HTML Element Reference

HTML Tags Ordered Alphabetically

Tag	Description
<!--...-->	Defines a comment
<!DOCTYPE>	Defines the document type
<a>	Defines a hyperlink
<abbr>	Defines an abbreviation or an acronym
<acronym>	Not supported in HTML5. Use <abbr> instead. Defines an acronym
<address>	Defines contact information for the author/owner of a document
<applet>	Not supported in HTML5. Use <embed> or <object> instead. Defines an embedded applet
<area>	Defines an area inside an image-map
<article>	Defines an article
<aside>	Defines content aside from the page content
<audio>	Defines sound content
<b>	Defines bold text
<base>	Specifies the base URL/target for all relative URLs in a document
<basefont>	Not supported in HTML5. Use CSS instead. Specifies a default color, size, and font for all text in a document
<bdi>	Isolates a part of text that might be formatted in a different direction from other text outside it
<bdo>	Overrides the current text direction
<big>	Not supported in HTML5. Use CSS instead. Defines big text
<blockquote>	Defines a section that is quoted from another source
<body>	Defines the document's body
 	Defines a single line break
<button>	Defines a clickable button
<canvas>	Used to draw graphics, on the fly, via scripting (usually JavaScript)
<caption>	Defines a table caption
<center>	Not supported in HTML5. Use CSS instead. Defines centered text
<cite>	Defines the title of a work
<code>	Defines a piece of computer code
<col>	Specifies column properties for each column within a <colgroup>

	element
<colgroup>	Specifies a group of one or more columns in a table for formatting
<data>	Links the given content with a machine-readable translation
<datalist>	Specifies a list of pre-defined options for input controls
<dd>	Defines a description/value of a term in a description list
<del>	Defines text that has been deleted from a document
<details>	Defines additional details that the user can view or hide
<dfn>	Represents the defining instance of a term
<dialog>	Defines a dialog box or window
<dir>	Not supported in HTML5. Use instead. Defines a directory list
<div>	Defines a section in a document
<dl>	Defines a description list
<dt>	Defines a term/name in a description list
<em>	Defines emphasized text
<embed>	Defines a container for an external (non-HTML) application
<fieldset>	Groups related elements in a form
<figcaption>	Defines a caption for a <figure> element
<figure>	Specifies self-contained content
<font>	Not supported in HTML5. Use CSS instead. Defines font, color, and size for text
<footer>	Defines a footer for a document or section
<form>	Defines an HTML form for user input
<frame>	Not supported in HTML5. Defines a window (a frame) in a frameset
<frameset>	Not supported in HTML5. Defines a set of frames
<h1> to <h6>	Defines HTML headings
<head>	Defines information about the document
<header>	Defines a header for a document or section
<hr>	Defines a thematic change in the content
<html>	Defines the root of an HTML document
<i>	Defines a part of text in an alternate voice or mood
<iframe>	Defines an inline frame
<img>	Defines an image
<input>	Defines an input control
<ins>	Defines a text that has been inserted into a document
<kbd>	Defines keyboard input
<label>	Defines a label for an <input> element
<legend>	Defines a caption for a <fieldset> element

Tag	Description
	Defines a list item
<link>	Defines the relationship between a document and an external resource (most used to link to style sheets)
<main>	Specifies the main content of a document
<map>	Defines a client-side image-map
<mark>	Defines marked/highlighted text
<meta>	Defines metadata about an HTML document
<meter>	Defines a scalar measurement within a known range (a gauge)
<nav>	Defines navigation links
<noframes>	Not supported in HTML5. Defines an alternate content for users that do not support frames
<noscript>	Defines an alternate content for users that do not support client-side scripts
<object>	Defines an embedded object
	Defines an ordered list
<optgroup>	Defines a group of related options in a drop-down list
<option>	Defines an option in a drop-down list
<output>	Defines the result of a calculation
<p>	Defines a paragraph
<param>	Defines a parameter for an object
<picture>	Defines a container for multiple image resources
<pre>	Defines preformatted text
<progress>	Represents the progress of a task
<q>	Defines a short quotation
<rp>	Defines what to show in browsers that do not support ruby annotations
<rt>	Defines an explanation/pronunciation of characters (for East Asian typography)
<ruby>	Defines a ruby annotation (for East Asian typography)
<s>	Defines text that is no longer correct
<samp>	Defines sample output from a computer program
<script>	Defines a client-side script
<section>	Defines a section in a document
<select>	Defines a drop-down list
<small>	Defines smaller text
<source>	Defines multiple media resources for media elements (<video> and <audio>)
<span>	Defines a section in a document
<strike>	Not supported in HTML5. Use <del> or <s> instead. Defines strikethrough text
<strong>	Defines important text

<style>	Defines style information for a document
<sub>	Defines subscripted text
<summary>	Defines a visible heading for a <details> element
<sup>	Defines superscripted text
<svg>	Defines a container for SVG graphics
<table>	Defines a table
<tbody>	Groups the body content in a table
<td>	Defines a cell in a table
<template>	Defines a template
<textarea>	Defines a multiline input control (text area)
<tfoot>	Groups the footer content in a table
<th>	Defines a header cell in a table
<thead>	Groups the header content in a table
<time>	Defines a date/time
<title>	Defines a title for the document
<tr>	Defines a row in a table
<track>	Defines text tracks for media elements (<video> and <audio>)
<tt>	Not supported in HTML5. Use CSS instead. Defines teletype text
<u>	Defines text that should be stylistically different from normal text
	Defines an unordered list
<var>	Defines a variable
<video>	Defines a video or movie
<wbr>	Defines a possible line-break

All HTML Attributes

Attribute	Belongs to	Description
accept	<input>	Specifies the types of files that the server accepts (only for type="file")
accept-charset	<form>	Specifies the character encodings that are to be used for the form submission
accesskey	Global Attributes	Specifies a shortcut key to activate/focus an element
action	<form>	Specifies where to send the form-data when a form is submitted
align	Not supported in HTML 5.	Specifies the alignment according to surrounding elements. Use CSS instead
alt	<area>, <img>, <input>	Specifies an alternate text when the original element fails to display
async	<script>	Specifies that the script is executed asynchronously (only for external scripts)
autocomplete	<form>, <input>	Specifies whether the <form> or the <input> element should have autocomplete enabled
autofocus	<button>, <input>, <select>, <textarea>	Specifies that the element should automatically get focus when the page loads
autoplay	<audio>, <video>	Specifies that the audio/video will start playing as soon as it is ready
bgcolor	Not supported in HTML 5.	Specifies the background color of an element. Use CSS instead
border	Not supported in HTML 5.	Specifies the width of the border of an element. Use CSS instead
charset	<meta>, <script>	Specifies the character encoding
checked	<input>	Specifies that an <input> element should be pre-selected when the page loads (for type="checkbox" or type="radio")
cite	<blockquote>, <del>, <ins>, <q>	Specifies a URL which explains the quote/deleted/inserted text
class	Global Attributes	Specifies one or more classnames for an element (refers to a class in a style

