Breaking Javascript

Events

by Shaun Kriel

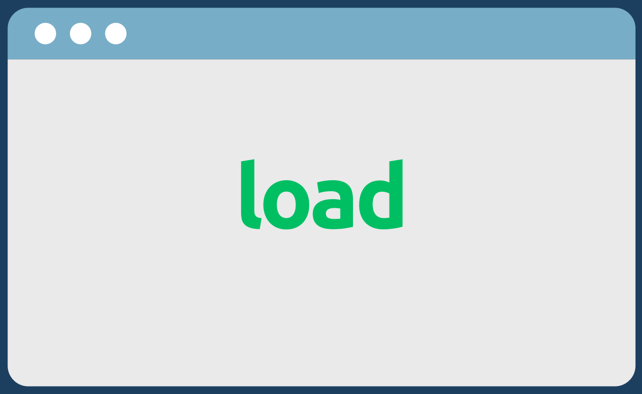
What is an Event in JavaScript ?

An event is like a signal that something has happened on your web page — like a mouse click, a keypress, or a page load.
JavaScript lets us "listen" for these events and respond when they happen.

Mouse Click Event

Keypress Event

Page Load Event



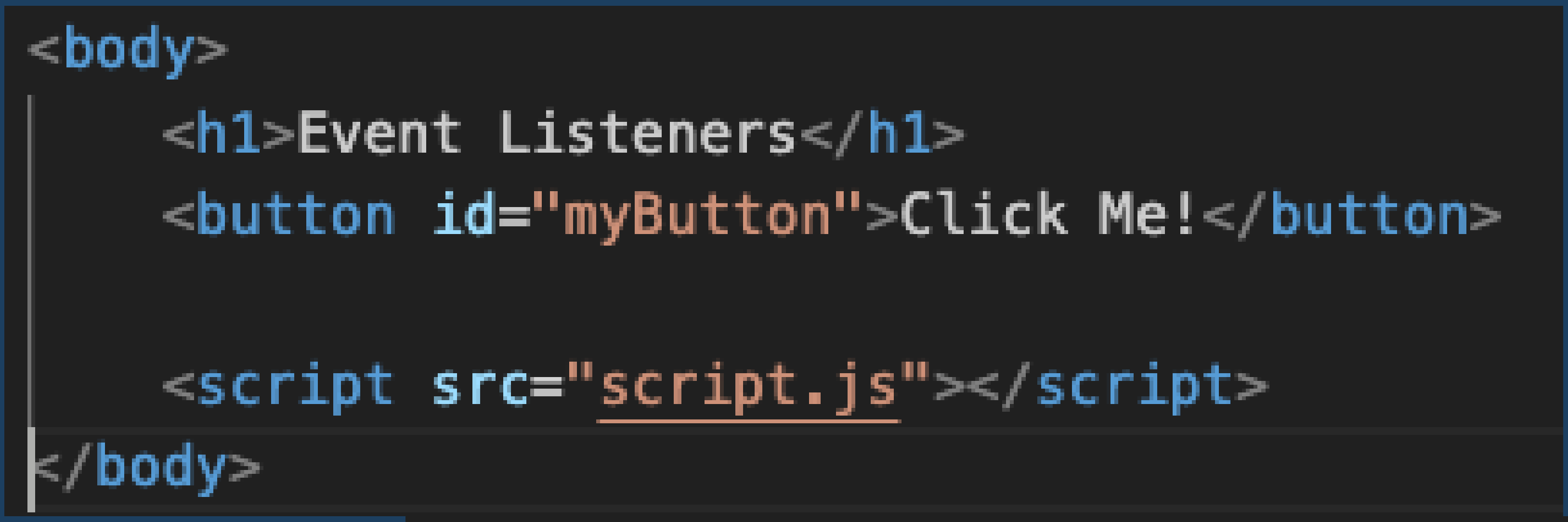
```
element.addEventListener('click', function() {...})

element.addEventListener('keydown', function(e) {...})

element.addEventListener('load', function() {...})
```

Types of 'events':

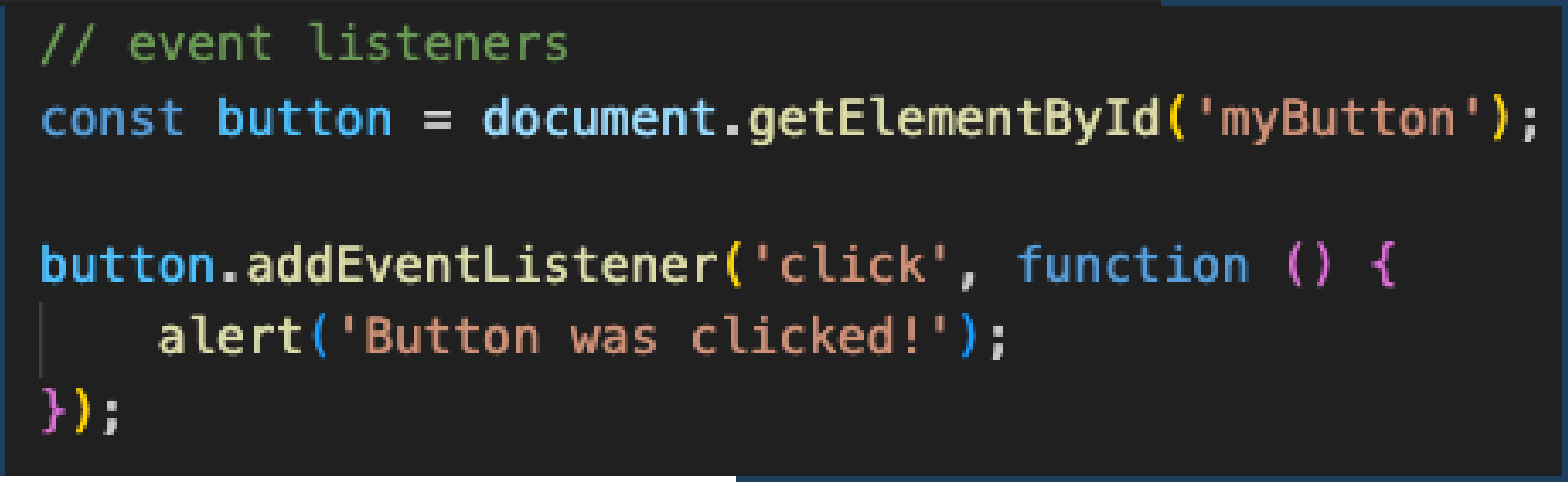
Event Listeners	Mouse Events
Event Objects	Keyboard Events
Event Keycodes	Input Events
Form Submission	Event Bubbling
Event Delegation	Window Events

Event Listeners

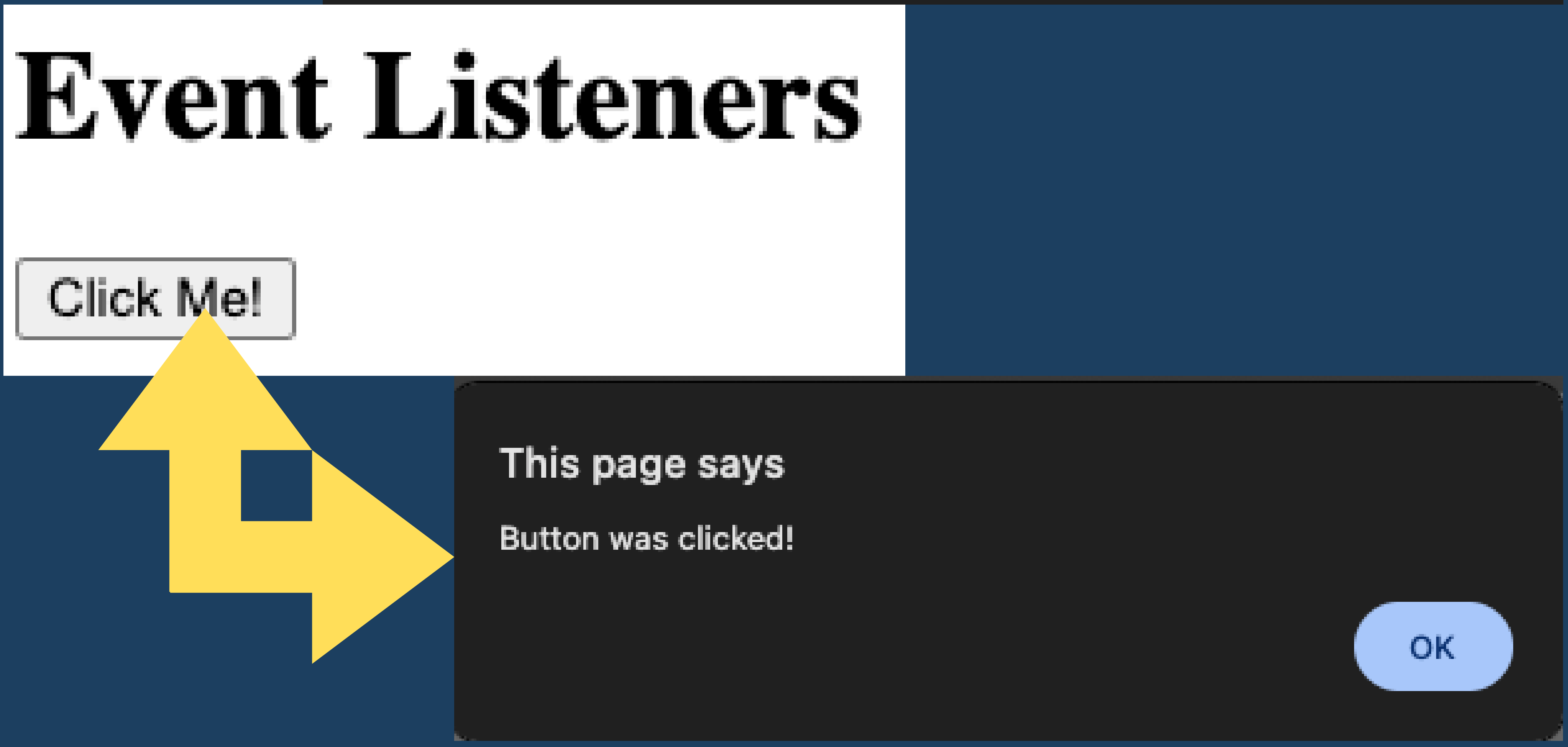
This is how we tell JavaScript to “watch” for a specific event.



```html
<body>
    <h1>Event Listeners</h1>
    <button id="myButton">Click Me!</button>

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



