



# Girls Who Code At Home

**Digital Art Rules Part 2**  
Reference Guide

## Digital Art Rules Part 2 Reference Guide

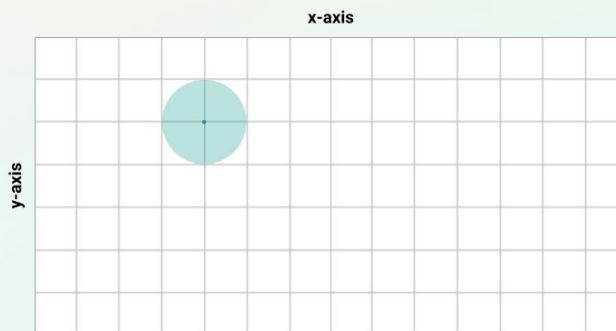


In this document you will find all of the answers to some of the questions in the activity. Follow along with the activity and when you see this icon, stop and check your ideas here.

### Step 3: Write your first function

#### Coordinate System Review

Let's say your friend gives you an instruction to draw a circle on a piece of paper. You could either just draw the circle or ask for more information. If you ask for more information, you might inquire: where on the paper? How big? What color? A perfect circle or more of an oval? To answer the question of where, your friend might say "A third of the way from the left side and three-fourths of the way down towards the bottom." That kind of instruction might work if you are talking to a human and don't need to be specific. But it won't work for a computer. Computers want you to be specific and precise with the rules you give them.



#### The coordinate system

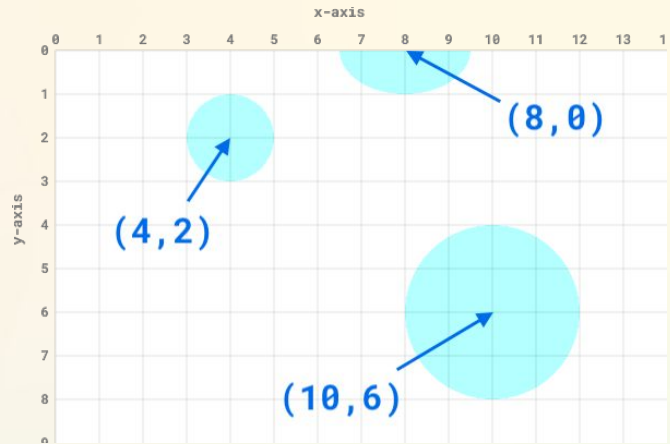
We can use the coordinate system to tell our program where to display an element on our computer screen. The coordinate system is a system that uses one or more numbers to identify the location of a point in space. They can be on a 2D plane or in 3D space. Coordinate planes (i.e. 2D) have an x-axis that runs horizontally and a y-axis that runs vertically to form a grid. They use an ordered pair to signify a point: (x position, y position).

#### Pixel perfect

Each pixel on your screen has a unique address in the coordinate system. In order to draw pixels to the screen, we must give our program the x coordinate (i.e. the location on the x-axis) and the y coordinate (i.e. the location on the y-axis) of the pixel.

### Step 3: Write your first function(cont.)

The origin or (0, 0) on the screen coordinate system is located at the top left corner. As you move right on the screen, the value of the x-coordinate increases. As you move down on the screen the value of the y-coordinate increases. This may appear a little different than the coordinate system layouts you have seen in math class, where the origin is in the center or at the bottom left corner.



### Step 5: Practice translating an example image

The new code added after each section is in teal.

#### Program the square

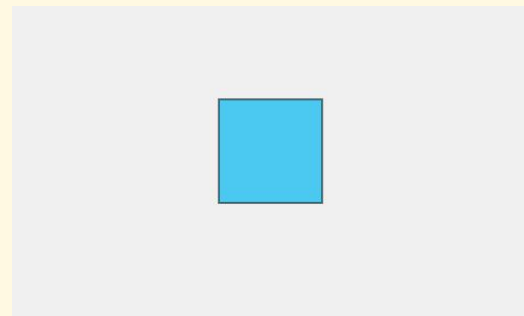
##### CODE

```
// Only runs once
function setup() {
  createCanvas(500, 300);
}

// Runs over and over in a loop
function draw() {
  // Set the background
  background(240);

  // Draw the square
  rectMode(CENTER);
  fill(76, 201, 240);
  rect(250, 140, 100, 100);
```