Attribute	Elements	Description
		sheet)
color	Not supported in HTML 5.	Specifies the text color of an element. Use CSS instead
cols	<textarea>	Specifies the visible width of a text area
colspan	<td>, <th>	Specifies the number of columns a table cell should span
content	<meta>	Gives the value associated with the http-equiv or name attribute
contenteditable	Global Attributes	Specifies whether the content of an element is editable or not
controls	<audio>, <video>	Specifies that audio/video controls should be displayed (such as a play/pause button etc)
coords	<area>	Specifies the coordinates of the area
data	<object>	Specifies the URL of the resource to be used by the object
data-*	Global Attributes	Used to store custom data private to the page or application
datetime	<del>, <ins>, <time>	Specifies the date and time
default	<track>	Specifies that the track is to be enabled if the user's preferences do not indicate that another track would be more appropriate
defer	<script>	Specifies that the script is executed when the page has finished parsing (only for external scripts)
dir	Global Attributes	Specifies the text direction for the content in an element
dirname	<input>, <textarea>	Specifies that the text direction will be submitted
disabled	<button>, <fieldset>, <input>, <optgroup>, <option>, <select>, <textarea>	Specifies that the specified element/group of elements should be disabled
download	<a>, <area>	Specifies that the target will be downloaded when a user clicks on the hyperlink
draggable	Global Attributes	Specifies whether an element is draggable or not
dropzone	Global Attributes	Specifies whether the dragged data is

		copied, moved, or linked, when dropped
enctype	<form>	Specifies how the form-data should be encoded when submitting it to the server (only for method="post")
for	<label>, <output>	Specifies which form element(s) a label/calculation is bound to
form	<button>, <fieldset>, <input>, <label>, <meter>, <object>, <output>, <select>, <textarea>	Specifies the name of the form the element belongs to
formaction	<button>, <input>	Specifies where to send the form-data when a form is submitted. Only for type="submit"
headers	<td>, <th>	Specifies one or more headers cells a cell is related to
height	<canvas>, <embed>, <iframe>, <img>, <input>, <object>, <video>	Specifies the height of the element
hidden	Global Attributes	Specifies that an element is not yet, or is no longer, relevant
high	<meter>	Specifies the range that is considered to be a high value
href	<a>, <area>, <base>, <link>	Specifies the URL of the page the link goes to
hreflang	<a>, <area>, <link>	Specifies the language of the linked document
http-equiv	<meta>	Provides an HTTP header for the information/value of the content attribute
id	Global Attributes	Specifies a unique id for an element
ismap	<img>	Specifies an image as a server-side image-map
kind	<track>	Specifies the kind of text track
label	<track>, <option>, <optgroup>	Specifies the title of the text track
lang	Global Attributes	Specifies the language of the element's content
list	<input>	Refers to a <datalist> element that contains pre-defined options for an <input> element
loop	<audio>, <video>	Specifies that the audio/video will start over again, every time it is finished
low	<meter>	Specifies the range that is considered to be a low value

max	\<input\>, \<meter\>, \<progress\>	Specifies the maximum value
maxlength	\<input\>, \<textarea\>	Specifies the maximum number of characters allowed in an element
media	\<a\>, \<area\>, \<link\>, \<source\>, \<style\>	Specifies what media/device the linked document is optimized for
method	\<form\>	Specifies the HTTP method to use when sending form-data
min	\<input\>, \<meter\>	Specifies a minimum value
multiple	\<input\>, \<select\>	Specifies that a user can enter more than one value
muted	\<video\>, \<audio\>	Specifies that the audio output of the video should be muted
name	\<button\>, \<fieldset\>, \<form\>, \<iframe\>, \<input\>, \<map\>, \<meta\>, \<object\>, \<output\>, \<param\>, \<select\>, \<textarea\>	Specifies the name of the element
novalidate	\<form\>	Specifies that the form should not be validated when submitted
onabort	\<audio\>, \<embed\>, \<img\>, \<object\>, \<video\>	Script to be run on abort
onafterprint	\<body\>	Script to be run after the document is printed
onbeforeprint	\<body\>	Script to be run before the document is printed
onbeforeunload	\<body\>	Script to be run when the document is about to be unloaded
onblur	All visible elements.	Script to be run when the element loses focus
oncanplay	\<audio\>, \<embed\>, \<object\>, \<video\>	Script to be run when a file is ready to start playing (when it has buffered enough to begin)
oncanplaythrough	\<audio\>, \<video\>	Script to be run when a file can be played all the way to the end without pausing for buffering
onchange	All visible elements.	Script to be run when the value of the element is changed
onclick	All visible elements.	Script to be run when the element is being clicked
oncontextmenu	All visible elements.	Script to be run when a context menu is triggered
oncopy	All visible elements.	Script to be run when the content of the element is being copied
oncuechange	\<track\>	Script to be run when the cue changes

		in a <track> element
oncut	All visible elements.	Script to be run when the content of the element is being cut
ondblclick	All visible elements.	Script to be run when the element is being double-clicked
ondrag	All visible elements.	Script to be run when the element is being dragged
ondragend	All visible elements.	Script to be run at the end of a drag operation
ondragenter	All visible elements.	Script to be run when an element has been dragged to a valid drop target
ondragleave	All visible elements.	Script to be run when an element leaves a valid drop target
ondragover	All visible elements.	Script to be run when an element is being dragged over a valid drop target
ondragstart	All visible elements.	Script to be run at the start of a drag operation
ondrop	All visible elements.	Script to be run when dragged element is being dropped
ondurationchange	<audio>, <video>	Script to be run when the length of the media changes
onemptied	<audio>, <video>	Script to be run when something bad happens and the file is suddenly unavailable (like unexpectedly disconnects)
onended	<audio>, <video>	Script to be run when the media has reach the end (a useful event for messages like "thanks for listening")
onerror	<audio>, <body>, <embed>, <img>, <object>, <script>, <style>, <video>	Script to be run when an error occurs
onfocus	All visible elements.	Script to be run when the element gets focus
onhashchange	<body>	Script to be run when there has been changes to the anchor part of the a URL
oninput	All visible elements.	Script to be run when the element gets user input
oninvalid	All visible elements.	Script to be run when the element is invalid
onkeydown	All visible elements.	Script to be run when a user is pressing a key
onkeypress	All visible elements.	Script to be run when a user presses a

