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. Here we will see a few examples to understand a relation between Event and JavaScript −

onclick Event Type

This is the most frequently used event type which occurs when a user clicks the left button of his mouse. You can put your validation, warning etc., against this event type.

Example

```
<html>
  <head>
    <script type = "text/javascript">
      <!--
        function sayHello() {
          alert("Hello World")
        }
      //-->
    </script>
  </head>

  <body>
    <p>Click the following button and see result</p>
    <form>
      <input type = "button" onclick = "sayHello()" value = "Say Hello" />
    </form>
  </body>
```

```
</html>
```

onsubmit Event Type

onsubmit is an event that occurs when you try to submit a form. You can put your form validation against this event type.

Example

The following example shows how to use onsubmit. Here we are calling a validate() function before submitting a form data to the webserver. If validate() function returns true, the form will be submitted, otherwise it will not submit the data.

```
<html>
  <head>
    <script type = "text/javascript">
      <!--
        function validation() {
          all validation goes here
          .........
          return either true or false
        }
      //-->
    </script>
  </head>

  <body>
    <form method = "POST" action = "t.cgi" onsubmit =
"return validate()">
      .......
      <input type = "submit" value = "Submit" />
    </form>
  </body>
</html>
```

onmouseover and onmouseout

These two event types will help you create nice effects with images or even with text as well. The onmouseover event triggers when you bring your mouse over any element and the onmouseout triggers when you move your mouse out from that element. Try the following example.

```
<html>
  <head>
```

```
    <script type = "text/javascript">
      <!--
        function over() {
          document.write ("Mouse Over");
        }
        function out() {
          document.write ("Mouse Out");
        }
      //-->
    </script>
  </head>

  <body>
    <p>Bring your mouse inside the division to see the result:</p>
    <div onmouseover = "over()" onmouseout = "out()">
      <h2> This is inside the division </h2>
    </div>
  </body>
</html>
```

HTML 5 Standard Events

The standard HTML 5 events are listed here for your reference. Here script indicates a Javascript function to be executed against that event.

Attribute		Description
Offline		Triggers when the document goes offline
Onabort		Triggers on an abort event
onafterprint		Triggers after the document is printed
onbeforeonload		Triggers before the document loads
onbeforeprint		Triggers before the document is printed
onblur		Triggers when the window loses focus
oncanplay		Triggers when media can start play, but might has to stop for buffering
oncanplaythrough		Triggers when media can be played to the end, without stopping for buffering
onchange		Triggers when an element changes

onclick		Triggers on a mouse click
oncontextmenu		Triggers when a context menu is triggered
ondblclick		Triggers on a mouse double-click
ondrag		Triggers when an element is dragged
ondragend		Triggers at the end of a drag operation
ondragenter		Triggers when an element has been dragged to a valid drop target
ondragleave		Triggers when an element is being dragged over a valid drop target
ondragover		Triggers at the start of a drag operation
ondragstart		Triggers at the start of a drag operation

ondrop		Triggers when dragged element is being dropped
ondurationchange		Triggers when the length of the media is changed
onemptied		Triggers when a media resource element suddenly becomes empty.
onended		Triggers when media has reach the end
onerror		Triggers when an error occur
onfocus		Triggers when the window gets focus
onformchange		Triggers when a form changes
onforminput		Triggers when a form gets user input
onhaschange		Triggers when the document has change
oninput		Triggers when an element gets user input

onininvalid		Triggers when an element is invalid
onkeydown		Triggers when a key is pressed
onkeypress		Triggers when a key is pressed and released
onkeyup		Triggers when a key is released
onload		Triggers when the document loads
onloadeddata		Triggers when media data is loaded
onloadedmetadata		Triggers when the duration and other media data of a media element is loaded
onloadstart		Triggers when the browser starts to load the media data
onmessage		Triggers when the message is triggered

onmousedown		Triggers when a mouse button is pressed
onmousemove		Triggers when the mouse pointer moves
onmouseout		Triggers when the mouse pointer moves out of an element
onmouseover		Triggers when the mouse pointer moves over an element
onmouseup		Triggers when a mouse button is released
onmousewheel		Triggers when the mouse wheel is being rotated
onoffline		Triggers when the document goes offline
onoine		Triggers when the document comes online
ononline		Triggers when the document comes online

onpagehide		Triggers when the window is hidden
onpageshow		Triggers when the window becomes visible
onpause		Triggers when media data is paused
onplay		Triggers when media data is going to start playing
onplaying		Triggers when media data has start playing
onpopstate		Triggers when the window's history changes
onprogress		Triggers when the browser is fetching the media data
onratechange		Triggers when the media data's playing rate has changed
onreadystatechange		Triggers when the ready-state changes

onredo		Triggers when the document performs a redo
onresize		Triggers when the window is resized
onscroll		Triggers when an element's scrollbar is being scrolled
onseeked		Triggers when a media element's seeking attribute is no longer true, and the seeking has ended
onseeking		Triggers when a media element's seeking attribute is true, and the seeking has begun
onselect		Triggers when an element is selected
onstalled		Triggers when there is an error in fetching media data
onstorage		Triggers when a document loads

onsubmit		Triggers when a form is submitted
onsuspend		Triggers when the browser has been fetching media data, but stopped before the entire media file was fetched
ontimeupdate		Triggers when media changes its playing position
onundo		Triggers when a document performs an undo
onunload		Triggers when the user leaves the document
onvolumechange		Triggers when media changes the volume, also when volume is set to "mute"
onwaiting		Triggers when media has stopped playing, but is expected to resume