**Lesson 2** Function Documentation

**How can we create a reference page for a custom function?**

| **Overview** | |
| --- | --- |
| In this lesson, students will create documentation for their custom functions so anyone can use and understand them. They will then learn how to create a class library of custom functions in p5. | |
| **Lesson Objectives** | |
| Students will be able to   * Explain what a code library is and why it is useful * Import a new library into a p5 sketch * Create documentation for a custom function | |
| **Suggested Duration** | |
| One period (45 minutes) | |
| **Blueprint Foundations Student Outcomes (**https://blueprint.cs4all.nyc/outcomes/) | |
| Abstraction Analyze | **Describe how** I might use patterns to express an idea. |
| Algorithms Analyze | **Describe how** instructions can have different outputs depending on inputs. |
| Algorithms  Prototype | **Demonstrate** the benefit of using an event, conditional, or loop in my prototype. |
| Algorithms Prototype | **Explain how** a function I prototyped can be used by someone else. |
| **Vocabulary** | |
| * **JavaScript Library**: A file that contains many JavaScript functions to help accomplish specific tasks, like p5.js. | |
| **Planning Notes** | |
| * Before this lesson, copy and paste students’ custom functions into a JavaScript file, ideally with the students’ names commented above the function they created. * **Do not** include setup() and draw() in this file--only the custom functions and any comments associated with the function. You can create a new JavaScript file by using a code editor like Atom or Sublime, or a built-in application like TextEdit or Notepad and **saving with a .js file extension**.   + Note: If you’re using TextEdit on a Mac, you’ll need to choose Format → Make Plain Text when trying to save as a .js file.   + You may also wish to add your own simple “test” function to the file (e.g., a function that writes the word “TEST” in the console or on the canvas) for you to use during the **Teacher Demo**. * Once the functions are in a JavaScript file, share it with your students electronically (e.g., by uploading the file to Google Classroom). * Additionally, you can host the file online and **generate a link** you can share with your students. For example, you can save the file in a GitHub repository and share it using jsDelivr, or you could use a site like [yourjavascript.com](http://yourjavascript.com/). * Figure out how you would like students to share the links to their reference pages with the entire class so students can review each other’s function documentation. | |
| **Resources** | |
| * Video Tutorial: [Multiple JS files](https://www.youtube.com/watch?v=Yk18ZKvXBj4) | |
| **Assessments** | |
| * During the **Teacher Demo**, check for the ability to:   + Import a new library into a p5 sketch * Assess the **Student Activity**. Check for the ability to:   + Create documentation for a custom function * Assess the **Wrap Up**. Check for the ability to:   + Explain what a code library is and why it is useful   + Describe where libraries are added to an html file   + Explain why documentation is a necessary part of a code library | |

| **Do Now:** |
| --- |
| * **[Design Journal]** Have students answer the following prompt:   + What information would someone else need to know in order to use a custom function that you created? Review the documentation for shape functions like [ellipse()](http://p5js.org/reference/#/p5/ellipse) to help you answer. |
| **Discussion:** |
| * Tell students that in today’s lesson, they will get to try out each other’s custom functions by importing their own class library into their p5 sketches. * Remind students that p5 is just **one example** of a JavaScript library. p5 doesn’t exist in a vacuum! There are coders all over the world creating libraries and tools in JavaScript and other programming languages. For example, [Aframe](https://aframe.io/) is a JavaScript library that makes it easy to create environments in Virtual Reality. For example, the [Access Mars](https://accessmars.withgoogle.com/) project was made using Aframe:      * Explain that programming libraries are are made for a specific **purpose**. Often, programmers are inspired to create a new library when they want to make it easier for others to create certain types of projects. For example, some libraries make it easier to animate objects on a web page or create web-based games. The purpose of your class’s custom function library is for students to share and test each other’s work. * One way that students might use libraries in your class is when designing web pages. Explain that if they’re up for a challenge, they can import and use a **CSS library** to make their pages look more modern and stylish. |
| **Teacher Demo: Adding New Libraries** |
| * Tell students to open up a new sketch.   + Ask: Do you remember which **file** contains the link to the p5 library? * Remind students that the main p5 library (along with the p5 DOM and sound libraries) is linked in the **index.html** file inside the <head> tag. * Now explain that you’re going to add your custom function library by uploading it to p5 using the same process students used to upload images. Share the function library you compiled with students and tell them to download it to their computers.   + Ask a student to walk you through the process for uploading images as the class follows along. When the student reaches the step where the image file is usually selected (or drag-and-dropped), select the library instead.   + Ask students if they notice anything interesting about the file that appears in the sidebar. Is there a way to tell that it’s a JavaScript library and not an image? *Answer: The .js extension lets us know that it’s a JavaScript file, just like sketch.js* * Now students will be able to click on the JavaScript library in the sidebar and view all their classmates’ functions. Try to use a function from the library (either your own test function or one made by a student) inside the sketch.js file and hit the play button. You will get an error:      * + Turn and Talk: Why do we get an error when we try to use this function inside our sketch.js file? * Students might point out that the two JavaScript files aren’t “connected”. Explain that in order for the sketch.js file to access those functions, the library needs to be added to index.html just like the main p5 library. Explain that it should be added inside the <head> tags because similar to the preload() function, it makes sure that all of the files are loaded **before** the sketch is run.      * Return to the sketch.js file. Before running the test function, ask the following:   + Ask: How can we double-check the syntax of this function? What parameters, if any, should we use? * Explain that in order to know how to use this function, students would have to scroll through the entire library, locate the function, and decipher the source code. * Students know that when they’re using a new p5 function, they can **consult the p5 reference page** for more information. In order for your class’s library to be useful, students will need to make documentation that their peers can reference when trying out their awesome functions! |
| **Student Activity: DIY Reference Page** |
| * Ask students to share their responses from the **Do Now**. * Using these responses as a jumping-off point, instruct students to create documentation pages with a similar format to the p5 reference site (see image below). These pages should contain:   + Function name   + Student’s name   + Example drawn on canvas and a line of text with the function call   + Description of what the function does   + Syntax of the function   + Parameters needed, including data type (e.g., number or string)      * Optionally, students may include a shout-out to a peer who helped debug their code, or links to other p5 functions used inside their custom functions (like [line()](https://p5js.org/reference/#/p5/line) or [ellipse()](https://p5js.org/reference/#/p5/ellipse), for example). * When students are finished, they should be paired off to peer-review each other’s documentation by testing out each other’s functions. * Once their reference page has been reviewed, students can share the link to their page with the whole class. * As an extra challenge, early finishers can work together to make a Google Site (or even a separate p5 webpage!) that compiles each separate reference page into a single master document as a resource for the class. |
| **Wrap Up** |
| * **[Design Journal]** Students should answer the following:   + Where in the index.html file are libraries added?   + Why do programmers make libraries?   + Why is documentation a necessary part of a code library? |
| **Extensions: N/A** |
|  |