+++ TEACHING GUIDE +++
MODULE 5: GRIDS AND PIXELS

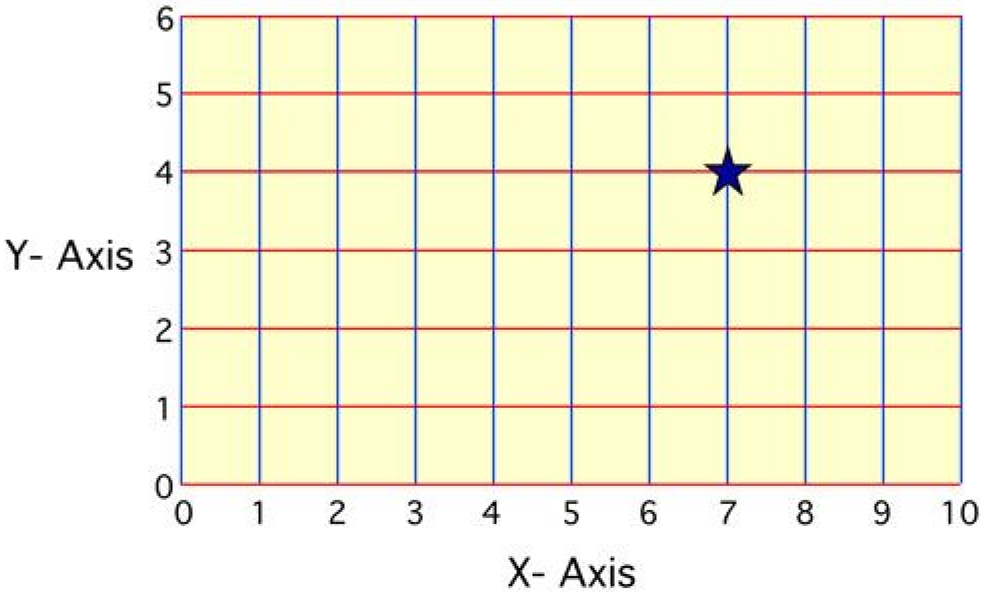
Agenda:

1. Coordinate systems (15 min)
2. What are pixels? (10 min)
3. The p5.js editor interface (10 min)
4. Your first sketch (15 min)

Coordinate systems

[*After introducing coordinate systems, draw a grid on the board (or projector or whatever you have access to) and have the learners take turns identifying various points on the grid. Be sure that everyone is comfortable pointing to some place on the grid and naming that point's X-Y coordinates*]

Even if you don't remember the phrase "coordinate system", you've probably seen them before. Our first introduction to coordinate systems is usually in a high school maths class. The most common coordinate system is a grid just like the one below. The X-Axis is the one that goes from left to right (sometimes called the horizontal axis), and the Y-Axis is the one that goes up and down (the vertical axis).



In maths class, you might be asked to identify the X-Y coordinates of the star. What are the star's coordinates?

As the maths gets more complicated, you have to draw lines and equations using the coordinate system. You might see formulas that look like this: *$y = 2x + 7$*

It's great if you know what this equation means and how to draw it on the grid, but it's not necessary for basic code art! To start making code art, you don't need to any complicated equations.

You only need to be familiar with coordinate systems and how they work. You should be able to look at any point on the grid and say what it's X-Y coordinate is.

