

What is an Event?

- Javascript has events to provide a dynamic interface to a webpage. These events are hooked to elements in the Document Object Model(DOM).
- 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.

Introduction to Event Handling

- Event Handling is a software routine that processes actions, such as keystrokes and mouse movements.
- It is the receipt of an event at some event handler from an event producer and subsequent processes.

Functions of Event Handling

- Event Handling identifies where an event should be forwarded.
- It makes the forward event.
- It receives the forwarded event.
- It takes some kind of appropriate action in response, such as writing to a log, sending an error or recovery routine or sending a message.
- The event handler may ultimately forward the event to an event consumer.

1. **onclick events:** This is a mouse event and provokes any logic defined if the user clicks on the element it is bound to.

Code #1:

```
<!doctype html>
<html>
  <head>
    <script>
      function hiThere() {
        alert('Hi there!');
      }
    </script>
  </head>
  <body>
    <button type="button" onclick="hiThere()">Click me event</button>
```

```
</body>
</html>
```

2. **onkeyup event:** This event is a keyboard event and executes instructions whenever a key is released after pressing.

Code #2:

```
<!doctype html>
<html>
  <head>
    <script>
      var a = 0;
      var b = 0;
      var c = 0;
      function changeBackground() {
        var x = document.getElementById('bg');
        bg.style.backgroundColor = 'rgb('+a+', '+b+', '+c+')';
        a += 1;
        b += a + 1;
        c += b + 1;
        if (a > 255) a = a - b;
        if (b > 255) b = a;
        if (c > 255) c = b;
      }
    </script>
  </head>
  <body>
    <input id="bg" onkeyup="changeBackground()"
      placeholder="write something" style="color:#fff">
  </body>
</html>
```

3. **onmouseover event:** This event corresponds to hovering the mouse pointer over the element and its children, to which it is bound to.

Code #3:

```
<!doctype html>
<html>
  <head>
    <script>
      function hov() {
        var e = document.getElementById('hover');
        e.style.display = 'none';
      }
    </script>
  </head>
  <body>
    <div id="hover" onmouseover="hov()"
      style="background-color:green;height:200px;width:200px;">
```

```
</div>
</body>
</html>
```

Green square gets disappear after mouse is taken over it.

- 4. onmouseout event:** Whenever the mouse cursor leaves the element which handles a mouseout event, a function associated with it is executed.

Code #4:

```
<!doctype html>
<html>
<head>
<script>
function out() {
var e = document.getElementById('hover');
e.style.display = 'none';
}
</script>
</head>
<body>
<div id="hover" onmouseout="out()"
style="background-color:green;height:200px;width:200px;">
</div>
</body>
</html>
```

Green square will disappear after mouse is taken over it and removed after some time.

- 5. onchange event:** This event detects the change in value of any element listing to this event.

Code #5:

```
<!doctype html>
<html>
<head></head>
<body>
<input onchange="alert(this.value)" type="number">
</body>
</html>
```

- 6. onload event:** When an element is loaded completely, this event is evoked.

Code #6:

```
<!doctype html>
<html>
<head></head>
<body>

</body>
```

```
</html>
```

7) onfocus event: An element listing to this event executes instructions whenever it receives focus.

Code #7:

```
<!doctype html>
<!doctype html>
<html>
  <head>
    <script>
      function focused() {
        var e = document.getElementById('inp');
        if (confirm('Got it?')) {
          e.blur();
        }
      }
    </script>
  </head>
  <body>
    <p>Take the focus into the input box below:</p>
    <input id="inp" onfocus="focused()">
  </body>
</html>
```

7. onblur event: This event is evoked when an element loses focus.

Code #8:

```
<!doctype html>
<html>
  <head></head>
  <body>
    <p>Write something in the input box and then click elsewhere
      in the document body.</p>
    <input onblur="alert(this.value)">
  </body>
</html>
```

Event Handlers

Event Handler	Description
onAbort	It executes when the user aborts loading an image.
onBlur	It executes when the input focus leaves the field of a text, textarea or a select option.

onChange	It executes when the input focus exits the field after the user modifies its text.
onClick	In this, a function is called when an object in a button is clicked, a link is pushed, a checkbox is checked or an image map is selected. It can return false to cancel the action.
onError	It executes when an error occurs while loading a document or an image.
onFocus	It executes when input focus enters the field by tabbing in or by clicking but not selecting input from the field.
onLoad	It executes when a window or image finishes loading.
onMouseOver	The JavaScript code is called when the mouse is placed over a specific link or an object.
onMouseOut	The JavaScript code is called when the mouse leaves a specific link or an object.
onReset	It executes when the user resets a form by clicking on the reset button.
onSelect	It executes when the user selects some of the text within a text or textarea field.
onSubmit	It calls when the form is submitted.
onUnload	It calls when a document is exited.

Handling Events from Body Elements

Example: the `load` event - triggered when the loading of a document is completed

```
<!DOCTYPE html>
<!-- load.html
    A document for load.js
-->
<html lang = "en">
<head>
    <title> load.html </title>
    <meta charset = "utf-8" />
    <script type = "text/javascript" src = "load.js" >
    </script>
</head>
<body onload="load_greeting();">
<!-- use the onload attribute of <body> to specify the event handler. -
->
    <p />
</body>
</html>
```

```
// load.js
//   An example to illustrate the load event

// The onload event handler

function load_greeting () {
    alert("You are visiting the home page of \n" +
        "Pete's Pickled Peppers \n" + "WELCOME!!!");
}
```