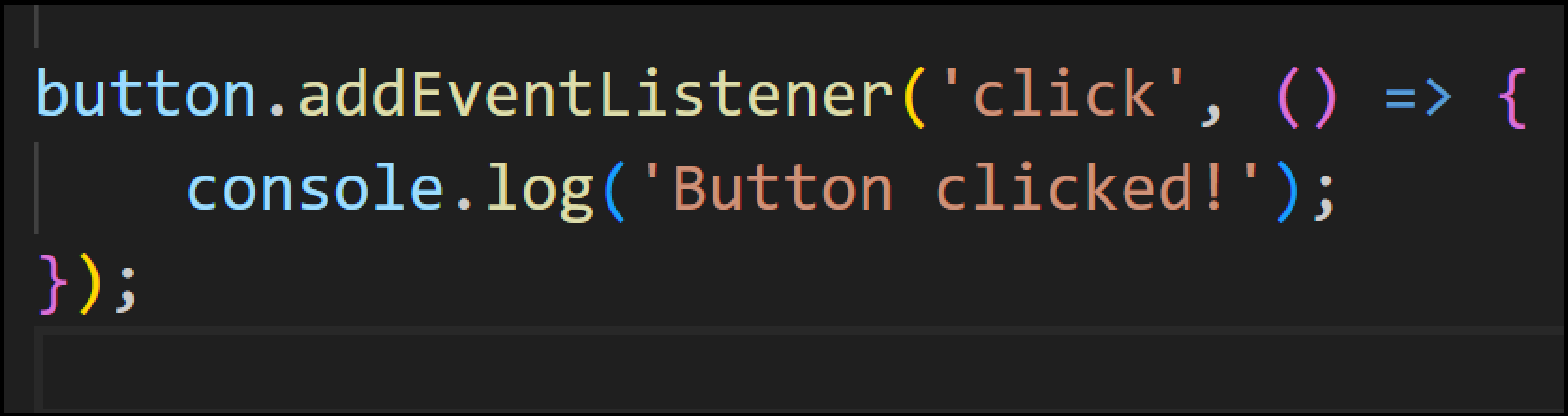
Event Handling in JS

addEventListener :

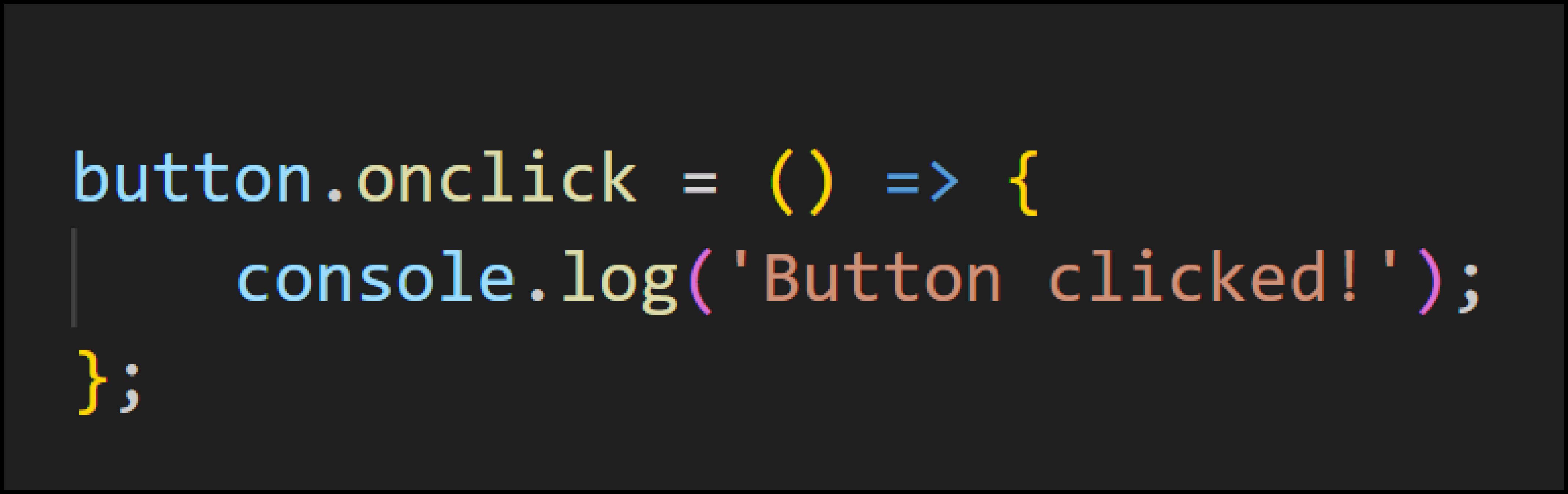
The preferred method for attaching event handlers. It allows multiple event listeners and avoids overwriting existing ones.



```
button.addEventListener('click', () => {
    console.log('Button clicked!');
});
```