```javascript
// event listeners
const button = document.getElementById('myButton');

button.addEventListener('click', function () {
    alert('Button was clicked!');
});
```



Event Listeners

This page says

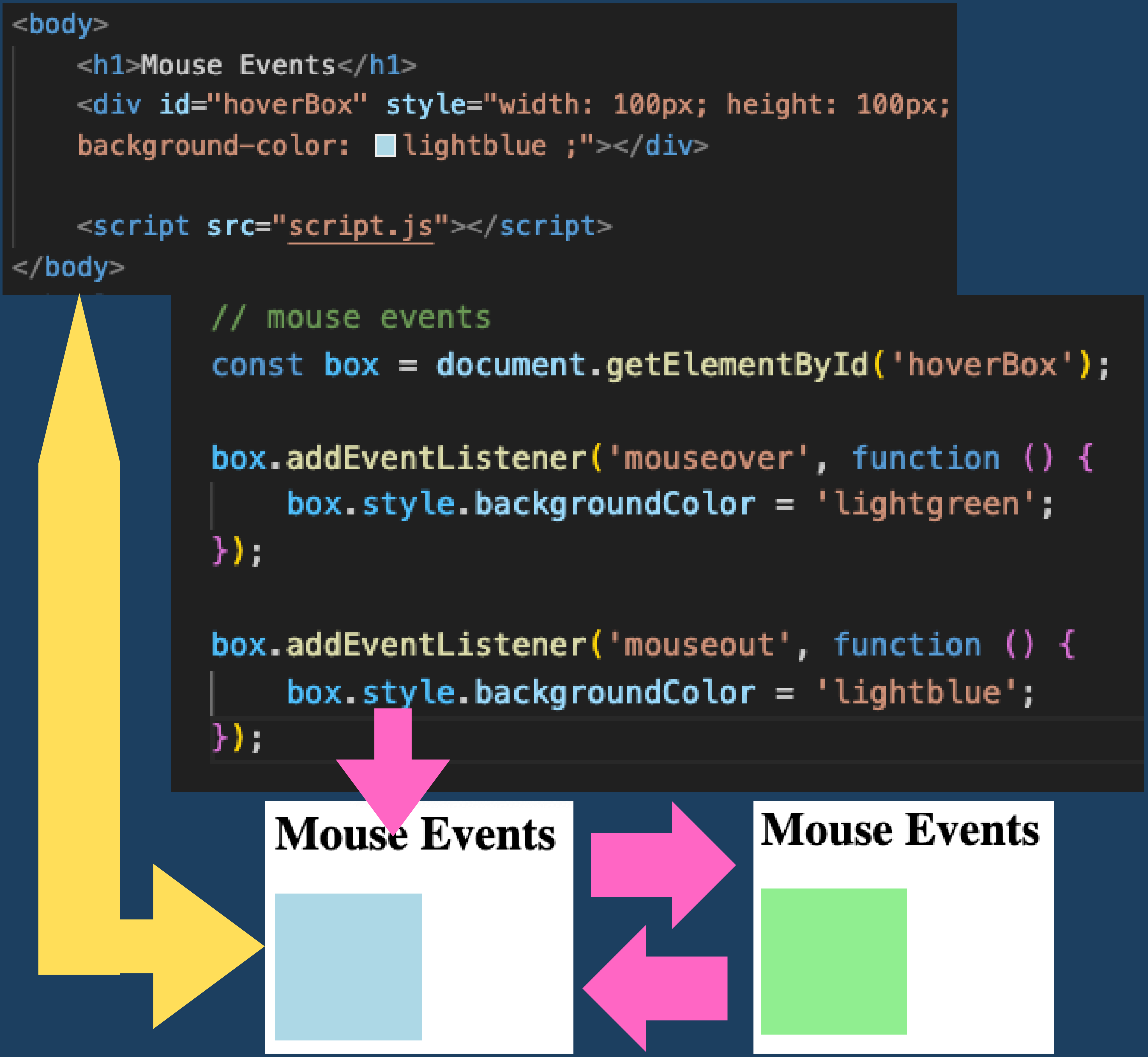
Button was clicked!

Explanation:

- **getElementById('myButton') finds the button in the HTML.**
- **addEventListener('click', function() {...}) tells the browser: "When this button is clicked, run this code."**
- **alert('Button was clicked!') shows a pop-up.**

Mouse Events

Mouse events happen when you use the mouse - click, move, hover



```html
<body>
    <h1>Mouse Events</h1>
    <div id="hoverBox" style="width: 100px; height: 100px;
    background-color: lightblue ;"></div>

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

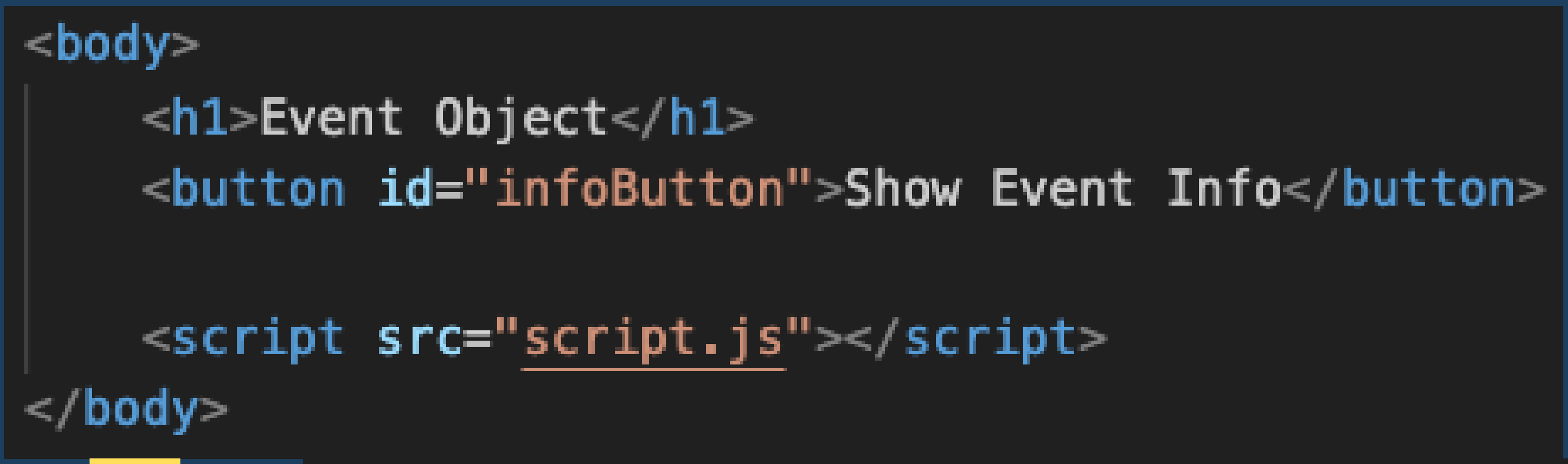
```javascript
// mouse events
const box = document.getElementById('hoverBox');

box.addEventListener('mouseover', function () {
    box.style.backgroundColor = 'lightgreen';
});

box.addEventListener('mouseout', function () {
    box.style.backgroundColor = 'lightblue';
});
```

Explanation:

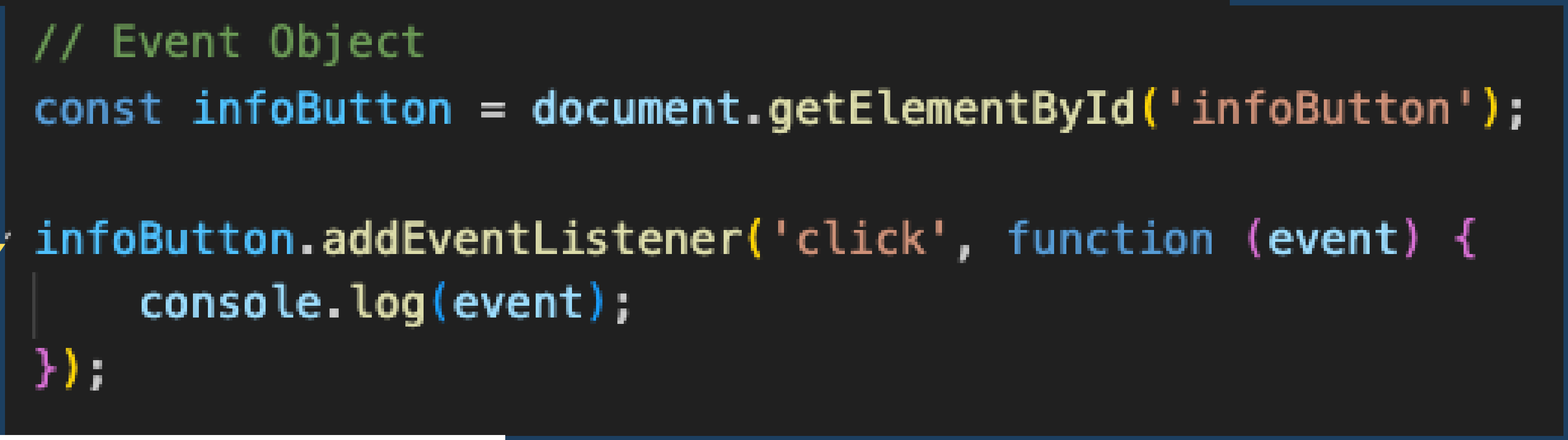
- **When the mouse moves over the box "mouseover", it turns green.**
- **When the mouse leaves the box "mouseout", it turns blue again.**

Event Object

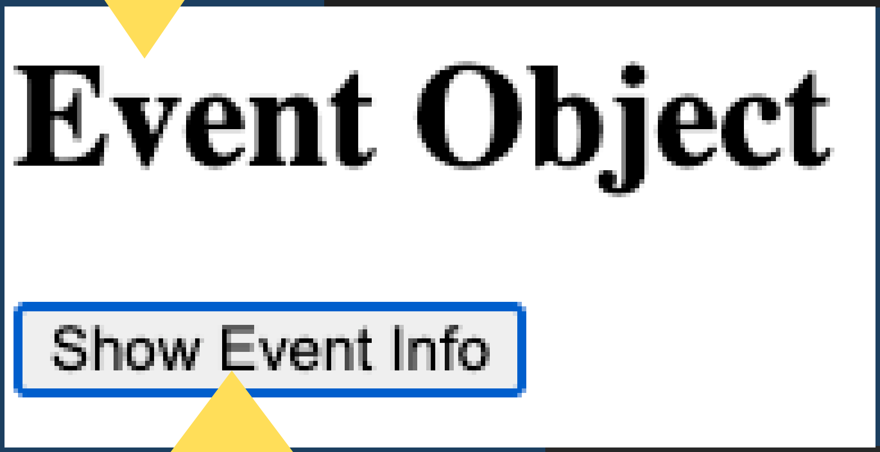
When an event happens, JS automatically gives us an **event object** with lots of details (which button was clicked, which key was pressed).

