



HTML DOM Events

[< Previous](#)[Next >](#)

HTML DOM Events

HTML DOM events allow JavaScript to register different event handlers on elements in an HTML document.

Events are normally used in combination with functions, and the function will not be executed before the event occurs (such as when a user clicks a button).

Event	Belongs To	Description
<u>abort</u>	<u>UiEvent</u> , <u>Event</u>	The event occurs when the loading of a media is aborted
<u>afterprint</u>	<u>Event</u>	The event occurs when a page has started printing, or if the print dialogue box has been closed
<u>animationend</u>	<u>AnimationEvent</u>	The event occurs when a CSS animation has completed
<u>animationiteration</u>	<u>AnimationEvent</u>	The event occurs when a CSS animation is repeated
<u>animationstart</u>	<u>AnimationEvent</u>	The event occurs when a CSS animation has started

<u>beforeprint</u>	<u>Event</u>	The event occurs when a page is about to be printed
<u>beforeunload</u>	<u>UiEvent</u> , <u>Event</u>	The event occurs before the document is about to be unloaded
<u>blur</u>	<u>FocusEvent</u>	The event occurs when an element loses focus
<u>canplay</u>	<u>Event</u>	The event occurs when the browser can start playing the media (when it has buffered enough to begin)
<u>canplaythrough</u>	<u>Event</u>	The event occurs when the browser can play through the media without stopping for buffering
<u>change</u>	<u>Event</u>	The event occurs when the content of a form element, the selection, or the checked state have changed (for <input>, <select>, and <textarea>)
<u>click</u>	<u>MouseEvent</u>	The event occurs when the user clicks on an element
<u>contextmenu</u>	<u>MouseEvent</u>	The event occurs when the user right-clicks on an element to open a context menu
<u>copy</u>	<u>ClipboardEvent</u>	The event occurs when the user copies the content of an element
<u>cut</u>	<u>ClipboardEvent</u>	The event occurs when the user cuts the content of an element
<u>dblclick</u>	<u>MouseEvent</u>	The event occurs when the user double-clicks on an element
<u>drag</u>	<u>DragEvent</u>	The event occurs when an element is being dragged
<u>dragend</u>	<u>DragEvent</u>	The event occurs when the user has finished dragging an element
<u>dragenter</u>	<u>DragEvent</u>	The event occurs when the dragged element enters the drop target

<u>dragleave</u>	<u>DragEvent</u>	The event occurs when the dragged element leaves the drop target
<u>dragover</u>	<u>DragEvent</u>	The event occurs when the dragged element is over the drop target
<u>dragstart</u>	<u>DragEvent</u>	The event occurs when the user starts to drag an element
<u>drop</u>	<u>DragEvent</u>	The event occurs when the dragged element is dropped on the drop target
<u>durationchange</u>	<u>Event</u>	The event occurs when the duration of the media is changed
<u>ended</u>	<u>Event</u>	The event occurs when the media has reach the end (useful for messages like "thanks for listening")
<u>error</u>	<u>ProgressEvent</u> , <u>UiEvent</u> , <u>Event</u>	The event occurs when an error occurs while loading an external file
<u>focus</u>	<u>FocusEvent</u>	The event occurs when an element gets focus
<u>focusin</u>	<u>FocusEvent</u>	The event occurs when an element is about to get focus
<u>focusout</u>	<u>FocusEvent</u>	The event occurs when an element is about to lose focus
<u>hashchange</u>	<u>HashChangeEvent</u>	The event occurs when there has been changes to the anchor part of a URL
<u>input</u>	<u>InputEvent</u> , <u>Event</u>	The event occurs when an element gets user input
<u>invalid</u>	<u>Event</u>	The event occurs when an element is invalid
<u>keydown</u>	<u>KeyboardEvent</u>	The event occurs when the user is pressing a key
<u>keypress</u>	<u>KeyboardEvent</u>	The event occurs when the user presses a key