onclick:

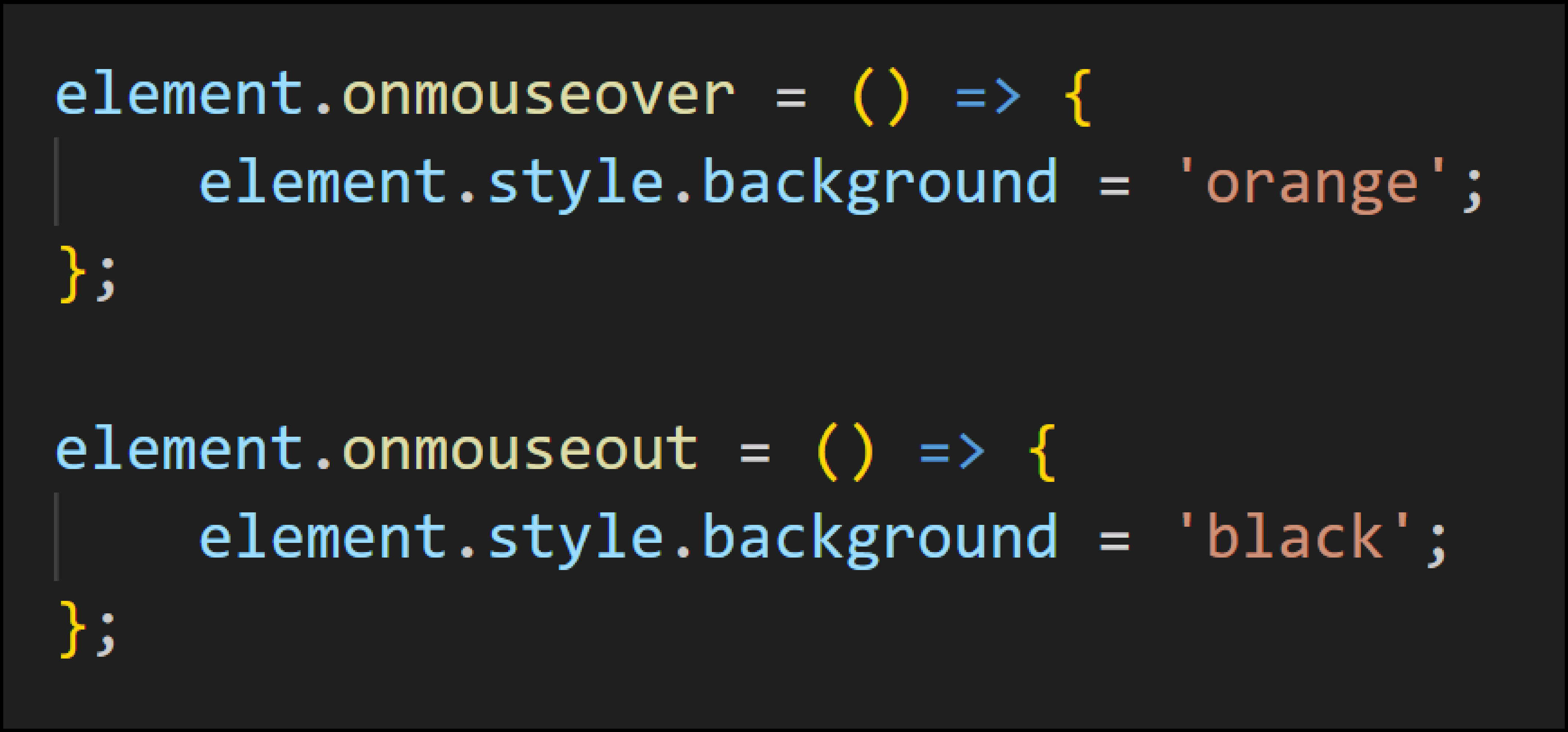
Assigns a function directly to the onclick property of an element. Be cautious, as it can overwrite existing click handlers.



```
button.onclick = () => {
    console.log('Button clicked!');
};
```

