

Events Handling

Mouse Events:

- click: Triggered when the user clicks an element.
- dblclick: Triggered when the user double-clicks an element.
- mousedown, mouseup: Triggered when the mouse button is pressed or released over an element.
- mousemove: Triggered when the mouse pointer moves over an element.
- mouseover, mouseout: Triggered when the mouse enters or leaves an element.

Keyboard Events:

- keydown, keyup: Triggered when a key is pressed or released while the focus is on an element.
- keypress: Triggered when a key is pressed, but not for keys like Shift, Ctrl, or Alt.

Form Events:

- submit: Triggered when a form is submitted.
- change: Triggered when the value of an input, select, or textarea element changes.
- input: Similar to change, but triggered immediately when the value changes (works well with real-time input).

Focus Events:

- focus, blur: Triggered when an element gains or loses focus.
- focusin, focusout: Similar to focus and blur, but they bubble up through the DOM.

Window Events:

- load: Triggered when the entire page (including all dependent resources) has finished loading.
- unload: Triggered when the user navigates away from the page.
- resize: Triggered when the browser window is resized.

Document Events:

- DOMContentLoaded: Triggered when the HTML document has been completely loaded and parsed, without waiting for stylesheets, images, and subframes to finish loading.
- readystatechange: Deprecated. Triggered when the readyState property of the document changes.

Media Events:

- play, pause: Triggered when a media element (like an audio or video player) starts or stops playing.
- timeupdate: Triggered as the playback position of a media element changes.

Touch Events (for mobile devices):

- touchstart, touchmove, touchend: Triggered when a touch point is placed on the touch surface, moved, or removed.

Drag and Drop Events:

- dragstart, dragenter, dragover, dragleave, dragend, drop: Events related to dragging and dropping elements.

KeyPress Event

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Keyboard Events Example</title>

```
</head>
```

```
<body>
```

```
<script>
```

```
    // Function to handle keydown event
```

```
    function handleKeyDown(event) {
```

```
        // Get the pressed key
```

```
        const pressedKey = event.key;
```

```
        // Display the pressed key in the console
```

```
        console.log("Key pressed: " + pressedKey);
```

```
    }
```

```
    // Add a keydown event listener to the document
```

```
    document.addEventListener("keydown", handleKeyDown);
```

```
</script>
```

```
</body>
```

```
</html>
```

Form submit Event :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Form Submit Event Example</title>
</head>
<body>

  <!-- HTML form with input fields -->
  <form id="myForm">
    <label for="username">Username:</label>
    <input type="text" id="username" name="username" required>

    <label for="password">Password:</label>
    <input type="password" id="password" name="password"
required>

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

  <script>
    // Function to handle form submission
    function handleFormSubmit(event) {
      // Prevent the default form submission behavior
      event.preventDefault();
```

```
// Get form data
const formData = new FormData(event.target);

// Log form data to the console
console.log("Form Data:");
for (const entry of formData) {
  console.log(entry[0] + ": " + entry[1]);
}

// Add a submit event listener to the form

document.getElementById("myForm").addEventListener("submit",
handleFormSubmit);
</script>

</body>
</html>
```