JavaScript HTML DOM EventListener

## The addEventListener() method

<html>

<body>

<h2>JavaScript addEventListener()</h2>

<p>This example uses the addEventListener() method to attach a click event to a button.</p>

<button id="myBtn">Try it</button>

<p id="demo"></p>

<script>

document.getElementById("myBtn").addEventListener("click", displayDate);

function displayDate() {

document.getElementById("demo").innerHTML = Date();

}

</script></body></html>

The addEventListener() method attaches an event handler to the specified element.

The addEventListener() method attaches an event handler to an element without overwriting existing event handlers.

You can add many event handlers to one element.

You can add many event handlers of the same type to one element, i.e two "click" events.

You can add event listeners to any DOM object not only HTML elements. i.e the window object.

The addEventListener() method makes it easier to control how the event reacts to bubbling.

When using the addEventListener() method, the JavaScript is separated from the HTML markup, for better readability and allows you to add event listeners even when you do not control the HTML markup.

You can easily remove an event listener by using the removeEventListener() method.

## Syntax

element.addEventListener(event, function, useCapture);

The first parameter is the type of the event (like "click" or "mousedown").

The second parameter is the function we want to call when the event occurs.

The third parameter is a boolean value specifying whether to use event bubbling or event capturing. This parameter is optional.

Note that you don't use the "on" prefix for the event; use "click" instead of "onclick".

## Add an Event Handler to an Element

### Example

Alert "Hello World!" when the user clicks on an element:

You can also refer to an external "named" function:

### Example

Alert "Hello World!" when the user clicks on an element:

<html>

<body>

<h2>JavaScript addEventListener()</h2>

<p>This example uses the addEventListener() method to execute a function when a user clicks on a button.</p>

<button id="myBtn">Try it</button>

<script>

document.getElementById("myBtn").addEventListener("click", myFunction);

function myFunction() {

alert ("Hello World!");

}

</script></body>

</html>

## Add Many Event Handlers to the Same Element

The addEventListener() method allows you to add many events to the same element, without overwriting existing events:

<html>

<body>

<h2>JavaScript addEventListener()</h2>

<p>This example uses the addEventListener() method to add two click events to the same button.</p>

<button id="myBtn">Try it</button>

<script>

var x = document.getElementById("myBtn");

x.addEventListener("click", myFunction);

x.addEventListener("click", someOtherFunction);

function myFunction() {

alert ("Hello World!");

}

function someOtherFunction() {

alert ("This function was also executed!");

}

</script></body></html>

You can add events of different types to the same element:

### Example

<html>

<body>

<h2>JavaScript addEventListener()</h2>

<p>This example uses the addEventListener() method to add many events on the same button.</p>

<button id="myBtn">Try it</button>

<p id="demo"></p>

<script>

var x = document.getElementById("myBtn");

x.addEventListener("mouseover", myFunction);

x.addEventListener("click", mySecondFunction);

x.addEventListener("mouseout", myThirdFunction);

function myFunction() {

document.getElementById("demo").innerHTML += "Moused over!<br>";

}

function mySecondFunction() {

document.getElementById("demo").innerHTML += "Clicked!<br>";

}

function myThirdFunction() {

document.getElementById("demo").innerHTML += "Moused out!<br>";

}

</script>

</body>

</html>