

## What is an Event?

- Events are things that happen, usually user actions, that are associated with an object.
- JavaScript's interaction with HTML is handled through events that occur when the user or browser manipulates a page.
- When the page loads, that is an event. When the user clicks a button, that click, too, is an event. Another example of events is like pressing any key, closing window, resizing 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 to occur.
- Events are a part of the Document Object Model (DOM) Level 3 and every HTML element have a certain set of events which can trigger JavaScript Code.
- Examples of events
  - > click-A mouse click
  - > load-A web page or an image loading
  - > mouseover- Mousing over a hot spot on the web page
  - > select- Selecting an input box in an HTML form
  - > submit- Submitting an HTML form
  - > A keystroke

### **event handler:**

- The "event handler" is a command that is used to specify actions in response to an event.
- We can write our event handlers in Javascript or vbscript and can specify these event handlers as a value of event tag attribute.
- An event handler executes a segment of a code based on certain events occurring within the application, such as onLoad, onClick.
- JavaScript event handlers can be divided into two parts:
  - interactive event handlers
  - non-interactive event handlers

- An interactive event handler is the one that depends on the user interactivity with the form or the document. For example, onMouseOver is an interactive event handler because it depends on the users action with the mouse.
- On the other hand non-interactive event handler would be onLoad, because this event handler would automatically execute JavaScript code without the user's interactivity.
- Event handlers are embedded in documents as attributes of html tag to which you assign JavaScript code. The syntax is:

`<htmltag eventhandler="JavaScript code">`

For ex.

`<body onLoad="hello()">`

- Here are all the event handlers available in JavaScript:

| Event Handler | USED IN                              | Description  |
|---------------|--------------------------------------|--|
| onabort       | Image                                | Loading of an image is interrupted                             |
| onload        | windows, image                       | Script runs when a HTML document loads                         |
| onunload      | Window, Document body                | Script runs when a HTML document unloads. User Exits the page. |
| Onchange      | Select lists, text, textarea         | Script runs when user changes the value of element.            |
| Onsubmit      | form                                 | Script runs when the form is submitted                         |
| Onreset       | form                                 | Script runs when the form is reset                             |
| Onselect      | text, textarea                       | Script runs when the form's element is selected                |
| Onblur        | Window & all form elements           | Script runs when the form's element loses focus                |
| Onfocus       | Window & all form elements           | Script runs when the form's element gets focus                 |
| onkeydown     | Documents, images, links, text areas | Script runs when key is pressed                                |
| onkeypress    | Documents, images, links, text areas | Script runs when key is pressed and released                   |

|         |                                     |                                  |
|---------|-------------------------------------|----------------------------------|
| Onkeyup | Documents, images, links, textareas | Script runs when key is released |
|---------|-------------------------------------|----------------------------------|

|             |   |   |
|-------------|---|---|
| OnClick     | Button, radio button, checkbox, submit button, reset button, link | Script runs when a user click form element or link            |
| ondblclick  | Button, radio button, checkbox, submit button, reset button, link | Script runs when a user double-click mouse                    |
| onmousedown | Documents, buttons, links   | Script runs when mouse button is pressed                      |
| onmousemove |   | Script runs when mouse pointer moves                          |
| onmouseout  | Area, link  | Script runs when mouse pointer moves out of an form's element |
| onmouseover | Link  | Script runs when mouse pointer moves over an element          |
| onmouseup   | Documents, buttons, links   | Script runs when mouse button is released                     |
| Error       | Images, windows   | The loading of a document or image causes an error.           |

**onAbort:**

An onAbort event handler executes JavaScript code when the user aborts loading an image.

[See Example:](#)

<HTML>

<TITLE>Example of onAbort Event Handler</TITLE>

<HEAD>

</HEAD>

<BODY>

<H3>Example of onAbort Event Handler</H3>

<b>Stop the loading of this image and see what happens:</b><p>

<IMG SRC="reaz.gif" onAbort="alert('You stopped the loading the image!')">

</BODY>

| Event   | Chrome        | IE (older version) | Firefox       | Safari        | Opera         |
|---------|---------------|--------------------|---------------|---------------|---------------|
| onabort | Not supported | Yes                | Not supported | Not supported | Not supported |

</HTML>

Here, an alert() method is called using the onAbort event handler when the user aborts loading the image.