##### RESULT



## Step 5: Practice translating an example image (cont.)

### Program the circle

#### CODE

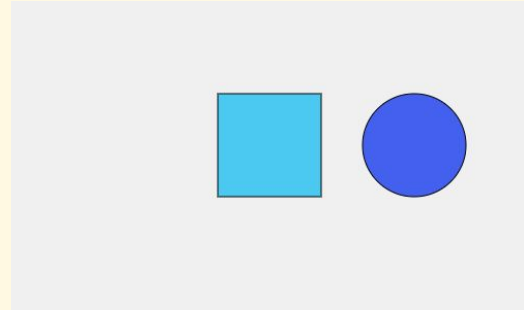
```
// Only draw() is included

function draw() {
  // Set the background
  background(240);

  // Draw the square
  rectMode(CENTER);
  fill(76, 201, 240);
  rect(250,140,100,100);

  // Draw the circle
  fill(67, 97, 238);
  ellipse(390,140,100,100);
```

#### RESULT



### Program the triangle

#### CODE

```
// Only draw() is included

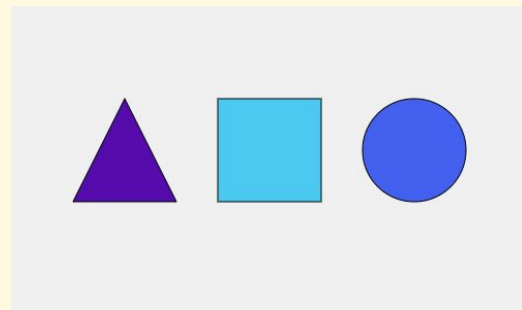
function draw() {
  // Set the background
  background(240);

  // Draw the square
  rectMode(CENTER);
  fill(76, 201, 240);
  rect(250,140,100,100);

  // Draw the circle
  fill(67, 97, 238);
  ellipse(390,140,100,100);

  // Draw the triangle
  fill(86, 11, 173);
  triangle(110,90,160,190,60,190);
```

#### RESULT



## Step 5: Practice translating an example image (cont.)

### Program the line

#### CODE

```
// Only draw() is included

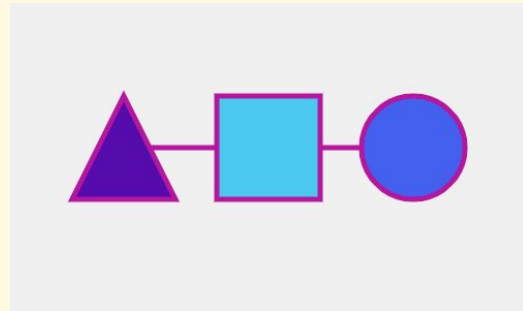
function draw() {
  // Set the background
  background(240);

  // Draw the line
  stroke(181, 23, 158);
  strokeWeight(5);
  line(110,140,390,140);

  // Draw the square
  rectMode(CENTER);
  fill(76, 201, 240);
  rect(250,140,100,100);

  // Draw the circle
  fill(67, 97, 238);
  ellipse(390,140,100,100);
```

#### RESULT



## Step 5: Practice translating an example image (cont.)

### Program the text

#### CODE

```
// Only draw() is included

function draw() {
  // Set the background
  background(240);

  // Draw the line
  stroke(181, 23, 158);
  strokeWeight(5);
  line(110,140,390,140);

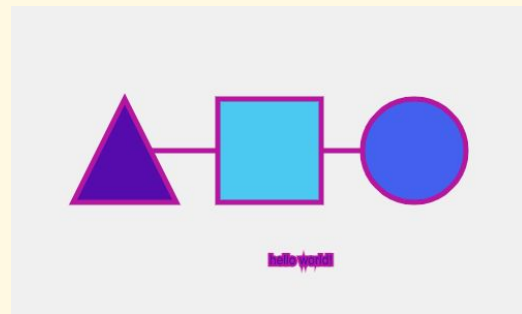
  // Draw the square
  rectMode(CENTER);
  fill(76, 201, 240);
  rect(250,140,100,100);

  // Draw the circle
  fill(67, 97, 238);
  ellipse(390,140,100,100);

  // Draw the triangle
  fill(86, 11, 173);
  triangle(110,90,160,190,60,190);

  // Draw text
  text("hello world!",250,250);
}
```

