**Lesson 2** Conditional Statements

**How can we use mouse position to make a sketch interactive?**

| **Overview** | |
| --- | --- |
| In this lesson, students will use “if” statements to update their sketches when certain conditions are true. They will be introduced to the p5 system variables mouseX and mouseY and use them in their conditional statements. | |
| **Lesson Objectives** | |
| Students will be able to   * Write a basic “if” statement * Use “if” statements to update the background, fill, and parameters of shapes * Make a sketch interactive by creating thresholds for mouseX and mouseY | |
| **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. |
| 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** | |
| * **Boolean Expression**: A logical expression that evaluates to true or false. * **Relational Operator**: Tests the relationship between to values to see if it is true or false. Some examples are: greater than (>), less than (<), greater than or equal to (>=), less than or equal to (<=). * **Conditional Statement**: Creates a block of code to be executed when a certain condition is met. For “if” statements, the code is only run when a condition is true. | |
| **Planning Notes** | |
| * This lesson uses greater-than and less-than symbols to compare numbers. [This resource](https://www.youtube.com/watch?v=nFsQA2Zvy1o&feature=youtu.be) may be useful for students who are not comfortable with mathematical inequalities. * In Lesson 3 there is a walkthrough of the **Student Activity** (Sunset Lab) so you will have more time to address students’ questions. * The **Wrap Up** uses an exit slip. Cut out the slips on the [print-out](https://docs.google.com/document/d/1_8FanJ1Ar9Wwegw3x0HTYxEc8N4rLe5noKRJDWKyXf4/edit) to distribute to your students, or create a Google Form instead. | |
| **Resources** | |
| * Video tutorial: [Conditional Statements](https://www.youtube.com/watch?v=1Osb_iGDdjk) * Video lesson: [Greater and Less Than Symbols](https://www.youtube.com/watch?v=nFsQA2Zvy1o&feature=youtu.be) | |
| **Assessments** | |
| * During the **Code Along**, check for the ability to:   + Identify the boolean expression in the conditional   + Explain how the sketch is affected when the condition is true or false * Assess the **Wrap Up** exit slip. Check for the ability to:   + Clearly comment the code   + Explain the two parts of the conditional--the boolean expression that is tested and the code block that runs when the expression is true | |

