|  |  |  |
| --- | --- | --- |
| **Event** | **Description** | **Applies** |
| onabort | For the IMG object, this event fires when the user aborts the downloading of the image. | IMG element. |
| onafterprint  *IE5+ only event* | Fires right after the page has finished printing. Useful for reverting a page back to its original look after onbeforeprint has modified it for printing. The following reloads the page after it's completed printing, undoing any changes made by onbeforeprint:  window.onafterprint=function(){  setTimeout("window.location.reload()",50) } | Window and BODY element. |
| onbeforeprint  *IE5+ only event* | Fires just prior to the page being printed or previewed for printing. Useful for altering the look/structure of a page for printing. The following hides the column with ID "sidebar" plus any FORMs just before it's printed:  window.onbeforeprint=function(){  document.getElementById("sidebar").style.display="none"  var allforms=document.getElementsByTagName("form")  for (var i=0; i<allforms.length; i++)   allforms[i].style.display="none" } | Window and BODY element. |
| onbeforeunload | Fires just before the onunload event when the window is unloaded. If any string is returned as part of this event handler's code, the browser will display a dialog box when the page is just about to be unloaded that gives the user a chance to cancel the action:  window.onbeforeunload=function(){  //do some clean up  return "Message from JavaScriptKit.com" }  http://www.javascriptkit.com/jsref/onbeforeunload.gif | The window and IFRAME elements. |
| onblur | Fires when an element loses input focus. In most non IE browsers, this event only works on form elements such as INPUT and TEXTAREA, whereas in IE, it also fires for additional elements as  well. | FORM elements in non IE browsers, most elements in IE. |
| oncut | Fires when the selection data is cut and added to the system's clipboard. Returning false prevents this data from being cut. The following disables any text within a TEXTAREA from being cut (copy is still OK):  <textarea oncut="return false"></textarea> | Editable elements such as TEXTAREA, INPUT etc. |
| onchange | Fires when the selection or contents of an element changes, then the focus is removed from the element. Occurs after the onblur event. Applies to FORM elements, most commonly the SELECT element.  [Here's an example](http://www.javascriptkit.com/jsref/events.shtml#selectmenu) that creates a select menu that navigates to the selected web site upon selecting it within the list. | Applies to FORM elements, most commonly the SELECT element. |
| onclick | Fires when an element is clicked. Occurs following a onmousedown and onmouseup event. | Most elements. |
| onclose | Fires when the current window is closed. | Window element. |
| oncontextmenu | Fires when the user right clicks the mouse and activates the context menu of the browser. Returning false prevents the default context menu from appearing. | Most elements. |
| oncopy | Fires when the selection data is copied and added to the system's clipboard. Returning false prevents this data from being copied. | Most elements. |
| ondblclick | Fires when an element is double clicked. Occurs following a onmousedown, onmouseup, onclick, then onmouseup event. | Most elements. |
| onerror | For the window object, this event fires when a JavaScript error has occurred for any script following it. By returning true inside this event, JavaScript errors on the page (if any) are suppressed. Three parameters are implicitly passed into the onerror object:  window.onerror=function(msg, url, linenumber){  var logerror='Error message: ' + msg + '. Url: ' + url + 'Line Number: ' + linenumber  alert(logerror)  return true }  For the IMG object, this event fires when an error has occurred with the loading of the image:  <image src="test.gif" onerror="alert('This image didn\'t download successfully')" /> | Window and IMG elements. |
| onfocus | Fires when an element receives focus, such as a INPUT or TEXTAREA element. [Here's an example](http://www.javascriptkit.com/jsref/events.shtml#onfocus) that empties the default value inside a FORM text field when the focus is set on it. | FORM elements in non IE browsers, most elements in IE. |
| onhashchange  FF3.5+ and IE8+ event only | Fires when there's been a change to the hash portion of the browser URL (anything following the # sign). To set the hash property of the browser URL, use [window.location.hash](http://www.javascriptkit.com/jsref/location.shtml). One way to persist actions occurring within an Ajax application is to update [location.hash](http://www.javascriptkit.com/jsref/location.shtml) each time so the modified URL is added to the browser's history list and accessible when the user clicks the "Back" button of the browser. onhashchange can be used to easily detect whenever a change to the hash property has been made, and react accordingly. | Window and BODY element. |
| onkeydown | Fires when a key on the keyboard is pressed down. Returning false disables input from the keyboard. For more info, see "[Keyboard and Mouse Events](http://www.javascriptkit.com/jsref/eventkeyboardmouse.shtml)". | Most elements. |
| onkeypress | Fires when a key on the keyboard is pressed. Returning false disables input from the keyboard. The following shows how to use onkeypress to limit a text box's content to only alphabetical keys plus BACKSPACE and SPACE keys:  <form> <input type="text" id="alphabox" size="25"></textarea> </form>  <script type="text/javascript">  document.getElementById("alphabox").onkeypress=function(e){ var e=window.event || e var keyunicode=e.charCode || e.keyCode //Allow alphabetical keys, plus BACKSPACE and SPACE return (keyunicode>=65 && keyunicode<=122 || keyunicode==8 || keyunicode==32)? true : false }  </script>  For more info, see "[Keyboard and Mouse Events](http://www.javascriptkit.com/jsref/eventkeyboardmouse.shtml)". | Most elements. |