#### RESULT



## Program the text (cont.)

## CODE

```
// Only draw() is included

function draw() {
  // Set the background
  background(240);

  // Draw the line
  stroke(181, 23, 158);
  strokeWeight(5);
  line(110,140,390,140);

  // Draw the square
  rectMode(CENTER);
  fill(76, 201, 240);
  rect(250,140,100,100);

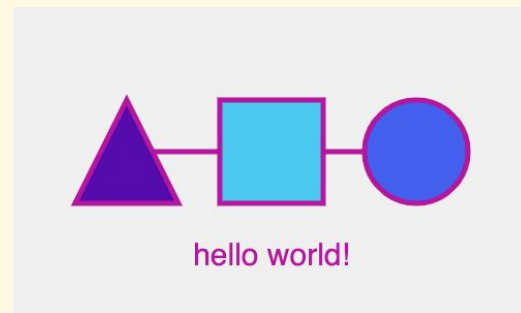
  // Draw the circle
  fill(67, 97, 238);
  ellipse(390,140,100,100);

  // Draw the triangle
  fill(86, 11, 173);
  triangle(110,90,160,190,60,190);

  // Remove outlines from this
  // point forward
  noStroke();

  // Draw text
  fill(181, 23, 158);
  textSize(30);
  textAlign(CENTER);
  text("hello world!",250,250);
}
```

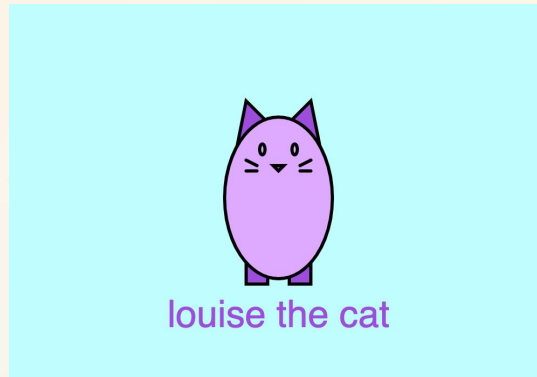
## RESULT



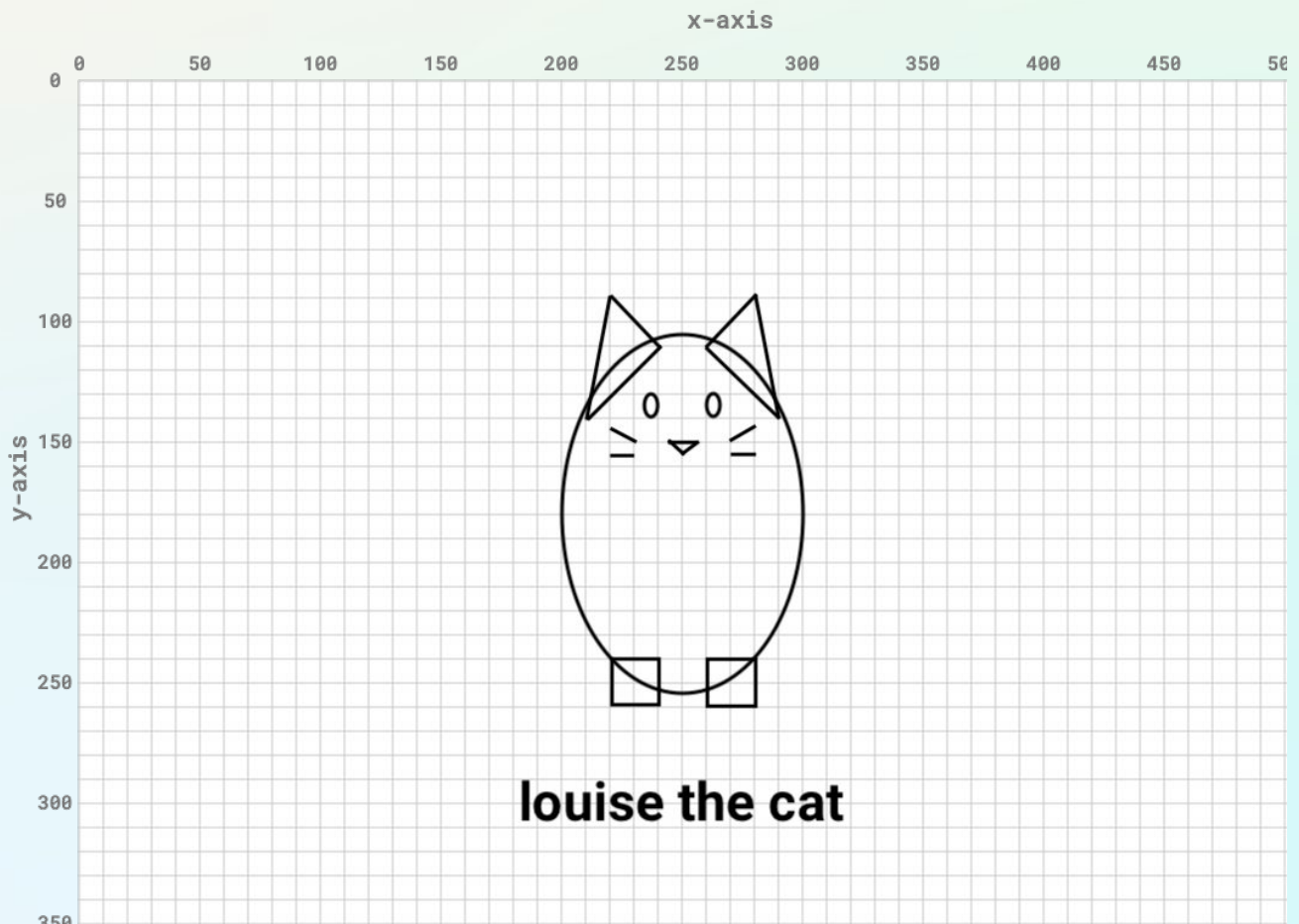
## Step 6: Translate your drawing to p5.js