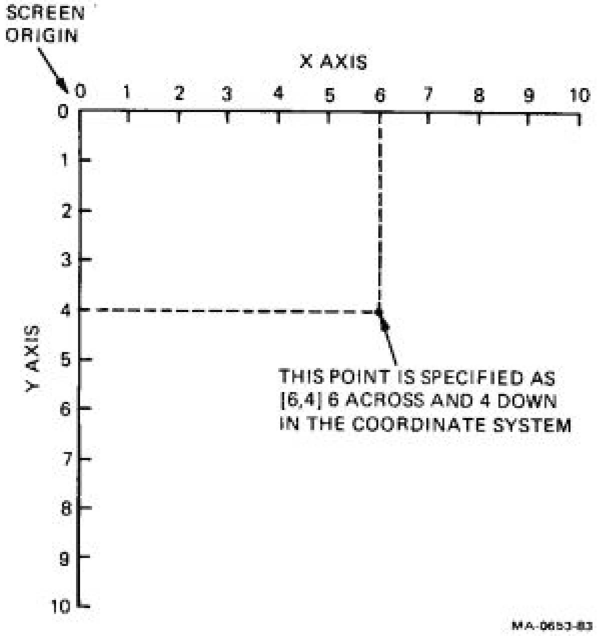
What are pixels?

[*This section introduces pixels by thinking of it as a huge coordinate system grid. The key point to emphasize is that the origin of the grid (0,0) is in the top left corner of the screen. So Y-coordinates increase in number as you move down the screen. Try drawing another grid on the board with this orientation and make sure the students are comfortable with identifying X-Y coordinates on this grid.*]

To make code art, first we have to understand that computers use a coordinate system just like the one you learned about in maths class. A computer screen is a rectangle and every point on the screen can be identified with an X-Y coordinate. You can think of a computer screen as a *huge* grid -- much bigger than the grids you used in maths class!

Even though the the grid and numbers are bigger, the idea is the same. You can choose any location on the screen and label it with an X-axis position and a Y-axis position. The only difference is that the origin (the 0-0 location) on a computer screen is in top left corner.

Move your mouse to the top, left corner of the screen as far as it can go. When the mouse is at the very edge, we can say that the mouse has a location of (0,0). As you move the mouse to the right along the top of the screen, you are increasing the mouse's X location. If you move the mouse down the screen, you are increasing the mouse's Y location. The grid on a computer looks like the coordinate system in the picture below.



On a computer, each point on the screen is called a pixel. Pixels are the units on the screen's coordinate system. For example, your computer screen may have 1,000 pixels on x-axis (from left to right) and 800 pixels on the y-axis (from top to bottom). In general, a bigger screen is just a bigger grid, so it has more pixels. A smaller screen has fewer pixels. The screen size (sometimes called its *resolution*) determines how much information, or how many pixels, a computer can display at one time.

To make make code art, you are going to learn to draw with pixels. In order to do that, you will have to instruct the computer which pixels you want to use for your drawing. Soon you will be able to use different colors, lines, and shapes. To start, you'll just use one shape.

The p5.js editor interface

[*Show the learners the interface and explain the relevant sections. Have the learners log in to their unique account they created last time (whether that's with their own email address or with the class email like ourcodingclass+benjamin@gmail.com.*