onmouseover / onmouseout

Handles mouseover and mouseout events, respectively

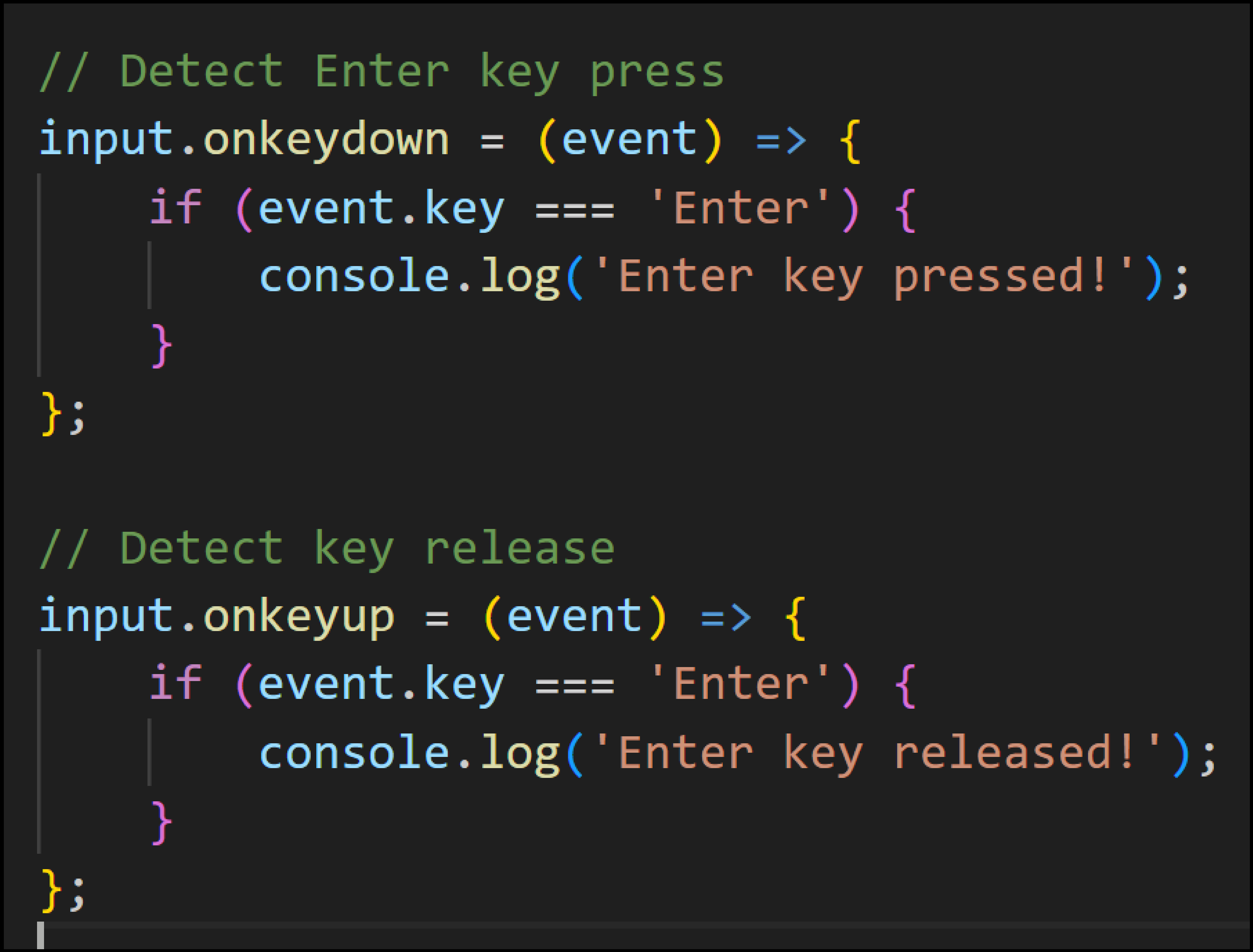


```
element.onmouseover = () => {
    element.style.background = 'orange';
};

element.onmouseout = () => {
    element.style.background = 'black';
};
```