### Sample Project: Cat (Easy)

If you don't have a drawing or instructions from Part 1, you can use this sample project to complete Part 1. This sample is easier as it includes fewer elements and none of the extensions.



### Coordinate Plane Drawing





## Step 6: Sample project Cat (cont.)

### Instruction Set

ELEMENT	VALUES	COLOR
Background		Light teal: R = 192, G = 253, B = 255
Ellipse	Center = (250,180) Width = 100 Height = 150	Light purple: R = 222, G = 170, B = 255
Left triangle	Point 1 = (260,110) Point 2 = (280,90) Point 3 = (290,140)	Dark purple: R = 157, G = 78, B = 221
Right triangle	Point 1 = (240,110) Point 2 = (220,90) Point 3 = (210,140)	Dark purple
Small ellipse (left)	Center = (235,135) Width = 5 Height = 10	Light purple
Small ellipse (right)	Center = (265,135) Width = 5 Height = 10	Light purple
Small triangle	Point 1 = (245,150) Point 2 = (255,150) Point 3 = (250,155)	Dark purple
Top line (left)	Point 1 = (230,150) Point 2 = (220,145)	Black
Bottom line (left)	Point 1 = (230,155) Point 2 = (220,155)	Black
Top line (right)	Point 1 = (270,150) Point 2 = (280,145)	Black
Bottom line (right)	Point 1 = (270,155) Point 2 = (280,155)	Black
Text	Text = "louise the cat" Center = (250,300)	Dark purple

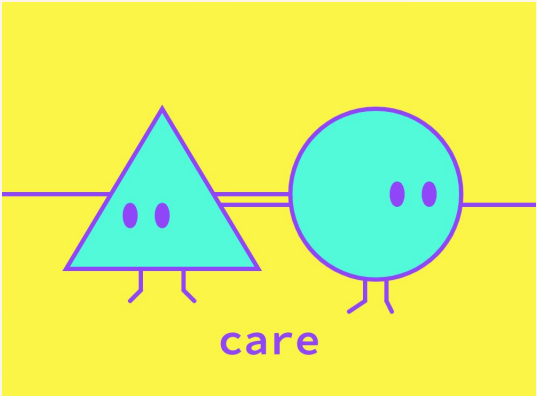
### Sample Code

You should only examine this code **after** completing the activity.