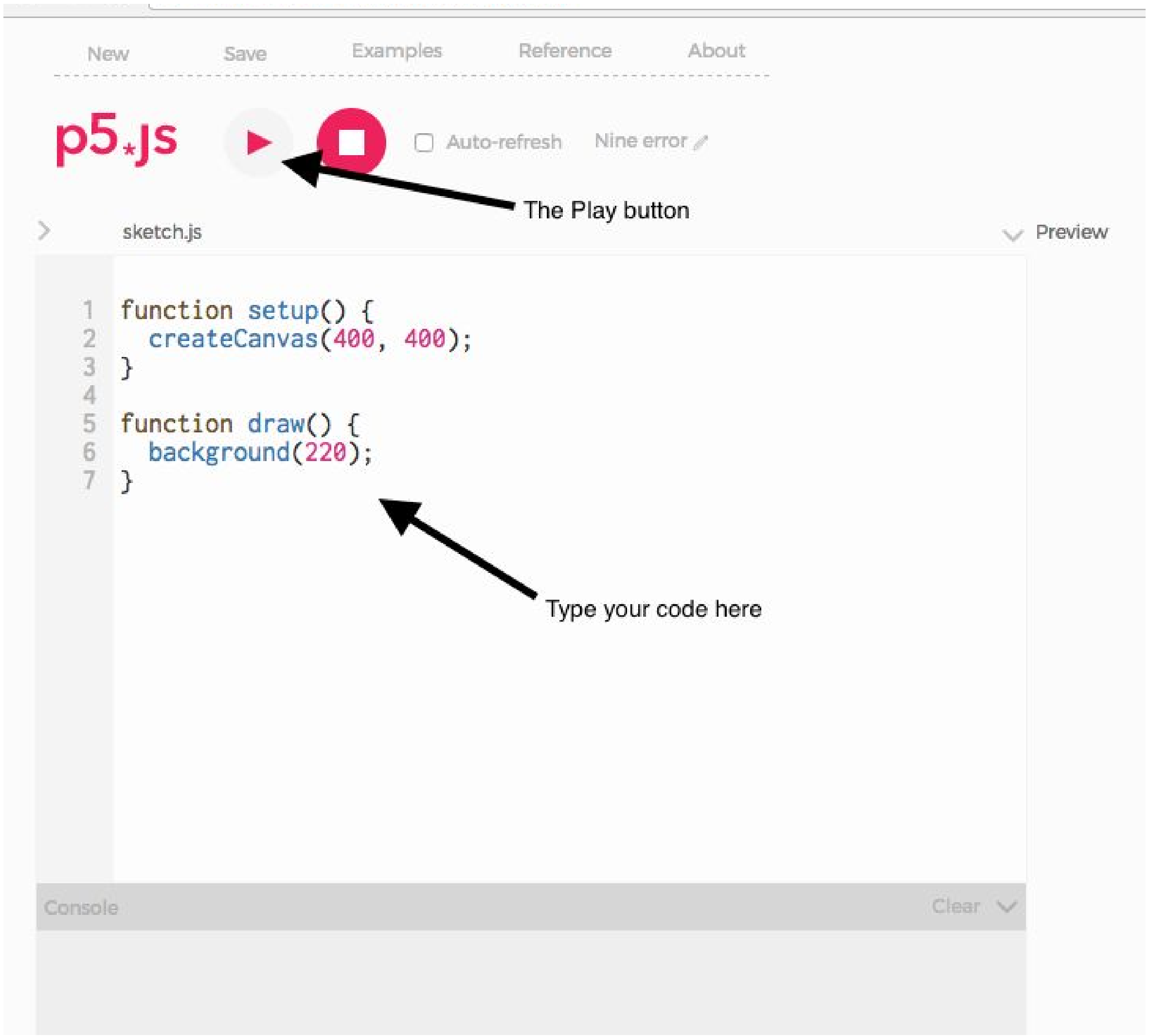
It is recommended that you use a standard format for usernames and passwords. Inevitably, a student will forget their username/password, so if you have them recorded somewhere you'll save yourself the hassle of resetting passwords or creating new accounts.]

Before we start drawing with code, we have to become familiar with a program that will allow us to draw. This program (also called a programming language) will translate our instructions into a visual output using pixels on the screen.

The name of the program we will use the called the p5.js web editor. To access it, open your Internet browser and type

editor.p5js.org

When the page loads, you will see a screen with a few buttons and a box with text in it. (The p5.js editor adds some basic code to get you started with your first sketch.)



