JavaScript addEventListener()

The **addEventListener()** method is used to attach an event handler to a particular element. It does not override the existing event handlers. Events are said to be an essential part of the JavaScript. A web page responds according to the event that occurred. Events can be user-generated or generated by API's. An event listener is a JavaScript's procedure that waits for the occurrence of an event.

The addEventListener() method is an inbuilt function of [JavaScript](https://www.javatpoint.com/javascript-tutorial). We can add multiple event handlers to a particular element without overwriting the existing event handlers.

Syntax

1. element.addEventListener(event, function, useCapture);

Although it has three parameters, the parameters ***event*** and ***function*** are widely used. The third parameter is optional to define. The values of this function are defined as follows.

Parameter Values

**event:** It is a required parameter. It can be defined as a string that specifies the event's name.

Note: Do not use any prefix such as "on" with the parameter value. For example, Use "click" instead of using "onclick".

**function:** It is also a required parameter. It is a [JavaScript function](https://www.javatpoint.com/javascript-function) which responds to the event occur.

**useCapture:** It is an optional parameter. It is a Boolean type value that specifies whether the event is executed in the bubbling or capturing phase. Its possible values are **true** and **false**. When it is set to true, the event handler executes in the capturing phase. When it is set to false, the handler executes in the bubbling phase. Its default value is **false**.

## **Event Bubbling or Event Capturing**

Now, we understand the use of the third parameter of JavaScript's addEventListener(), i.e., **useCapture.**

In HTML DOM, **Bubbling** and **Capturing** are the two ways of event propagation. We can understand these ways by taking an example.

Suppose we have a div element and a paragraph element inside it, and we are applying the **"click"** event to both of them using the **addEventListener()** method. Now the question is on clicking the paragraph element, which element's click event is handled first.

So, in **Bubbling,** the event of paragraph element is handled first, and then the div element's event is handled. It means that in bubbling, the inner element's event is handled first, and then the outermost element's event will be handled.

In **Capturing** the event of div element is handled first, and then the paragraph element's event is handled. It means that in capturing the outer element's event is handled first, and then the innermost element's event will be handled.

1. addEventListener(event, function, useCapture);

We can specify the propagation using the **useCapture** parameter. When it is set to false (which is its default value), then the event uses bubbling propagation, and when it is set to true, there is the capturing propagation.