**Lesson 2** Getting Started with the p5.js Web Editor

**How is the p5.js web editor structured?**

| **Overview** | |
| --- | --- |
| In this lesson students will be introduced to the p5.js web editor. They will explore its interface and begin drawing shapes. | |
| **Lesson Objectives** | |
| Students will be able to   * Create an account on the [p5.js web editor](https://editor.p5js.org/signup). * Explore the features of the p5 web editor * Change the file name of a p5 sketch * Describe the basic structure of a line of code in p5.js | |
| **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** | |
| * **Web Editor**: An online environment where programmers can write and test their code. * **p5 sketch**: The JavaScript file that uses p5 to create digital drawings and interactive web programs. * **Pixel**: Pixels are the dots that make up an image on a computer screen. They are the smallest element of a digital image. | |
| **Planning Notes** | |
| * Determine how you would like students to login to their p5 accounts. If they already have Gmail accounts, it is recommended that students [login here](https://editor.p5js.org/login) with their Gmail credentials. * Decide how you would like students to name each new p5 sketch. For example, students might call today’s sketch “U1L2 Basic Shapes” * In this lesson, students should practice submitting today’s sketch to you. Before class, figure out how you want to collect their projects (e.g., Google Classroom) * Now that students are using the web editor, it is useful to have a contingency plan in case of internet outages. Check out [this document](https://docs.google.com/document/d/15nRye7SNxK_YRD3lCCLHIEIFQo9xXcIfB9vDFvcAJy0/edit?usp=sharing) for instructions on how to use p5.js offline. | |
| **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.js Web Editor Interface](https://docs.google.com/document/d/1HrfrPa-_KEl9-PapthanHuDaynrsT0VRSwpajD2f00Y/edit?usp=sharing) * [p5.js Web Editor](https://editor.p5js.org/) | |
| **Assessments** | |
| * Assess the **Do Now** activity. Make sure every student has an account and has written down their username and password. * Circulate during the **student activity**. Check for the ability to:   + Edit existing lines of code.   + Make inferences about how code affects the visual output. * Assess the **wrap-up**. Make sure every student has shared their sketch with you. | |

| **Do Now** |
| --- |
| * Have students sign up for the p5.js Web Editor account [here](http://editor.p5js.org/signup). **Note:** If students are already using the Google Apps, instruct them to [login here](https://editor.p5js.org/login) with their Gmail account (different from the sign-up link). * Once students have logged in, let them explore the interface for 1 or 2 minutes. |
| **Discussion: p5.js Web Editor Overview** |
| * Display the p5.js web editor on the screen and ask students what they noticed when they explored the interface during the Do Now. * Explain to students that they will use the p5 editor to write and test their JavaScript programs, and point out its most important features: * **Text Editor**: Students will write their code inside the white text box. When a new sketch is created, some starter code will automatically appear, along with a random project name. * **Project Name:** Instruct students to change the project name so it corresponds to the lesson. In the example below, students would click on “Secretive powder” and type in something like “U1L1 Basic Shapes”      * **Play, Stop, Preview**: Instruct students to click the play button. This runs the code and creates a preview on the right. By default, p5’s starter code creates a canvas with a gray background. Instruct students to hit the stop button. The preview will disappear because it stops the program from running.   + Ask students to infer which line of code creates the canvas and which line of code sets the background. *Answer: The code on Line 2 creates the canvas and the code on Line 6 sets the background.* |
| **Teacher Demo: Simple program in p5.js Web Editor** |
| * Ask students to close their computers and pay attention to the demonstration. You may introduce your students to the **Hands and Eyeballs** protocol: Hands are in laps and eyes are on the teacher. * Demo 1:   Explain that “**createCanvas(400,400)**” creates a canvas, or drawing area, and also sets it to a specific size. Change the numbers in the parentheses and run the code. Try this multiple times with both numbers and ask the following questions:   * + What do you think the first number is for? *Answer: the width of the canvas*   + What do you think the second number is for? *Answer: the height of the canvas*   Ask students how they usually measure the size of an object in real life. Our p5 canvas is not measured in inches or feet, but **pixels**, the dots that make up a digital image. The default canvas size is 400 pixels wide by 400 pixels tall.   * Demo 2:   Add **point(100, 200);** under **background(220);** and run the sketch. It may look like nothing has changed, but if you zoom in to your screen you should see a single black dot on the canvas. Explain that by default, this point is the size of one tiny pixel! A canvas that is 400 x 400 is made up of 160 *thousand* pixels.   * Demo 3:   Tell students that in order to create visual artwork with p5, they will begin by adding shapes.  Add **ellipse(200, 200, 50, 50);** under **background(220);** and run the code. For students who are not fluent typers, point out that parentheses are typed with Shift + 9 and Shift + 0.  Explain that changing the numbers will change how the shape is drawn. In the following activity, students should think about what each number represents. |
| **Student Activity** |
| * Display the following lines of code along with the prompts for exploration and questions. Explain to students that they will type each line of code under **background(220);**   + Lines of code (linked [here](https://editor.p5js.org/mparker/sketches/vi3RqdwQd)):     - **ellipse(200, 200, 100, 100);**     - **rect(200, 200, 50, 50);**     - **line(100, 150, 300, 270);**   + Try:     - One at a time, change a number inside the parentheses and hit the play button       * **[Design Journal]** What do you think each number is for? Write your findings in your design journal.     - To change the order of the lines of code (e.g., type rect before ellipse)       * **[Design Journal]** What did you notice?     - To make more than one of each shape       * **[Design Journal]** What did you notice?   + Bonus: if there is some time left in the period encourage students to try the following:     - Draw a horizontal line, a vertical line, a diagonal line.     - Draw three ellipses next to each other.     - Draw three rectangles stacked vertically on top of each other. |
| **Wrap Up** |
| * Show students how to get a shareable link to a project.   + First save the project:: File --> Save (The shortcut is CTRL/Command + S)   + Copy the main URL: Highlight the address and copy it by pressing CTRL/Command + C   + Paste the URL: Show students where they should paste the link (CTRL/Command + V) in order to share the sketch with you |
| **Extensions** |
| N/A |