

# JavaScript -Introduction to Events

## What is an Event ?

JavaScript's interaction with HTML is handled through events that occur when the user or the browser manipulates a page.

When the page loads, it is called an event. When the user clicks a button, that click too is an event. Other examples include events like pressing any key, closing a window, resizing a window, etc.

Developers can use these events to execute JavaScript coded responses, which cause buttons to close windows, messages to be displayed to users, data to be validated, and virtually any other type of response imaginable.

Events are a part of the Document Object Model (DOM) Level 3 and every HTML element contains a set of events which can trigger JavaScript Code.

Please go through this small tutorial for a better understanding [HTML Event Reference](#).

## JavaScript Event Handlers

Event handler can be used as an attribute of the HTML element. It takes the inline JavaScript or function execution code as a value.

Whenever any event triggers, it invokes the inline JavaScript code or executes the callback function to perform the particular action.

In simple terms, it is used to handle the events.

## Syntax

Users can follow the syntax below to use event handlers with HTML elements.

```
<div eventHandler = "JavaScript_code"> </div>
```

In the above syntax, you need to replace the 'eventHandler' with the actual event handler like 'onclick', 'onmouseover', etc. The 'JavaScript\_code' should execute the function or run JavaScript inline.

## Example: Inline JavaScript with Event Handlers

In the code below, we have created the <button> element. Also, we have used the

'onclick' event handler to capture the click event on the button.

We have written the inline JavaScript code to handle the event. In the inline JavaScript code, the 'this' keyword represents the <button> element, and we change the button's text color to red.

[Open Compiler](#)

```
<html>
<body>
  <h2> Click the button to Change its text's color </h2>
  <button onclick = "this.style.color='red'"> Click Me </button>
  <div id = "output"> </div>
</body>
</html>
```

## Example: Function with Event Handlers

In the code below, we have created the <div> element and given style into the <head> section.

We used the 'onclick' event handler with the <button> element, which calls the handleClick() function when the user clicks the button.

The handleClick() function takes the 'event' object as a parameter. In the handleClick() function, we change the background color of the <div> element using JavaScript.

[Open Compiler](#)

```
<html>
<head>
  <style>
    #test {
      width: 600px;
      height: 100px;
      background-color: red;
    }
  </style>
</head>
<body>
  <div id = "test"> </div> <br>
  <button onclick = "handleClick()"> Change Div Color </button>
```

```
<script>
  function handleClick(event) {
    var div = document.getElementById("test");
    div.style.backgroundColor = "blue";
  }
</script>
</body>
</html>
```

## Example: Multiple Functions with Event Handlers

In the code below, we have added the 'onmouseenter' event handler with the <div> element. We call the changeFontSize() and changeColor() functions when a user enters the mouse cursor in the <div> element.

The changeFontSize() function changes the size of the text, and changeColor() function changes the color of the text.

This way, you can invoke the multiple functions on the particular event.

[Open Compiler](#)

```
<html>
<head>
  <style>
    #test {
      font-size: 15px;
    }
  </style>
</head>
<body>
  <h2> Hover over the below text to customize the font. </h2>
  <div id = "test" onmouseenter = "changeFontSize(); changeColor();" > Hello Wo
  <script>
    function changeFontSize(event) {
      document.getElementById("test").style.fontSize = "25px";
    }
    function changeColor(event) {
      document.getElementById("test").style.color = "red";
    }
  </script>
</body>
```

&lt;/html&gt;

Learn **JavaScript** in-depth with real-world projects through our **JavaScript certification course**. Enroll and become a certified expert to boost your career.

## JavaScript Event Object

The function that handles the event takes the 'event' object as a parameter. The 'event' object contains the information about the event and the element on which it occurred.

There are various properties and methods are also available which can be used with the event object to get information.

Object	Description
Event	It is a parent of all event objects.

Here is the list of different types of event objects. Each event object contains various events, methods, and properties.

Object/Type	Handles
AnimationEvent	It handles the CSS animations.
ClipboardEvent	It handles the changes of the clipboard.
DragEvent	It handles the drag-and-drop events.
FocusEvent	To handle the focus events.
HashChangeEvent	It handles the changes in the anchor part of the URL.
InputEvent	To handle the form inputs.
KeyboardEvent	To handle the keyboard interaction by users.
MediaEvent	It handles the media-related events.
MouseEvent	To handle the mouse interaction by users.
PageTransitionEvent	To handle the navigation between web pages.
PopStateEvent	It handles the changes in the page history.
ProgressEvent	To handle the progress of the file loading.
StorageEvent	To handle changes in the web storage.

TouchEvent	To handle touch interaction on the device's screen.
TransitionEvent	To handle CSS transition.
UiEvent	To handle the user interface interaction by users.
WheelEvent	To handle the mouse-wheel interaction by users.