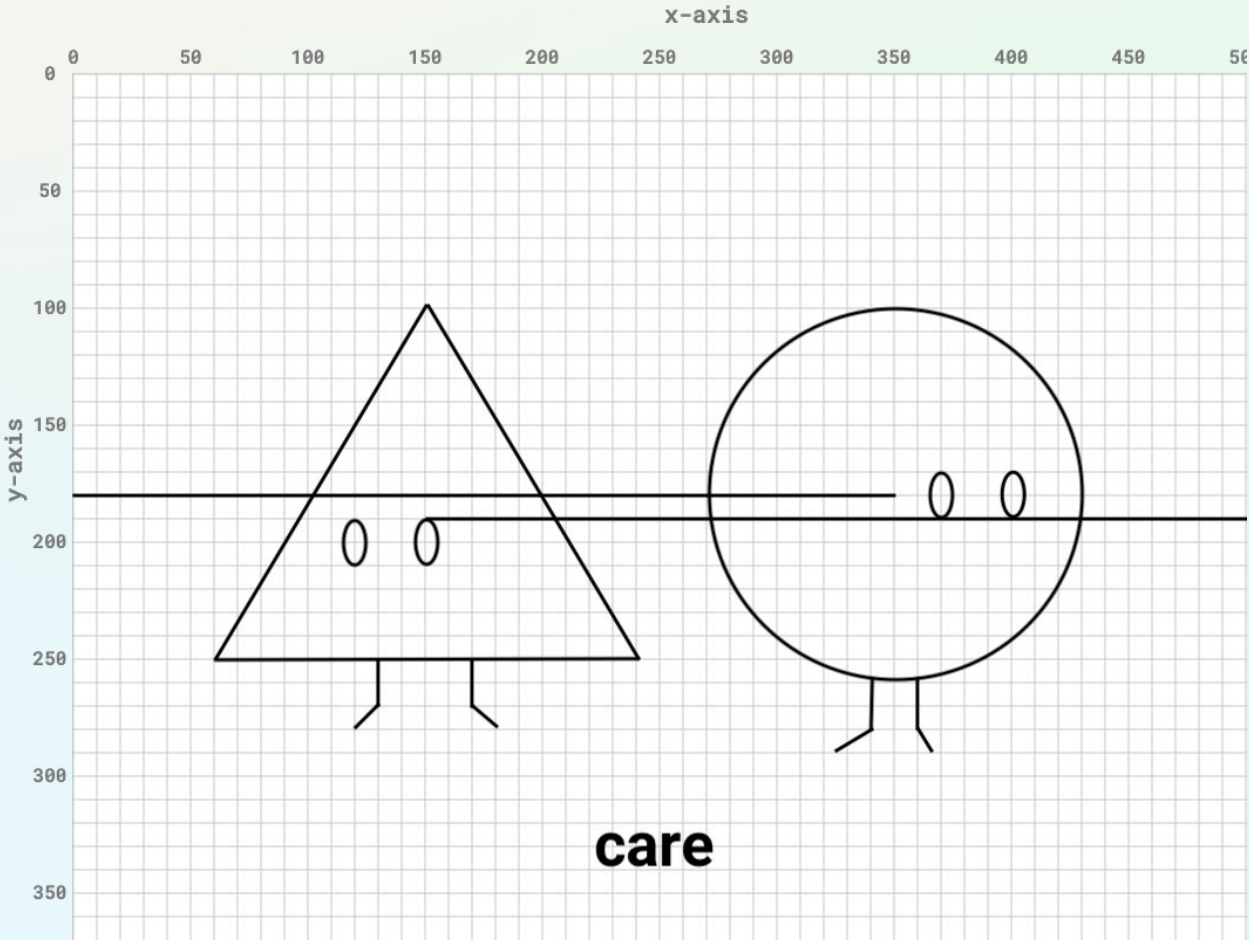
→ [Digital Art Rules Sample Project - Cat](#)

**Sample Project: Care (Challenging)**

If you don't have a drawing or instructions from Part 1, you can use this sample project to complete Part 2. This sample is more challenging as it includes more elements and the extensions.