		key
onkeyup	All visible elements.	Script to be run when a user releases a key
onload	<body>, <iframe>, <img>, <input>, <link>, <script>, <style>	Script to be run when the element is finished loading
onloadeddata	<audio>, <video>	Script to be run when media data is loaded
onloadedmetadata	<audio>, <video>	Script to be run when meta data (like dimensions and duration) are loaded
onloadstart	<audio>, <video>	Script to be run just as the file begins to load before anything is actually loaded
onmousedown	All visible elements.	Script to be run when a mouse button is pressed down on an element
onmousemove	All visible elements.	Script to be run as long as the mouse pointer is moving over an element
onmouseout	All visible elements.	Script to be run when a mouse pointer moves out of an element
onmouseover	All visible elements.	Script to be run when a mouse pointer moves over an element
onmouseup	All visible elements.	Script to be run when a mouse button is released over an element
onmousewheel	All visible elements.	Script to be run when a mouse wheel is being scrolled over an element
onoffline	<body>	Script to be run when the browser starts to work offline
ononline	<body>	Script to be run when the browser starts to work online
onpagehide	<body>	Script to be run when a user navigates away from a page
onpageshow	<body>	Script to be run when a user navigates to a page
onpaste	All visible elements.	Script to be run when the user pastes some content in an element
onpause	<audio>, <video>	Script to be run when the media is paused either by the user or programmatically
onplay	<audio>, <video>	Script to be run when the media has started playing
onplaying	<audio>, <video>	Script to be run when the media has started playing

onpopstate	<body>	Script to be run when the window's history changes.
onprogress	<audio>, <video>	Script to be run when the browser is in the process of getting the media data
onratechange	<audio>, <video>	Script to be run each time the playback rate changes (like when a user switches to a slow motion or fast forward mode).
onreset	<form>	Script to be run when a reset button in a form is clicked.
onresize	<body>	Script to be run when the browser window is being resized.
onscroll	All visible elements.	Script to be run when an element's scrollbar is being scrolled
onsearch	<input>	Script to be run when the user writes something in a search field (for <input="search">)
onseeked	<audio>, <video>	Script to be run when the seeking attribute is set to false indicating that seeking has ended
onseeking	<audio>, <video>	Script to be run when the seeking attribute is set to true indicating that seeking is active
onselect	All visible elements.	Script to be run when the element gets selected
onstalled	<audio>, <video>	Script to be run when the browser is unable to fetch the media data for whatever reason
onstorage	<body>	Script to be run when a Web Storage area is updated
onsubmit	<form>	Script to be run when a form is submitted
onsuspend	<audio>, <video>	Script to be run when fetching the media data is stopped before it is completely loaded for whatever reason
ontimeupdate	<audio>, <video>	Script to be run when the playing position has changed (like when the user fast forwards to a different point in the media)
ontoggle	<details>	Script to be run when the user opens or closes the <details> element
onunload	<body>	Script to be run when a page has unloaded (or the browser window has been closed)

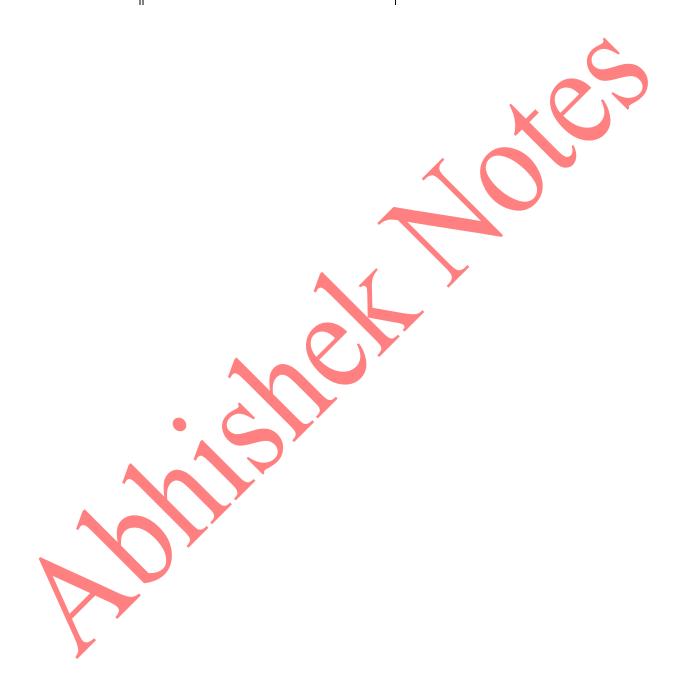|		
onvolumechange	<audio>, <video>	Script to be run each time the volume of a video/audio has been changed
onwaiting	<audio>, <video>	Script to be run when the media has paused but is expected to resume (like when the media pauses to buffer more data)
onwheel	All visible elements.	Script to be run when the mouse wheel rolls up or down over an element
open	<details>	Specifies that the details should be visible (open) to the user
optimum	<meter>	Specifies what value is the optimal value for the gauge
pattern	<input>	Specifies a regular expression that an <input> element's value is checked against
placeholder	<input>, <textarea>	Specifies a short hint that describes the expected value of the element
poster	<video>	Specifies an image to be shown while the video is downloading, or until the user hits the play button
preload	<audio>, <video>	Specifies if and how the author thinks the audio/video should be loaded when the page loads
readonly	<input>, <textarea>	Specifies that the element is read-only
rel	<a>, <area>, <link>	Specifies the relationship between the current document and the linked document
required	<input>, <select>, <textarea>	Specifies that the element must be filled out before submitting the form
reversed		Specifies that the list order should be descending (9,8,7...)
rows	<textarea>	Specifies the visible number of lines in a text area
rowspan	<td>, <th>	Specifies the number of rows a table cell should span
sandbox	<iframe>	Enables an extra set of restrictions for the content in an <iframe>
scope	<th>	Specifies whether a header cell is a header for a column, row, or group of columns or rows
selected	<option>	Specifies that an option should be pre-selected when the page loads
shape	<area>	Specifies the shape of the area

