**Lesson 5** Text Inputs

**How can we make a web page dynamic?**

| **Overview** | |
| --- | --- |
| In this lesson students discuss the difference between static and dynamic content. They will also add input fields to a web page and display the text by pressing a button. | |
| **Lesson Objectives** | |
| Students will be able to   * Describe the differences between static and dynamic content on the web * Add input fields to a web page with different text types * Modify the HTML of a web page using values from text inputs | |
| **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. |
| Networks Analyze | **Explain how** consent, privacy and security impact my trust in an application. |
| Networks Prototype | **Explain how** I used at least three different markup tags to build a website. |
| **Vocabulary** | |
| * **Input** - Information captured from user interaction | |
| **Planning Notes** | |
| * For the **Discussion**, figure out which websites you’d like to show your students as examples of static and dynamic websites. Make sure that your school whitelists/unblocks the sites you choose. Some examples are:   + [p5 Reference Site](https://p5js.org/) - static   + [Thisissand](https://thisissand.com/) - static   + [Instagram](https://www.instagram.com/explore/tags/p5js/?hl=en) posts with the #p5js hashtag - dynamic | |
| **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 the **Student Activity**. Check for the ability to:   + Add input fields to a web page with different text types   + Modify the HTML of a web page using the values from text inputs * Assess the **Wrap Up**. Check for the ability to:   + Describe the difference between static and dynamic content   + Explain that the text from an input is “connected” to the button through the behavior defined in a callback function   + Describe the security limitations of inputs that use the “password” type | |

| **Do Now:** |
| --- |
| * **[Design Journal]** Based on the following definitions of the words **static** and **dynamic**, what do you think makes a *web page* static or dynamic?   + Static: showing little change; characterized by lack of movement, animation, or progression   + Dynamic: marked by usually continuous activity or change |
| **Discussion:** |
| * Have students share their responses from the **Do Now**. * Explain that in order for a website (or for the content on a website) to be considered dynamic, the HTML on a page will change *while it’s running*.   + Ask: What makes it possible to change the HTML of a page using JavaScript (or other programming languages)? *Answer: The DOM* * Give your students some examples of websites, asking them whether these sites are static or dynamic. Here are some that you might use:   + [p5 Reference Site](https://p5js.org/) - This is a static web page. None of the content changes in real time, and everything on the page was “hard-coded” using HTML and CSS.   + [Thisissand](https://thisissand.com/) - This may surprise some of your students, but this web page is static. Even though the application was programmed using JavaScript and can respond to user interaction, **the JavaScript is not affecting the DOM**. Because none of the HTML elements on the page are changing, it is not a dynamic site.   + [Instagram](https://www.instagram.com/explore/tags/p5js/?hl=en) posts with the #p5js hashtag - This is a dynamic web page. As users create more posts with the #p5js hashtag, the DOM will automatically update the HTML to include these new images and videos. |
| **Teacher Demo: Text Inputs** |
| * To transition to the day’s demo, explain that capturing text that a user types is a necessary part of most dynamic websites. In the previous Instagram example, that page was generated after using a search bar with a **text input field**:      * Build out [this example](https://editor.p5js.org/mparker/sketches/iEVOI8yZ9) with your students that modifies an HTML element using the text from an input field. It may be helpful to share this [starter code](https://editor.p5js.org/mparker/sketches/JllO6u2X4) with the button and header pre-coded to save time during the demo. As students code along, emphasize the following:   + To create a text input field, you’ll need to use a new function called createInput(). It has two optional parameters. The first allows you to put default text inside the box (usually an instruction to the user, like, “Write here”). The second describes the type of text that will be entered.   + To capture the text that the user types into the box, you need to use a new object function called value() and save the result inside a variable.   + The button and the text input have no inherent relationship to each other. If you want something to happen with the text when the button is pressed, you need to use value() inside the button’s callback function.   + If you’re not using the canvas in an example, you can hide it by saving it inside a variable and using the hide() function. * Once you’ve completed the demo with your students, change the second argument on line 8 from “text” to “password”.   + Ask students what they notice and why it might be happening |
| **Student Activity: Password Compromised!** |
| * For this activity, students will practice creating text inputs and displaying their values by creating a sketch [like this one](https://editor.p5js.org/mparker/present/B0LKN33h) that displays a user’s login information, including their password, when they hit a button. Sketches should include the following:   + HTML headers with the name and description of an imaginary website   + An HTML paragraph with a prompt to create a new account.   + A text input with the argument type “text” that asks for a username   + A text input with the argument type “password” that asks for a password   + A “submit” button that displays the username and password on the screen when it is clicked. * Note: When testing their sketches, tell students to only type fake passwords, not ones they actually use to login to sites. * Students who finish early can:   + Make the password as gigantic as possible by changing the styling of the HTML using CSS.   + Use the [p5.speech](https://github.com/IDMNYU/p5.js-speech) library to make the computer speak the password out loud! Copy the code below into the <head> of the index.html file, and use the speak() function on the text you’d like spoken:   <script src="https://cdn.jsdelivr.net/gh/IDMNYU/p5.js-speech/lib/p5.speech.js"></script> |
| **Wrap Up** |
| * **[Design Journal]** Have students answer the following prompts:   + In your own words, what is the main difference between a static and dynamic web page?   + How do you “connect”a button to a text input using code? In today’s examples, why was the user’s text displayed when the button was pressed?   + Does using the password type for a text input make that information safe? Why or why not? |
| **Extensions: N/A** |
|  |