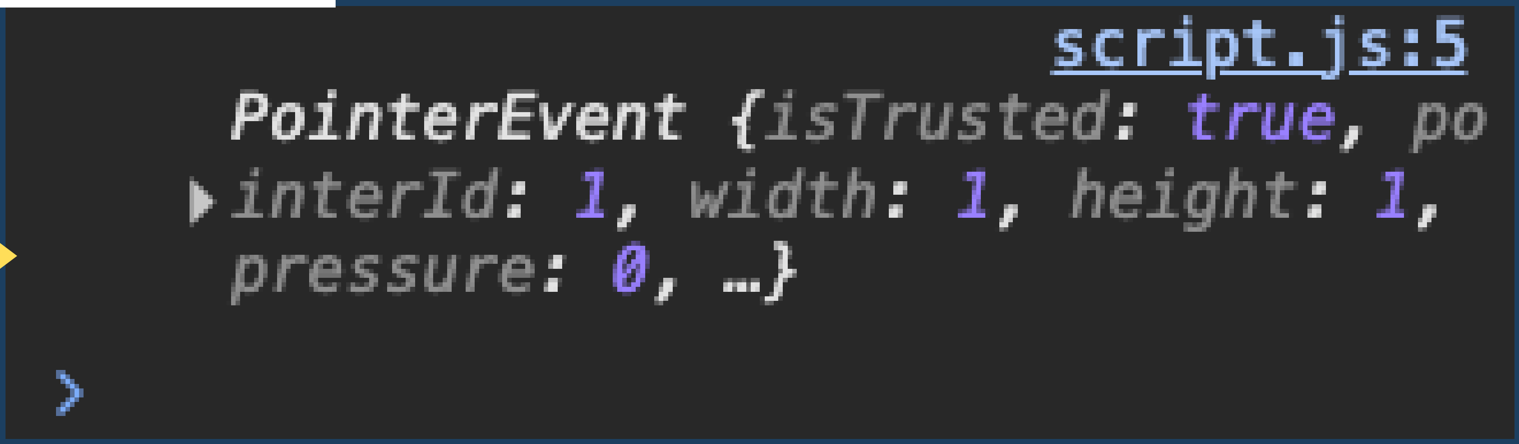


```html
<body>
    <h1>Event Object</h1>
    <button id="infoButton">Show Event Info</button>

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



```javascript
// Event Object
const infoButton = document.getElementById('infoButton');

infoButton.addEventListener('click', function (event) {
    console.log(event);
});
```





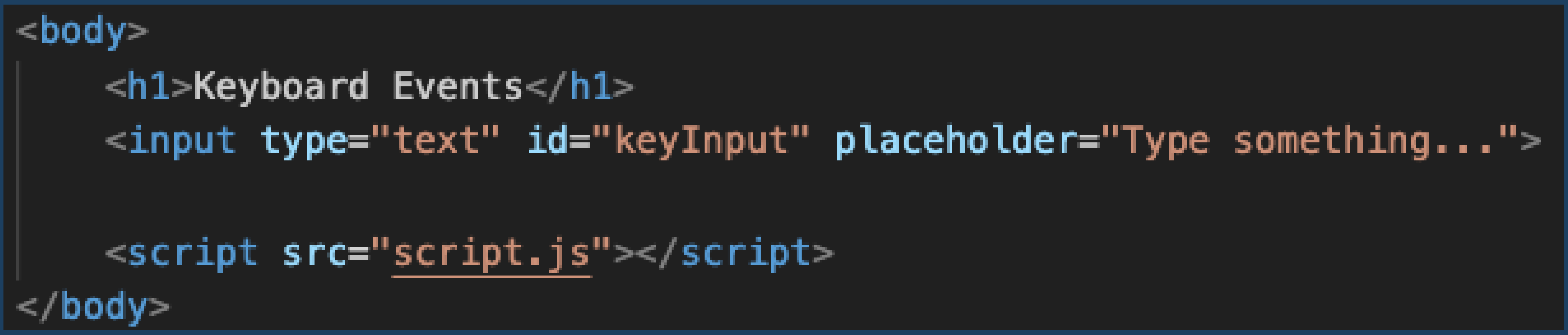
```
                                              script.js:5
  PointerEvent {isTrusted: true, po
▸ interId: 1, width: 1, height: 1,
  pressure: 0, …}
>
```

Explanation:

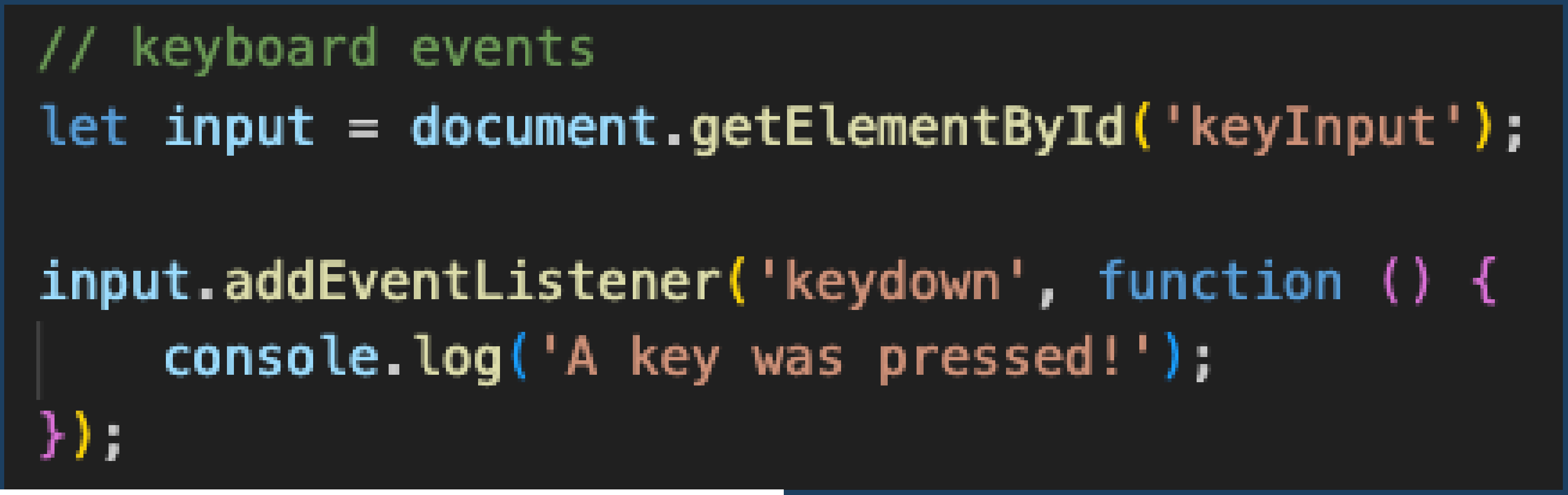
- **The event object contains info like the event type (click), where it happened, and more.**
- **console.log(event) shows all that info in the Console (right-click → Inspect → Console).**

Keyboard Events

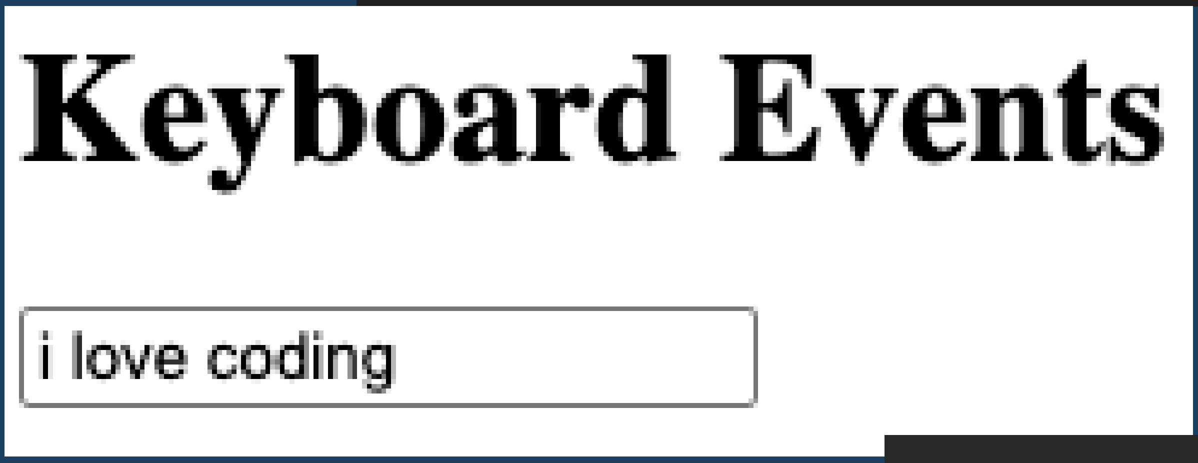
You can listen for when a user **presses** or **releases** a key.