size	\<input>, \<select>	Specifies the width, in characters (for \<input>) or specifies the number of visible options (for \<select>)
sizes	\<img>, \<link>, \<source>	Specifies the size of the linked resource
span	\<col>, \<colgroup>	Specifies the number of columns to span
spellcheck	Global Attributes	Specifies whether the element is to have its spelling and grammar checked or not
src	\<audio>, \<embed>, \<iframe>, \<img>, \<input>, \<script>, \<source>, \<track>, \<video>	Specifies the URL of the media file
srcdoc	\<iframe>	Specifies the HTML content of the page to show in the \<iframe>
srclang	\<track>	Specifies the language of the track text data (required if kind="subtitles")
srcset	\<img>, \<source>	Specifies the URL of the image to use in different situations
start	\	Specifies the start value of an ordered list
step	\<input>	Specifies the legal number intervals for an input field
style	Global Attributes	Specifies an inline CSS style for an element
tabindex	Global Attributes	Specifies the tabbing order of an element
target	\<a>, \<area>, \<base>, \<form>	Specifies the target for where to open the linked document or where to submit the form
title	Global Attributes	Specifies extra information about an element
translate	Global Attributes	Specifies whether the content of an element should be translated or not
type	\<a>, \<button>, \<embed>, \<input>, \<link>, \<menu>, \<object>, \<script>, \<source>, \<style>	Specifies the type of element
usemap	\<img>, \<object>	Specifies an image as a client-side image-map
value	\<button>, \<input>, \ , \<option>, \<meter>, \<progress>, \<param>	Specifies the value of the element
width	\<canvas>, \<embed>, \<iframe>,	Specifies the width of the element

	<img>, <input>, <object>, <video>	
wrap	<textarea>	Specifies how the text in a text area is to be wrapped when submitted in a form

HTML Global Attributes

HTML attributes give elements meaning and context.

The global attributes below can be used on **any** HTML element.

HTML Global Attributes

Attribute	Description
accesskey	Specifies a shortcut key to activate/focus an element
class	Specifies one or more classnames for an element (refers to a class in a style sheet)
contenteditable	Specifies whether the content of an element is editable or not
data-*	Used to store custom data private to the page or application
dir	Specifies the text direction for the content in an element
draggable	Specifies whether an element is draggable or not
dropzone	Specifies whether the dragged data is copied, moved, or linked, when dropped
hidden	Specifies that an element is not yet, or is no longer, relevant
id	Specifies a unique id for an element
lang	Specifies the language of the element's content
spellcheck	Specifies whether the element is to have its spelling and grammar checked or not
style	Specifies an inline CSS style for an element
tabindex	Specifies the tabbing order of an element
title	Specifies extra information about an element
translate	Specifies whether the content of an element should be translated or not

HTML Event Attributes

Global Event Attributes

HTML has the ability to let events trigger actions in a browser, like starting a JavaScript when a user clicks on an element.

To learn more about programming events, please visit our JavaScript tutorial.

Below are the global event attributes that can be added to HTML elements to define event actions.

Window Event Attributes

Events triggered for the window object (applies to the <body> tag):

Attribute	Value	Description
onafterprint	*script*	Script to be run after the document is printed
onbeforeprint	*script*	Script to be run before the document is printed
onbeforeunload	*script*	Script to be run when the document is about to be unloaded
onerror	*script*	Script to be run when an error occurs
onhashchange	*script*	Script to be run when there has been changes to the anchor part of the a URL
onload	*script*	Fires after the page is finished loading
onmessage	*script*	Script to be run when the message is triggered
onoffline	*script*	Script to be run when the browser starts to work offline
ononline	*script*	Script to be run when the browser starts to work online
onpagehide	*script*	Script to be run when a user navigates away from a page
onpageshow	*script*	Script to be run when a user navigates to a page
onpopstate	*script*	Script to be run when the window's history changes
onresize	*script*	Fires when the browser window is resized
onstorage	*script*	Script to be run when a Web Storage area is updated
onunload	*script*	Fires once a page has unloaded (or the browser window has been closed)

Form Events

Events triggered by actions inside a HTML form (applies to almost all HTML elements, but is most used in form elements):

Attribute	Value	Description
onblur	*script*	Fires the moment that the element loses focus
onchange	*script*	Fires the moment when the value of the element is changed
oncontextmenu	*script*	Script to be run when a context menu is triggered
onfocus	*script*	Fires the moment when the element gets focus
oninput	*script*	Script to be run when an element gets user input
oninvalid	*script*	Script to be run when an element is invalid
onreset	*script*	Fires when the Reset button in a form is clicked
onsearch	*script*	Fires when the user writes something in a search field (for <input="search">)
onselect	*script*	Fires after some text has been selected in an element
onsubmit	*script*	Fires when a form is submitted

Keyboard Events

Attribute	Value	Description
onkeydown	*script*	Fires when a user is pressing a key
onkeypress	*script*	Fires when a user presses a key
onkeyup	*script*	Fires when a user releases a key

MouseEvents

Attribute	Value	Description
onclick	*script*	Fires on a mouse click on the element
ondblclick	*script*	Fires on a mouse double-click on the element
onmousedown	*script*	Fires when a mouse button is pressed down on an element
onmousemove	*script*	Fires when the mouse pointer is moving while it is over an element
onmouseout	*script*	Fires when the mouse pointer moves out of an element
onmouseover	*script*	Fires when the mouse pointer moves over an element
onmouseup	*script*	Fires when a mouse button is released over an element
onmousewheel	*script*	Deprecated. Use the onwheel attribute instead

onwheel	*script*	Fires when the mouse wheel rolls up or down over an element

Drag Events

Attribute	Value	Description
ondrag	*script*	Script to be run when an element is dragged
ondragend	*script*	Script to be run at the end of a drag operation
ondragenter	*script*	Script to be run when an element has been dragged to a valid drop target
ondragleave	*script*	Script to be run when an element leaves a valid drop target
ondragover	*script*	Script to be run when an element is being dragged over a valid drop target
ondragstart	*script*	Script to be run at the start of a drag operation
ondrop	*script*	Script to be run when dragged element is being dropped
onscroll	*script*	Script to be run when an element's scrollbar is being scrolled

Clipboard Events

Attribute	Value	Description
oncopy	*script*	Fires when the user copies the content of an element
oncut	*script*	Fires when the user cuts the content of an element
onpaste	*script*	Fires when the user pastes some content in an element

Media Events

Events triggered by medias like videos, images and audio (applies to all HTML elements, but is most common in media elements, like <audio>, <embed>, <img>, <object>, and <video>).

Tip: Look at our HTML Audio and Video DOM Reference for more information.

