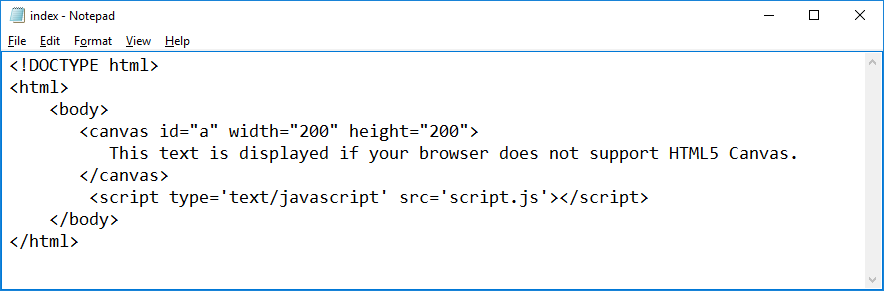
**Javascript Drawing Commands**

Javascript is a web-based version of the Java programming language. It is used to handle higher-end function through a web page interface with the user.

A javascript can be either directly embedded into your .html document or linked, similar to CSS.

Let’s look at a web page that utilizes a javascript:

**index.html**



This is the name of the javascript file

There are two new tags that we notice, <canvas> and <script>

<canvas>

This tag is used to define the size of the area in which the drawing will appear. We also specify an id (“a”), which will be referenced in our javascript file. This allows us to create multiple canvases on our web pages that can link to different javascripts. However, we will only work with a single canvas in our web pages.

<script>

This tag is used to tell the html page the name of the javascript file to load. Javascript files end in **.js** In the above code, the name of the javascript file is *script.js*.

Javascript – First two lines

var a\_canvas = document.getElementById("a");

var draw = a\_canvas.getContext("2d");

These two command will always begin your javascript drawing file. The connect the script to the canvas id you set up on your HTML page (“a”) and create a tool called “draw” that you will use for all of your drawing commands.

**Setting the Color**

draw.fillStyle="#FF0000"; - Sets the fill color of the object drawn

draw.strokeStyle="#FF0000"; - Sets the color for outlined shapes

**Drawing Lines**

draw.beginPath(); Starts a new series of drawing commands.

draw.moveTo(20, 20); Points to an (X, Y) location within the canvas, not the web page.

draw.lineTo(100, 20); Will draw a line from the current location to the (X, Y) location specified. You can use several lineTo statement to create polygons.

draw.stroke(); Actually draws the line(s) that you defined.

Draw.fill(); Draws the lines and fills in the area inside of them.

draw.closePath(); Ends this group of draw commands. Will cause the last point to reconnect back to the first.

**Drawing Rectangles**

draw.strokeRect(X, Y, width, height); Creates a rectangle at location (X, Y) with size (width, height) (no fill)

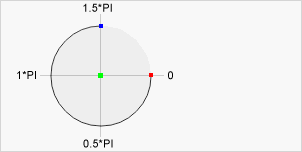
draw.fillRect(20,20,150,100); Same as strokeRect, only the rectangle will be filled in.

draw.clearRect(50, 50, 20, 25); Clears out part of another rectangle

**Drawing Arcs**

draw.beginPath();

draw.arc(X, Y, radius, startAngle, endAngle); Draws an arc centered around point(X, Y) that extends from the starting angle to the ending angle, with a radius of *radius*