```html
<body>
    <h1>Keyboard Events</h1>
    <input type="text" id="keyInput" placeholder="Type something...">

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

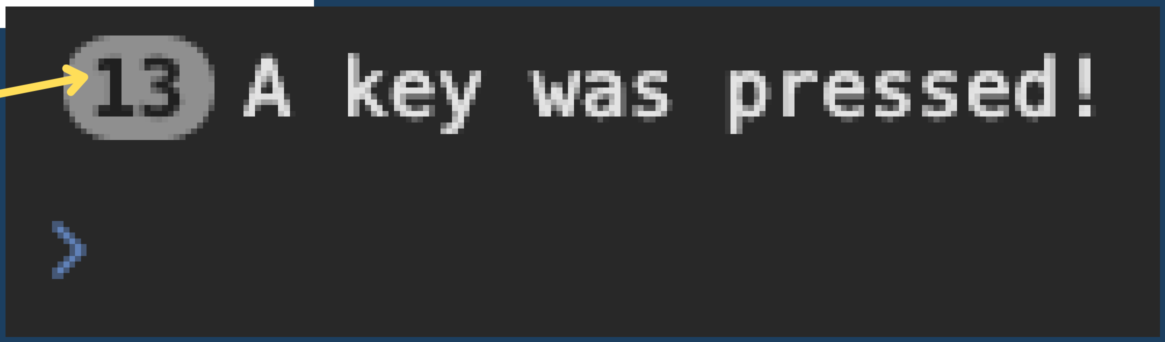


```javascript
// keyboard events
let input = document.getElementById('keyInput');

input.addEventListener('keydown', function () {
    console.log('A key was pressed!');
});
```



Keyboard Events

i love coding



```
13 A key was pressed!
>
```

number of characters incl spaces

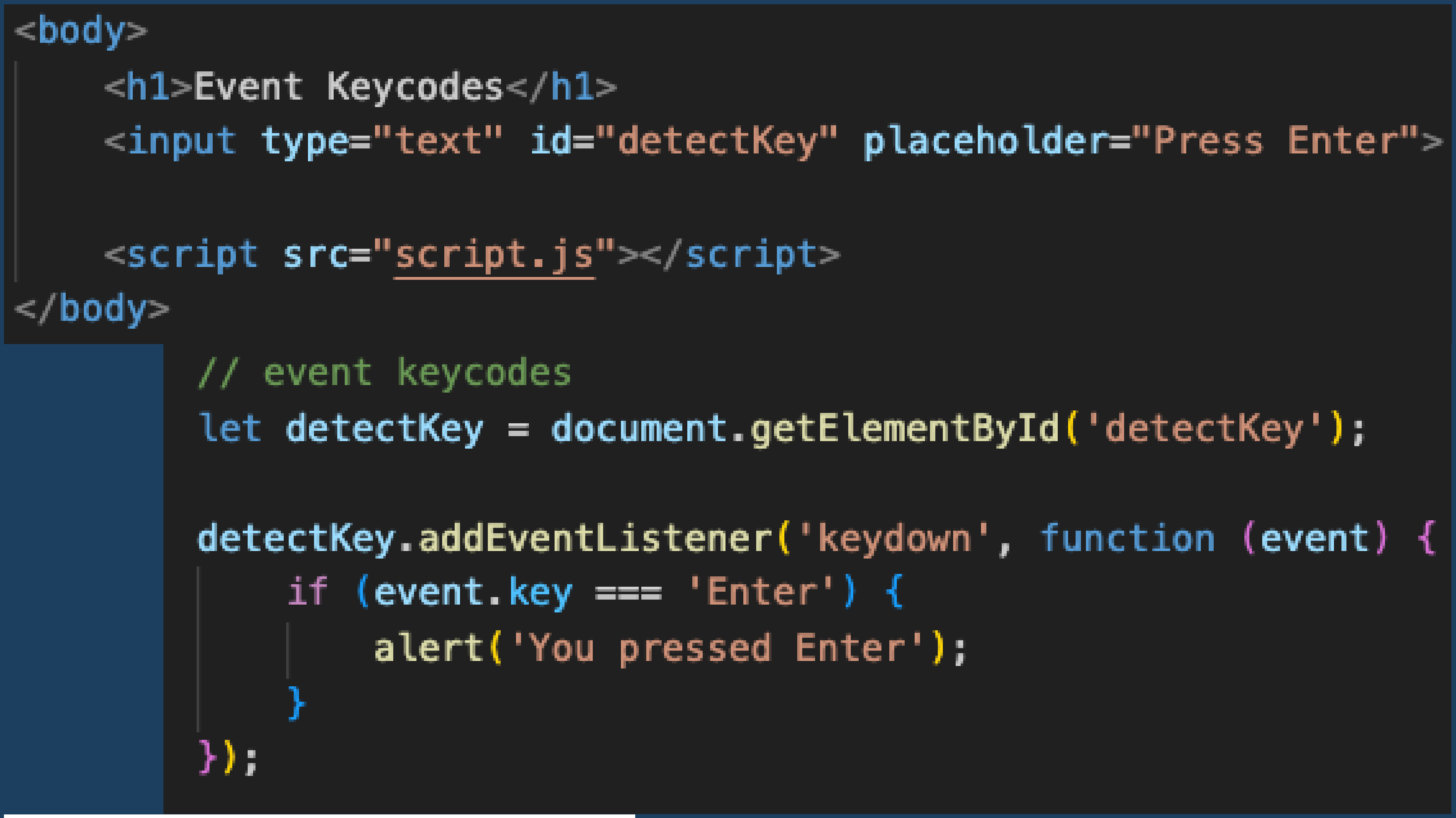
(right-click → Inspect → Console).

Explanation:

- **keydown runs as soon as a key is pressed.**
- **Try typing — every key press shows a message in the console.**

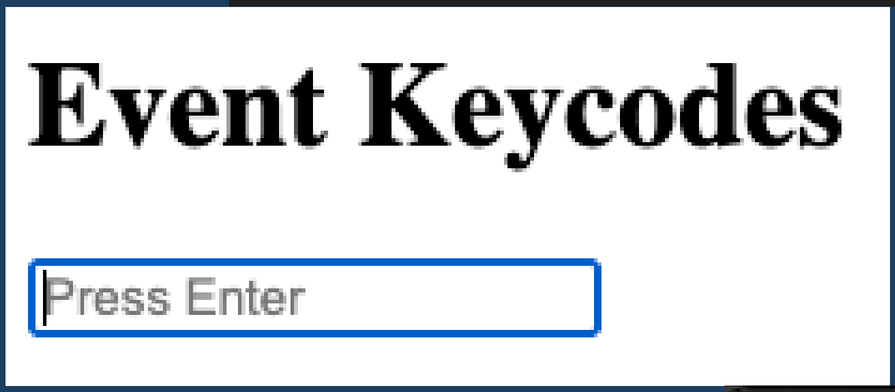
Event Keycodes

You can find **which key** was pressed using event.key or event.keyCode.

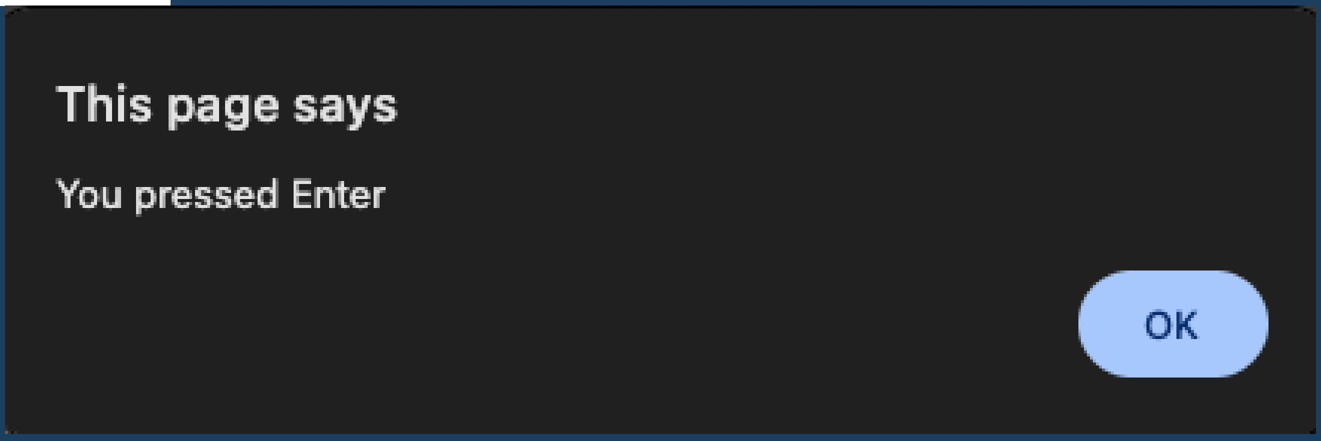


```html
<body>
    <h1>Event Keycodes</h1>
    <input type="text" id="detectKey" placeholder="Press Enter">

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

```javascript
// event keycodes
let detectKey = document.getElementById('detectKey');

