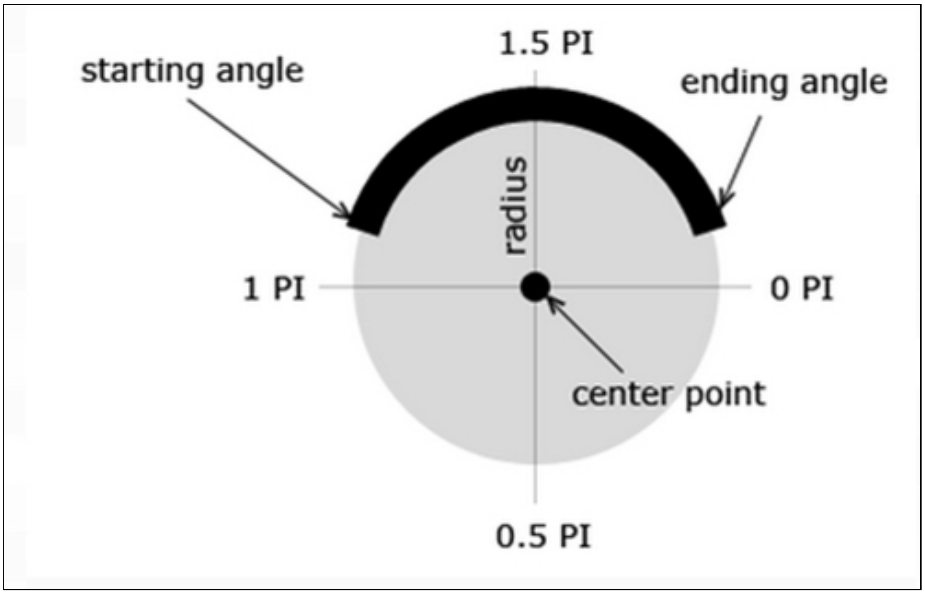
Drawing circles and arcs

The `ctx.arc(cx, cy, radius, startAngle, endAngle, drawInverse)` method is useful for drawing arcs of circles. It takes the center of the circle/arc, its radius, the starting angle of the arc (turning clockwise), the ending angle of the arc, and an optional parameter we will talk about later.

Note: figures borrowed from http://www.html5canvastutorials.com/tutorials/html5-canvas-arcs/



TYPICAL USAGE

Typical usage for drawing an arc/circle/ellipse is:

```
ctx.arc(centerX, centerY, radius, startAngle, endAngle); // clockwise drawing
ctx.arc(centerX, centerY, radius, startAngle, endAngle, false);
```

The angles are in radians (between `0` and `2*Math.PI`). The arc is drawn clockwise. Beware that this may not seem natural if you're used to the trigonometric order.

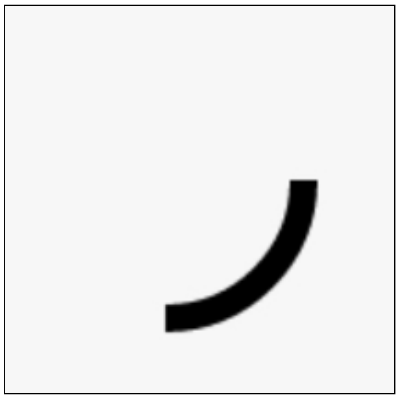
The last parameter is optional and has a value of `false` by default. If `true`, instead of drawing an arc of circle that corresponds to the parameters, it will draw its complementary. See the examples below to see the difference.

EXAMPLE 1: DRAWING AN ARC WITH RADIUS = 50, STARTING ANGLE = 0, END ANGLE = PI/2

Try this example online: http://jsbin.com/tikite/1/edit

```
ctx.beginPath();
// we ommited the last parameter
ctx.arc(100, 75, 50, 0, Math.PI/2);

ctx.lineWidth = 10;
ctx.stroke();
```

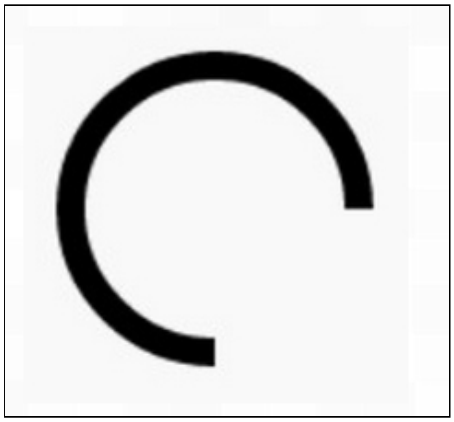
Here is the result:



If we change the last parameter (we omitted it, so it took a value of `false` by default):

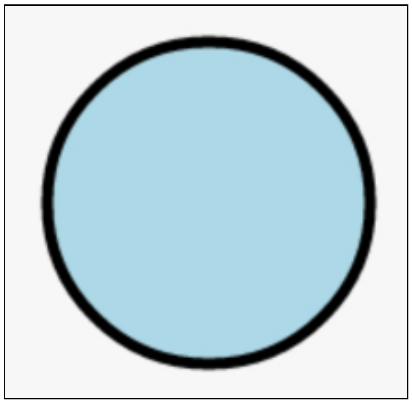
```
ctx.beginPath();
// we ommited the last parameter
ctx.arc(100, 75, 50, 0, Math.PI/2, true);

ctx.lineWidth = 10;
ctx.stroke();
```

Then, the result is the "complementary" of the previous arc:



EXAMPLE 2: DRAWING A FULL CIRCLE (FILLED + OUTLINED)

Try this example: http://jsbin.com/gazuba/2/edit



Source code:

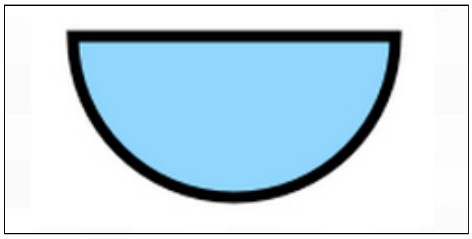
```
    var canvas = document.getElementById("myCanvas");
    var ctx = canvas.getContext("2d");
    var centerX = canvas.width / 2;
    var centerY = canvas.height / 2;
    var radius = 70;
    ctx.beginPath();
    // Add to the path a full circle (from 0 to 2PI)
10. ctx.arc(centerX, centerY, radius, 0, 2*Math.PI, false);
    // With path drawing you can change the context
    // properties until a call to stroke() or fill() is performed
    ctx.fillStyle = "lightBlue";
    // Draws the filled circle in light blue
    ctx.fill();
     // Prepare for the outline
    ctx.lineWidth = 5;
```

```
20.  ctx.strokeStyle = "black";
     // draws AGAIN the path (the circle), this
     // time in wireframe
     ctx.stroke();
     // Notice we called context.arc() only once ! And drew it twice
     // with different styles
```

Notice that we called `ctx.arc()` only once! And drew it twice, with different styles, with calls to `ctx.stroke()` and `ctx.fill()`. Each call drew the defined path in wireframe and in filled mode!

PROPOSED PROJECTS

Project 1 - modify the previous example on http://jsbin.com/gazuba/2/edit in order to get:



Project 2 - make a small program that draws a smiling head like that (or make something better!)

KNOWLEDGE CHECK 3.4.8 (NOT GRADED)

```
ctx.beginPath();
ctx.moveTo(100, 100);
ctx.lineTo(200, 200);
```

```
ctx.arc(500, 500, 100, 0, 2*Math.PI);
ctx.stroke();
```

Will the circle above be connected to the last extremity of the line drawn from (100, 100) to (200, 200)?

- Yes
- no