Attribute	Value	Description
onabort	*script*	Script to be run on abort
oncanplay	*script*	Script to be run when a file is ready to start playing (when it has buffered enough to begin)
Oncanplaythrough	*script*	Script to be run when a file can be played all the way to the end without pausing for buffering

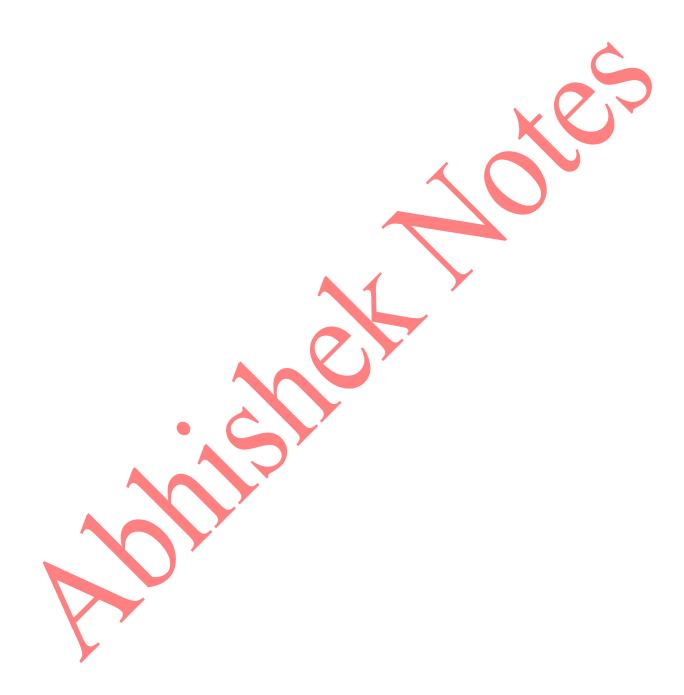|		
Oncuechange	*script*	Script to be run when the cue changes in a <track> element
Ondurationchange	*script*	Script to be run when the length of the media changes
Onemptied	*script*	Script to be run when something bad happens and the file is suddenly unavailable (like unexpectedly disconnects)
Onended	*script*	Script to be run when the media has reach the end (a useful event for messages like "thanks for listening")
Onerror	*script*	Script to be run when an error occurs when the file is being loaded
Onloadeddata	*script*	Script to be run when media data is loaded
Onloadedmetadata	*script*	Script to be run when meta data (like dimensions and duration) are loaded
Onloadstart	*script*	Script to be run just as the file begins to load before anything is actually loaded
Onpause	*script*	Script to be run when the media is paused either by the user or programmatically
Onplay	*script*	Script to be run when the media is ready to start playing
Onplaying	*script*	Script to be run when the media actually has started playing
Onprogress	*script*	Script to be run when the browser is in the process of getting the media data
onratechange	*script*	Script to be run each time the playback rate changes (like when a user switches to a slow motion or fast forward mode)
onseeked	*script*	Script to be run when the seeking attribute is set to false indicating that seeking has ended
onseeking	*script*	Script to be run when the seeking attribute is set to true indicating that seeking is active
onstalled	*script*	Script to be run when the browser is unable to fetch the media data for whatever reason
onsuspend	*script*	Script to be run when fetching the media data is stopped before it is completely loaded for whatever reason
ontimeupdate	*script*	Script to be run when the playing position has changed (like when the user fast forwards to a different point in the media)
onvolumechange	*script*	Script to be run each time the volume is changed which (includes setting the volume to "mute")
onwaiting	*script*	Script to be run when the media has paused but is expected to resume (like when the media pauses to buffer more data)

Misc Events

Attribute	Value	Description
ontoggle	*script*	Fires when the user opens or closes the <details> element

HTML Canvas Reference

Description

The HTML5 `<canvas>` tag is used to draw graphics, on the fly, via scripting (usually JavaScript).

However, the `<canvas>` element has no drawing abilities of its own (it is only a container for graphics) - you must use a script to actually draw the graphics.

The `getContext()` method returns an object that provides methods and properties for drawing on the canvas.

This reference will cover the properties and methods of the `getContext("2d")` object, which can be used to draw text, lines, boxes, circles, and more - on the canvas.

Browser Support

The numbers in the table specify the first browser version that fully supports the element.

Element					
`<canvas>`	4.0	9.0	2.0	3.1	9.0

Internet Explorer 9, Firefox, Opera, Chrome, and Safari support `<canvas>` and its properties and methods.

Note: Internet Explorer 8 and earlier versions, do not support the `<canvas>` element.

Colors, Styles, and Shadows

Property	Description
fillStyle	Sets or returns the color, gradient, or pattern used to fill the drawing
strokeStyle	Sets or returns the color, gradient, or pattern used for strokes
shadowColor	Sets or returns the color to use for shadows
shadowBlur	Sets or returns the blur level for shadows
shadowOffsetX	Sets or returns the horizontal distance of the shadow from the shape

shadowOffsetY	‖	Sets or returns the vertical distance of the shadow from the shape

Method	Description
createLinearGradient()	Creates a linear gradient (to use on canvas content)
createPattern()	Repeats a specified element in the specified direction
createRadialGradient()	Creates a radial/circular gradient (to use on canvas content)
addColorStop()	Specifies the colors and stop positions in a gradient object

Line Styles

Property	Description
lineCap	Sets or returns the style of the end caps for a line
lineJoin	Sets or returns the type of corner created, when two lines meet
lineWidth	Sets or returns the current line width
miterLimit	Sets or returns the maximum miter length

Rectangles

Method	Description
rect()	Creates a rectangle
fillRect()	Draws a "filled" rectangle
strokeRect()	Draws a rectangle (no fill)
clearRect()	Clears the specified pixels within a given rectangle

Paths

Method	Description
fill()	Fills the current drawing (path)
stroke()	Actually draws the path you have defined
beginPath()	Begins a path, or resets the current path
moveTo()	Moves the path to the specified point in the canvas, without creating a line
closePath()	Creates a path from the current point back to the starting point
lineTo()	Adds a new point and creates a line to that point from the last specified point in the canvas
clip()	Clips a region of any shape and size from the original canvas
quadraticCurveTo()	Creates a quadratic Bézier curve
bezierCurveTo()	Creates a cubic Bézier curve

arc()	Creates an arc/curve (used to create circles, or parts of circles)
arcTo()	Creates an arc/curve between two tangents
isPointInPath()	Returns true if the specified point is in the current path, otherwise false

Transformations

Method	Description
scale()	Scales the current drawing bigger or smaller
rotate()	Rotates the current drawing
translate()	Remaps the (0,0) position on the canvas
transform()	Replaces the current transformation matrix for the drawing
setTransform()	Resets the current transform to the identity matrix. Then runs transform()

Text

Property	Description
font	Sets or returns the current font properties for text content
textAlign	Sets or returns the current alignment for text content
textBaseline	Sets or returns the current text baseline used when drawing text

Method	Description
fillText()	Draws "filled" text on the canvas
strokeText()	Draws text on the canvas (no fill)
measureText()	Returns an object that contains the width of the specified text

Image Drawing

Method	Description
drawImage()	Draws an image, canvas, or video onto the canvas

Pixel Manipulation

Property	Description
width	Returns the width of an ImageData object
height	Returns the height of an ImageData object
data	Returns an object that contains image data of a specified ImageData object

Method	Description
createImageData()	Creates a new, blank ImageData object
getImageData()	Returns an ImageData object that copies the pixel data for the specified rectangle on a canvas
putImageData()	Puts the image data (from a specified ImageData object) back onto the canvas

Compositing

Property	Description
globalAlpha	Sets or returns the current alpha or transparency value of the drawing
globalCompositeOperation	Sets or returns how a new image are drawn onto an existing image

Other

Method	Description
save()	Saves the state of the current context
restore()	Returns previously saved path state and attributes
createEvent()	
getContext()	
toDataURL()	

HTML Audio/Video DOM Reference

HTML Audio and Video DOM Reference

The HTML5 DOM has methods, properties, and events for the `<audio>` and `<video>` elements.