**Coordinate Plane Drawing**



## Instruction Set

ELEMENT	VALUES	COLOR
Background		Yellow: R = 250, G = 245, B = 70
Top horizontal line	Point 1 = (0,180) Point 2 = (350,180)	Purple: R = 145, G = 70, B = 250
Bottom horizontal line	Point 1 = (150,190) Point 2 = (550,190)	Purple
Triangle	Point 1 = (110,90) Point 2 = (160,190) Point 3 = (60,190)	Teal: R = 82, G = 250, B = 218
Left ellipse (Triangle)	Center = (120,200) Width = 10 Height = 20	Purple
Right ellipse (Triangle)	Center = (150,200) Width = 10 Height = 20	Purple
Left line (Triangle leg)	Point 1 = (130,250) Point 2 = (130,270)	Purple
Left line (Triangle foot)	Point 1 = (130,270) Point 2 = (120,280)	Purple
Right line (Triangle leg)	Point 1 = (170,250) Point 2 = (170,270)	Purple
Right line (Triangle foot)	Point 1 = (170,270) Point 2 = (180,280)	Purple
Circle	Center = (350,180) Width = 160 Height = 160	Teal
Left ellipse (Circle)	Center = (370,180) Width = 10 Height = 20	Purple
Right ellipse (Circle)	Center = (400,180) Width = 10 Height = 20	Purple
Left line (Circle leg)	Point 1 = (360,260) Point 2 = (360,280)	Purple
Left line (Circle foot)	Point 1 = (360,280) Point 2 = (365,290)	Purple
Right line (Circle leg)	Point 1 = (340,260) Point 2 = (340,280)	Purple
Right line (Circle foot)	Point 1 = (340,280) Point 2 = (325,290)	Purple
Text	Text = "care" Center = (250,330)	Purple

## Sample Code

You should only examine this code **after** completing the activity.

→ [Digital Art Rules Sample Project - Care \(with Extensions\)](#)

## Function Reference

SHAPES	
FUNCTION	DESCRIPTION
<pre>rect(x, y, width, height);</pre>	<ul style="list-style-type: none"><li>→ <b>rect</b>: The function name. <i>To learn more, see the <a href="#">rect()</a> entry in the p5.js Reference</i></li><li>→ <b>()</b>: We use parentheses to tell our program that it needs to call the function. Sometimes we include parameters or inputs in the function inside our parentheses.</li><li>→ <b>x</b>: The x-coordinate of the rectangle.</li><li>→ <b>y</b>: The y-coordinate of the rectangle.</li><li>→ <b>width</b>: Sets the width of the ellipse in pixels.</li><li>→ <b>height</b>: Sets the height of the ellipse in pixels.</li><li>→ <b>,</b>: We use commas to separate the different parameters or inputs in the functions.</li><li>→ <b>;</b>: All lines of code in p5.js must end with a semicolon.</li></ul>
<pre>rectMode(CENTER);</pre>	<ul style="list-style-type: none"><li>→ <b>rect</b>: The function name. <i>To learn more, see the <a href="#">rectMode()</a> entry in the p5.js Reference.</i></li><li>→ <b>()</b>: We use parentheses to tell our program that it needs to call the function. Sometimes we include parameters or inputs in the function inside our parentheses.</li><li>→ <b>CENTER</b>: Interprets the first two parameters as the shape's center point. Note that this is in all caps and is case sensitive.</li><li>→ <b>;</b>: All lines of code in p5.js must end with a semicolon.</li></ul>
<pre>ellipse(x, y, width, height);</pre>	<ul style="list-style-type: none"><li>→ <b>ellipse</b>: The function name. Ellipse is another word for oval. <i>To learn more, see the <a href="#">ellipse()</a> entry in the p5.js Reference</i></li><li>→ <b>()</b>: We use parentheses to tell our program that it needs to call the function. Sometimes we include parameters or inputs in the function inside our parentheses.</li><li>→ <b>x</b>: The x-coordinate at the center of the ellipse.</li><li>→ <b>y</b>: The y-coordinate at the center of the ellipse.</li><li>→ <b>width</b>: Sets the width of the ellipse in pixels.</li><li>→ <b>height</b>: Sets the height of the ellipse in pixels.</li><li>→ <b>,</b>: We use commas to separate the different parameters or inputs in the functions.</li><li>→ <b>;</b>: All lines of code in p5.js must end with a semicolon.</li></ul>

