# Week 5 HTML.

## Working with HTML5 Canvas:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>HTML5 Exercises</title>
    <script src="lab5Exercises.js"></script>
</head>
<body>
  <canvas id="myCanvas" height="640" width="480"></canvas>

<script>
    var exerciseNo = prompt( "Enter exercise number (1-4):");
    lab5Exercises( parseInt( exerciseNo ) );
</script>
</body>
</html>
```

lab5.html

```
// jde, added 28/10/17, solutions to exercises covering HTML5 exercise
var lab5Exercises = function( exNo ) {

    function ex1() {

        function drawShapes(){
            // 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');

                // Filled triangle
                ctx.beginPath();
                ctx.moveTo(25,25);
                ctx.lineTo(105,25);
                ctx.lineTo(25,105);
                ctx.fill();

                // Stroked triangle
                ctx.beginPath();
                ctx.moveTo(125,125);
                ctx.lineTo(125,45);
                ctx.lineTo(45,125);
                ctx.closePath();
                ctx.stroke();
            }

            else {
                alert('Update your browser, eg Safari or Firefox 1.5+ to see
this demo.');
            }
        }

        drawShapes();
    }
```

```
function ex2() {

    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');

            for (i=0;i<10;i++){
                ctx.lineWidth = 1+i;
                ctx.beginPath();
                ctx.moveTo(5+i*14,5);
                ctx.lineTo(5+i*14,140);
                ctx.stroke();
            }

        }

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

    drawShape();
};

function ex3() {
    function drawSquare(canvasId, startX, startY, length) {
        // get the canvas element using the DOM
        var canvas = document.getElementById(canvasId);
        // 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(startX, startY);
            ctx.lineTo(startX, startY + length);
            ctx.lineTo(startX + length, startY + length);
            ctx.lineTo(startX + length, startY);
            ctx.lineTo(startX, startY);
            ctx.stroke();
        }
    }

    function drawRectangle(canvasId, startX, startY, length, width) {
        // get the canvas element using the DOM
        var canvas = document.getElementById(canvasId);
        // 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(startX, startY);
            ctx.lineTo(startX, startY + width);
            ctx.lineTo(startX + length, startY + width);
            ctx.lineTo(startX + length, startY);
            ctx.lineTo(startX, startY);
            ctx.stroke();
        }
    }
```

```
function drawEquiLateralTriangle(canvasId, startX, startY, length) {
    // get the canvas element using the DOM
    var canvas = document.getElementById(canvasId);
    // 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(startX, startY);
        ctx.lineTo(startX, startY + length);
        ctx.lineTo(startX + length, startY + (length / 2.0) );
        ctx.lineTo(startX, startY);
        ctx.stroke();
    }
}

function drawCircle( canvasId, startX, startY, radius) {
    // get the canvas element using the DOM
    var canvas = document.getElementById(canvasId);

    // 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(startX, startY - (radius / 2.0 ));

        var x, y, angle;
        for (i = 0; i < 360; i++) {
            angle = ( i * Math.PI ) / 180.0;
            x = startX + ( radius * Math.cos( angle ) );
            y = startY + ( radius * Math.sin( angle ) );
            ctx.lineTo(x, y);
        }
        ctx.stroke();
    }
}

function drawLines( canvasId ) {
    // get the canvas element using the DOM
    var canvas = document.getElementById(canvasId);

    // 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');

        for (i = 0; i < 10; i++) {
            ctx.linelength = 1 + i;
            ctx.beginPath();
            ctx.moveTo(5 + i * 14, 5);
            ctx.lineTo(5 + i * 14, 140);
            ctx.stroke();
        }
    }

}

function drawShapes(canvasId) {
    var canvas = document.getElementById(canvasId);
    // Make sure we don't execute when canvas isn't supported
    if (canvas.getContext) {
        drawLines(canvasId);
        drawSquare(canvasId, 120, 120, 150);
```

```
                drawRectangle(canvasId, 320, 120, 150, 70);
                drawEquiLateralTriangle(canvasId, 320, 20, 80 );
                drawCircle( canvasId, 120, 120, 45 );
            } else {
                alert("Canvas isn't supported");
            }
        }

        drawShapes( "myCanvas");
    };

    function ex4() {
        function geoLocation() {
            navigator.geolocation.getCurrentPosition(
                function(position) {
                    alert( position.coords.latitude + ", " +
position.coords.longitude );
                    var loc = position.coords.latitude + ", " +
position.coords.longitude;
                    console.log( loc );
                },
                function(error) { alert("call failed"); }
            );
        }

        geoLocation();
    }

    console.log( "Attempting Exercise:" + exNo );
    switch ( exNo ) {
        case 1 :
            ex1();
            break;
        case 2 :
            ex2();
            break;
        case 3 :
            ex3();
            break;
        case 4 :
            ex4();
            break;
    }
};
```

lab5Exercises.js

That's all for now but keep practicing.