These methods, properties, and events allow you to manipulate `<audio>` and `<video>` elements using JavaScript.

HTML Audio/Video Methods

Method	Description
addTextTrack()	Adds a new text track to the audio/video
canPlayType()	Checks if the browser can play the specified audio/video type
load()	Re-loads the audio/video element
play()	Starts playing the audio/video
pause()	Pauses the currently playing audio/video

HTML Audio/Video Properties

Property	Description
audioTracks	Returns an AudioTrackList object representing available audio tracks
autoplay	Sets or returns whether the audio/video should start playing as soon as it is loaded
buffered	Returns a TimeRanges object representing the buffered parts of the audio/video
controller	Returns the MediaController object representing the current media controller of the audio/video
controls	Sets or returns whether the audio/video should display controls (like play/pause etc.)
crossOrigin	Sets or returns the CORS settings of the audio/video
currentSrc	Returns the URL of the current audio/video
currentTime	Sets or returns the current playback position in the audio/video (in seconds)
defaultMuted	Sets or returns whether the audio/video should be muted by default
defaultPlaybackRate	Sets or returns the default speed of the audio/video playback
duration	Returns the length of the current audio/video (in seconds)
ended	Returns whether the playback of the audio/video has ended or not

error	Returns a MediaError object representing the error state of the audio/video
loop	Sets or returns whether the audio/video should start over again when finished
mediaGroup	Sets or returns the group the audio/video belongs to (used to link multiple audio/video elements)
muted	Sets or returns whether the audio/video is muted or not
networkState	Returns the current network state of the audio/video
paused	Returns whether the audio/video is paused or not
playbackRate	Sets or returns the speed of the audio/video playback
played	Returns a TimeRanges object representing the played parts of the audio/video
preload	Sets or returns whether the audio/video should be loaded when the page loads
readyState	Returns the current ready state of the audio/video
seekable	Returns a TimeRanges object representing the seekable parts of the audio/video
seeking	Returns whether the user is currently seeking in the audio/video
src	Sets or returns the current source of the audio/video element
startDate	Returns a Date object representing the current time offset
textTracks	Returns a TextTrackList object representing the available text tracks
videoTracks	Returns a VideoTrackList object representing the available video tracks
volume	Sets or returns the volume of the audio/video

HTML Audio/Video Events

Event	Description
abort	Fires when the loading of an audio/video is aborted
canplay	Fires when the browser can start playing the audio/video
canplaythrough	Fires when the browser can play through the audio/video without stopping for buffering
durationchange	Fires when the duration of the audio/video is changed
emptied	Fires when the current playlist is empty
ended	Fires when the current playlist is ended
error	Fires when an error occurred during the loading of an audio/video
loadeddata	Fires when the browser has loaded the current frame of the audio/video
loadedmetadata	Fires when the browser has loaded meta data for the audio/video
loadstart	Fires when the browser starts looking for the audio/video

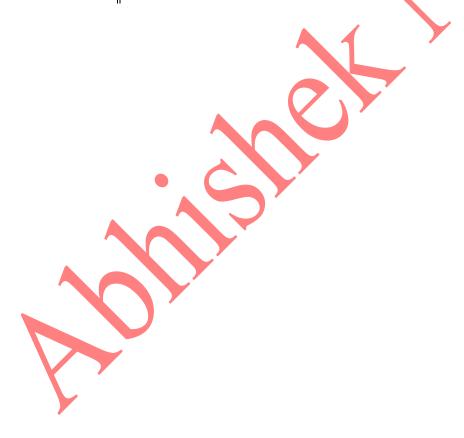|	
pause	Fires when the audio/video has been paused
play	Fires when the audio/video has been started or is no longer paused
playing	Fires when the audio/video is playing after having been paused or stopped for buffering
progress	Fires when the browser is downloading the audio/video
ratechange	Fires when the playing speed of the audio/video is changed
seeked	Fires when the user is finished moving/skipping to a new position in the audio/video
seeking	Fires when the user starts moving/skipping to a new position in the audio/video
stalled	Fires when the browser is trying to get media data, but data is not available
suspend	Fires when the browser is intentionally not getting media data
timeupdate	Fires when the current playback position has changed
volumechange	Fires when the volume has been changed
waiting	Fires when the video stops because it needs to buffer the next frame

HTML URL Encoding Reference

URL encoding converts characters into a format that can be transmitted over the Internet.

URL - Uniform Resource Locator

Web browsers request pages from web servers by using a URL.

The URL is the address of a web page, like: **https://www.w3schools.com.**

URL Encoding (Percent Encoding)

URLs can only be sent over the Internet using the [ASCII character-set](#).

Since URLs often contain characters outside the ASCII set, the URL has to be converted into a valid ASCII format.

URL encoding replaces unsafe ASCII characters with a "%" followed by two hexadecimal digits.

URLs cannot contain spaces. URL encoding normally replaces a space with a plus (+) sign or with %20.

Try It Yourself

If you click the "Submit" button below, the browser will URL encode the input before it is sent to the server. A page at the server will display the received input.

Hello Günter

Try some other input and click Submit again.

URL Encoding Functions

In JavaScript, PHP, and ASP there are functions that can be used to URL encode a string.

PHP has the rawurlencode() function, and ASP has the Server.URLEncode() function.

In JavaScript you can use the **encodeURIComponent()** function.

Click the "URL Encode" button to see how the JavaScript function encodes the text.

Hello Günter

Hello%20G%C3%BCnter

Note: The JavaScript function encodes space as %20.

ASCII Encoding Reference

Your browser will encode input, according to the character-set used in your page.

The default character-set in HTML5 is UTF-8.