<u>keyup</u>	<u>KeyboardEvent</u>	The event occurs when the user releases a key
<u>load</u>	<u>UiEvent</u> , <u>Event</u>	The event occurs when an object has loaded
<u>loadeddata</u>	<u>Event</u>	The event occurs when media data is loaded
<u>loadedmetadata</u>	<u>Event</u>	The event occurs when meta data (like dimensions and duration) are loaded
<u>loadstart</u>	<u>ProgressEvent</u>	The event occurs when the browser starts looking for the specified media
<u>message</u>	<u>Event</u>	The event occurs when a message is received through the event source
<u>mousedown</u>	<u>MouseEvent</u>	The event occurs when the user presses a mouse button over an element
<u>mouseenter</u>	<u>MouseEvent</u>	The event occurs when the pointer is moved onto an element
<u>mouseleave</u>	<u>MouseEvent</u>	The event occurs when the pointer is moved out of an element
<u>mousemove</u>	<u>MouseEvent</u>	The event occurs when the pointer is moving while it is over an element
<u>mouseover</u>	<u>MouseEvent</u>	The event occurs when the pointer is moved onto an element, or onto one of its children
<u>mouseout</u>	<u>MouseEvent</u>	The event occurs when a user moves the mouse pointer out of an element, or out of one of its children
<u>mouseup</u>	<u>MouseEvent</u>	The event occurs when a user releases a mouse button over an element
mousewheel	<u>WheelEvent</u>	Deprecated. Use the <u>wheel</u> event instead
<u>offline</u>	<u>Event</u>	The event occurs when the browser starts to work offline
<u>online</u>	<u>Event</u>	The event occurs when the browser starts

		to work online
<u>open</u>	<u>Event</u>	The event occurs when a connection with the event source is opened
<u>pagehide</u>	<u>PageTransitionEvent</u>	The event occurs when the user navigates away from a webpage
<u>pageshow</u>	<u>PageTransitionEvent</u>	The event occurs when the user navigates to a webpage
<u>paste</u>	<u>ClipboardEvent</u>	The event occurs when the user pastes some content in an element
<u>pause</u>	<u>Event</u>	The event occurs when the media is paused either by the user or programmatically
<u>play</u>	<u>Event</u>	The event occurs when the media has been started or is no longer paused
<u>playing</u>	<u>Event</u>	The event occurs when the media is playing after having been paused or stopped for buffering
popstate	<u>PopStateEvent</u>	The event occurs when the window's history changes
<u>progress</u>	<u>Event</u>	The event occurs when the browser is in the process of getting the media data (downloading the media)
<u>ratechange</u>	<u>Event</u>	The event occurs when the playing speed of the media is changed
<u>resize</u>	<u>UiEvent</u> , <u>Event</u>	The event occurs when the document view is resized
<u>reset</u>	<u>Event</u>	The event occurs when a form is reset
<u>scroll</u>	<u>UiEvent</u> , <u>Event</u>	The event occurs when an element's scrollbar is being scrolled
<u>search</u>	<u>Event</u>	The event occurs when the user writes something in a search field (for <input="search">)

<u>seeked</u>	<u>Event</u>	The event occurs when the user is finished moving/skipping to a new position in the media
<u>seeking</u>	<u>Event</u>	The event occurs when the user starts moving/skipping to a new position in the media
<u>select</u>	<u>UiEvent</u> , <u>Event</u>	The event occurs after the user selects some text (for <input> and <textarea>)
<u>show</u>	<u>Event</u>	The event occurs when a <menu> element is shown as a context menu
<u>stalled</u>	<u>Event</u>	The event occurs when the browser is trying to get media data, but data is not available
storage	<u>StorageEvent</u>	The event occurs when a Web Storage area is updated
<u>submit</u>	<u>Event</u>	The event occurs when a form is submitted
<u>suspend</u>	<u>Event</u>	The event occurs when the browser is intentionally not getting media data
<u>timeupdate</u>	<u>Event</u>	The event occurs when the playing position has changed (like when the user fast forwards to a different point in the media)
<u>toggle</u>	<u>Event</u>	The event occurs when the user opens or closes the <details> element
<u>touchcancel</u>	<u>TouchEvent</u>	The event occurs when the touch is interrupted
<u>touchend</u>	<u>TouchEvent</u>	The event occurs when a finger is removed from a touch screen
<u>touchmove</u>	<u>TouchEvent</u>	The event occurs when a finger is dragged across the screen
<u>touchstart</u>	<u>TouchEvent</u>	The event occurs when a finger is placed on a touch screen