## Browser Support

### onBlur:

An onBlur event handler executes JavaScript code when input focus leaves the field of a text, textarea, or a select option. For windows, frames and framesets the event handler executes JavaScript code when the window loses focus. In windows you need to specify the event handler in the <BODY> attribute.

For example:

```
<BODY BGCOLOR='#ffffff' onBlur="document.bgcolor='#000000'">
```

**Note:** On a Windows platform, the onBlur event does not work with <FRAMESET>.

[See Example:](#)

```
<html>
  <head>
    <script type="text/javascript">
      function upperCase() {
        var x=document.getElementById("fname").value
        document.getElementById("fname").value=x.toUpperCase()
      }
    </script>
  </head>
```

```
<body>
  Enter your name:
  <input type="text" id="fname" onblur="upperCase()">
</body>
</html>
```

**onFocus:**

The onfocus event occurs when an element gets focus.

The onfocus event is most often used with <input>, <select>, and <a>.

In windows you need to specify the event handler in the <BODY> attribute.

For example:

```
<BODY BGCOLOR="#ffffff" onFocus="document.bgcolor='#000000'">
```

[See Example:](#)

```
<HTML>
  <TITLE>Example of onFocus Event Handler</TITLE>
  <HEAD></HEAD>
  <BODY>
    <H3>Example of onFocus Event Handler</H3>
    Click your mouse in the text box:<br>
    <form name="myform">
      <input type="text" name="data" value="" size=10
      onFocus='alert("You focused the textbox!!")'>
    </form>
  </BODY>
</HTML>
```

In the above example, when you put your mouse on the text box, an alert() message displays a message.

# Onchange:

The onchange event occurs when the value of an element has been changed.  
For radiobuttons and checkboxes, the onchange event occurs when the checked state has been changed.

[See Example:](#)

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function myFunction() {
        var x = document.getElementById("fname");
        x.value = x.value.toUpperCase();
      }
    </script>
  </head>
  <body>

    Enter your name: <input type="text" id="fname"
    onchange="myFunction()">

    <p>When you leave the input field, a function is triggered which
    transforms the input text to upper case.</p>

  </body>
</html>
```

## onClick:

The onclick event occurs when the user clicks on an element.

The **onClick** event handler is activated by a click on a form element. This can mean a radio or check button, but also submit, reset, or a user-defined button. In our example, if you click on a form element, a message should appear that tells you which element you clicked. Here is the source code:

```
<html>
<head>
  <title>Title of the Page</title>
  <script language="JavaScript">
    function message(element)
    {
      alert("You clicked the " + element + " element!")
    }
  </script>
</head>
<body>
  <form>
    <input type="radio" name="Radio" onClick="message('Radio Button
    1')">Option 1<br>
    <input type="radio" name="Radio" onClick="message('Radio Button
    2')">Option 2<br>
    <input type="checkbox" onClick="message('Checkbutton')">Check
    Button<br>
    <input type="submit" value="Send" onClick="message('Send Button')">
    <input type="reset" value="Reset" onClick="message('Reset Button')">
    <input type="button" value="Mine" onClick="message('My very own
    Button')">
  </form>
</body>
</html>
```

onLoad:

An onLoad event occurs when a window or image finishes loading. For windows, this event handler is specified in the BODY attribute of the window. In an image, the event handler will execute handler text when the image is loaded.

For example:

```
<IMG NAME="myimage" SRC="http://rhoque.com/ad_rh.jpg" onLoad="alert('You loaded myimage')">
```

See Example:

```
<HTML>
  <TITLE>Example of onLoad Event Handler</TITLE>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      function hello()
      {
        alert("Hello there...\nThis is an example of onLoad.");
      }
    </SCRIPT>
  </HEAD>
  <BODY onLoad="hello()">
    <H3>Example of onLoad Event Handler</H3>
  </BODY>
</HTML>
```

The example shows how the function hello() is called by using the onLoad event handler.

### **onmouseover and onmouseout:**

These two event types will help you to create nice effects with images or even with text as well. The *onmouseover* event occurs when you bring your mouse over any element and the *onmouseout* occurs when you take your mouse out from that element.

onMouseOver and onMouseOut are often used to create "animated" buttons.



See Example:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
    <script language="JavaScript">

      picture1=new Image
      picture1.src="picture1.png"
      picture2=new Image
      picture2.src="picture2.png"

    </script>
  </head>

  <body>
    <a onMouseOver="document.picture.src=picture2.src"
      onMouseOut="document.picture.src=picture1.src">
       </a>
  </body>
</html>
```

**onReset:**

A onReset event handler executes JavaScript code when the user resets a form by clicking on the reset button.

See Example:

```
<HTML>
  <TITLE>Example of onReset Event Handler</TITLE>
  <HEAD></HEAD>
<BODY>
  <H3> Example of onReset Event Handler </H3>
```

Please type something in the text box and press the reset button:<br>

```
<form name="myform" onReset="alert('This will reset the form!')">  
  <input type="text" name="data" value="" size="20">  
  <input type="reset" Value="Reset Form" name="myreset">  
</form>
```

```
</BODY>
```

```
</HTML>
```

In the above example, when you push the button, "Reset Form" after typing something, the alert method displays the message, "This will reset the form!"

onSelect:

A onSelect event handler executes JavaScript code when the user selects some of the text within a text or textarea field.

See Example:

```
<HTML>
```

```
  <TITLE>Example of onSelect Event Handler</TITLE>
```

```
  <HEAD></HEAD>
```

```
  <BODY>
```

```
    <H3>Example of onSelect Event Handler</H3>
```

```
    Select the text from the text box:<br>
```

```
    <form name="myform">
```

```
      <input type="text" name="data" value="Select This" size=20  
      onSelect="alert('This is an example of onSelect!!')">
```

```
    </form>
```

```
</BODY>
```

```
</HTML>
```

In the above example, when you try to select the text or part of the text, the alert method displays the message, "This is an example of onSelect!!".

## *Onsubmit:*

Another most important event type is *onsubmit*. This event occurs when you try to submit a form. So you can put your form validation against this event type.

Here is simple example showing its usage. 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.

### Example:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>

</head>
<body>
<form method="POST" name="myForm" action="mouse_over_out.html"
onsubmit="return validate()">
.....<br/>
<input type="text" name="uname"/> <br/>
<input type="submit" value="Submit" />
</form>
<script language="javascript" type="text/javascript">

function validate()
{

    var x = document.forms["myForm"]["uname"].value;
    if (x == null || x == "")
    {
        alert("Name must be filled out");
        return false;
    }
}
</script>
</body>
</html>
```

# onkeypress

Execute a JavaScript when a user presses a key:  
<input type="text" onkeypress="myFunction()">

The onkeypress event occurs when the user presses a key (on the keyboard).

```
<!DOCTYPE html>
<html>
  <body>

    <p>A function is triggered when the user is pressing a key in the input
    field.</p>

    <input type="text" onkeypress="myFunction()">

    <script>
      function myFunction() {
        alert("You pressed a key inside the input field");
      }
    </script>

  </body>
</html>
```

## onkeydown Event

Execute a JavaScript when a user is pressing a key:

<input type="text" onkeydown="myFunction()">

For ex.

```
<!DOCTYPE html>
<html>
<body>

<p>A function is triggered when the user is pressing a key in the input field.</p>

<input type="text" onkeydown="myFunction()">

<script>
function myFunction() {
  alert("You pressed a key inside the input field");
}
</script>
```

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

### onkeyup Event

Execute a JavaScript when a user releases a key:

```
<input type="text" onkeyup="myFunction()">
```

The onkeyup event occurs when the user releases a key (on the keyboard).

```
<!DOCTYPE html>  
<html>  
<body>
```

<p>A function is triggered when the user releases a key in the input field. The function transforms the character to upper case.</p>

Enter your name: <input type="text" id="fname" onkeyup="myFunction()">

```
<script>  
function myFunction() {  
    var x = document.getElementById("fname");  
    x.value = x.value.toUpperCase();  
}  
</script>  
  
</body>  
</html>
```