Character	From Windows-1252	From UTF-8
space	%20	%20
!	%21	%21
"	%22	%22
#	%23	%23
$	%24	%24
%	%25	%25
&	%26	%26
'	%27	%27
(	%28	%28
)	%29	%29
*	%2A	%2A
+	%2B	%2B
,	%2C	%2C

-		%2D	%2D
.		%2E	%2E
/		%2F	%2F
0		%30	%30
1		%31	%31
2		%32	%32
3		%33	%33
4		%34	%34
5		%35	%35
6		%36	%36
7		%37	%37
8		%38	%38
9		%39	%39
:		%3A	%3A
;		%3B	%3B
<		%3C	%3C
=		%3D	%3D
>		%3E	%3E
?		%3F	%3F
@		%40	%40
A		%41	%41
B		%42	%42
C		%43	%43
D		%44	%44
E		%45	%45
F		%46	%46
G		%47	%47
H		%48	%48
I		%49	%49
J		%4A	%4A
K		%4B	%4B
L		%4C	%4C
M		%4D	%4D
N		%4E	%4E
O		%4F	%4F
P		%50	%50
Q		%51	%51
R		%52	%52
S		%53	%53

T	%54	%54
U	%55	%55
V	%56	%56
W	%57	%57
X	%58	%58
Y	%59	%59
Z	%5A	%5A
[	%5B	%5B
\	%5C	%5C
]	%5D	%5D
^	%5E	%5E
_	%5F	%5F
`	%60	%60
a	%61	%61
b	%62	%62
c	%63	%63
d	%64	%64
e	%65	%65
f	%66	%66
g	%67	%67
h	%68	%68
i	%69	%69
j	%6A	%6A
k	%6B	%6B
l	%6C	%6C
m	%6D	%6D
n	%6E	%6E
o	%6F	%6F
p	%70	%70
q	%71	%71
r	%72	%72
s	%73	%73
t	%74	%74
u	%75	%75
v	%76	%76
w	%77	%77
x	%78	%78
y	%79	%79
z	%7A	%7A

Char	Code 1	Code 2
{	%7B	%7B
|	%7C	%7C
}	%7D	%7D
~	%7E	%7E
	%7F	%7F
`	%80	%E2%82%AC
•	%81	%81
‚	%82	%E2%80%9A
ƒ	%83	%C6%92
„	%84	%E2%80%9E
…	%85	%E2%80%A6
†	%86	%E2%80%A0
‡	%87	%E2%80%A1
ˆ	%88	%CB%86
‰	%89	%E2%80%B0
Š	%8A	%C5%A0
‹	%8B	%E2%80%B9
Œ	%8C	%C5%92
•	%8D	%C5%8D
Ž	%8E	%C5%BD
•	%8F	%8F
•	%90	%C2%90
'	%91	%E2%80%98
'	%92	%E2%80%99
"	%93	%E2%80%9C
"	%94	%E2%80%9D
•	%95	%E2%80%A2
–	%96	%E2%80%93
—	%97	%E2%80%94
~	%98	%CB%9C
™	%99	%E2%84
š	%9A	%C5%A1
›	%9B	%E2%80
œ	%9C	%C5%93
•	%9D	%9D
ž	%9E	%C5%BE
Ÿ	%9F	%C5%B8
	%A0	%C2%A0
¡	%A1	%C2%A1

¢	%A2	%C2%A2
£	%A3	%C2%A3
¤	%A4	%C2%A4
¥	%A5	%C2%A5
¦	%A6	%C2%A6
§	%A7	%C2%A7
¨	%A8	%C2%A8
©	%A9	%C2%A9
ª	%AA	%C2%AA
«	%AB	%C2%AB
¬	%AC	%C2%AC
	%AD	%C2%AD
®	%AE	%C2%AE
¯	%AF	%C2%AF
°	%B0	%C2%B0
±	%B1	%C2%B1
²	%B2	%C2%B2
³	%B3	%C2%B3
´	%B4	%C2%B4
µ	%B5	%C2%B5
¶	%B6	%C2%B6
·	%B7	%C2%B7
¸	%B8	%C2%B8
¹	%B9	%C2%B9
º	%BA	%C2%BA
»	%BB	%C2%BB
¼	%BC	%C2%BC
½	%BD	%C2%BD
¾	%BE	%C2%BE
¿	%BF	%C2%BF
À	%C0	%C3%80
Á	%C1	%C3%81
Â	%C2	%C3%82
Ã	%C3	%C3%83
Ä	%C4	%C3%84
Å	%C5	%C3%85
Æ	%C6	%C3%86
Ç	%C7	%C3%87
È	%C8	%C3%88

É	%C9	%C3%89
Ê	%CA	%C3%8A
Ë	%CB	%C3%8B
Ì	%CC	%C3%8C
Í	%CD	%C3%8D
Î	%CE	%C3%8E
Ï	%CF	%C3%8F
Ð	%D0	%C3%90
Ñ	%D1	%C3%91
Ò	%D2	%C3%92
Ó	%D3	%C3%93
Ô	%D4	%C3%94
Õ	%D5	%C3%95
Ö	%D6	%C3%96
×	%D7	%C3%97
Ø	%D8	%C3%98
Ù	%D9	%C3%99
Ú	%DA	%C3%9A
Û	%DB	%C3%9B
Ü	%DC	%C3%9C
Ý	%DD	%C3%9D
Þ	%DE	%C3%9E
ß	%DF	%C3%9F
à	%E0	%C3%A0
á	%E1	%C3%A1
â	%E2	%C3%A2
ã	%E3	%C3%A3
ä	%E4	%C3%A4
å	%E5	%C3%A5
æ	%E6	%C3%A6
ç	%E7	%C3%A7
è	%E8	%C3%A8
é	%E9	%C3%A9
ê	%EA	%C3%AA
ë	%EB	%C3%AB
ì	%EC	%C3%AC
í	%ED	%C3%AD
î	%EE	%C3%AE
ï	%EF	%C3%AF

ð	%F0	%C3%B0
ñ	%F1	%C3%B1
ò	%F2	%C3%B2
ó	%F3	%C3%B3
ô	%F4	%C3%B4
õ	%F5	%C3%B5
ö	%F6	%C3%B6
÷	%F7	%C3%B7
ø	%F8	%C3%B8
ù	%F9	%C3%B9
ú	%FA	%C3%BA
û	%FB	%C3%BB
ü	%FC	%C3%BC
ý	%FD	%C3%BD
þ	%FE	%C3%BE
ÿ	%FF	%C3%BF

URL Encoding Reference

The ASCII control characters **%00-%1F** were originally designed to control hardware devices.

Control characters have nothing to do inside a URL.

