

## Mushroom Makers - Lesson 2: CSS Styling & JS Interactivity

Ready to turn your text story into a playable Choose-Your-Own-Adventure RPG? Let's connect CSS and JavaScript!

### Step 1: Connecting CSS and JS to HTML

Ensure your *index.html* file links your CSS stylesheet in the `<head>` (between `<head>` and `</head>`) and your JavaScript script right before `</body>` (copy the parts in **bold** into your *index.html* file):

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Choose Your Own Adventure</title>

<!-- Link CSS -->
<link rel="stylesheet" href="style.css">

</head>
<body>

<!--Your Story Content Goes Here -->

<!-- Link JavaScript -->
<script src="script.js">
</script>

</body>
</html>
```

### Step 2: Updating our HTML to prepare for interactivity (html)

Replace the code in your *index.html* with this interactive story structure (you can change the words of the story however you like):

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Mushroom Quest</title>
<link rel="stylesheet" href="style.css">
</head>
```

```

<body>

<h1 id="main-title">🍄 The Spore Citadel</h1>

<!-- SCENE 1 -->
<div id="scene1" class="story-box">
  <p class="story-text">
    You stand before two glowing caves. The left cave smells like spicy soup. The right
    cave hums with strange energy.
  </p>
  <button class="choice-btn" onclick="goToScene('scene1', 'scene2')">Enter the Spicy
  Cave</button>
  <button class="choice-btn" onclick="goToScene('scene1', 'scene3')">Enter the
  Humming Cave</button>
</div>

<!-- SCENE 2 (Initially Hidden) -->
<div id="scene2" class="story-box hidden">
  <p class="story-text">
    You enter the spicy cave and find a friendly Goblin Chef cooking Mushroom Stew. He
    offers you a bowl!
  </p>
  <button class="choice-btn" onclick="goToScene('scene2', 'scene1')">Go Back to
  Entrance</button>
</div>

<!-- SCENE 3 (Initially Hidden) -->
<div id="scene3" class="story-box hidden">
  <p class="story-text">
    A giant Robo-Mushroom blocks your path! Its red visor blinks dangerously.
  </p>
  <button class="choice-btn" onclick="goToScene('scene3', 'scene1')">Run Back
  Outside!</button>
</div>

<script src="script.js"></script>
</body>
</html>

```

### Notes:

`<!-- Blabla -->` is an html comment. It doesn't affect the code; it is just a note to help you read your code and find parts of your code later. These are very useful, especially when your project becomes more complex.

The **onclick** connects to the JavaScript that we will write later. First, let's style our HTML with some CSS.

### Step 3: Styling with CSS (style.css)

We can use CSS to make your html output more beautiful. In CSS:

- Target an **id** using # so it will look like: `#main-title`
- Target a **class** using . so it will look like: `.story-text`

Open style.css file and add styles:

```
/* Page Background and Fonts */
body {
  background-color: #1a1a2e;
  color: #e0e0e0;
  font-family: 'Courier New', monospace;
  padding: 20px;
}

/* Styling an ID */
#main-title {
  color: #ff4757;
  text-shadow: 2px 2px #5f27cd;
}

/* Styling Classes */
.story-box {
  background-color: #16213e;
  border: 2px solid #00d2d3;
  border-radius: 8px;
  padding: 20px;
  margin-bottom: 20px;
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.5);
}

.story-text {
  font-size: 18px;
  line-height: 1.5;
}

/* RPG Buttons */
.choice-btn {
  background-color: #ff9f43;
  color: #1a1a2e;
  border: none;
  padding: 10px 15px;
  font-weight: bold;
  cursor: pointer;
}
```

```
border-radius: 4px;
margin-right: 10px;
}

.choice-btn:hover {
background-color: #ee5253;
color: #fff;
}

/* Helper class to hide scenes */
.hidden {
display: none;
}
```

**Note:**

*/\* Blabla \*/* is a CSS comment. It doesn't affect the code; it is just a note to help you read your code and find parts of your code later. These are very useful, especially when your project becomes more complex.

Look at the Mushroom Makers - CSS Quick Reference Dictionary worksheet. You can play with different styles. Try out different ways to make your html output look more interesting.

#### Step 4: Adding Interactivity with JavaScript (script.js)

We can use JavaScript to change scenes when a button is clicked.

Add this function to your *script.js*:

```
// Function to switch scenes
function goToScene(hideSceneId, showSceneId) {
// Hide the current scene
document.getElementById(hideSceneId).classList.add('hidden');

// Show the next scene
document.getElementById(showSceneId).classList.remove('hidden');
}
```

**Note:**

*// Blabla* is a JavaScript comment. It doesn't affect the code; it is just a note to help you read your code and find parts of your code later. These are very useful, especially when your project becomes more complex.

What this JavaScript does:

A **function** in JavaScript is a small instruction for your computer. You give the function a name and put actions inside it. The code inside the function does not run right away—it waits until you **call** its name.

In our game, we created a function called *goToScene(hideSceneId, showSceneId)*. This does the following:

- **The Name:** *goToScene* tells the browser what action to do.
- **The Inputs (Parameters):** *hideSceneId* and *showSceneId* act as **placeholders** (or special **variables**). When you define the function, these placeholders wait for real information. When a player clicks a button, you send two specific scene names into these variables. So, for example, in SCENE 1 the first button does `onclick="goToScene('scene1', 'scene2')"` which tells *hideSceneId* to hide scene 1, and tells *showSceneId* to show scene 2. Notice how the code in *index.html* is communicating with the code in *script.js*.
- **The Action:** The function hides the old scene and shows the new scene on your screen instantly.

Instead of writing the same steps for every single button in your story, you write the function **once** in JavaScript and re-use it whenever a player makes a choice.

### Challenge! 🏆

1. Can you add a **Scene 4** and connect new buttons to create a branching multi-ending adventure game?
2. Try adding an image to each part of the story or changing the CSS styles.