<u>transitionend</u>	<u>TransitionEvent</u>	The event occurs when a CSS transition has completed
<u>unload</u>	<u>UiEvent</u> , <u>Event</u>	The event occurs once a page has unloaded (for <body>)
<u>volumechange</u>	<u>Event</u>	The event occurs when the volume of the media has changed (includes setting the volume to "mute")
<u>waiting</u>	<u>Event</u>	The event occurs when the media has paused but is expected to resume (like when the media pauses to buffer more data)
<u>wheel</u>	<u>WheelEvent</u>	The event occurs when the mouse wheel rolls up or down over an element

HTML DOM Event Properties and Methods

Property/Method	Belongs To	Description
<u>altKey</u>	<u>MouseEvent</u>	Returns whether the "ALT" key was pressed when the mouse event was triggered
<u>altKey</u>	<u>KeyboardEvent</u> , <u>TouchEvent</u>	Returns whether the "ALT" key was pressed when the key event was triggered
<u>animationName</u>	<u>AnimationEvent</u>	Returns the name of the animation
<u>bubbles</u>	<u>Event</u>	Returns whether or not a specific event is a bubbling event
<u>button</u>	<u>MouseEvent</u>	Returns which mouse button was pressed when the mouse event was triggered
<u>buttons</u>	<u>MouseEvent</u>	Returns which mouse buttons were pressed when the mouse

		event was triggered
<u>cancelable</u>	<u>Event</u>	Returns whether or not an event can have its default action prevented
<u>charCode</u>	<u>KeyboardEvent</u>	Returns the Unicode character code of the key that triggered the onkeypress event
changeTouches	<u>TouchEvent</u>	Returns a list of all the touch objects whose state changed between the previous touch and this touch
<u>clientX</u>	<u>MouseEvent</u> , <u>TouchEvent</u>	Returns the horizontal coordinate of the mouse pointer, relative to the current window, when the mouse event was triggered
<u>clientY</u>	<u>MouseEvent</u> , <u>TouchEvent</u>	Returns the vertical coordinate of the mouse pointer, relative to the current window, when the mouse event was triggered
clipboardData	<u>ClipboardData</u>	Returns an object containing the data affected by the clipboard operation
code	<u>KeyboardEvent</u>	Returns the code of the key that triggered the event
composed	<u>Event</u>	Returns whether the event is composed or not
<u>createEvent()</u>	<u>Event</u>	Creates a new event
<u>ctrlKey</u>	<u>MouseEvent</u>	Returns whether the "CTRL" key was pressed when the mouse event was triggered
<u>ctrlKey</u>	<u>KeyboardEvent</u> , <u>TouchEvent</u>	Returns whether the "CTRL" key was pressed when the key event was triggered
<u>currentTarget</u>	<u>Event</u>	Returns the element whose

		event listeners triggered the event
data	<u>InputEvent</u>	Returns the inserted characters
dataTransfer	<u>DragEvent</u> , <u>InputEvent</u>	Returns an object containing the data being dragged/dropped, or inserted/deleted
<u>defaultPrevented</u>	<u>Event</u>	Returns whether or not the preventDefault() method was called for the event
deltaX	<u>WheelEvent</u>	Returns the horizontal scroll amount of a mouse wheel (x-axis)
deltaY	<u>WheelEvent</u>	Returns the vertical scroll amount of a mouse wheel (y-axis)
deltaZ	<u>WheelEvent</u>	Returns the scroll amount of a mouse wheel for the z-axis
deltaMode	<u>WheelEvent</u>	Returns a number that represents the unit of measurements for delta values (pixels, lines or pages)
<u>detail</u>	<u>UiEvent</u>	Returns a number that indicates how many times the mouse was clicked
<u>elapsedTime</u>	<u>AnimationEvent</u>	Returns the number of seconds an animation has been running
<u>elapsedTime</u>		Returns the number of seconds a transition has been running
<u>eventPhase</u>	<u>Event</u>	Returns which phase of the event flow is currently being evaluated
getTargetRanges()	<u>InputEvent</u>	Returns an array containing target ranges that will be

