Here's a detailed assignment for the "DOM Manipulation" topic that focuses on using JavaScript to dynamically change the content of a webpage in response to user events.

---

## \*\*Assignment: DOM Manipulation\*\*

### \*\*Objective\*\*

The objective of this assignment is to practice DOM manipulation using JavaScript. You will create a webpage that responds to user interactions by dynamically updating its content.

### \*\*Task Description\*\*

You are required to create a simple HTML page that utilizes JavaScript to change the content of specific elements when an event occurs, such as a button click. This assignment will help you understand how to manipulate the DOM and enhance user experience through interactive features.

### \*\*Requirements\*\*

1. \*\*HTML Structure\*\*:

- Create a new HTML file (e.g., `dom-manipulation.html`) with the following elements:

- A header (`<h1>`) with a title for your page.

- A paragraph (`<p>`) with initial text that will be updated.

- A button (`<button>`) that triggers the content change when clicked.

- Optional: Add additional elements (e.g., a list or image) to demonstrate more complex changes.

2. \*\*JavaScript Functionality\*\*:

- Write a `<script>` section in your HTML file or link to an external JavaScript file (e.g., `dom-manipulation.js`).

- Use the following requirements to implement DOM manipulation:

\*\*a. Event Listener\*\*:

- Add an event listener to the button that listens for the `click` event.

\*\*b. Content Update\*\*:

- Use `document.getElementById()` to select the paragraph element by its ID.

- Update the `innerHTML` property of the paragraph to change its text when the button is clicked.

- Optionally, you can also change the text of the header or add additional content.

3. \*\*Interactive Features\*\*:

- Experiment with additional features, such as:

- Changing the style (e.g., color or font size) of the updated text using inline styles.

- Adding a countdown or timer that updates the text every few seconds.

- Allowing multiple clicks to change the content to different messages.

### \*\*Example HTML Structure\*\*

Here’s a sample structure of how your HTML might look:

```html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>DOM Manipulation Example</title>

<link rel="stylesheet" href="styles.css"> <!-- Optional CSS link -->

</head>

<body>

<h1 id="page-title">Welcome to My Webpage</h1>

<p id="text-content">This is the initial content of the paragraph.</p>

<button id="change-text-button">Click to Change Text</button>

<script src="dom-manipulation.js"></script> <!-- Link to your JavaScript file -->

</body>

</html>

```

### \*\*Example JavaScript Code\*\*

Here’s a sample JavaScript code snippet to help you get started:

```javascript

// dom-manipulation.js

// Function to change the text content

function changeText() {

// Select the paragraph element by ID

var paragraph = document.getElementById("text-content");

// Update the innerHTML of the paragraph

paragraph.innerHTML = "The content has been updated! Click the button again to change it again.";

}

// Add event listener to the button

document.getElementById("change-text-button").addEventListener("click", changeText);

```

### \*\*Submission Guidelines\*\*

- Submit your HTML file (`dom-manipulation.html`) and your JavaScript file (`dom-manipulation.js`).

- Ensure that your JavaScript code is functioning correctly and updates the content as expected.

- Validate your HTML and JavaScript for any errors.

### \*\*Evaluation Criteria\*\*

- Proper implementation of event listeners and DOM manipulation techniques.

- Clarity of code organization and use of comments to explain the functionality.

- Creativity in the design and functionality of the interactive elements on the page.

- Overall user experience and responsiveness of the webpage.

### \*\*Additional Tips\*\*

- Use browser developer tools to debug and test your JavaScript code.

- Experiment with different event types (e.g., `mouseover`, `dblclick`) to see how they affect the interaction.

---

This assignment encourages students to explore the basics of DOM manipulation, allowing them to create interactive and dynamic web pages using JavaScript.