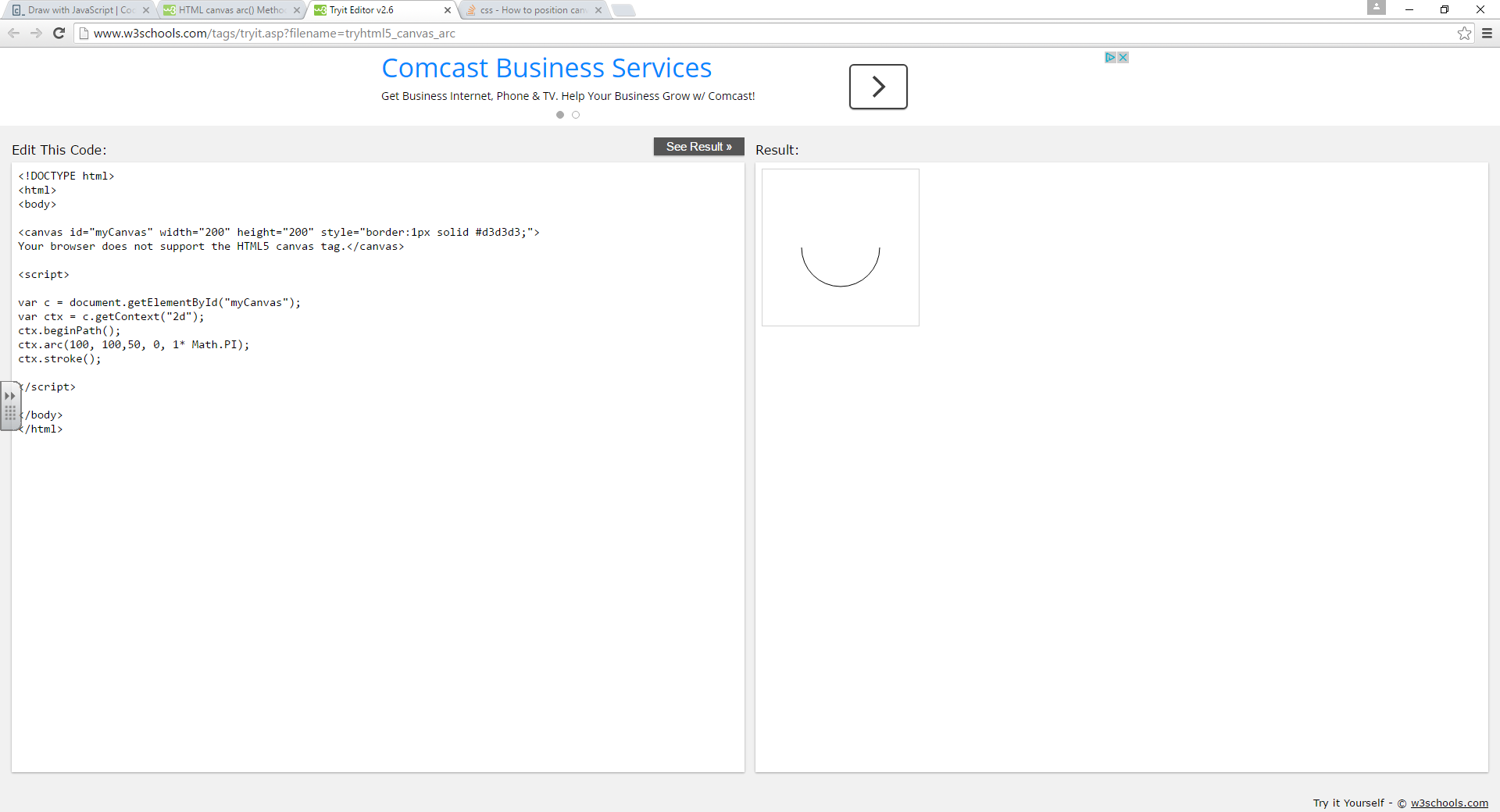
draw.stroke();

Angles range from 0 to 2 \* PI. Use the image to right to assist you.

Example:

draw.beginPath();

draw.arc(100, 100, 50, 0, 1\*Math.PI);

draw.stroke();

For a canvas of size 200 x 200, the result would look like this:

\*Note: Using draw.fill() instead of draw.stroke would fill in the arc.

**Drawing Text**

draw.font = "30px Arial";

draw.fillText("Hello World",10,50);

draw.strokeText(“Hello World”, 10, 50);

Now create these drawings using javascript. Create a new html page for each with the same filename. They are worth 20 points each.

target.js android.js greenlantern.js