The Play button

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

function draw() {
  background(220);
}
```

Type your code here

On the left side of the screen, you will see the “code editor”. This is where you can type your instructions to tell the program what to do (also known as “code”).

When you’ve written some code, you press the “Play button” near the top to run your program. Just to the right of the Play button is the Stop button. If you program is running, you can press that button to stop it.

On the right side of the screen is where you’ll see your program or drawing once you press the Play button. You will also see a menu of options above like play button. We will explore some of these later.

Your first sketch

[*The term ellipse might be new to some people. Emphasize that it is just another word for circle or oval. Depending on the group, it may be worth reviewing the difference between circle and* *oval. The biggest obstacle of this section is making sure they put their first line of code in the right place. It should be a separate line, after the background() line and before the closing curly bracket. For now be sure everyone understands where the code they type needs to go. They'll be introduced to the difference between Setup and Draw functions in a later lesson.*]

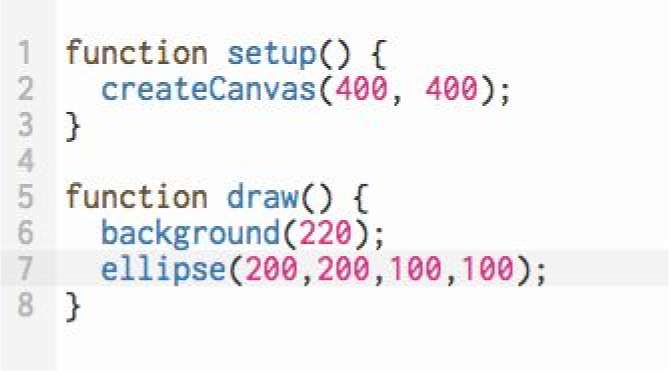
Now, it's finally time to make your first drawing with code! We will start very simply, using just one function, called the "ellipse function." As you might expect, the ellipse function allows you to draw an ellipse (sometimes called an oval or a circle).

Try adding this code on a separate line right after the line of code that says "background(220);" on line 6 (your new code must be inside the curly brackets { }).

ellipse(200,200,100,100);

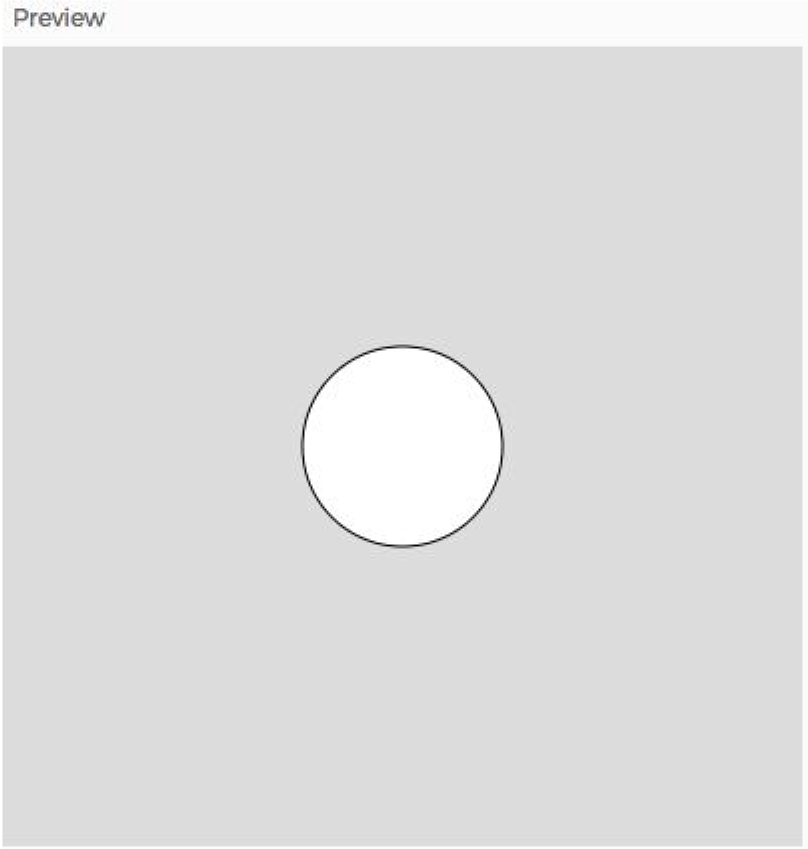
Make sure that you end your line of code with the semi-colon ;
This key is just to the right of the L key on your keyboard.
Also make sure that your new line of code is on a separate line and inside the curly brackets { }

Your screen should look like this:



