**Lesson 3** Functions and Parameters

**What are functions and parameters?**

| **Overview** | |
| --- | --- |
| In this lesson students will learn about p5.js functions and parameters and the use of the p5.js reference. In addition, they will learn to use program flow to change the order in which shapes are drawn. | |
| **Lesson Objectives** | |
| Students will be able to   * Call functions with different parameters * Use execution flow to layer shapes * Leave comments on their code * Consult the [p5.js online documentation](https://p5js.org/reference/) | |
| **Suggested Duration** | |
| One period (45 minutes) | |
| **Blueprint Foundations Student Outcomes (**https://blueprint.cs4all.nyc/outcomes/) | |
| Abstraction  Analyze | **Give examples** of specific patterns in something I can see, do or touch. |
| Abstraction  Prototype | **Describe** different things I tried in order to achieve a goal. |
| Algorithms  Analyze | **Describe how** instructions can have different outputs depending on inputs. |
| Programming  Analyze | **Experiment** with the commands of a programming language. |
| **Vocabulary** | |
| * **Function**: Lines of code that perform specific tasks * **Parameters**: Parameters describe the types of values that a function needs in order to run * **Arguments**: The actual values (often numbers) that a programmer uses to call a function | |
| **Planning Notes** | |
| * Print the following material for each student for the Do Now activity   + [Deconstruct Lines worksheet](https://docs.google.com/document/d/1Cv036F93vJYvO2nvALQmcurmwcKtIPt10lcBsMHcxhI/edit?usp=sharing) | |
| **Resources** | |
| * Video tutorial: [1.2 Basics of Drawing](https://www.youtube.com/watch?v=D1ELEeIs0j8) | [Code](https://github.com/CodingRainbow/Rainbow-Code/tree/master/p5.js/1.1_p5.js_basics_of_drawing) * p5 documentation: <https://p5js.org/reference/> * Student activity [starter code](https://editor.p5js.org/cs4all/sketches/VEtFoyxmg) and [solution code](https://editor.p5js.org/cs4all/sketches/4gEwGEj5g) | |
| **Assessments** | |
| * Assess the **Do Now** activity. Check for the ability to:   + Identify x and y coordinates using p5’s coordinate system. * Assess the **Student Activity**. Check for the ability to:   + Use program flow to draw shapes in a certain order   + Use parameters to change how shapes are drawn   + Leave comments on their code | |

| **Do Now** |
| --- |
| * Students should grab a copy of the [Deconstruct Lines worksheet](https://docs.google.com/document/d/1Cv036F93vJYvO2nvALQmcurmwcKtIPt10lcBsMHcxhI/edit?usp=sharing) and begin filling it out. |
| **Discussion: Functions and Parameters** |
| * Open the [sketch](https://editor.p5js.org/mparker/sketches/vi3RqdwQd) that students used in the last student activity in Lesson 2 and display it on the screen. * p5.js functions   Explain that the lines of code that they used to draw an ellipse, rectangle, and line are **functions**:   * + Functions are lines of code that perform specific tasks. The code for the shape functions isn’t in the sketch - it’s in the p5 library. We use can use this code inside our sketch by **calling** the function’s name, followed by parentheses.      * Parameters Explain that many functions require **parameters** in order to work, which are the values inside the parentheses.   + It’s impossible to memorize the parameters for all the functions that exist, so programmers always need to look up this information online. Show students how to find the parameters of an ellipse by going to the [p5.js reference page](https://p5js.org/reference/) and using CTRL/Command+F to find the [ellipse function](https://p5js.org/reference/#/p5/ellipse).   + Scroll down to the Syntax and Parameters section, which shows that the four main parameters of an ellipse are x, y, width, and height. Note: Students won’t be using the “detail” parameter in class, but they can always feel free to try out new parameters and test if it changes the sketch.   + Inside the sketch, change the values inside the ellipse function to test out what you learned from the reference page.      * Parameters vs. Arguments Explain to students that another word that is often used instead of parameter is **argument**. Their meanings are very similar, but the main difference is that parameters **describe** the values that are used inside a function, while arguments are the **actual values** used when a function is called.   + Try typing “ellipse(x, y ,width, height)” and running the code. It won’t work, because the ellipse function needs actual **number values** to “understand” where to draw the ellipse on the canvas and how big it should be. The parameters tell us what the numbers mean, not which numbers to use.   + Note: It is fine to let students know that they can use “parameters” and “arguments” interchangeably when talking about their code.      * setup() and draw() functions     Explain the following about the setup() and draw() functions:   * + These functions are special because p5 calls them automatically when a sketch is run. They don’t have parameters, so the parentheses are empty.   + We use the setup() function to *set up* a sketch. In order to draw anything in p5, we need to make a canvas and give it a size (like 400 x 400) by calling createCanvas() **inside** setup(). When you hit play, anything that is inside the setup() function will run **one time**.   + In this unit, all of our shape functions will be called inside draw().   + The code inside the draw() function actually runs in a **loop**. Every function we use inside draw is being called over and over again until the program is stopped. The loop happens so fast that the preview looks like one image, but in reality, the shapes are constantly being drawn on top of each other. Note: This loop will become relevant when students add color to their shapes, and later when they learn to animate shapes. |
| **Teacher Demo: Layering shapes** |
| * Show students these images and ask the following questions:     **rect(x, y, width, height); ellipse(x, y, width, height);**   * + If I want to draw a 100x100 rectangle on the center of a 400x400 canvas, what should be the four parameters for the rect() function?   + If I want to draw a 100x100 ellipse on the center of the same canvas, what should be the four parameters for the ellipse() function? * Open the p5.js editor and draw two shapes based on students’ answers. The code should like the one below:      * Point out that the ellipse is on top of the rectangle, and it’s because the program draws the rectangle first then ellipse. Explain that everything in the draw() loop (and the p5 sketch as a whole) will run from top to bottom. |
| **Student Activity: Recreate p5 sketches** |
| * Display the sketch below on the screen. Share the starter code and the line of code for the sketch. These lines of code are not in the order so students will need to reorder the code to recreate the sketch. * Before students begin, open the starter code and show students how to leave comments in their code.   + The lines with “//” before them are comments.      * + These lines are **ignored** by the computer   + Comments are useful to note which part of code does what.   + If your program does not behave as you expect, it is helpful to turn off lines of code one by one by “commenting them out” (you can also comment out a whole section by highlighting the code and typing CTRL or Command+?)      * Ask students to recreate the sketch and leave comments to indicate which lines are for face, left eye, right eye and nose.  | Sketch [Starter code](https://editor.p5js.org/cs4all/sketches/VEtFoyxmg)  [Solution](https://editor.p5js.org/cs4all/sketches/4gEwGEj5g) for teacher reference | **Lines of code for the sketch:**  rect(250, 100, 50, 50);  rect(100, 100, 50, 50);  rect(150, 150, 100, 100);  ellipse(275, 125, 20, 20);  ellipse(200, 200, 350, 300);  ellipse(125, 125, 20, 20);  ellipse(200, 200, 100, 100); | | --- | --- | |
| **Wrap Up** |
| * Remind students how to get a shareable link to their projects and explain how they can share it with you. Ask them to send you the sketch they worked on today. |
| **Extensions** |
| N/A |