Dynamically Sizing Shapes

13 min

So far, we've only been using static values as arguments when drawing shapes. However, you can use dynamic values to create drawings that adjust to a canvas's width and height. With dynamic values, you don't have to keep track of the exact values of your arguments—they will automatically adjust for you!

Before using variables for our drawings, it's helpful to know that drawing function arguments can use *operators* like `+`, `-`, `*`, and `/`, to calculate the values. Consider the following:

```
ellipse(40, 100, 40, 60);
```

Above code can be rewritten using operators:

```
ellipse(20 + 20, 120 - 20, 20 * 2, 120/2);
```

This might not seem like much now, but in a bit, you will see how powerful using operators can be.

p5.js also allows you to use *variables* as function arguments. The p5.js library provides useful built-in variables, including `width` and `height` that store the width and height of the p5.js canvas.

Consider this ellipse drawn using static values:

```
function setup(){
   createCanvas(400, 400);
}

function draw(){
   ellipse(200, 200, 400, 400);
}
```

If you were to resize the canvas while keeping the same dimensions for the ellipse, you would have to change the values of the `createCanvas()` function and then change the four arguments of the `ellipse()` function.

Instead, you can provide built-in variables as arguments to the `ellipse()` function, so that values will automatically change based on the canvas dimensions.

```
function setup(){
   createCanvas(400, 400);
```

```
}

function draw() {
  ellipse(width/2, height/2, width, height);
}
```

The biggest change here is using the `width` and `height` variables to calculate the position and size of the ellipse. `width/2` gives you the horizontal center of the canvas, and `height/2` gives you the vertical center. Similarly, `width` returns the full width of the canvas and `height` of the canvas.

Now, to change the size of the canvas and keep the relative size and position of the ellipse, you would only have to change the arguments of the `createCanvas()` function. You don't have to change anything else because the value of the `width` and `height` variables will automatically change.

Instructions

1.

Replace the static width value for all four circles with the `circleWidth` variable.
Hint
Replace the constant width of `50` given as the third argument of the `circle()` function with the `circleWidth` variable.

2.

Reposition each circle so that they are at the center of each quadrant of the canvas. Calculate the positions using the `width` and `height` variables.
Hint
To dynamically calculate the x and y positions using the `width` and `height` variables, think about where the circles should go in proportion to the canvas's width and height. For example, to draw the circle at the center of the top right quadrant, the x coordinate would be three-quarters of the canvas width, and the y coordinate would be one-quarter of the canvas height.

sketch.js

```
let  circleWidth = 100;

function setup() {
  createCanvas(400, 400);
  background(220);

  line(width/2, 0, width/2, height);
```

```
  line(0, height/2, width, height/2);
}

function draw() {

  // TODO: Move circle to top-left quadrant
  circle(width/4, height/4, circleWidth);

  // TODO: Move circle to top-right quadrant
  circle(3*width/4, height/4, circleWidth);

  // TODO: Move circle to bottom-left quadrant
  circle(width/4, height - height/4, circleWidth);

  // TODO: Move circle to bottom-right quadrant
  circle(width - width/4, height - height/4, circleWidth);

}
```