| onkeyup | Fires when a key on the keyboard is released after being depressed. For more info, see "[Keyboard and Mouse Events](http://www.javascriptkit.com/jsref/eventkeyboardmouse.shtml)". | Most elements. |
| onload | Fires when an element has loaded, typically, the window, IFRAME, or image element. | The window, IFRAME, and image element. |
| onmousedown | Fires when mouse button is pressed inside an element. | Most elements. |
| onmouseup | Fires when mouse button is released inside an element. | Most elements. |
| onmouseover | Fires when mouse moves over an element. | Most elements. |
| onmouseout | Fires when mouse moves out of an element. | Most elements. |
| onmousemove | Fires when mouse moves about inside an element. | Most elements. |
| onmousewheel  *Supported in IE6+, Opera, and Safari event. In FF, use addEventListener("DOMMouseScroll", funtionref, useCapture)* | Fires when the user moves the mouse wheel. The event object  wheelDelta property returns a positive or negative number indicating the amount the wheel has rotated plus the direction (up or down). The number returned is always in multiples of 120, where a value of 120 means the mouse wheel has been moved up one "click", while -120 means down one "click". If the user quickly moves the mouse wheel 3 clicks upwards for example, wheelDelta returns 720.  FF doesn't support the onmousewheel event, but rather, the equivalent DOMMouseScroll event that can only be binded to an element via scripting (ie: using element.addEventListener()). The event object property detail returns the distance travelled by the mouse wheel in multiples of 1, where 1 means the mouse wheel has been moved **down** one "click", while -1 means **up** one "click".  The following table better demonstrates the value released by onmousewheel and DOMMouseScroll to the event property wheelDelta and detail, respectively, each time the mouse wheel is rolled:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Event** | **Up 1 click** | **Up 2 clicks** | **Down 1 click** | **Down 2 clicks** | | **e.wheelDelta ( onmousewheel)** | 120 | 240 | -120 | -240 | | **e.detail (DOMMouseScroll)** | -1 | -2 | 1 | 2 |   [Click here](http://www.javascriptkit.com/jsref/events.shtml#mousewheeldemo) for a demo that contracts/ expands a DIV vertically as you move the mousewheel up and down within the DIV. | Most elements. |
| onmouseenter  *IE5+ only event* | Fires when the mouse first moves into an element, without firing again when the mouse also moves into inner elements inside this target (unlike onmouseover). This event does NOT bubble upwards. | Most elements. |
| onmouseleave  *IE5+ only event* | Fires when the mouse formally moves out of an element, without firing repeatedly as the mouse moves into inner elements inside this target (unlike onmouseover). This event does NOT bubble upwards. | Most elements. |
| onoffline | Fires when the browser is working offline. | BODY element |
| ononline | Fires when the browser is working online. | BODY element |
| onpaste | Fires when the selection data is pasted from the system clipboard back into the document. Returning false prevents this data from being pasted. | Editable elements such as TEXTAREA, INPUT etc. |
| onreadystatechange | Fires whenever the readyState property changes, allowing you to react to different stages of an asynchronous Ajax request. In some browsers such as Firefox, onreadystatechange fires multiple times while readyState is 3, letting you react as the data is downloading, such as to create a progress bar. IE (as of IE7) only fires onreadystatechange once during readyState of 3. The "readyState" property keeps track of the current stage of the request by returning one of the following values:   * 0: uninitialized * 1: loading * 2: loaded * 3: interactive * 4: complete   The following demonstrates checking the state of an Ajax request and waiting for it to be fully loaded before executing any follow up code:  var myrequest=new ajaxRequest() myrequest.onreadystatechange=function(){  if (**myrequest.readyState==4**){ //if request has completed   if (myrequest.status==200 || window.location.href.indexOf("http")==-1){ //if request was successful (versus "page not found" etc) or if run offline (rely on readyState of 4 alone then)    //do something with returned data   }  } }  In IE, onreadystatechange is also defined for regular elements on the page such as the document, and fires during each stage of the element's loading. | Asynchronous Ajax request. In IE, it is also applicable to most elements on the page. |
| onreset | Fires when the user resets a form by clicking on the "reset" button of a FORM. | The FORM element. |
| onresize | Fires when the window or an IFRAME is resized. In IE, onresize also fires for additional elements such as a DIV or SPAN when its dimensions have changed. | Window and IFRAME elements in non IE browsers, most elements in IE. |
| onscroll | Fires when an element is scrolled via its scrollbar. Typically applied to the window or elements with CSS set to overflow:scroll. | Scrollable elements. |
| onunload | Fires when the window or IFRAME is unloaded. | The window and IFRAME elements. |
| onselect | Fires when user selects some text within a TEXTAREA or INPUT. | TEXTAREA and INPUT elements. |
| onsubmit | Fires when the user submits a form by clicking on the "submit" button of a FORM. Returning false cancels the form submission.  Note that the FORM [submit()](http://www.javascriptkit.com/jsref/form.shtml) method does not trigger this event when used to submit a form. | The FORM element. |

