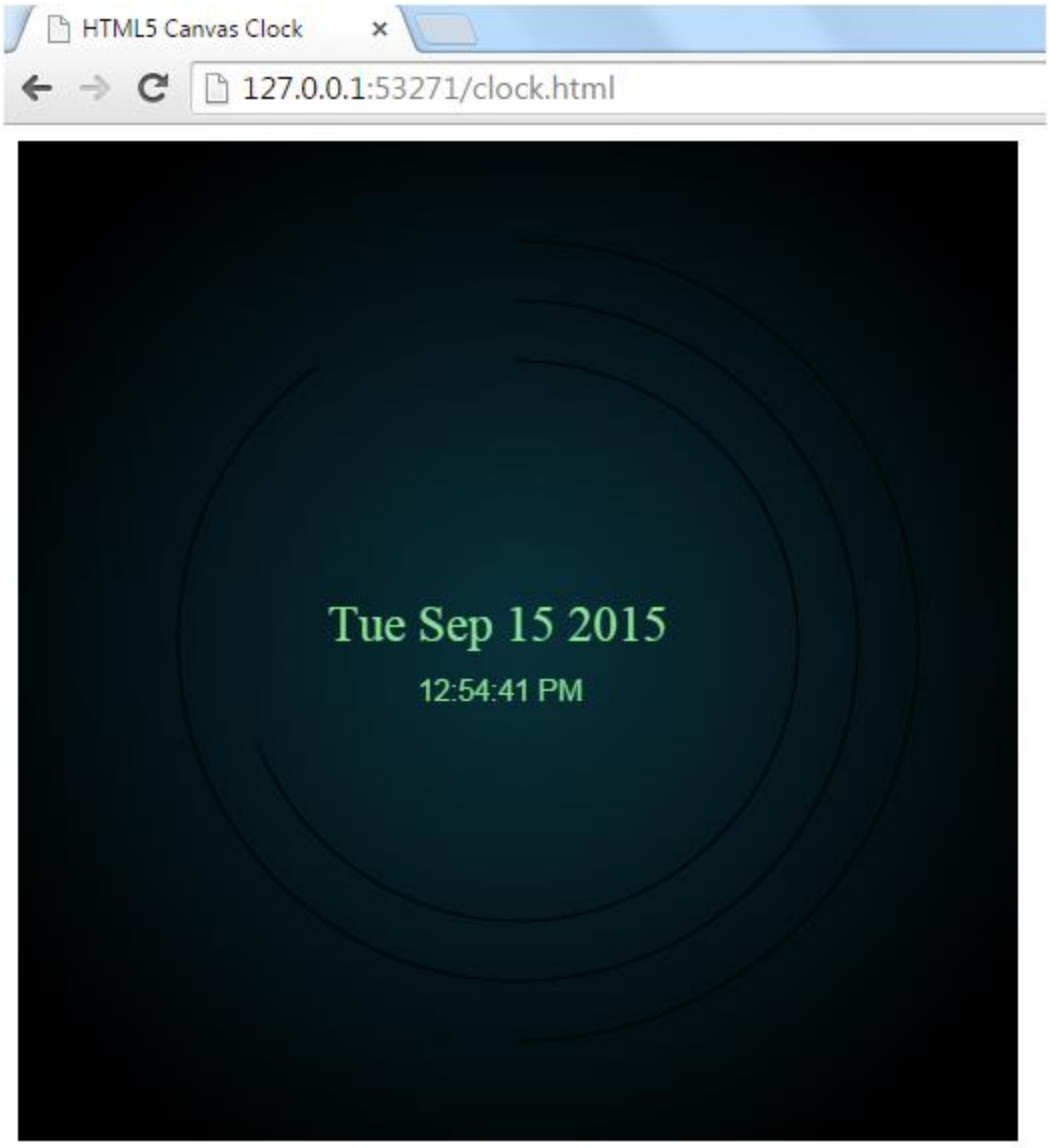




Next we are going to style our clock a bit by adding background gradient to it.

```
  // Background
gradient = ctx.createRadialGradient(250,250,5,250,250,300);
gradient.addColorStop(0,'#09303a');
gradient.addColorStop(1, '#000000');
ctx.fillStyle = gradient;
ctx.fillRect(0,0,500,500);
```