		affected by the insertion/deletion
getModifierState()	<u>MouseEvent</u>	Returns an array containing target ranges that will be affected by the insertion/deletion
inputType()	<u>InputEvent</u>	Returns the type of the change (i.e "inserting" or "deleting")
isComposing	<u>InputEvent</u> , <u>KeyboardEvent</u>	Returns whether the state of the event is composing or not
<u>isTrusted</u>	<u>Event</u>	Returns whether or not an event is trusted
<u>key</u>	<u>KeyboardEvent</u>	Returns the key value of the key represented by the event
key	<u>StorageEvent</u>	Returns the key of the changed storage item
<u>keyCode</u>	<u>KeyboardEvent</u>	Returns the Unicode character code of the key that triggered the onkeypress event, or the Unicode key code of the key that triggered the onkeydown or onkeyup event
<u>location</u>	<u>KeyboardEvent</u>	Returns the location of a key on the keyboard or device
lengthComputable	<u>ProgressEvent</u>	Returns whether the length of the progress can be computable or not
loaded	<u>ProgressEvent</u>	Returns how much work has been loaded
<u>metaKey</u>	<u>MouseEvent</u>	Returns whether the "META" key was pressed when an event was triggered
<u>metaKey</u>	<u>KeyboardEvent</u> , <u>TouchEvent</u>	Returns whether the "meta" key was pressed when the key

		event was triggered
MovementX	<u>MouseEvent</u>	Returns the horizontal coordinate of the mouse pointer relative to the position of the last mousemove event
MovementY	<u>MouseEvent</u>	Returns the vertical coordinate of the mouse pointer relative to the position of the last mousemove event
newValue	<u>StorageEvent</u>	Returns the new value of the changed storage item
<u>newURL</u>	<u>HasChangeEvent</u>	Returns the URL of the document, after the hash has been changed
offsetX	<u>MouseEvent</u>	Returns the horizontal coordinate of the mouse pointer relative to the position of the edge of the target element
offsetY	<u>MouseEvent</u>	Returns the vertical coordinate of the mouse pointer relative to the position of the edge of the target element
oldValue	<u>StorageEvent</u>	Returns the old value of the changed storage item
<u>oldURL</u>	<u>HasChangeEvent</u>	Returns the URL of the document, before the hash was changed
onemptied		The event occurs when something bad happens and the media file is suddenly unavailable (like unexpectedly disconnects)
<u>pageX</u>	<u>MouseEvent</u>	Returns the horizontal coordinate of the mouse pointer, relative to the

		document, when the mouse event was triggered
<u>pageY</u>	<u>MouseEvent</u>	Returns the vertical coordinate of the mouse pointer, relative to the document, when the mouse event was triggered
<u>persisted</u>	<u>PageTransitionEvent</u>	Returns whether the webpage was cached by the browser
<u>preventDefault()</u>	<u>Event</u>	Cancels the event if it is cancelable, meaning that the default action that belongs to the event will not occur
<u>propertyName</u>	<u>AnimationEvent</u> , <u>TransitionEvent</u>	Returns the name of the CSS property associated with the animation or transition
<u>pseudoElement</u>	<u>AnimationEvent</u> , <u>TransitionEvent</u>	Returns the name of the pseudo-element of the animation or transition
<u>region</u>	<u>MouseEvent</u>	Returns the name of the pseudo-element of the animation
<u>relatedTarget</u>	<u>MouseEvent</u>	Returns the element related to the element that triggered the mouse event
<u>relatedTarget</u>	<u>FocusEvent</u>	Returns the element related to the element that triggered the event
<u>repeat</u>	<u>KeyboardEvent</u>	Returns whether a key is being hold down repeatedly, or not
<u>screenX</u>	<u>MouseEvent</u>	Returns the horizontal coordinate of the mouse pointer, relative to the screen, when an event was triggered
<u>screenY</u>	<u>MouseEvent</u>	Returns the vertical coordinate of the mouse pointer, relative