## Function Reference (cont.)

SHAPES (CONT.)	
FUNCTION	DESCRIPTION
<pre>triangle(x1, y1, x2, y2,          x3, y3);</pre>	<ul style="list-style-type: none"> <li>→ <b>triangle</b>: The function name. To learn more, see the <a href="#">triangle() entry in the p5.js Reference</a></li> <li>→ <b>()</b>: We use parentheses to tell our program that it needs to call the function. Sometimes we include parameters or inputs in the function inside our parentheses.</li> <li>→ <b>x1</b>: The x-coordinate of the first point.</li> <li>→ <b>y1</b>: The y-coordinate of the first point.</li> <li>→ <b>x2</b>: The x-coordinate of the second point.</li> <li>→ <b>y2</b>: The y-coordinate of the second point.</li> <li>→ <b>x3</b>: The x-coordinate of the third point.</li> <li>→ <b>y3</b>: The y-coordinate of the third point.</li> <li>→ <b>,</b>: We use commas to separate the different parameters or inputs in the functions.</li> <li>→ <b>;</b>: All lines of code in p5.js must end with a semicolon.</li> </ul>
<pre>quad(x1, y1, x2, y2, x3,      y3, x4, y4);</pre> <p><i>Note that the points should be listed clockwise or counterclockwise</i></p>	<ul style="list-style-type: none"> <li>→ <b>quad</b>: The function name. To learn more, see the <a href="#">quad() entry in the p5.js Reference</a></li> <li>→ <b>()</b>: We use parentheses to tell our program that it needs to call the function. Sometimes we include parameters or inputs in the function inside our parentheses.</li> <li>→ <b>x1</b>: The x-coordinate of the first point.</li> <li>→ <b>y1</b>: The y-coordinate of the first point.</li> <li>→ <b>x2</b>: The x-coordinate of the second point.</li> <li>→ <b>y2</b>: The y-coordinate of the second point.</li> <li>→ <b>x3</b>: The x-coordinate of the third point.</li> <li>→ <b>y3</b>: The y-coordinate of the third point.</li> <li>→ <b>x4</b>: The x-coordinate of the fourth point.</li> <li>→ <b>y4</b>: The y-coordinate of the fourth point.</li> <li>→ <b>,</b>: We use commas to separate the different parameters or inputs in the functions.</li> <li>→ <b>;</b>: All lines of code in p5.js must end with a semicolon.</li> </ul>

## Function Reference (cont.)

LINE	
FUNCTION	DESCRIPTION
<code>line(x1, y1, x2, y2);</code>	<ul style="list-style-type: none"> <li>→ <b>line</b>: The function name. <i>To learn more, see the <a href="#">line()</a> entry in the p5.js Reference</i></li> <li>→ <b>()</b>: We use parentheses to tell our program that it needs to call the function. Sometimes we include parameters or inputs in the function inside our parentheses.</li> <li>→ <b>x1</b>: The x-coordinate of the first point.</li> <li>→ <b>y1</b>: The y-coordinate of the first point.</li> <li>→ <b>x2</b>: The x-coordinate of the second point.</li> <li>→ <b>y2</b>: The y-coordinate of the second point.</li> <li>→ <b>,</b>: We use commas to separate the different parameters or inputs in the functions.</li> <li>→ <b>;</b>: All lines of code in p5.js must end with a semicolon.</li> </ul>
<code>strokeWeight(weight);</code>	<ul style="list-style-type: none"> <li>→ <b>strokeWeight</b>: The function name. <i>To learn more, see the <a href="#">strokeWeight()</a> entry in the p5.js Reference</i></li> <li>→ <b>()</b>: We use parentheses to tell our program that it needs to call the function. Sometimes we include parameters or inputs in the function inside our parentheses.</li> <li>→ <b>weight</b>: The width of the stroke used for lines, points and the border around shapes. All widths are set in units of pixels.</li> <li>→ <b>;</b>: All lines of code in p5.js must end with a semicolon.</li> </ul>
<code>noStroke();</code>	<ul style="list-style-type: none"> <li>→ <b>noStroke</b>: The function name. <i>To learn more, see the <a href="#">noStroke()</a> entry in the p5.js Reference</i></li> <li>→ <b>()</b>: We use parentheses to tell our program that it needs to call the function. Sometimes we include parameters or inputs in the function inside our parentheses.</li> <li>→ <b>,</b>: We use commas to separate the different parameters or inputs in the functions.</li> <li>→ <b>;</b>: All lines of code in p5.js must end with a semicolon.</li> </ul>