detectKey.addEventListener('keydown', function (event) {
    if (event.key === 'Enter') {
        alert('You pressed Enter');
    }
});
```



Event Keycodes



This page says

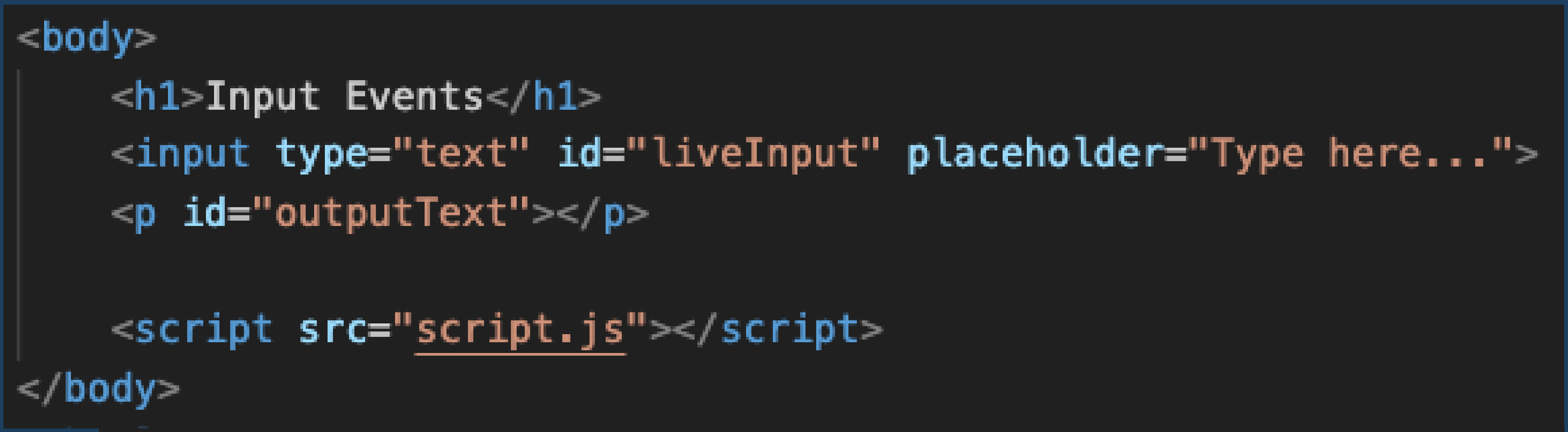
You pressed Enter

Explanation:

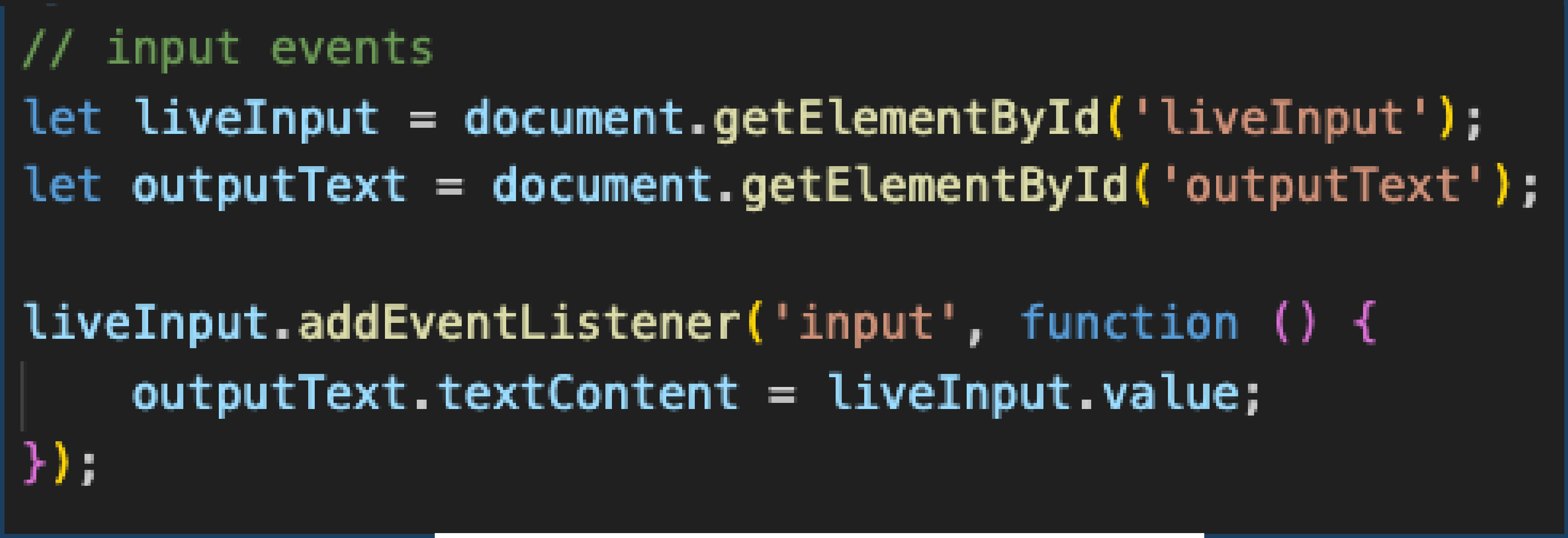
- **event.key gives the name of the key.**
- **If the user presses Enter, it shows an alert.**

Input Events

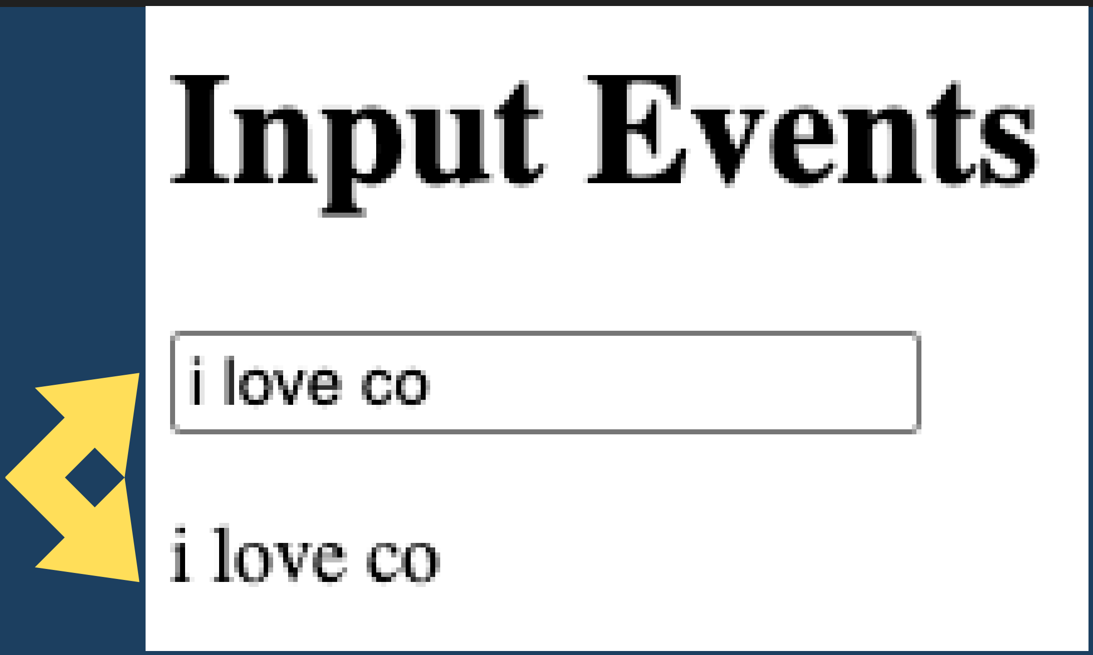
Input events happen when a user **types** or **edits** an input field.



```html
<body>
    <h1>Input Events</h1>
    <input type="text" id="liveInput" placeholder="Type here...">
    <p id="outputText"></p>

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



```javascript
// input events
let liveInput = document.getElementById('liveInput');
let outputText = document.getElementById('outputText');

liveInput.addEventListener('input', function () {
    outputText.textContent = liveInput.value;
});
```



Input Events

i love co

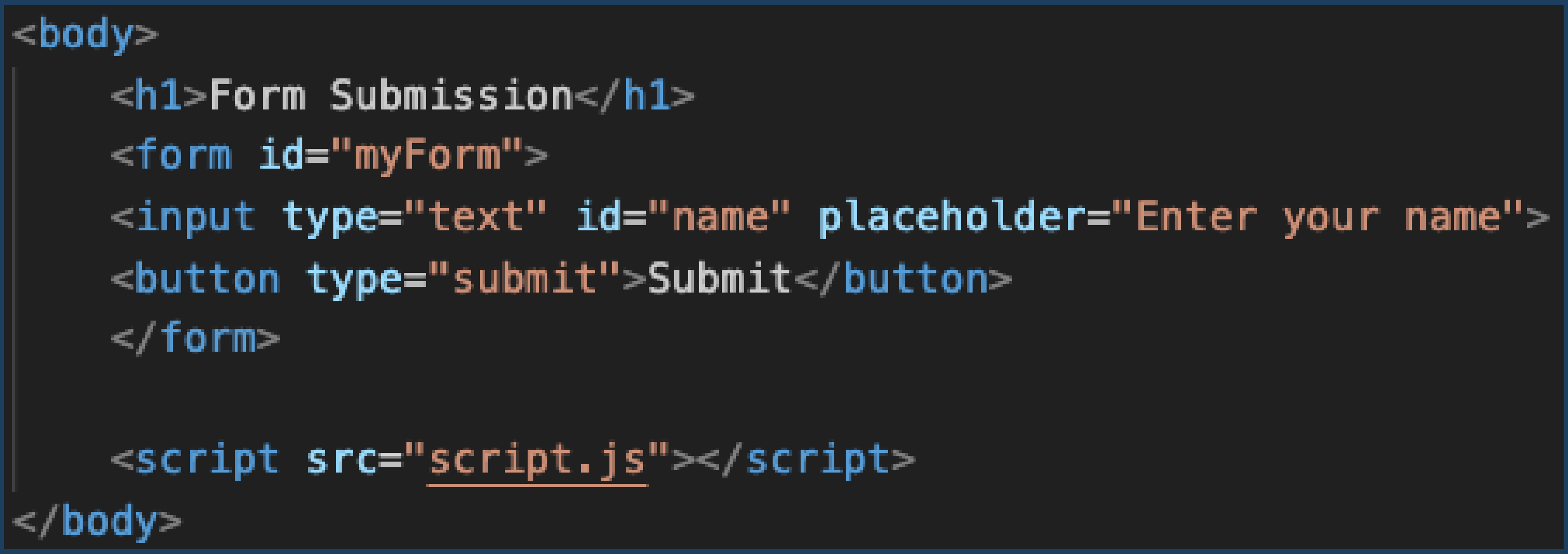
i love co

Explanation:

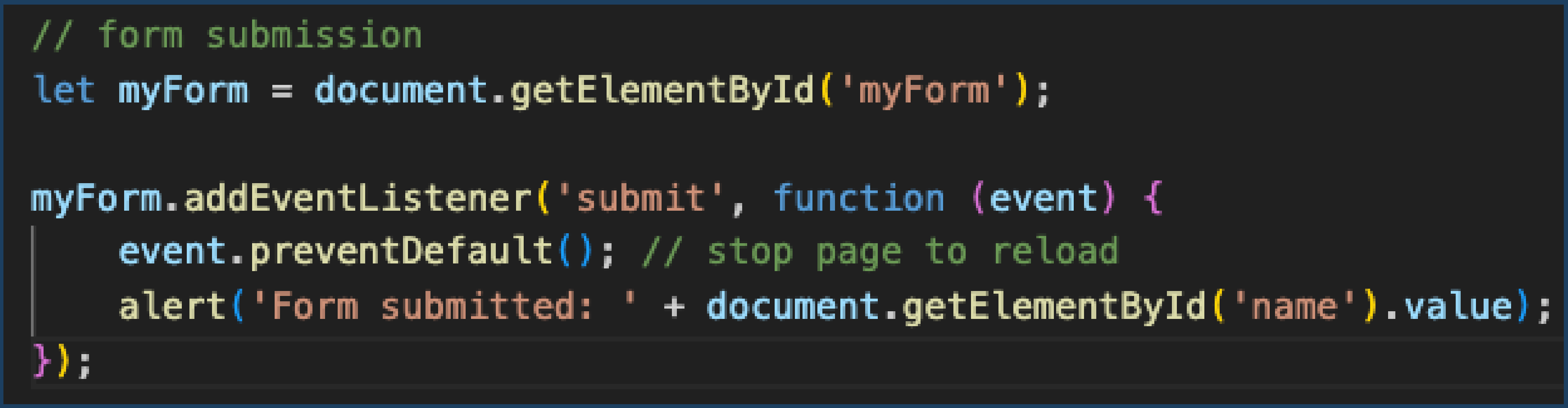
- **As the user types, the text below updates live.**
- **input event fires whenever the input field changes.**

Form Submission

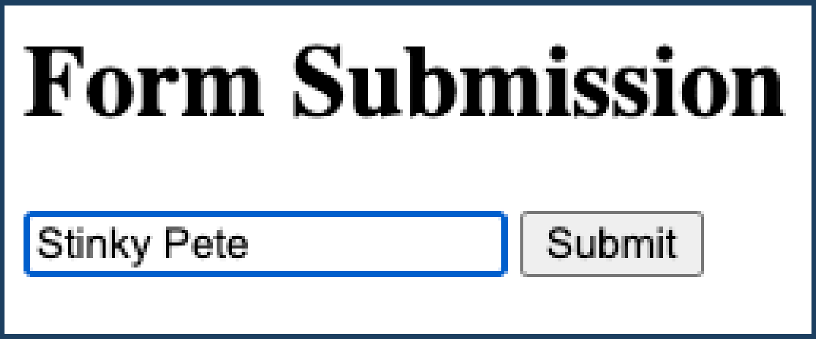
When you submit a form, an event happens, you can catch it to **prevent page** reload and **handle data**.



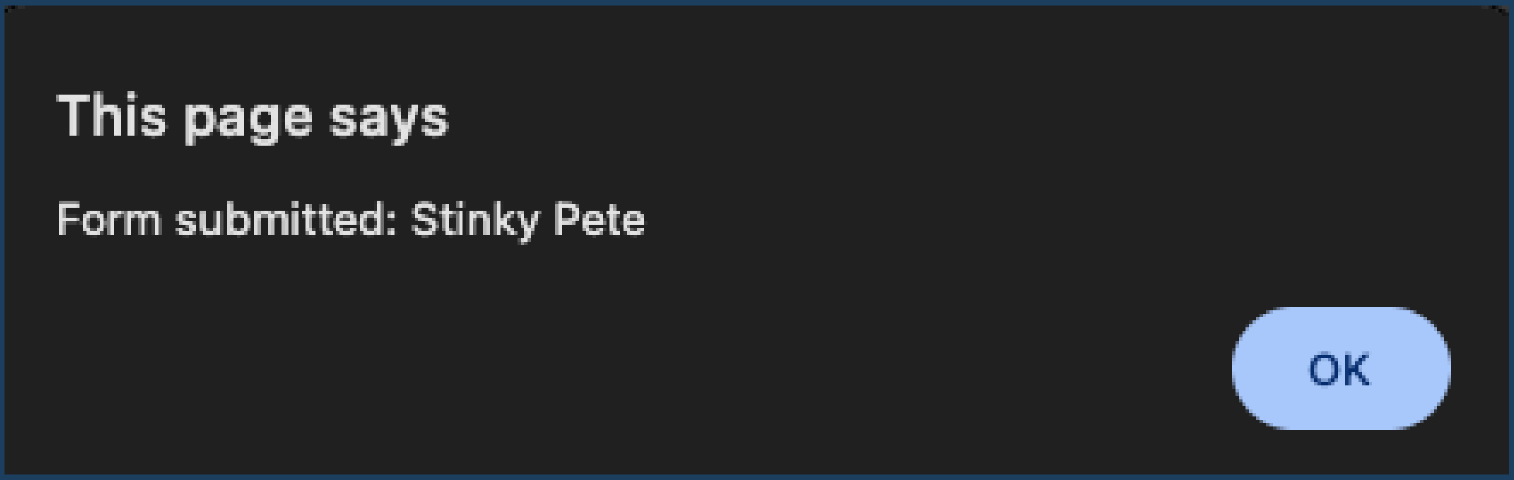
```html
<body>
    <h1>Form Submission</h1>
    <form id="myForm">
    <input type="text" id="name" placeholder="Enter your name">
    <button type="submit">Submit</button>
    </form>

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



```javascript
// form submission
let myForm = document.getElementById('myForm');

myForm.addEventListener('submit', function (event) {
    event.preventDefault(); // stop page to reload
    alert('Form submitted: ' + document.getElementById('name').value);
});
```



Form Submission



This page says

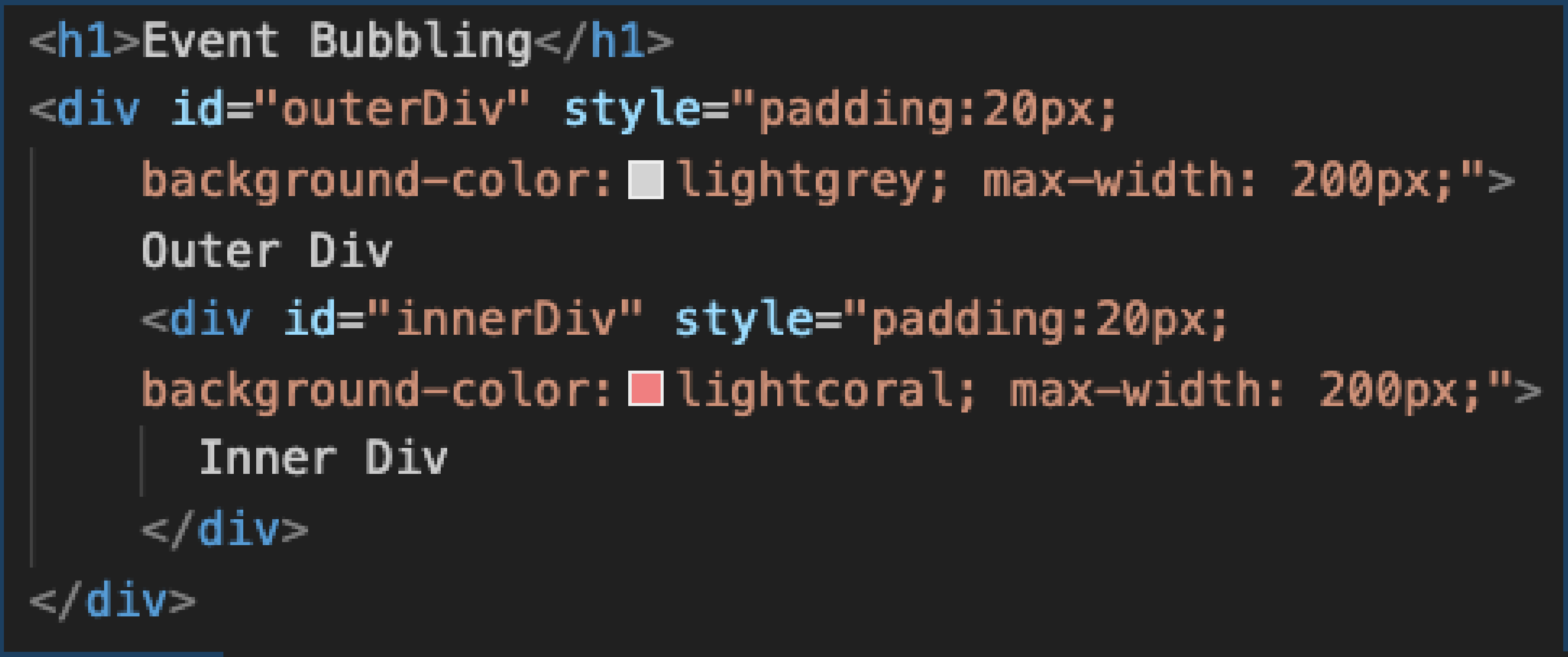
Form submitted: Stinky Pete

Explanation:

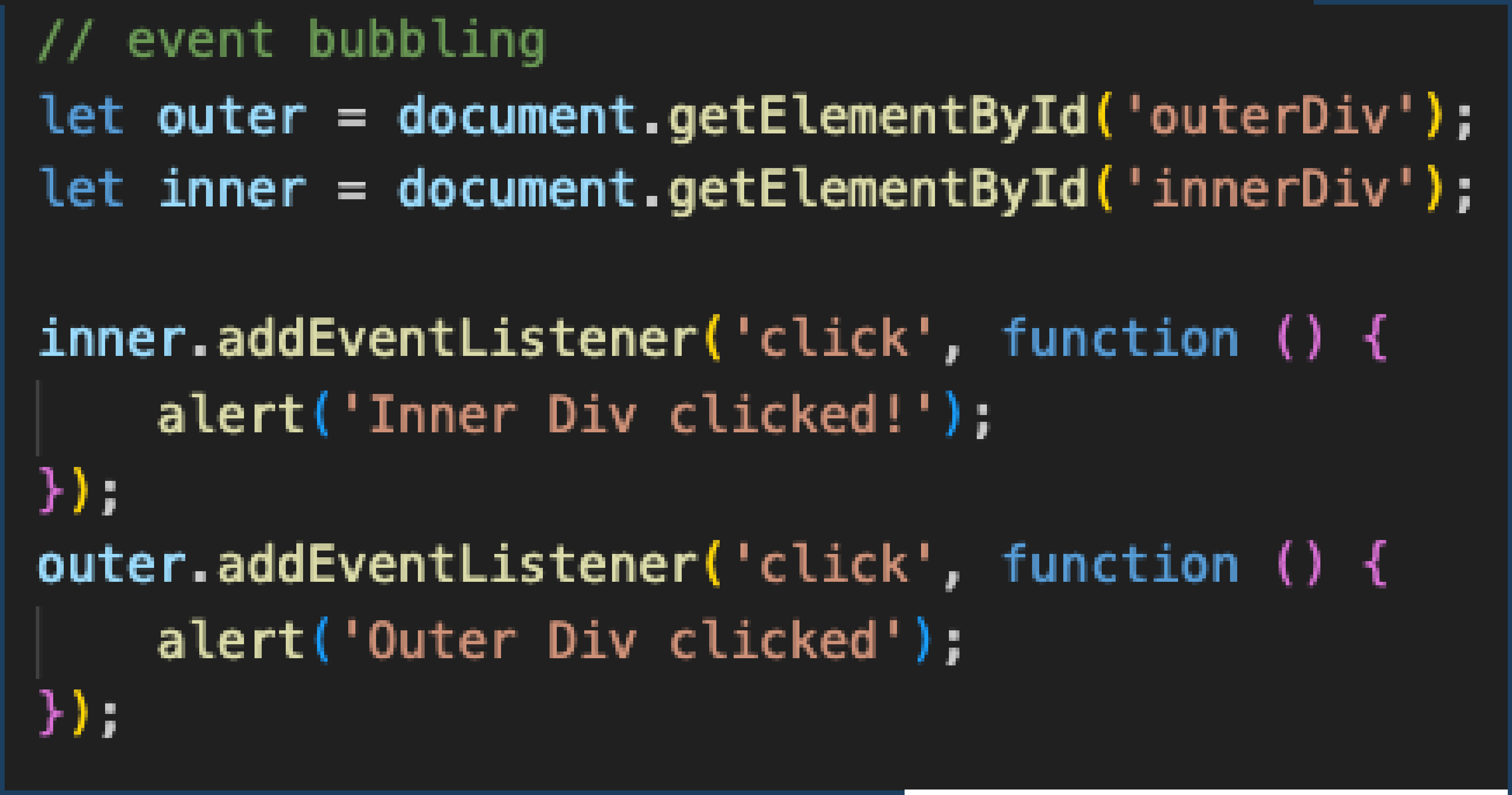
- event.preventDefault() **stops the page from refreshing.**
- **You can access form data and do whatever you want!**

Event Bubbling

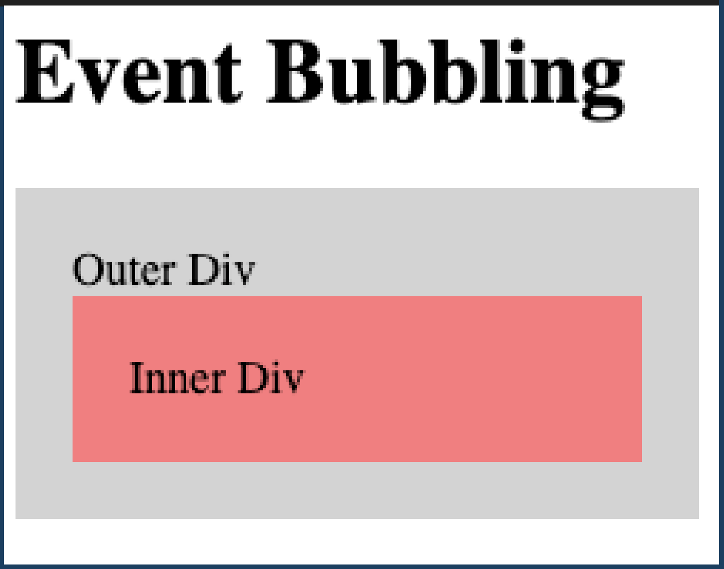
Event bubbling means events start from the **deepest** element and bubble up to the top.



