**Lesson 1** Variables

**How can we use variables to make programs more flexible?**

| **Overview** | |
| --- | --- |
| In this lesson, students will expand on their knowledge of built-in p5 variables to write and use their own variables, making their programs more flexible. In the student activity, they will follow pair-programming protocols to recreate a poster in p5. | |
| **Lesson Objectives** | |
| Students will be able to   * Describe what a variable is and how it is used * Declare and assign values to variables * Use mathematical operations on variables * Use pair programming to solve coding challenges | |
| **Suggested Duration** | |
| One or two periods (45 - 90 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  Communicate | **Compare and Contrast** my instructions with other instructions that complete the same task. |
| Programming  Communicate | **Present** the challenges and benefits of using tools to program collaboratively. |
| **Vocabulary** | |
| * **Variable** - In computer science, a variable is a container that holds information, like numbers or text. * **Assignment Operator** - Used to change the information stored inside a variable. The “equals sign” is one example of an assignment operator. | |
| **Planning Notes** | |
| * Students will be expected to **pair program** in today’s student activity. Refer to [this document](https://drive.google.com/file/d/1QtskhgXG5w5bsOrFXawprd62MQPJrGsz/view?usp=sharing) for tips on teaching pair programming.   + Decide in advance how students will share their sketches with their partner at the end of the lesson. * To give students more practice, print out copies of the Rocky poster [planning grid](https://docs.google.com/document/d/1-H-LyLza7iK9Aha6qt1v60w2YvOMpsteb_2pyDPNcZw/edit?usp=sharing) for the Extension activity. Note that the canvas height and “shadow” do not exactly match the sketch, but this won’t affect the “stairs” that students need to plan. | |
| **Resources** | |
| * [Video](https://www.youtube.com/watch?v=vgkahOzFH2Q) on Pair Programming * [Tips](https://drive.google.com/file/d/1QtskhgXG5w5bsOrFXawprd62MQPJrGsz/view?usp=sharing) for teaching Pair Programming * Video tutorial: [2.2 Variables in p5.js (make your own)](https://www.youtube.com/watch?v=Bn_B3T_Vbxs&index=6&list=PLRqwX-V7Uu6Zy51Q-x9tMWIv9cueOFTFA) | [Code](https://github.com/CodingRainbow/Rainbow-Code/tree/master/p5.js/2.2_Variables_in_p5.js_user_defined) * [Expressions, Variables and Loops](https://creative-coding.decontextualize.com/expressions-variables-and-loops/) by Allison Parrish (ignore the loops section). * Getting Started With p5.js: Chapter 4, examples 4.1, 4.2 | [Code](https://github.com/lmccart/gswp5.js-code/tree/master/04_Variables) * [Color Transparency](https://p5js.org/learn/color.html) (after RGB Color section) | |
| **Assessments** | |
| * Circulate during **Independent Practice**. Check for the ability to:   + Create a new variable above setup()   + Initialize the variable with a starting value   + Use the variable inside function calls * Assess the **Student Activity**. Check for the ability to:   + Use mathematical operations on variables   + Follow pair programming protocols | |

