Using Functions

As you probably know, functions help make your code reusable. They also ensure your code isn't duplicated unnecessarily throughout your app. You place the code you want to re-use inside a function, and that function is what you tell the rest of your app to call. That is pretty straighforward.

When it comes to drawing stuff on the `canvas`, the biggest advantage of functions is that you can pass arguments that your drawing code then uses to customize what gets drawn. Before I bore you any further, here is the full code for the circles example:

HTML   JavaScript

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

for (var i = 0; i < 40; i++) {
  var r = Math.round(15 + Math.random() * 150);
  drawCircle(r);
}

function drawCircle(radius) {
  var xPos = Math.round(Math.random() * myCanvas.width);
  var yPos = Math.round(Math.random() * myCanvas.height);

  context.beginPath();
  context.arc(xPos, yPos, radius, 0, 2 * Math.PI, true);
  context.fillStyle = "rgba(41, 170, 255, .1)";
  context.fill();
}
```

Take a few moments to walk through this code and see how the various pieces of it work. The biggest chunk of code (and the star) is the `drawCircle` function:

```
function drawCircle(radius) {
   var xPos = Math.round(Math.random() * myCanvas.width);
   var yPos = Math.round(Math.random() * myCanvas.height);

   context.beginPath();
   context.arc(xPos, yPos, radius, 0, 2 * Math.PI, true);
   context.fillStyle = "rgba(41, 170, 255, .1)";
   context.fill();
}
```

This function neatly contains everything our `canvas` needs to get a circle drawn on the screen. It takes care of defining the circle, placing it in a random position on the screen, setting the size using the passed in argument for the radius, and ultimately drawing the whole thing with a semi-transparent blue color.

We call this function from inside the `for` loop just a few lines earlier:

```
for (var i = 0; i < 40; i++) {
   var r = Math.round(15 + Math.random() * 150);
```

```
    drawCircle(r);
}
```

This loop runs 40 times. Each time it runs, it calls the `drawCircle` function with a random value passed in for the radius. At the end of each of these calls, a blue circle gets placed somewhere on the screen. If we wanted to customize what gets drawn further, we can modify the `drawCircle` function and add more arguments really easily. If we ever added more entry points that would draw more circles, we can just call `drawCircle` directly without duplicating any code. Gotta love functions!