```html
<h1>Event Bubbling</h1>
<div id="outerDiv" style="padding:20px;
    background-color: lightgrey; max-width: 200px;">
    Outer Div
    <div id="innerDiv" style="padding:20px;
    background-color: lightcoral; max-width: 200px;">
      Inner Div
    </div>
</div>
```



```javascript
// event bubbling
let outer = document.getElementById('outerDiv');
let inner = document.getElementById('innerDiv');

inner.addEventListener('click', function () {
    alert('Inner Div clicked!');
});
outer.addEventListener('click', function () {
    alert('Outer Div clicked');
});
```



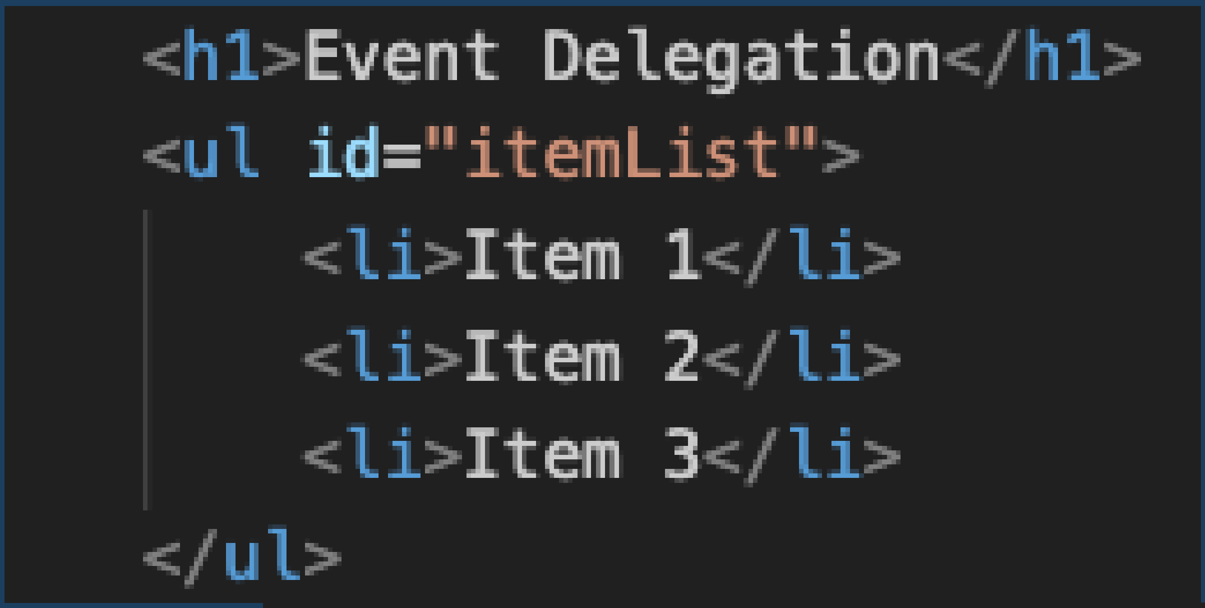
```
Event Bubbling

Outer Div
Inner Div
```

Explanation:

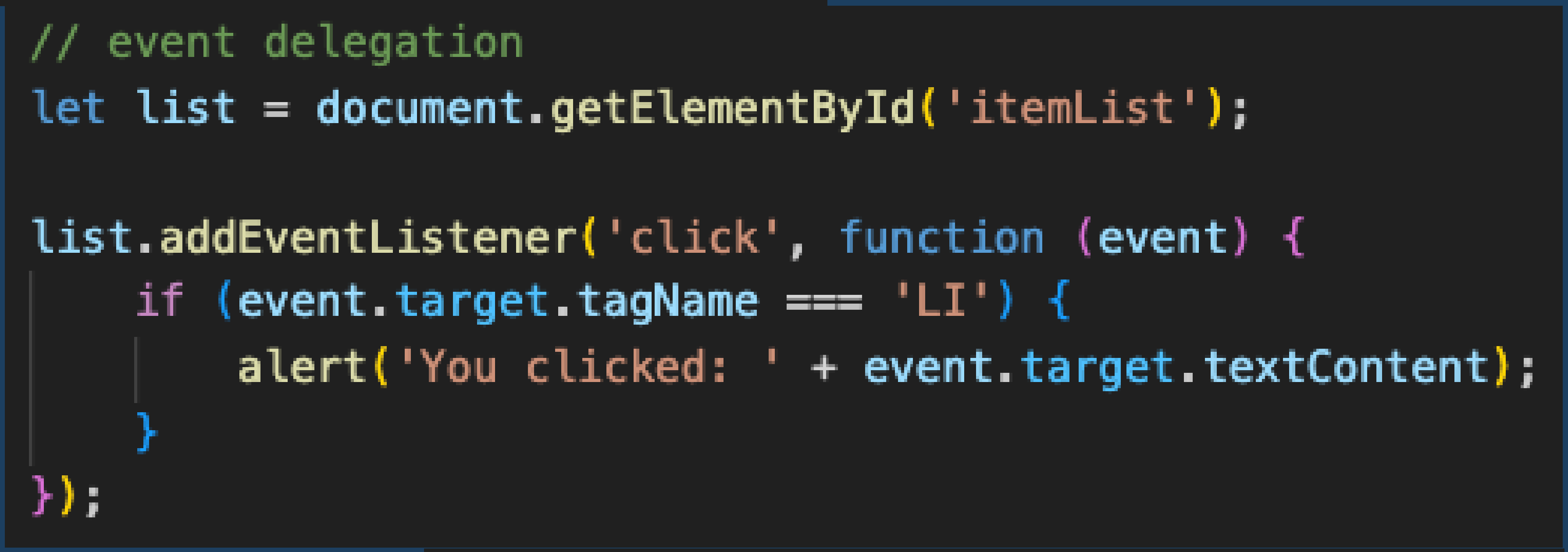
- **Click the Inner Div → first Inner alert shows, then Outer.**
- **Because the event bubbles up through the elements.**

Event Delegation

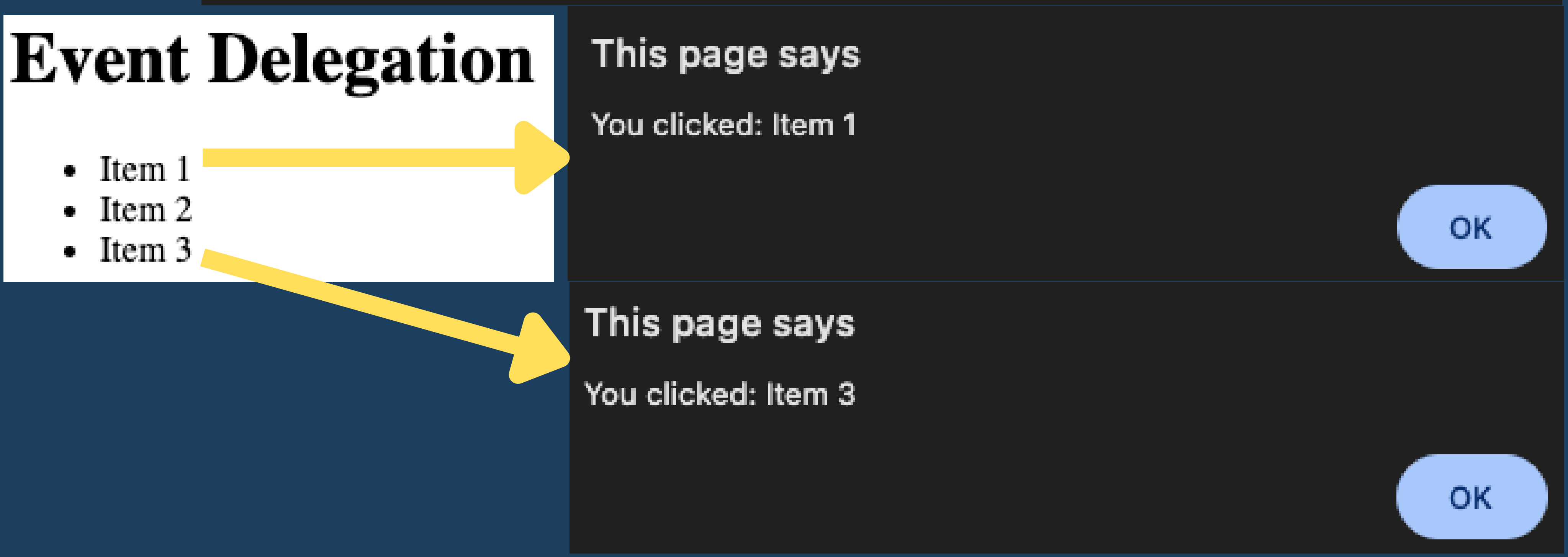
Instead of putting a listener on **every child** element, you can listen on the **parent** and check what was clicked.



