**Lesson 4** Buttons

**How can we use HTML buttons to control a p5 sketch?**

| **Overview** | |
| --- | --- |
| In this lesson students will learn how to add a button to a webpage and create a callback function that is triggered by the button. | |
| **Lesson Objectives** | |
| Students will be able to   * Use the p5 DOM library to create a button * Define and use callback functions * Use buttons to control the output on a p5 canvas | |
| **Suggested Duration** | |
| One period (45 minutes) | |
| **Blueprint Foundations Student Outcomes (**https://blueprint.cs4all.nyc/outcomes/) | |
| Algorithms  Prototype | **Explain why** I used specific instructions to complete a task. |
| Programming Communicate | **Discuss** what can and cannot be done with a specific set of commands. |
| Algorithms Communicate | **Compare and contrast** my instructions with other instructions that complete the same task. |
| **Vocabulary** | |
| * **Callback function:** A function that is passed to another function as an argument | |
| **Planning Notes** | |
| * N/A | |
| **Resources** | |
| * Video tutorial: [Interacting with the DOM using Sliders, Buttons and Text Inputs](https://www.youtube.com/watch?v=587qclhguQg) * Video tutorial: [Handling DOM Events with Callbacks](https://www.youtube.com/watch?v=NcCEzzd9BGE) | |
| **Assessments** | |
| * Assess students’ responses during the **Teacher Demo**. Check for the ability to:   + Compare and contrast mousePressed() as a standalone function and as a function attached to a button object. * Assess the **Student Activity**. Check for the ability to:   + Use the p5 DOM library to create HTML buttons   + Define and use callback functions   + Use buttons to control the output on a p5 canvas | |

| **Do Now:** |
| --- |
| * **[Design Journal]** What do you observe about the button in [this sketch](https://editor.p5js.org/mparker/present/XBz8OlO9s)? In your design journals, list the important features and behaviors of digital and physical buttons. |
| **Discussion:** |
| * Have students share their responses from the **Do Now**. * Ask: What are some similarities and differences of digital and physical buttons? * You can also use this as an opportunity to discuss **abstraction**: How have digital buttons been *abstracted* from physical ones? |
| **Teacher Demo: Worst DOM Game Ever** |
| * Build out [this sketch](https://editor.p5js.org/mparker/sketches/WVDFb8GOF) with your students to create “The Worst DOM Game Ever”. Emphasize the following:   + Step 1: Call the createButton() function and save it inside a variable. This function is just like createP() in that it uses the DOM to generate an HTML element with JavaScript.   + Step 2: Use **dot syntax** to attach the mousePressed() function to the button variable so the program knows when the button has been pressed.   + Step 3: Create a **custom function** that contains the code for the behavior of the button. This function controls what the button *does*.   + Step 4: Use this function as an argument *inside* mousePressed(). When a function is passed as an argument, it is referred to as a **callback function**. * Have students compare “The Worst DOM Game Ever” to its [earlier version](https://editor.p5js.org/mparker/sketches/2B8fmfvX-) from Unit 2   + Ask: How is mousePressed() different in each sketch? What are the advantages of attaching mousePressed() to a button instead of using it as a standalone function?   + Expand on student responses by explaining that a digital button provides *feedback*. Seeing the button “push” down in response to the mouse reassures the user that the action was registered by the computer. |
| **Student Activity:** |
| * Have students choose one of the following [Mild](https://editor.p5js.org/mparker/sketches/UqdIYPM4T), [Medium](https://editor.p5js.org/mparker/sketches/8QjD_81ou), or [Spicy](https://editor.p5js.org/mparker/sketches/HIjShoX6j) options to work on during the rest of the class (examples are linked above for teacher reference):   + Infinite Buttons (Mild): Make a button that creates more buttons when clicked! For an extra challenge, keep a counter of how many buttons you’ve made. You can also try to make the buttons appear in random locations on the page by attaching the [position()](https://p5js.org/reference/#/p5.Element/position) method to the new buttons.   + Winking Button (Medium): Make a simple emoji with two eyes, and create a button that makes one eye “wink” open and shut.   + Winking Button (Spicy): Same challenge as above, but make the winking motion a smooth animation from open to close. |
| **Wrap Up** |
| * Make sure students save their sketches and share them with you before they leave. |
| **Extensions: N/A** |
|  |