| **Do Now: Kinesthetic Conditionals** |
| --- |
| Give your students instructions to act out in the form of conditional statements. For example:   * “*If* it is true that your birthday comes before June, *then* nod your head” * “*If* it is true that you wear glasses, *then* stand up” * “*If* it is true that you listen to Beyonce, *then* snap your fingers”   Have students write down their own conditionals in their notebooks. They should follow this structure: “*If* it is true that (fact), *then* (perform action).” Then call on students to read their conditional statements out loud while the class acts out the instruction. Note: To prevent actions from getting too wild, you may want to constrain students to a menu of actions that they are allowed to perform. |
| **Discussion: Conditional Statements** |
| * Explain to students that they will apply the same logic in the Do Now to their code using conditional statements. They will begin by using one type of conditional statement, an “if” statement. Open up a new sketch to show them the anatomy of an “if” statement:      * Boolean Expressions   + Model changing the background color of the sketch:      * + You can use the following language to explain that in programming, the “something” is called a Boolean Expression: “A Boolean expression shows a comparison, often between numbers. It can only be **true** or **false**.”   Now update the code to use a boolean expression:     * + Explain that because it is *always* true that 6 is greater than 5, the background color will remain black.   Now change the boolean expression to (6 < 5):     Ask students for predictions about what will happen to the code. After running the sketch, explain that the code block is **ignored** and the background will *never* be black because the expression (6 < 5) is false.   + Tell students to think about the boolean expression as a **test** of true or false: “When we write 6 < 5, we’re not saying that six really is less than five. We’re asking the computer to *test* if the first number is smaller than the second number. Since 6 is not less than 5, the expression is false, and the code inside the brackets is skipped over.” * Review: mouseX and mouseY   + You can use the following language to explain the limitations of code you just wrote: “Our goal is to make our p5 sketches interactive, which means we want users to be able to *change* the sketch. The two boolean expressions we just used are *always* true or *always* false, so the sketch can’t change over time. So instead of using plain numbers like 5 and 6 in our expressions, we’re going to use the **variables** mouseX and mouseY.”   + Call on a student to remind the class what information is stored in mouseX and mouseY. Explain that by using these variables inside conditionals, students can change the behavior of a sketch depending on where the mouse is on the screen. |
| **Teacher Demo: Conditional Statements with mouseX** |
| * Code Along: As a class, write out the following code:      * Check For Understanding   + Ask students to identify the boolean expression. *Answer: mouseX > 300*   + Ask: When is the boolean expression true? How is the sketch affected? *Answer: The expression is true when the mouse is more than 300 pixels to the right of the canvas. This causes the background to turn white.*   + Ask: When is the boolean expression false? How is the sketch affected? *Answer: The expression is false when the mouse is less than 300 pixels to the right of the canvas. The background stays black, because the code block in our conditional is ignored.* * Scaffold: It may be helpful to add text(mouseX, 20, 20) so students can constantly track the x position of the mouse. * Independent Practice   + Instruct students to experiment by changing the values in the boolean expression. For example, ask them to test if mouseX is greater than 150 pixels, or if mouseX is *less* than 200 pixels. * Code Along (continued)   + As a class, expand on the previous example by building out the code below. Explain that you can change more than one aspect of the sketch inside a single if statement. In this example, both the background and the fill change when mouseX is greater than 300.      * Turn and Talk   + Move the ellipse function so it occurs **before** the if statement on Line 9. Have students discuss in pairs what they notice about the sketch, and what is the reason for the change. Students should remember from the previous unit that a fill function must be called *before* a shape function for the shape to have that color. |
| **Student Activity: Sunset Lab** |
| This lab expands on the previous in-class example by using conditionals to make a “sunset”:   * Share this [starter code](https://editor.p5js.org/mparker/sketches/fFHwiRs-O) with your students for students to duplicate and save. * Point out that in this example, the yPosition variable is **initialized** inside the draw loop instead of at the top of the sketch. * To make it easier for students to create colors using RGB values, they can use this [color picker tool](https://www.w3schools.com/colors/colors_rgb.asp). * Students should follow the prompts below:  1. Make the default background light blue 2. Make the fill yellow 3. Change the yPosition variable to 100 4. Change mouseX to mouseY 5. Inside the if statement:    1. Change the background to dark blue    2. Change the fill to white    3. Write a new line of code that changes the yPosition variable to 350 6. BONUS: Make the moon look like a crescent moon! Write another conditional for when mouseY is greater than 200, draw a dark blue ellipse with the same y position as the moon, but an x position slightly to the left.   [Solution](https://editor.p5js.org/mparker/sketches/bN91HQkRL) for teacher reference.  Extension: Note that in this example, the same condition is being tested in two different if statements. If students point out this repetition, challenge them to recreate the sketch using *only one* if statement. See [solution](https://editor.p5js.org/mparker/sketches/QRUiAM-PV) for teacher reference.  Students may not have time to finish the Sunset Lab. Manage their expectations and let them know that they should get to as many prompts as they can without rushing. There will be an opportunity to review the lab as a class in the next lesson. |
| **Wrap Up** |
| * Students should save and name their sketches according to class convention. * Distribute the [exit slip](https://docs.google.com/document/d/1_8FanJ1Ar9Wwegw3x0HTYxEc8N4rLe5noKRJDWKyXf4/edit) and collect it before students are dismissed (see code below). Students should write their responses on lines 12 - 15. Emphasize the importance of sharing their thought process rather than getting the “right answer.” |