```
1  function setup() {
2    createCanvas(400, 400);
3  }
4  
5  function draw() {
6    background(220);
7    ellipse(200,200,100,100);
8  }
```

Once you've entered this new code, press the Play button at the top.
Now it should display a circle in the center of the grey box on the right:



Congratulations! You’ve just made your first program.

It isn’t very exciting yet, so let’s take a look at how the ellipse function works. Then you can modify it to make your drawing more unique.

The ellipse function

[*You may want to demonstrate changing the X and Y parameters and running the program again, especially if you have a projector, so students see how changing those numbers affects the position of the circle on the canvas. The key concept here is that every learner is comfortable with modifying the parameters and running the program each time the make a change. If they understand what the changes will look like before they hit run and see the result, that’s even better.*

Emphasize that the width and height are values in pixels, just like the x and y values that determine the ellipse’s position. Understanding that the canvas/sketch is a big pixel grid is helpful in mastering the basic shape functions and making their ideas appear correctly on the canvas.]

The ellipse function uses 4 numbers to display an circle. These numbers are called parameters. You can change these numbers to affect what the ellipse looks like.

Here’s what you typed and what each parameter does:

```
ellipse(200,200,100,100);
ellipse(x,   y,   w,   h);                    [x position, y position, width, height]
```

The first and second numbers tell the program where to display the ellipse. You told the program to display the center of the ellipse at the X-position of 200, and the Y-position of 200. That makes the center of the ellipse is 200 pixels to the right of the drawing’s origin, and 200 pixels down from the origin. Remember that the origin for your drawing is the top left corner of the grey box where it displays.

The third and fourth numbers tell the program how big the ellipse should be. In this case, you told the program to make the ellipse 100 pixels wide and 100 pixels high.

Try to change one or two of the numbers in the ellipse function and press the Play button again. See how the different numbers you enter change the position or size of the ellipse?

Change more numbers and press the Play button. For now, keep all the numbers you try between 0 and 400.

Exercise: Make your own ellipse sketch