| **Do Now:** |
| --- |
| * Share [this example](https://editor.p5js.org/mparker/sketches/m02nd3EAo) with students and have them answer the following:  Imagine a friend wanted you to help them create a poster for their archery club. Right now, the bullseye is in the lower left.  1. How many arguments (parameters) would you have to change to move the bullseye to the center, and what would the new numbers be? *Answer: 6 numbers. The x and y coordinates of each ellipse would be (200, 250)* 2. Where do you see repetition in the parameters of this sketch? *Answer: The x and y position of each ellipse is the same.* |
| **Discussion: Variables** |
| Variables   * Pull up the bullseye [example](https://editor.p5js.org/mparker/sketches/m02nd3EAo) from the Do Now   + Ask students how to update the code to move the bullseye to the center, and live-code those changes. * Tell students that you’ve changed your mind and you think the design will look better in the upper left corner   + Ask: How many more changes need to be made? * Explain that right now, this code is repetitive and difficult to change. To make this code more flexible and easy to update, students will need to use **variables**. |
| **Teacher Demo: Variables** |
| * In Unit 1, students used built-in p5 variables mouseX, mouseY, width and height. In this lesson, they will be making custom variables. Use the bullseye example to illustrate how to make a custom variable:  1. **Decide:** Which values should be replaced with variables? Not all values need variables - only those that are likely to *change*. In this example, it makes sense to use variables for the x and y parameters to make it easier to change the bullseye’s location. 2. **Declare:** Use the keyword “var” to let the program know you’re making a variable. Then decide on a name that makes sense for the variable. This name should begin with a lowercase letter and **not** be a p5 keyword (for example, “width” or “fill” cannot be used). Type this abovethe setup() function. Note: In some examples, students may come across the keyword **let**, so let them know that these are equivalent. 3. **Initialize:** Assign a starting value to the variable. Using 200 will center the bullseye horizontally. 4. **Use:** Instead of repeating numbers, use the variable. Right now, xPosition *contains* the number 200, so the ellipse function uses a location of 200 for the x parameter.  * Point out that so far, this has not saved any time or effort. However, if you decide that you want to move the bullseye to the left or the right, you only need to change one number—the number assigned to the variable at the very top of the code.      * Turn and Talk: In your own words, explain to a neighbor why the bullseye moves when the value of the xPosition variable changes. Think about how each ellipse function uses the variable on lines 11, 13, and 15. Then share out to the whole class. * Independent Practice: Students should practice on their own by duplicating the bullseye sketch and creating a new variable called yPosition. * Ask: Do you remember how we centered shapes using **width** and **height**? *Answer: We divided width and height by 2.* * Explain that students can use any mathematical operation on variables that contain numbers, like division, multiplication, addition or subtraction. This can help using variables to express relationships between numbers.   + Turn and Talk: Discuss the relationship between the size of each circle in the bullseye. What numbers do you need to add to 50 to make it the same size as the bigger circles? How could you multiply 50 to make it the same size as the bigger circles? * Make a new variable called “size” and set it equal to 50. Then update each ellipse function as follows:      * Try changing the size variable to different numbers, like 10 or 250:      * To keep the ellipses in proportion, multiply by size instead of adding to it. Now when you change the size variable, the entire bullseye will grow and shrink by the same amount:     [Link to code](https://editor.p5js.org/mparker/sketches/RSEnQ-vxR) |
| **Student Activity: Recreate the Wimbledon Poster - Pair Programming** |
| * Students will recreate [this poster](https://i.pinimg.com/originals/04/03/52/0403527d7cec047230eb38156a3320ad.png) for the Wimbledon Tennis Tournament in pairs (here is a [video of highlights](https://www.youtube.com/watch?v=7mp9ou50gEA) from the women’s final in 2016). Share this [starter sketch](https://editor.p5js.org/mparker/sketches/i9h9TJXc3) and point out the following:   + There are two variables for each parameter with **changing** values. The “y” variable will control the y position of the tennis balls. The “opacity” variable will control how transparent each tennis ball is.   + By using a fourth argument inside fill(), you can change a shape’s transparency. This number should be between 0 and 255. 255 is 100% visible, and 0 is completely invisible.     Students should follow the prompts below to recreate the poster ([solution](https://editor.p5js.org/mparker/sketches/SXm6FOzRz) for teacher reference):   1. Pair Program: Recreate the Wimbledon poster by adding four tennis balls to the sketch 2. Duplicate the [starter code](https://editor.p5js.org/mparker/sketches/i9h9TJXc3) and save the copied sketch. 3. Draw four new ellipses using the “y” variable (the x position and size don’t change). 4. **Multiply** the y variable by different numbers to make the tennis balls match the poster.    1. HINT: You may have to multiply by numbers with **decimals**. 5. Write four fill() functions above each new ellipse. 6. **Subtract** from the opacity variable to make each new tennis ball more see-through than the last. 7. Uncomment Line 10 to call noStroke() and remove the outlines on the tennis balls.  * Note: Before students begin coding, introduce them to pair programming protocols:   1. There are two roles, Driver and Navigator (see details below).   2. These roles switch every 5 - 7 minutes   3. Coding only happens on **one computer**. The Navigator may use their computer to refer to the p5.js reference page or today’s slide deck.  | **Driver** | **Navigator** | | --- | --- | | * Uses the keyboard and mouse to complete task * Discusses changes with Navigator before typing * Focuses on line-by-line coding and syntax | * Hands off the keyboard and mouse * Constantly communicates with and gives directions to the Driver * Focuses on the “big picture” and debugging errors | |
| **Wrap Up** |
| * As a class, discuss the following:   + During pair programming, what is challenging about each role?   + Give a specific shout-out to your partner in the coding challenge. * Make sure students share their sketches with their partners and with you before they leave. |
| **Extensions: Recreate the Rocky Poster - Pair Programming** |
| * Give students extra practice by having them recreate the Rocky [movie poster](http://lthscomputerart2.weebly.com/uploads/9/8/2/3/9823286/5005273.jpg) using p5, which references this [famous scene](https://www.youtube.com/watch?v=_YYmfM2TfUA). Share [this sketch](https://editor.p5js.org/mparker/sketches/RDBooysDM) with students, in which the stairs are missing! Students should follow the steps below to recreate the poster ([solution](https://editor.p5js.org/mparker/sketches/IVR-quqw2) for teacher reference):      1. GOAL: Recreate the Rocky poster by adding at least three stairs in between the first “stair” and the “running man” 2. PLAN: Draw the steps on this [template](https://docs.google.com/document/d/1-H-LyLza7iK9Aha6qt1v60w2YvOMpsteb_2pyDPNcZw/edit?usp=sharing) first and write pseudocode 3. CODE: Using **pair programming**, draw the rectangles using the numbers one of the templates. 4. COMMENT: Change those rectangles into comments (like on Line 16) 5. COMPLETE: Redraw those rectangles with the variables at the top by using **addition** and **subtraction**. |