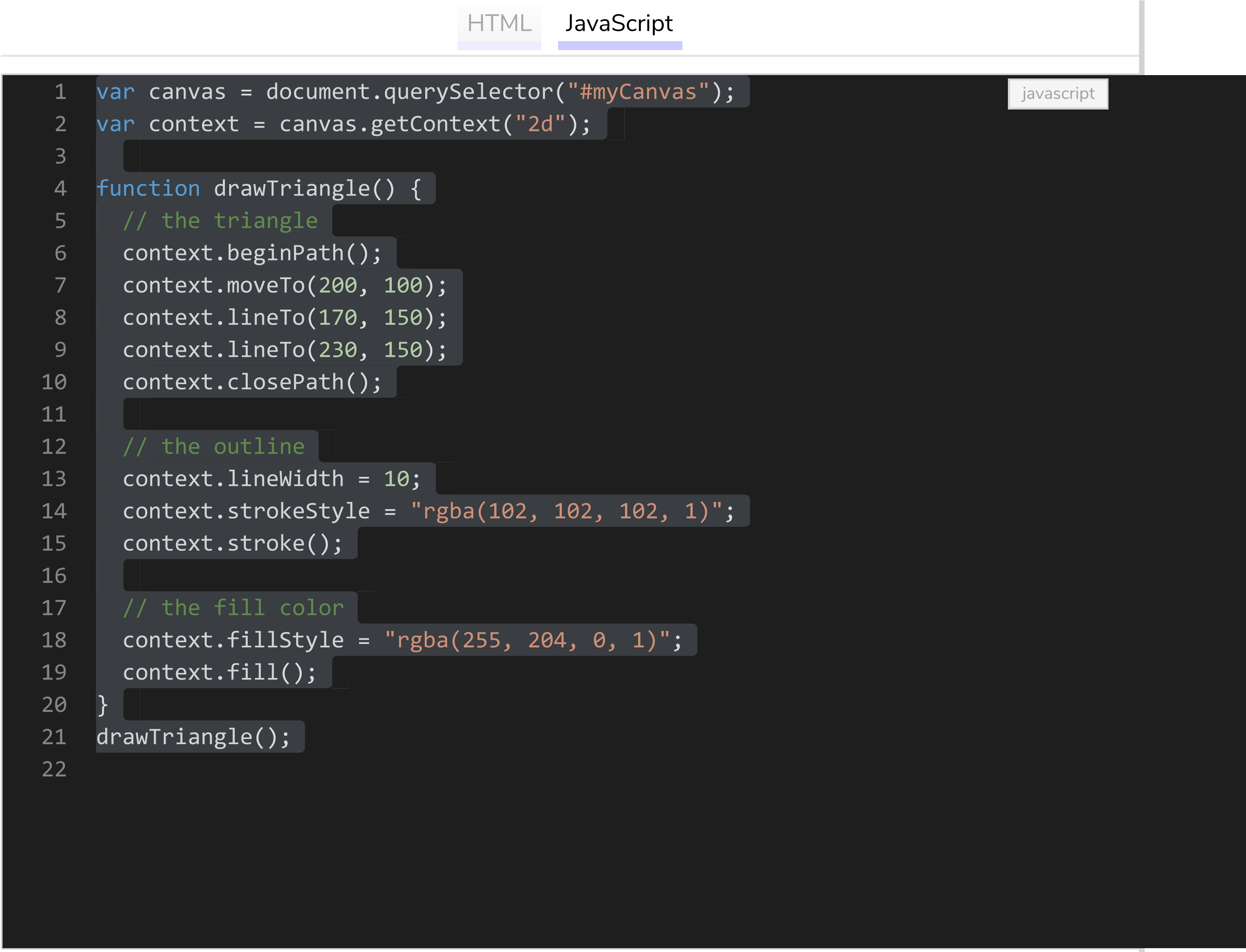
Displaying Our Triangle

Let’s start easy by first drawing our triangle. The way we are going to do that is by defining a function called `drawTriangle` that draws a triangle at a fixed position in our `canvas`.