```html
<h1>Event Delegation</h1>
<ul id="itemList">
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ul>
```



```javascript
// event delegation
let list = document.getElementById('itemList');

list.addEventListener('click', function (event) {
    if (event.target.tagName === 'LI') {
        alert('You clicked: ' + event.target.textContent);
    }
});
```



Event Delegation

- Item 1
- Item 2
- Item 3

This page says

You clicked: Item 1

This page says

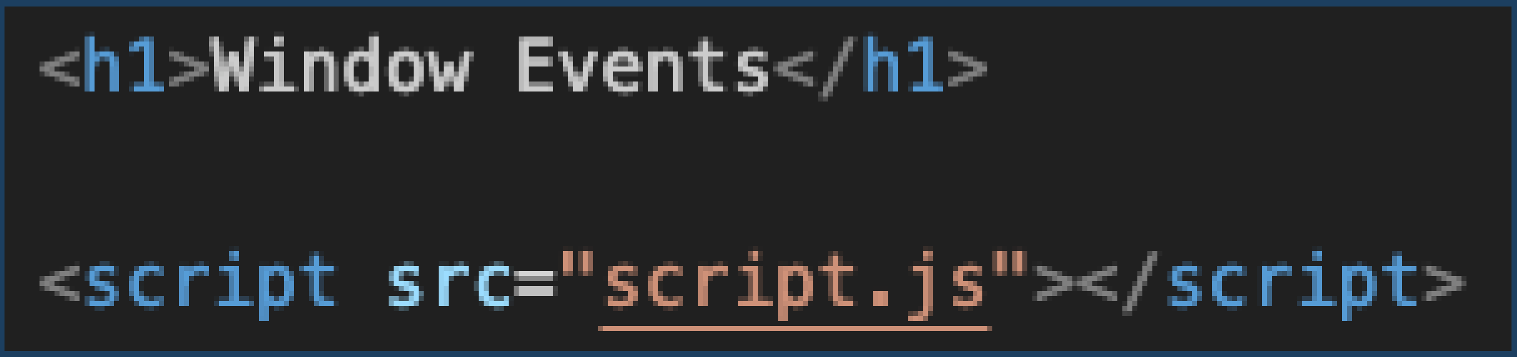
You clicked: Item 3

Explanation:

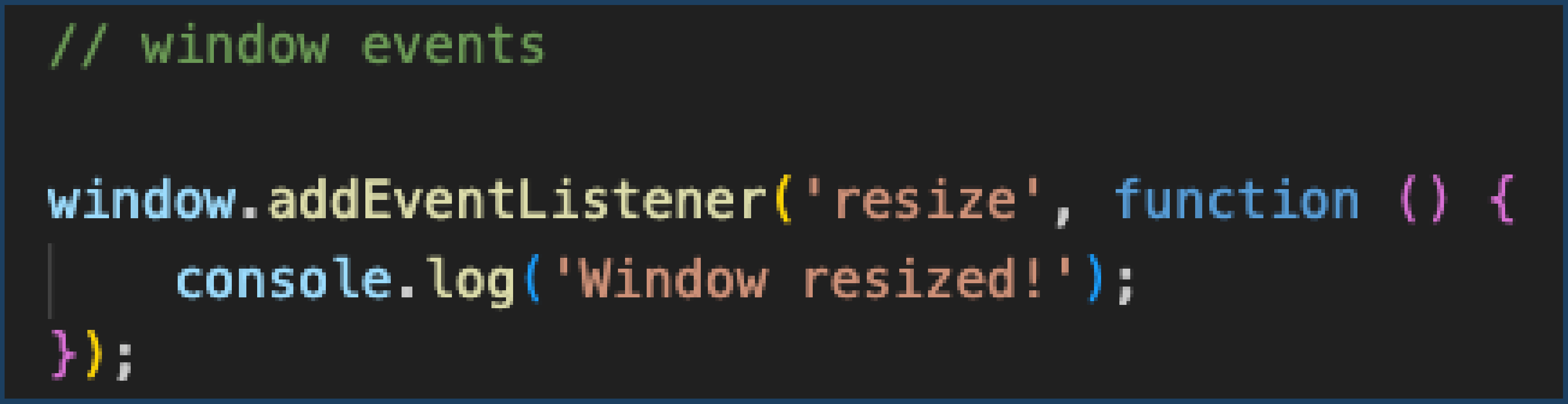
- **Only one listener on the ul.**
- **event.target is the exact item clicked (li).**
- **Saves memory when you have lots of elements!**

Window Events

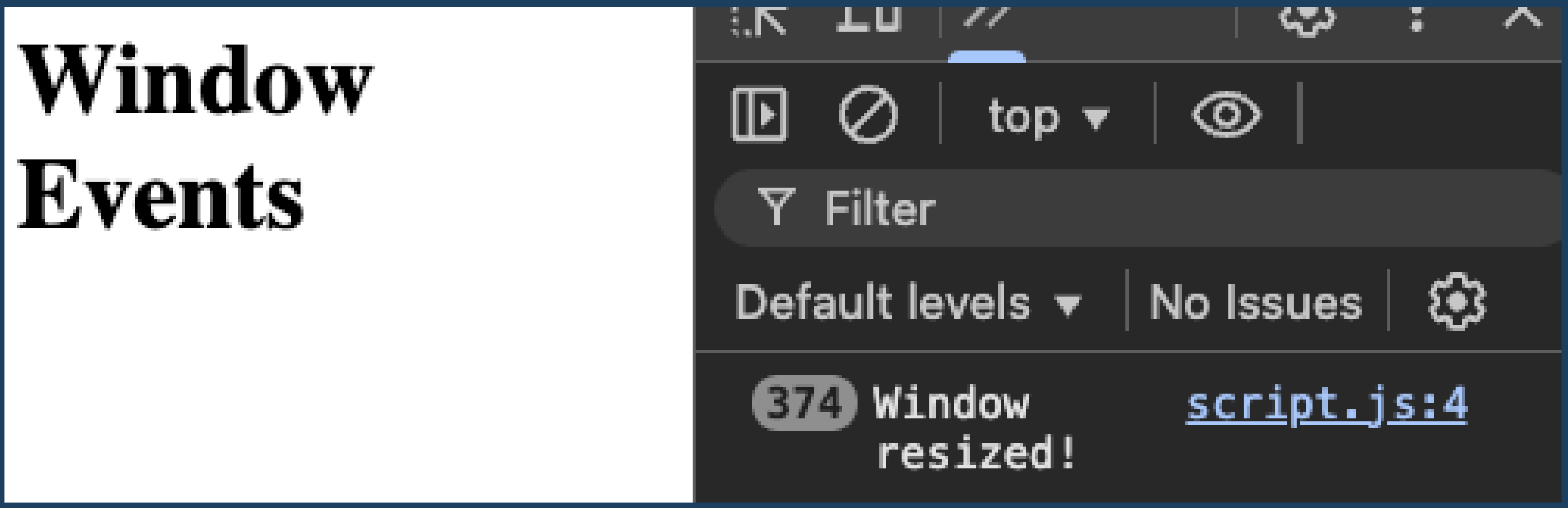
Window events happen to the **whole browser window-** triggered by actions realted to the browser window (like resizing or scrolling).



```html
<h1>Window Events</h1>

<script src="script.js"></script>
```



```javascript
// window events

window.addEventListener('resize', function () {
    console.log('Window resized!');
});
```



Window
Events

```
374 Window          script.js:4
    resized!
```

Common window events:

- **load: fires when the whole page has loaded**
- **resize: fires when the window is resized**
- **scroll: fires when the user scrolls in the document**
- **beforeunload: fires before user leaves the page**

Explanation:

- When you resize the **browser window**, a message appears in the console.