onkeydown / onkeyup:

Manages keydown and keyup events, respectively.

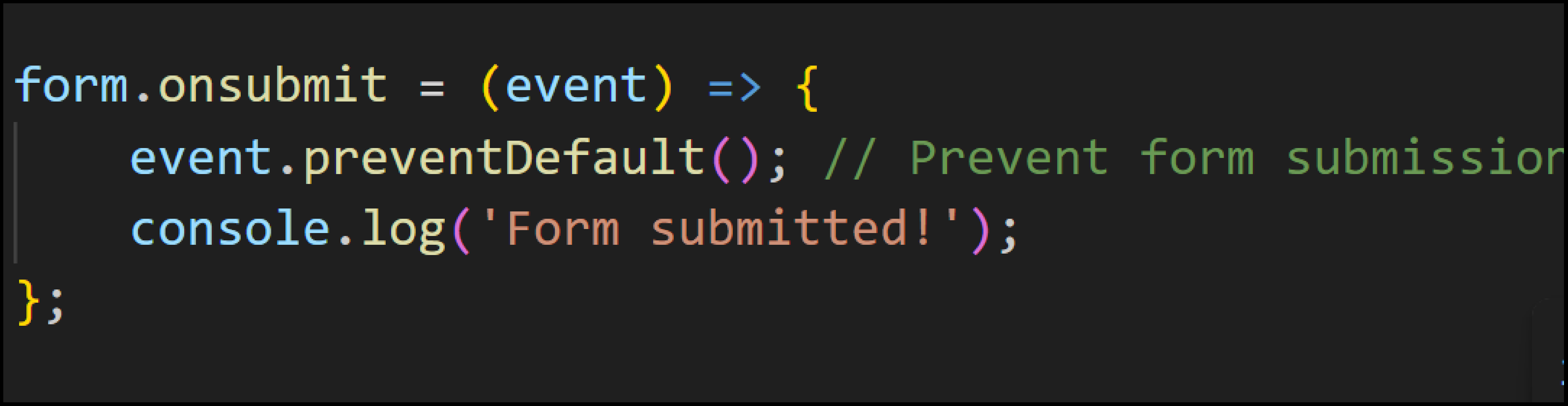


```
// Detect Enter key press
input.onkeydown = (event) => {
    if (event.key === 'Enter') {
        console.log('Enter key pressed!');
    }
};

// Detect key release
input.onkeyup = (event) => {
    if (event.key === 'Enter') {
        console.log('Enter key released!');
    }
};
```

onsubmit:

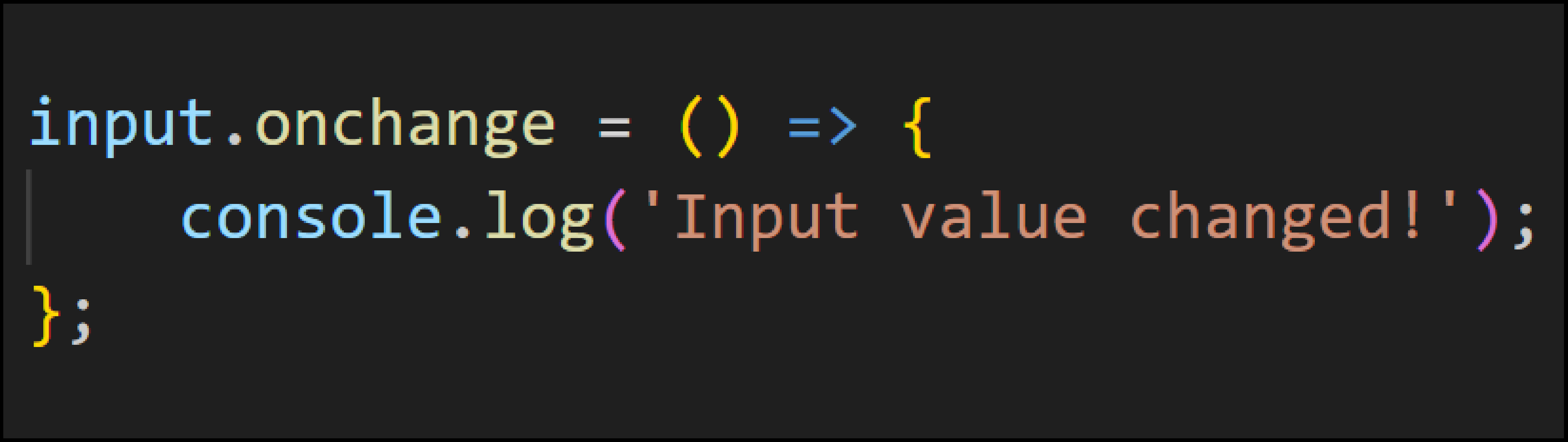
Deals with the submit event of a form element.



```
form.onsubmit = (event) => {
    event.preventDefault(); // Prevent form submission
    console.log('Form submitted!');
};
```

onchange:

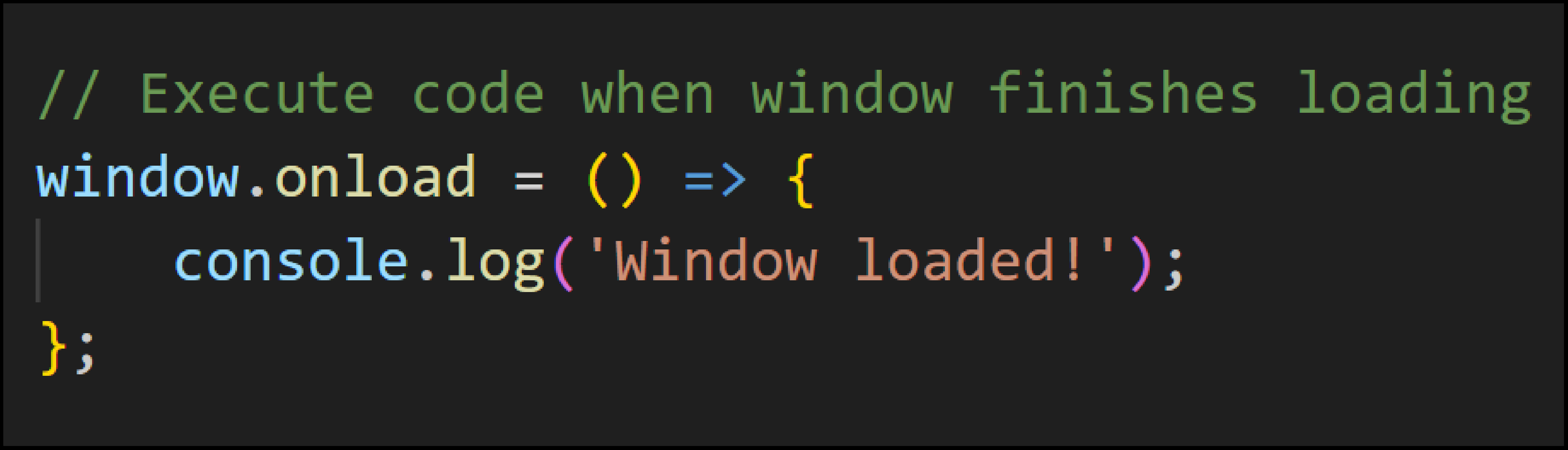
Responds to changes in the value of an input element.



```
input.onchange = () => {
    console.log('Input value changed!');
};
```

onload:

Handles the load event of a window or an image element.



```
// Execute code when window finishes loading
window.onload = () => {
    console.log('Window loaded!');
};
```