		to the screen, when an event was triggered
<u>shiftKey</u>	<u>MouseEvent</u>	Returns whether the "SHIFT" key was pressed when an event was triggered
<u>shiftKey</u>	<u>KeyboardEvent</u> , <u>TouchEvent</u>	Returns whether the "SHIFT" key was pressed when the key event was triggered
state	<u>PopStateEvent</u>	Returns an object containing a copy of the history entries
<u>stopImmediatePropagation()</u>	<u>Event</u>	Prevents other listeners of the same event from being called
stopPropagation()	<u>Event</u>	Prevents further propagation of an event during event flow
storageArea	<u>StorageEvent</u>	Returns an object representing the affected storage object
<u>target</u>	<u>Event</u>	Returns the element that triggered the event
targetTouches	<u>TouchEvent</u>	Returns a list of all the touch objects that are in contact with the surface and where the touchstart event occurred on the same target element as the current target element
<u>timeStamp</u>	<u>Event</u>	Returns the time (in milliseconds relative to the epoch) at which the event was created
total	<u>ProgressEvent</u>	Returns the total amount of work that will be loaded
touches	<u>TouchEvent</u>	Returns a list of all the touch objects that are currently in contact with the surface
<u>transitionend</u>	<u>TransitionEvent</u>	The event occurs when a CSS

		transition has completed
<u>type</u>	<u>Event</u>	Returns the name of the event
<u>url</u>	<u>StorageEvent</u>	Returns the URL of the changed item's document
<u>which</u>	<u>MouseEvent</u>	Returns which mouse button was pressed when the mouse event was triggered
<u>which</u>	<u>KeyboardEvent</u>	Returns the Unicode character code of the key that triggered the onkeypress event, or the Unicode key code of the key that triggered the onkeydown or onkeyup event
<u>view</u>	<u>UiEvent</u>	Returns a reference to the Window object where the event occurred

[< Previous](#)

[Next >](#)



COLOR PICKER



HOW TO

Tabs

Dropdowns

Accordions

Convert Weights

Animated Buttons

Side Navigation

Top Navigation

Modal Boxes
Progress Bars
Parallax
Login Form
HTML Includes
Google Maps
Range Sliders
Tooltips
Slideshow
Filter List
Sort List

SHARE



CERTIFICATES

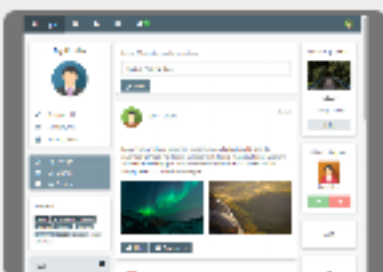
HTML, CSS, JavaScript, PHP, jQuery, Bootstrap and XML.

[Read More »](#)

Templates

by

W3.CSS



Templates

by

W3.CSS

[REPORT ERROR](#)
[PRINT PAGE](#)
[FORUM](#)
[ABOUT](#)

Top 10 Tutorials

[HTML Tutorial](#)
[CSS Tutorial](#)
[JavaScript Tutorial](#)
[W3.CSS Tutorial](#)
[Bootstrap Tutorial](#)
[SQL Tutorial](#)
[PHP Tutorial](#)
[jQuery Tutorial](#)
[Angular Tutorial](#)
[How To Tutorial](#)

Top 10 References

[HTML Reference](#)
[CSS Reference](#)
[JavaScript Reference](#)
[W3.CSS Reference](#)
[Bootstrap Reference](#)
[SQL Reference](#)
[PHP Reference](#)
[HTML Colors](#)
[jQuery Reference](#)
[AngularJS Reference](#)

Top 10 Examples

[HTML Examples](#)
[CSS Examples](#)

[JavaScript Examples](#)
[W3.CSS Examples](#)
[Bootstrap Examples](#)
[HTML DOM Examples](#)
[PHP Examples](#)
[jQuery Examples](#)
[Angular Examples](#)
[XML Examples](#)

Web Certificates

[HTML Certificate](#)
[CSS Certificate](#)
[JavaScript Certificate](#)
[jQuery Certificate](#)
[PHP Certificate](#)
[Bootstrap Certificate](#)
[XML Certificate](#)

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our [terms of use](#), [cookie and privacy policy](#). Copyright 1999-2018 by Refsnes Data. All Rights Reserved.

Powered by W3.CSS.