After setting clock background gradient ,the clock will look like as below

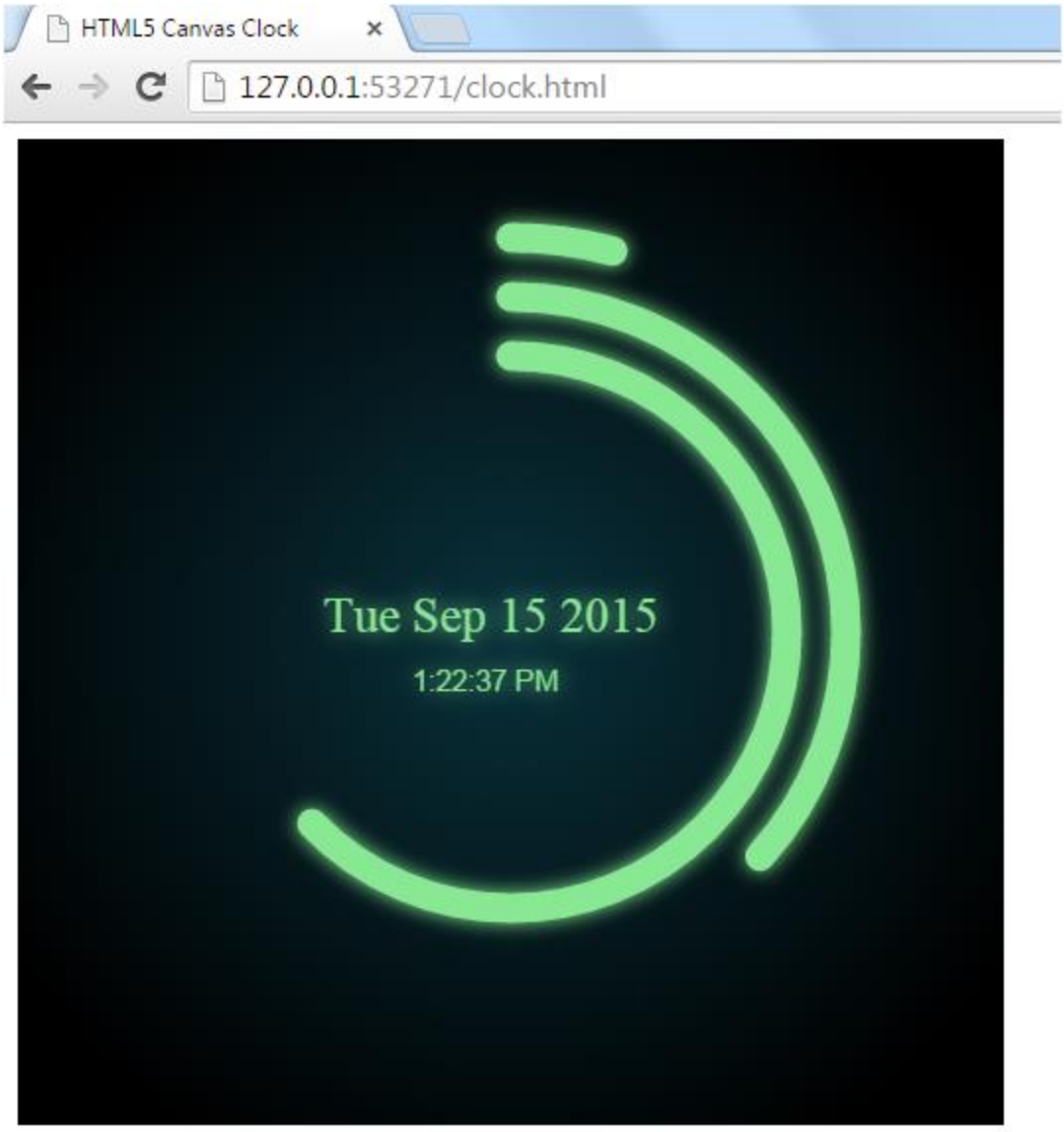


Note that its hard to see the hour, minute and second circles. So we will change the color for the canvas stroke and increase the line width.

```
ctx.strokeStyle = '#89E894';
ctx.lineWidth = 15;
ctx.lineCap = 'round';
ctx.shadowBlur = 15;
ctx.shadowColor = '#89E894';
```

Above we have set the lineWidth to 15 and strokeStyle to light green. Note that we have used lineCap property to round the circle arcs.

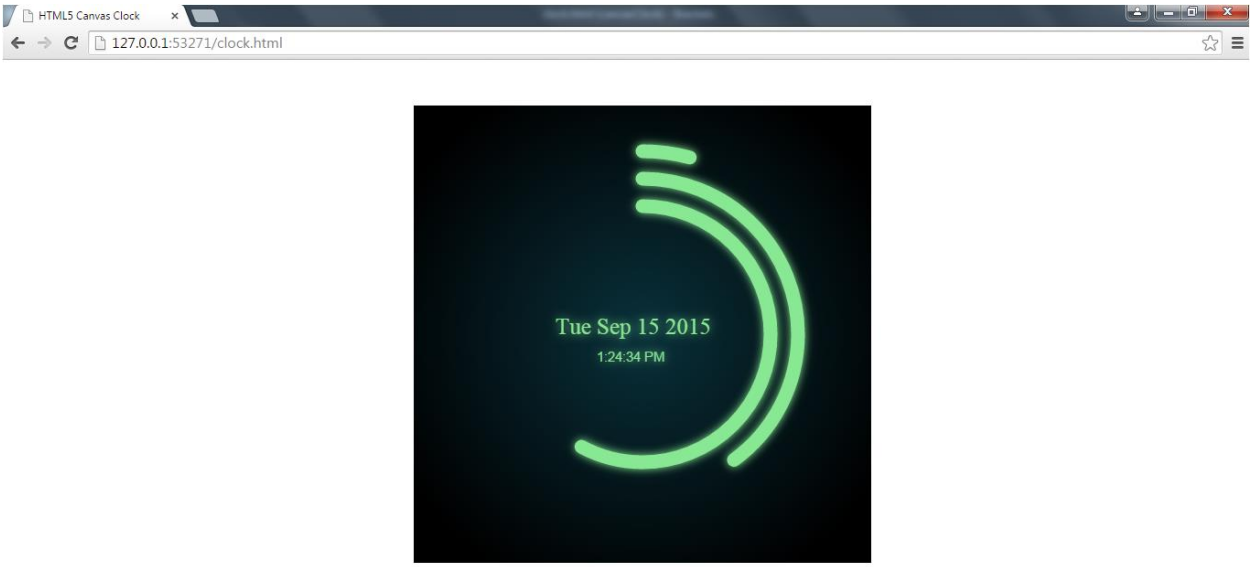
Below is how your clock will look like



Next we are going to position the canvas.

```
<div style="position:absolute; top:50px; left:450px; z-index:1">
 <canvas id="canvas" width="500" height="500"></canvas>
</div>
```

Below is the screenshot of the clock after positioning the canvas



Lets put a background image for clock.html page. After putting background image your clock will look like as shown below