[*A possible point of confusion for learners is that some ellipses may be hidden by others depending on the X-Y parameters they use. Drawing order will be covered in a future lesson, but it may be necessary to introduce the concept now (i.e. that the program will draw the ellipses in the order they typed the commands, so later calls to ellipse() may cover up previous ellipses drawn). This can be difficult to understand at first because all of the drawing takes place instantly, so learners may be confused if some ellipses are hidden or partly covered by other ellipses. To achieve the drawing they have in mind, they may have to change the order of their lines of code.*

To save the sketch, the learners must be logged into an account. If the class is using a group account, tell them to give the sketch a unique name with their name so they can easily find it on the group’s account.]

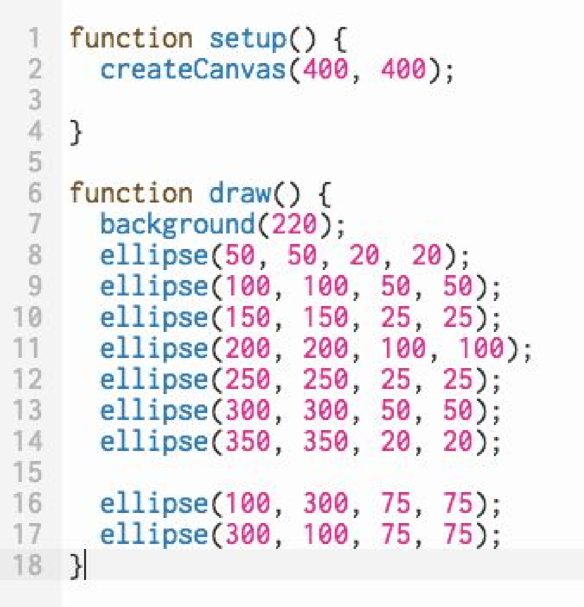
For today’s exercise, you will make a drawing using the ellipse function.

First, try adding a second line of code with another ellipse function. Give it a different X and Y location from the first ellipse. Press the Play button and notice that two ellipses now appear in the grey box.

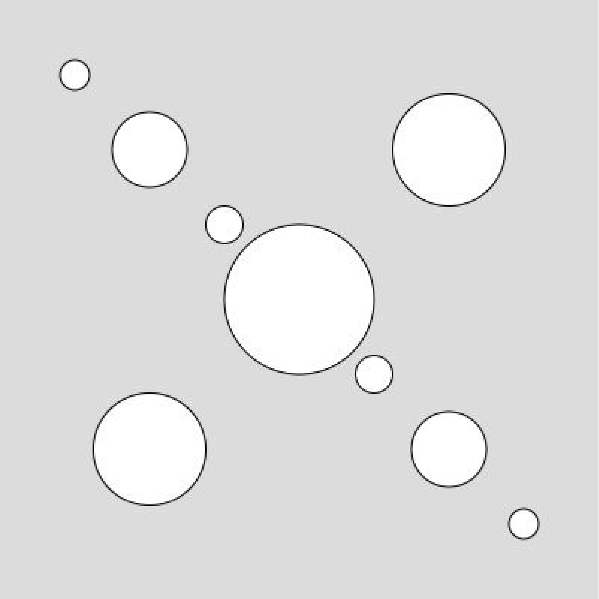
You can add as many lines of code with the ellipse function as you want. Experiment with different X and Y locations and different widths and heights. Try to make a nice pattern. Or see if you can draw a person or a face using the ellipse function.

If you are stuck, try drawing some circles on a piece of paper and then see if you can make the same drawing using code.

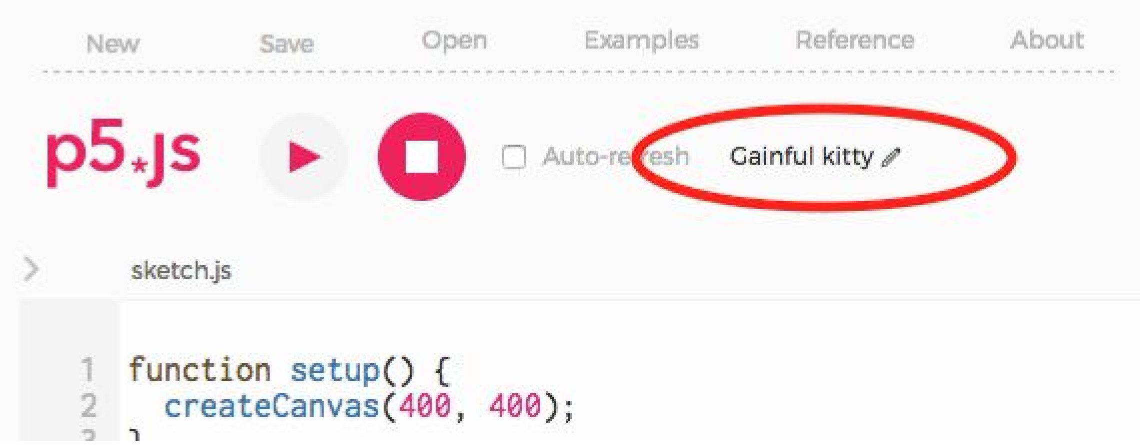
A simple example is given below:



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

}

function draw() {
  background(220);
  ellipse(50, 50, 20, 20);
  ellipse(100, 100, 50, 50);
  ellipse(150, 150, 25, 25);
  ellipse(200, 200, 100, 100);
  ellipse(250, 250, 25, 25);
  ellipse(300, 300, 50, 50);
  ellipse(350, 350, 20, 20);

  ellipse(100, 300, 75, 75);
  ellipse(300, 100, 75, 75);
}
```



Once you are happy with your sketch, it's time to save it. First, click on the two words next to the small pencil icon and enter a name for your drawing, like "circles", or "circles-yourname". Then, click on the "Save" button near the top of the screen.



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

Now your drawing will be saved on your account and you can open your drawing again to work on it wherever you have the Internet.