

JavaScript Dom Manipulation



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.

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





Slim JavaScript tips

Event Name

Explanation

Mouse Events

click dblclick mousedown mouseup mousemove

Occurs when the element is clicked.

Occurs when the element is double-clicked.

Occurs when the mouse button is pressed over an element.

Occurs when the mouse button is released over an element.

Occurs when the mouse pointer moves over an element.

Keyboard Events

k e y d o w n k e y u p

Occurs when a key on the keyboard is pressed down.

Occurs when a key on the keyboard is released.

Form Events

submit change focus blur Occurs when a form is submitted.

Occurs when the value of an input element changes.

Occurs when an element gets focus.

Occurs when an element loses focus.

Window Events

load unload resize scroll

Occurs when a page has finished loading.

Occurs when the user leaves the page.

Occurs when the browser window is resized.

Occurs when the user scrolls in the window.

Document Events

DOMContentLoaded readystatechange

Occurs when the document has been completely loaded and parsed.

Occurs when the readyState of the document changes.

Touch Events

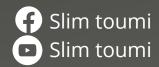
touchstart touchend touchmove Occurs when a touch point is placed on the touch surface.

Occurs when a touch point is removed from the touch surface.

Occurs when a touch point is dragged across the touch surface.

You can also create and dispatch custom events using the CustomEvent constructor.

Custom Events





Slim

Removing Event Listeners

```
const div = document.querySelector('div')
const clickHandler = () \Rightarrow {
    // 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.





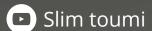
Stop Propagation

```
JS main.js

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

child.addEventListener('click', (event) \Rightarrow {
    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





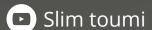
Default Action Suppression

```
JS main.js

const link = document.querySelector('a')

link.addEventListener('click', (event) \Rightarrow {
    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.





5

Hopefully You Found It Usefull!

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

like

Comment

Share



