From Nouns to Navigation: Madlibs HTML Adventures

Summary:
In this lesson, students will learn how to make a Madlib game using HTML and JavaScript. They will start by planning out their Madlib story, then create an HTML form with input fields for adjectives, nouns, verbs, and adverbs. Then they will write JavaScript code to generate the Madlib story based on the user's input.

Audience: 4-12th Graders with good keyboarding skills.  
Plugged or unplugged: Plugged

California State Standards for Language Arts:
- CCSS.ELA-LITERACY.W.4.3: Write narratives to develop real or imagined experiences or events using effective technique, descriptive details, and clear event sequences.
- CCSS.ELA-Literacy.SL.5.2 Summarize a written text read aloud or information presented in diverse media and formats, including visually, quantitatively, and orally.
- CCSS.ELA-Literacy.L.6.3 Use knowledge of language and its conventions when writing, speaking, reading, or listening.
- CCSS.ELA-LITERACY.W.9-10.3: Write narratives to develop real or imagined experiences or events using effective technique, well-chosen details, and well-structured event sequences.
- CCSS.ELA-LITERACY.W.11-12.3: Write narratives to develop real or imagined experiences or events using effective technique, well-chosen details, and well-structured event sequences.
- California State Standards for Computer Science:
- CSTA.CK-12.CSD.L2-05: Write programs that utilize variables to store and manipulate data.
- CSTA.CK-12.CSD.L2-11: Create interactive programs that use user input to make decisions.
Materials needed:
- A text editor (Notepad)
- A web browser
- An internet connection to download any necessary files or resources

Prerequisite skills:
- Open programs independently
- Copy+Paste Commands

Time Estimate:
45 minutes

Objectives:
1. Students will complete a form
2. Students will navigate the provided web page using either a screenreader or a screen magnification system/software
3. Students will edit a text document to add their own variables to a Madlibs HTML page

Instructional Sequence:

Step 1: Plan Your Madlib
Before students start coding their Madlib, they should plan it out. A Madlib is a game where one person asks another person to fill in the blank spaces in a story with random words, without knowing what the story is about. Start by writing a story with several blanks for nouns, verbs, adjectives, and adverbs. Students can use this template and modify it as needed.

Step 2: Create the HTML Form
```html
<!DOCTYPE html>
<html>
<head>
  <title>Page Title Here</title>
</head>
<body>
  <h1>Activity Name</h1>
  <p>Fill in the blanks to create a story:</p>
  <form>
    <label for="animal">Enter a type of animal:</label>
    <input type="text" id="animal" />
  </form>
</body>
</html>
```
These can be edited at will once the student has understood the relationship between the tags and the output text, and how everything is connected. One change at the top will impact everything at the bottom, so it needs to line up!

Step 3: Creating a script function!

First, an explanation! The button at the end of the form doesn't do anything unless it is told exactly what to do, when to do it, and under what conditions. For our example, we are just telling it what to do after it has been clicked. That's what onclick means. So the function "createStory" will run once the button is clicked, and needs to be activated by a script. The code is as follows:

```javascript
<script>
function createStory() {
    var yourAnimal = document.getElementById("animal").value;
    var yourName = document.getElementById("name1").value;
    var yourOccupation = document.getElementById("occupation").value;
    var yourAdj1 = document.getElementById("adj1").value;
    var yourName2 = document.getElementById("name2").value;
    var yourName3 = document.getElementById("name3").value;
    // Further logic here...
}
</script>
```
| Step 4 : Writing the Madib and putting in the variables! | This is the part where students can write out their Madlib under the code as follows: document.write( "Quotation marks are needed for the computer to display the text as text, and not try to interpret it as code. To add a variable, you would close the quotation marks and add" + yourAnimal + "the following sentence(s)." ); } </script>  

The beauty of this activity is that it can be as short or as long as you want it to be!  

The variables (var) can be changed to anything. For example, you can make it read:  

```javascript  
var donut1 =  
```

However, the donut1 variable is created in the script, and has to call the answer that the user put in at the top, in the form. So you would go back up to that line and change it to  

```html  
<label for="donut1">Enter a type of donut:</label>  
<input type="text" id="donut1" /></br>""  
```

NOTE: If you copy and paste code from a Google Doc to another text editor like Notepad, the quotation marks won't translate the same and the code likely will not work as intended. They need to either be replaced inside of the Notepad, or typed out.