## Function Reference (cont.)

COLOR	
FUNCTION	DESCRIPTION
<pre>background(redValue, greenValue, blueValue);</pre>	<ul style="list-style-type: none"> <li>→ <b>background</b>: The function name. <i>To learn more, see the <a href="#">background()</a> entry in the p5.js Reference</i></li> <li>→ <b>()</b>: We use parentheses to tell our program that it needs to call the function. Sometimes we include parameters or inputs in the function inside our parentheses.</li> <li>→ <b>redValue</b>: The red value between 0 and 255.</li> <li>→ <b>greenValue</b>: The green value between 0 and 255.</li> <li>→ <b>blueValue</b>: The blue value between 0 and 255.</li> <li>→ <b>;</b>: All lines of code in JavaScript must end with a semicolon.</li> </ul>
<pre>fill(redValue, greenValue, blueValue);</pre>	<ul style="list-style-type: none"> <li>→ <b>fill</b>: The function name. <i>To learn more, see the <a href="#">fill()</a> entry in the p5.js Reference</i></li> <li>→ <b>()</b>: We use parentheses to tell our program that it needs to call the function. Sometimes we include parameters or inputs in the function inside our parentheses.</li> <li>→ <b>redValue</b>: The red value between 0 and 255.</li> <li>→ <b>greenValue</b>: The green value between 0 and 255.</li> <li>→ <b>blueValue</b>: The blue value between 0 and 255.</li> <li>→ <b>;</b>: All lines of code in p5.js must end with a semicolon.</li> </ul>
<pre>stroke(redValue, greenValue, blueValue);</pre>	<ul style="list-style-type: none"> <li>→ <b>stroke</b>: The function name. <i>To learn more, see the <a href="#">stroke()</a> entry in the p5.js Reference</i></li> <li>→ <b>()</b>: We use parentheses to tell our program that it needs to call the function. Sometimes we include parameters or inputs in the function inside our parentheses.</li> <li>→ <b>redValue</b>: The red value between 0 and 255.</li> <li>→ <b>greenValue</b>: The green value between 0 and 255.</li> <li>→ <b>blueValue</b>: The blue value between 0 and 255.</li> <li>→ <b>;</b>: All lines of code in p5.js must end with a semicolon.</li> </ul>

## Function Reference (cont.)

TEXT	
FUNCTION	DESCRIPTION
<code>text(string, x, y);</code>	<ul style="list-style-type: none"> <li>→ <b>text</b>: The function name. <i>To learn more, see the <a href="#">text()</a> entry in the p5.js Reference</i></li> <li>→ <b>()</b>: We use parentheses to tell our program that it needs to call the function. Sometimes we include parameters or inputs in the function inside our parentheses.</li> <li>→ <b>string</b>: A string is a series of text characters that can be words, letters, or symbols. They must be surrounded by either single-quotation marks(') or double-quotation marks(").</li> <li>→ <b>x</b>: The x-coordinate of the text location.</li> <li>→ <b>y</b>: The y-coordinate of the text location.</li> <li>→ <b>,</b>: We use commas to separate the different parameters or inputs in the functions.</li> <li>→ <b>;</b>: All lines of code in p5.js must end with a semicolon.</li> </ul>
<code>textSize(size);</code>	<ul style="list-style-type: none"> <li>→ <b>textSize</b>: The function name. <i>To learn more, see the <a href="#">textSize()</a> entry in the p5.js Reference</i></li> <li>→ <b>()</b>: We use parentheses to tell our program that it needs to call the function. Sometimes we include parameters or inputs in the function inside our parentheses.</li> <li>→ <b>size</b>: The size of the letters in units of pixels</li> <li>→ <b>,</b>: We use commas to separate the different parameters or inputs in the functions.</li> <li>→ <b>;</b>: All lines of code in p5.js must end with a semicolon.</li> </ul>
<code>textAlign(horizontalAlign);</code>	<ul style="list-style-type: none"> <li>→ <b>textAlign</b>: The function name. <i>To learn more, see the <a href="#">textAlign()</a> entry in the p5.js Reference</i></li> <li>→ <b>()</b>: We use parentheses to tell our program that it needs to call the function. Sometimes we include parameters or inputs in the function inside our parentheses.</li> <li>→ <b>horizontalAlign</b>: Set the horizontal alignment using LEFT, CENTER, or RIGHT. Note that these values are in all caps and case sensitive.</li> <li>→ <b>,</b>: We use commas to separate the different parameters or inputs in the functions.</li> <li>→ <b>;</b>: All lines of code in p5.js must end with a semicolon.</li> </ul>