Using our usual example where we have a `canvas` element defined with an `id` value of **myCanvas**, ensure the contents of your `script` tag look as follows:



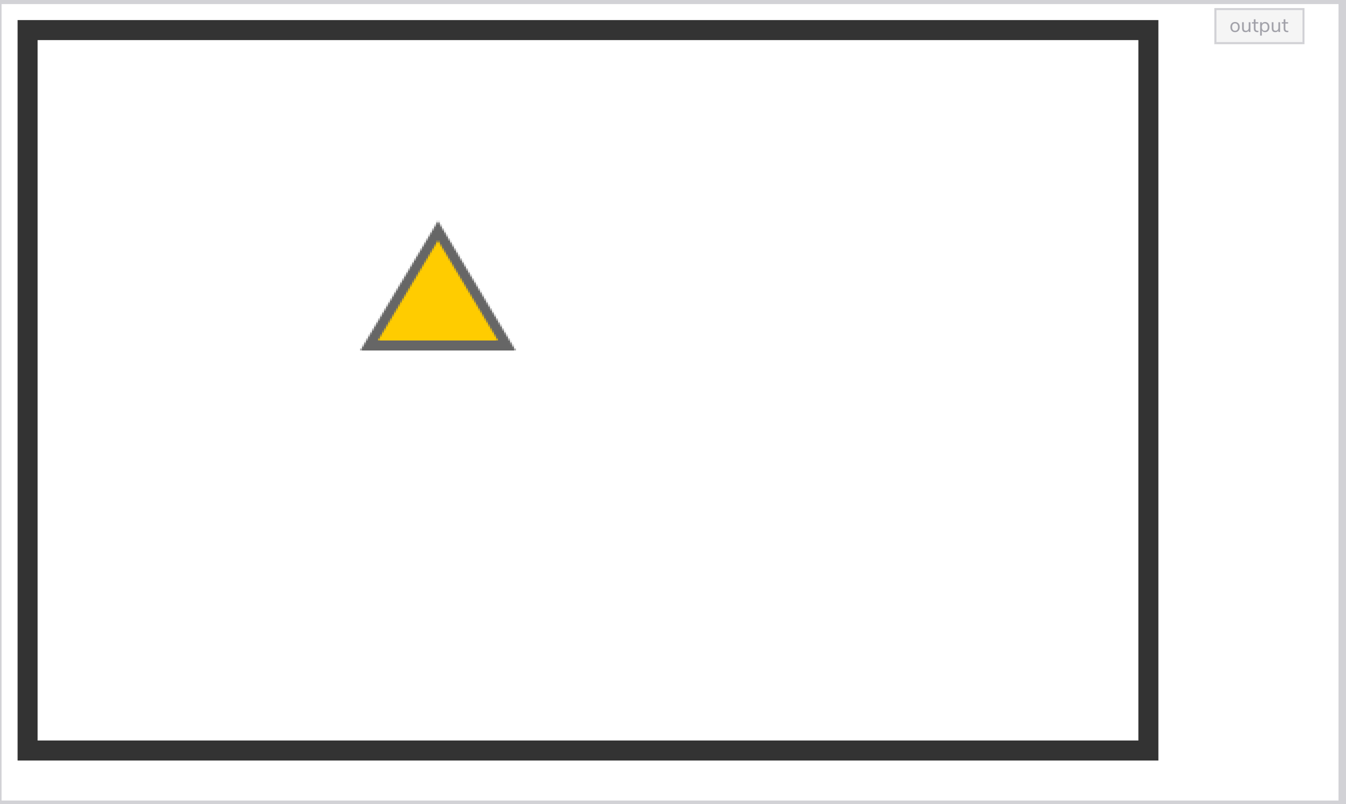
```javascript
var canvas = document.querySelector("#myCanvas");
var context = canvas.getContext("2d");

function drawTriangle() {
  // the triangle
  context.beginPath();
  context.moveTo(200, 100);
  context.lineTo(170, 150);
  context.lineTo(230, 150);
  context.closePath();

  // the outline
  context.lineWidth = 10;
  context.strokeStyle = "rgba(102, 102, 102, 1)";
  context.stroke();

  // the fill color
  context.fillStyle = "rgba(255, 204, 0, 1)";
  context.fill();
}
drawTriangle();

```



Once you have added this code to your document, go ahead and preview your document in your browser. If everything worked out properly, you'll see a yellow triangle displayed. There is nothing exciting going on with this code that you haven't seen before, but there is one thing I want to call out. Notice that our triangle is defined by the following X and Y values:

```
context.moveTo(200, 100);
context.lineTo(170, 150);
context.lineTo(230, 150);
```

It is these values that we'll eventually end up adjusting to accommodate our arrow key presses.