



JavaScript

Dom Manipulation

4

Event Handling
Essentials



Adding Event Listeners

JS main.js

```
const div = document.querySelector('div')

element.addEventListener('eventName', () => {
  // Handle the event
});
```

Explanation : **addEventListener** is a method in JavaScript that enables you to **listen** for a specific event on a DOM element.

Note : It is crucial to replace **eventName** in your JavaScript code with the specific **event** you intend to capture, as detailed on the **subsequent page**.



Event Name	Explanation
Mouse Events	
click	Occurs when the element is clicked.
dblclick	Occurs when the element is double-clicked.
mousedown	Occurs when the mouse button is pressed over an element.
mouseup	Occurs when the mouse button is released over an element.
mousemove	Occurs when the mouse pointer moves over an element.
Keyboard Events	
keydown	Occurs when a key on the keyboard is pressed down.
keyup	Occurs when a key on the keyboard is released.
Form Events	
submit	Occurs when a form is submitted.
change	Occurs when the value of an input element changes.
focus	Occurs when an element gets focus.
blur	Occurs when an element loses focus.
Window Events	
load	Occurs when a page has finished loading.
unload	Occurs when the user leaves the page.
resize	Occurs when the browser window is resized.
scroll	Occurs when the user scrolls in the window.
Document Events	
DOMContentLoaded	Occurs when the document has been completely loaded and parsed.
readystatechange	Occurs when the readyState of the document changes.
Touch Events	
touchstart	Occurs when a touch point is placed on the touch surface.
touchend	Occurs when a touch point is removed from the touch surface.
touchmove	Occurs when a touch point is dragged across the touch surface.
Custom Events	You can also create and dispatch custom events using the CustomEvent constructor.



Removing Event Listeners

JS main.js

```
const div = document.querySelector('div')
const clickHandler = () => {
  // Handle the click event
};
// Add the event listener
div.addEventListener('click', clickHandler);
// Remove the event listener
div.removeEventListener('click', clickHandler);
```

Explanation : **removeEventListener** method is used to **remove** a previously added **event listener** from a DOM element.



Stop Propagation

```
JS main.js

const child = document.getElementById('child');

child.addEventListener('click', (event) => {
  event.stopPropagation();
});
```

Explanation : **stopPropagation** is a method used within an event handler to prevent the event from **propagating** or **bubbling up** or **down** the DOM hierarchy



Default Action Suppression

JS main.js

```
const link = document.querySelector('a')

link.addEventListener('click', (event) => {
  event.preventDefault();
});
```

Explanation : **preventDefault** is a method used within an event handler to **suppress** or **prevent** the **default action** associated with an event. This is commonly used to **stop actions** like **navigating** to a new page when a link is **clicked**.





**Hopefully You Found It
Usefull!**

“Be sure to save this post so you
can come back to it later,”

like

Comment

Share

