**Easy Questions:**

1. **Basic Click Event Listener:**
   * Create a button that, when clicked, changes the text inside a <p> tag.

Example:

html

Copy code

<button id="myButton">Click Me</button>

<p id="text">Original Text</p>

Task: Write JavaScript to change the <p> content to "Text Changed!" on button click.

1. **Mouseover Event Listener:**
   * Create a <div> that changes its background color when the mouse hovers over it and resets when the mouse leaves.
2. **Keyboard Event Listener:**
   * Write a function that logs every key the user presses on their keyboard in the console.
3. **Adding Event Listeners to Multiple Elements:**
   * Given a list of items, attach a click event listener to each item. When clicked, log the item text in the console.

**Medium Questions:**

1. **Toggle Event Listener:**
   * Create a button that toggles the visibility of a <div>. When clicked, it hides the <div>, and when clicked again, it shows the <div> back.
2. **Event Listener Removal:**
   * Write a function to add an event listener to a button that logs "Clicked!" when clicked. Then, after the first click, remove the event listener so it doesn't log on subsequent clicks.
3. **Prevent Default Action:**
   * Create a form with a text input and submit button. Attach a submit event listener that prevents the default form submission and logs "Form Submission Prevented" instead.
4. **Event Delegation:**
   * Create a parent <div> that contains multiple child <button> elements. Attach a click event listener to the parent <div> to handle click events on any button without adding individual listeners to each button.

**Hard Questions:**

1. **Custom Event Listener:**
   * Create a custom event called customClick that fires when a user clicks a button. When the custom event is triggered, log "Custom Event Triggered!" in the console.
   * **Objective**: Create and dispatch a custom event when a specific action occurs (e.g., a user clicks a button three times).
   * **Instructions**:
   1. Create a button that triggers a custom event after being clicked three times.
   2. When the custom event is fired, display a message below the button.

html

Copy code

<button id="customEventButton">Click Me</button>

<p id="customEventMessage"></p>

<script>

// Write the JavaScript code here (handle custom event creation and dispatch)

</script>

1. **Event Propagation (Bubbling vs. Capturing):**
   * Create a nested structure with two <div> elements. Attach a click event listener to both the parent and child <div>. Demonstrate event propagation by logging messages for both capturing and bubbling phases when clicking the child <div>.
2. **Debouncing with Event Listeners:**
   * Implement a search input box that shows a list of suggestions based on what the user types. Use event debouncing to limit how often the suggestions are updated as the user types.
3. **Throttling a Scroll Event Listener:**
   * Attach a scroll event listener to the window that logs the current scroll position. Use throttling to limit how often the scroll position is logged, even when the user scrolls rapidly.

These exercises will help you practice event listener handling at various levels of complexity.

**Easy Level**

1. **Remove Event Listeners:**
   * Add an event listener to a button that logs "Clicked" to the console when clicked. After three clicks, remove the event listener.

**Hard Level**

1. **Event Propagation with Stop Propagation:**
   * Create a setup with a button inside a div. Attach click event listeners to both the button and the div.
   * Use event.stopPropagation() to stop the event from propagating to the div when the button is clicked.
2. **Multiple Event Types:**
   * Create an input field that changes its border color to green when focused and back to red when blurred.
3. **Throttling or Debouncing an Event:**
   * Implement a search box that logs input only after the user has stopped typing for 300ms (debouncing effect).

**JavaScript Events Assignment**

**Part 1: Easy**

**3. Input Event**

* **Objective**: Capture user input from a text field and display it in real-time.
* **Instructions**:
  1. Create an input field and a <p> element.
  2. As the user types into the input field, display the value inside the <p> element in real-time.

html

Copy code

<input type="text" id="inputField" placeholder="Type something here">

<p id="displayText"></p>

<script>

// Write the JavaScript code here

</script>

**Part 2: Medium**

**5. Form Submit Event**

* **Objective**: Prevent the default form submission and display the form data without reloading the page.
* **Instructions**:
  1. Create a form with an input field for the user's name and a submit button.
  2. On form submission, prevent the default action and display the entered name below the form without refreshing the page.

<form id="nameForm">

<input type="text" id="nameInput" placeholder="Enter your name">

<button type="submit">Submit</button>

</form>

<p id="nameDisplay"></p>

<script>

// Write the JavaScript code here

</script>

**Part 3: Hard**

**8. Drag and Drop Event**

* **Objective**: Implement a drag-and-drop functionality.
* **Instructions**:
  1. Create two <div> elements: one for the draggable item and another as the drop zone.
  2. Allow the user to drag the item from one <div> to another using drag-and-drop events (dragstart, dragover, drop).

html

Copy code

<div id="dragContainer">

<div id="dragItem" draggable="true">Drag Me!</div>

</div>

<div id="dropZone">Drop Here</div>

<script>

// Write the JavaScript code here

</script>

**9. Throttle Scroll Event**

* **Objective**: Optimize a scroll event using throttling.
* **Instructions**:
  1. Create a page with some content and a scrollable area.
  2. Add an event listener for the scroll event but throttle the event so it doesn't fire continuously. Display the scroll position every time the event fires.
  3. Use a throttling mechanism (either manually or using a utility like lodash.throttle).

html

Copy code

<div id="scrollableContent" style="height: 400px; overflow: scroll;">

<!-- Add a large block of text or content here to make the area scrollable -->

</div>

<p id="scrollPos">Scroll Position: 0</p>

<script>

// Write the JavaScript code here (include throttling mechanism)

</script>