Displaying the Circle

Ok, now that you know all about the `arc` function and how you can mentally draw your circle, it's time to draw it for realz. Let's say that that you have some starting code that looks as follows:

```
var mainCanvas = document.getElementById("myCanvas");
var mainContext = mainCanvas.getContext("2d");

var canvasWidth = mainCanvas.width;
var canvasHeight = mainCanvas.height;

function draw() {

}
draw();
```

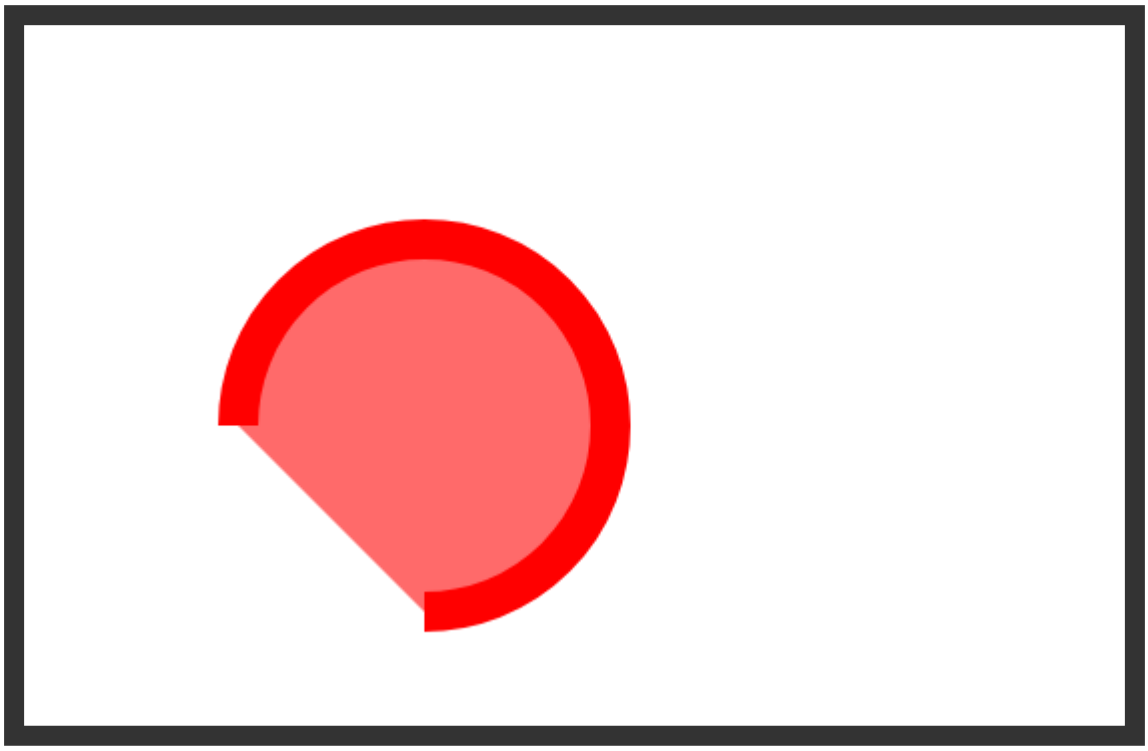
This code just takes care of just getting your `canvas` (whose id value is **myCanvas**) prepped for drawing content. There is nothing fancy going on here, so let's fix that. Modify your `draw` function by adding the following code:

HTML JavaScript

javascript

```
var mainCanvas = document.getElementById("myCanvas");
var mainContext = mainCanvas.getContext("2d");

var canvasWidth = mainCanvas.width;
var canvasHeight = mainCanvas.height;

function draw() {
    // draw the colored region
    mainContext.beginPath();
    mainContext.arc(200, 200, 93, Math.PI / 2, Math.PI, true);
    mainContext.fillStyle = '#FF6A6A';
    mainContext.fill();

    // draw the stroke
    mainContext.lineWidth = 20;
    mainContext.strokeStyle = '#FF0000';
    mainContext.stroke();
}

draw();
```

output



We've already seen this weird semi-circle in the previous section, it is nice to see it being drawn from actual code as opposed to just plain English words. Before we wrap things up, let's end (ironically) by looking at the code for how to draw a full circle:

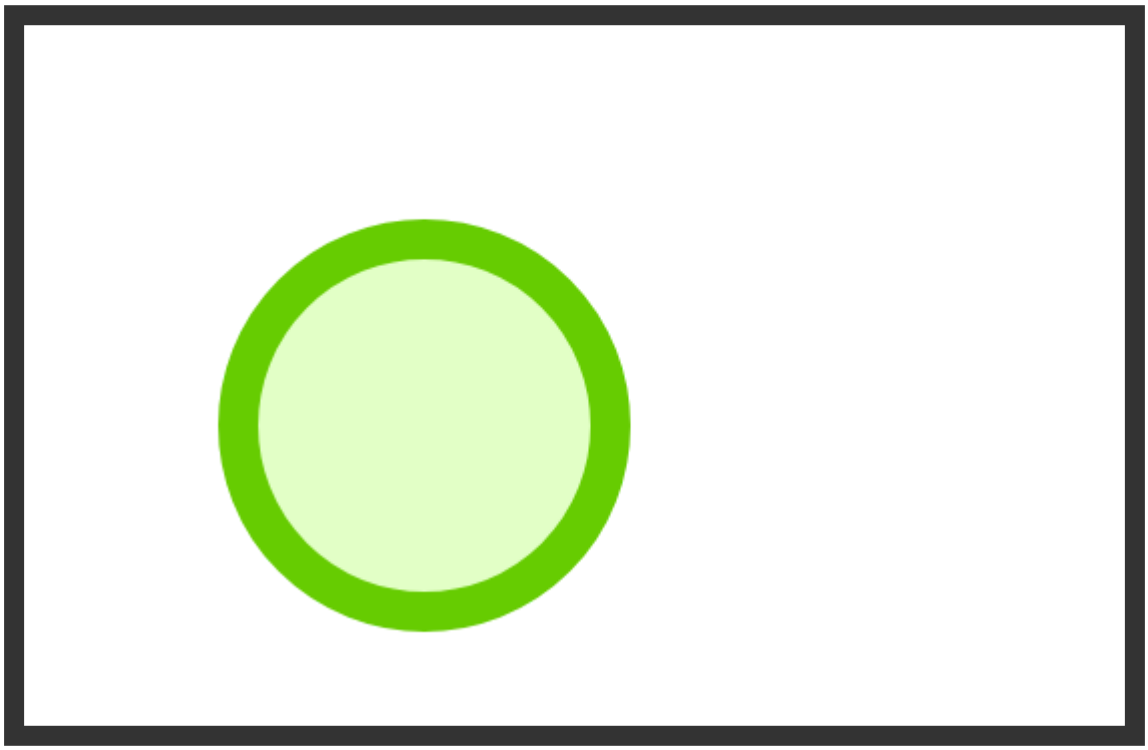
HTML JavaScript

javascript

```
var mainCanvas = document.getElementById("myCanvas");
var mainContext = mainCanvas.getContext("2d");

var canvasWidth = mainCanvas.width;
var canvasHeight = mainCanvas.height;

function draw() {
    // draw the colored region
    mainContext.beginPath();
    mainContext.arc(200, 200, 93, 0, 2 * Math.PI, true);
    mainContext.fillStyle = "#E2FFC6";
    mainContext.fill();

    // draw the stroke
    mainContext.lineWidth = 20;
    mainContext.strokeStyle = "#66CC01";
    mainContext.stroke();
}
```

```
20  draw();
```

output



You can tell by looking at the arguments we passed in to the `arc` method for why that is. The **startAngle** value is 0, and the **endAngle** value is 2π. We don't leave any room for any spoiled radians to go off and do something crazy.