**http://www.javascriptkit.com/jsref/bullet.gifExamples**

**http://www.javascriptkit.com/Red_CurlyC035.gifonfocus Example**

The following clears the default value of a form field when the user sets focus on it:

**Demo:**

<form>  
<input type="text" id="search" value="Search JavaScript Kit" style="width:200px" /> <input type="submit" value="Submit" />  
</form>  
  
<script type="text/javascript">  
  
var searchbox=document.getElementById("search")  
searchbox.onfocus=function(){  
 if (this.defaultValue==this.value)  
  this.value=""  
}  
  
</script>

**http://www.javascriptkit.com/Red_CurlyC035.gifonchange Example**

Here's a SELECT menu that navigates to a particular site immediately upon user selection, thanks to the onchange event handler:

**Demo:**

<form id="aform">  
<select id="mymenu" size="1">  
<option value="nothing" selected="selected">Select a site</option>  
<option value="http://www.dynamicdrive.com">Dynamic Drive</option>  
<option value="http://www.codingforums.com">Coding Forums</option>  
<option value="http://www.cssdrive.com">CSS Drive</option>  
</select>  
</form>  
  
<script type="text/javascript">  
  
var selectmenu=document.getElementById("mymenu")  
selectmenu.**onchange**=function(){ //run some code when "onchange" event fires  
 var chosenoption=this.options[this.selectedIndex] //this refers to "selectmenu"  
 if (chosenoption.value!="nothing"){  
  window.open(chosenoption.value, "", "") //open target site (based on option's value attr) in new window  
 }  
}  
  
</script>

**http://www.javascriptkit.com/Red_CurlyC035.gifonmousewheel Example**

The following example expands or contracts a DIV box whenever you move the mouse wheel downwards or upwards, respectively. It works in IE6+, FF3+, Opera 8+, and Safari3+:

**Demo:**

<div id="growbox" style="border:1px solid black; background:navy; width:300px; height:100px; color:white">Move Mouse Wheel to Expand/Contract me</div>  
  
<script type="text/javascript">  
  
var growbox=document.getElementById("growbox")  
  
function shrinkgrow(obj, e){  
 var evt=window.event || e //equalize event object  
 //\*\* Equalize value returned when mouse wheel is moved  
 //\*\* Result: -120 means down 1 click, 120 means up one click, and so on  
 var wheelclicks=evt.detail? evt.detail\*(-120) : evt.wheelDelta  
 var curheight=parseInt(obj.style.height)  
 var heightdelta=curheight\*0.1\*(wheelclicks/-120) //plus or minus 10% of current box height for each mousewheel "click"  
 if (wheelclicks<=-120 || wheelclicks>=120 && curheight>=100)  
  obj.style.height=curheight+heightdelta+"px"  
}  
  
 if (window.attachEvent)  
  growbox.attachEvent("onmousewheel", function(){shrinkgrow(growbox)}) //invoke function   
 else if (growbox.addEventListener)  
  growbox.addEventListener(/Firefox/i.test(navigator.userAgent)? "DOMMouseScroll" : "mousewheel", function(e){shrinkgrow(growbox, e)}, false);  
  
</script>