Copilot

Event handling in JavaScript allows you to execute code in response to user actions or browser events. Here are some common ways to handle events in HTML and JavaScript:

Inline Event Handling

You can add event handlers directly in your HTML elements using attributes like `onclick`, `onmouseover`, etc.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Event Handling Example</title>
</head>
<body>
    <button onclick="alert('Button clicked!')">Click Me</button>
</body>
</html>
```

Event Handling with JavaScript

A more flexible approach is to use JavaScript to add event listeners to elements. This method keeps your HTML clean and separates the structure from the behavior.

Example: Adding an Event Listener

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Event Handling Example</title>
</head>
<body>
    <button id="myButton">Click Me</button>

    <script>
        document.getElementById("myButton").addEventListener("click", function() {
            alert("Button clicked!");
        });
    </script>
</body>
</html>
```

Explanation

- **Inline Event Handling**: The `onclick` attribute is used directly in the HTML to call a JavaScript function or execute code when the button is clicked.
- **JavaScript Event Handling**: The `addEventListener` method is used to attach an event handler to the button. This method allows you to add multiple event listeners to a single element and remove them if needed.

Common Events

Here are some common events you might handle:

- `onclick`: When an element is clicked.
- `onmouseover`: When the mouse pointer is moved over an element.
- `onmouseout`: When the mouse pointer is moved out of an element.
- `onkeydown`: When a key is pressed down.
- `onload`: When the page has finished loading.

Would you like to see more examples or details on a specific type of event?