ASCII Character	Description	URL-encoding
NUL	null character	%00
SOH	start of header	%01
STX	start of text	%02
ETX	end of text	%03
EOT	end of transmission	%04
ENQ	Enquiry	%05
ACK	acknowledge	%06
BEL	bell (ring)	%07
BS	backspace	%08
HT	horizontal tab	%09
LF	line feed	%0A
VT	vertical tab	%0B
FF	form feed	%0C
CR	carriage return	%0D
SO	shift out	%0E

SI	shift in	%0F
DLE	data link escape	%10
DC1	device control 1	%11
DC2	device control 2	%12
DC3	device control 3	%13
DC4	device control 4	%14
NAK	negative acknowledge	%15
SYN	synchronize	%16
ETB	end transmission block	%17
CAN	cancel	%18
EM	end of medium	%19
SUB	substitute	%1A
ESC	escape	%1B
FS	file separator	%1C
GS	group separator	%1D
RS	record separator	%1E
US	unit separator	%1F

HTML Language Code Reference

ISO Language Codes

The HTML lang attribute can be used to declare the language of a Web page or a portion of a Web page. This is meant to assist search engines and browsers.

According to the W3C recommendation you should declare the primary language for each Web page with the lang attribute inside the <html> tag,

For Example:-

```
<html lang="en">
...
</html>

In XHTML, the language is declared inside the <html> tag as follows:

<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
...
</html>
```

ISO 639-1 Language Codes

ISO 639-1 defines abbreviations for languages.

In HTML and XHTML they can be used in the lang and xml:lang attributes.

See also: next reference for country codes.

Language	ISO Code
Abkhazian	ab
Afar	aa
Afrikaans	af
Akan	ak
Albanian	sq
Amharic	am
Arabic	ar
Aragonese	an
Armenian	hy

Assamese	as
Avaric	av
Avestan	ae
Aymara	ay
Azerbaijani	az
Bambara	bm
Bashkir	ba
Basque	eu
Belarusian	be
Bengali (Bangla)	bn
Bihari	bh
Bislama	bi
Bosnian	bs
Breton	br
Bulgarian	bg
Burmese	my
Catalan	ca
Chamorro	ch
Chechen	ce
Chichewa, Chewa, Nyanja	ny
Chinese	zh
Chinese (Simplified)	zh-Hans
Chinese (Traditional)	zh-Hant
Chuvash	cv
Cornish	kw
Corsican	co
Cree	cr
Croatian	hr
Czech	cs
Danish	da
Divehi, Dhivehi, Maldivian	dv
Dutch	nl
Dzongkha	dz
English	en
Esperanto	eo
Estonian	et
Ewe	ee
Faroese	fo
Fijian	fj

Finnish	fi
French	fr
Fula, Fulah, Pulaar, Pular	ff
Galician	gl
Gaelic (Scottish)	gd
Gaelic (Manx)	gv
Georgian	ka
German	de
Greek	el
Greenlandic	kl
Guarani	gn
Gujarati	gu
Haitian Creole	ht
Hausa	ha
Hebrew	he
Herero	hz
Hindi	hi
Hiri Motu	ho
Hungarian	hu
Icelandic	is
Ido	io
Igbo	ig
Indonesian	id, in
Interlingua	ia
Interlingue	ie
Inuktitut	iu
Inupiak	ik
Irish	ga
Italian	it
Japanese	ja
Javanese	jv
Kalaallisut, Greenlandic	kl
Kannada	kn
Kanuri	kr
Kashmiri	ks
Kazakh	kk
Khmer	km
Kikuyu	ki
Kinyarwanda (Rwanda)	rw

Kirundi	rn
Kyrgyz	ky
Komi	kv
Kongo	kg
Korean	ko
Kurdish	ku
Kwanyama	kj
Lao	lo
Latin	la
Latvian (Lettish)	lv
Limburgish (Limburger)	li
Lingala	ln
Lithuanian	lt
Luga-Katanga	lu
Luganda, Ganda	lg
Luxembourgish	lb
Manx	gv
Macedonian	mk
Malagasy	mg
Malay	ms
Malayalam	ml
Maltese	mt
Maori	mi
Marathi	mr
Marshallese	mh
Moldavian	mo
Mongolian	mn
Nauru	na
Navajo	nv
Ndonga	ng
Northern Ndebele	nd
Nepali	ne
Norwegian	no
Norwegian bokmål	nb
Norwegian nynorsk	nn
Nuosu	ii
Occitan	oc
Ojibwe	oj
Old Church Slavonic, Old Bulgarian	cu

Oriya	or
Oromo (Afaan Oromo)	om
Ossetian	os
Pāli	pi
Pashto, Pushto	ps
Persian (Farsi)	fa
Polish	pl
Portuguese	pt
Punjabi (Eastern)	pa
Quechua	qu
Romansh	rm
Romanian	ro
Russian	ru
Sami	se
Samoan	sm
Sango	sg
Sanskrit	sa
Serbian	sr
Serbo-Croatian	sh
Sesotho	st
Setswana	tn
Shona	sn
Sichuan Yi	ii
Sindhi	sd
Sinhalese	si
Siswati	ss
Slovak	sk
Slovenian	sl
Somali	so
Southern Ndebele	nr
Spanish	es
Sundanese	su
Swahili (Kiswahili)	sw
Swati	ss
Swedish	sv
Tagalog	tl
Tahitian	ty
Tajik	tg
Tamil	ta

Tatar	tt
Telugu	te
Thai	th
Tibetan	bo
Tigrinya	ti
Tonga	to
Tsonga	ts
Turkish	tr
Turkmen	tk
Twi	tw
Uyghur	ug
Ukrainian	uk
Urdu	ur
Uzbek	uz
Venda	ve
Vietnamese	vi
Volapük	vo
Wallon	wa
Welsh	cy
Wolof	wo
Western Frisian	fy
Xhosa	xh
Yiddish	yi, ji
Yoruba	yo
Zhuang, Chuang	za
Zulu	zu

HTML ISO Country Codes Reference

ISO Country Codes

ISO country codes define abbreviations for countries.

In HTML they can be used as an addition to the language value in the lang attribute.

The first two characters of a language code defines the **language**. (see previous reference)

The last two defines the **country**.

For Example:-

```
<html lang="en-US">
...
</html>
```

ISO Country Codes

Country	ISO Code
AFGHANISTAN	AF
ALBANIA	AL
ALGERIA	DZ
AMERICAN SAMOA	AS
ANDORRA	AD
ANGOLA	AO
ANTARCTICA	AQ
ANTIGUA AND BARBUDA	AG
ARGENTINA	AR
ARMENIA	AM
ARUBA	AW
AUSTRALIA	AU
AUSTRIA	AT
AZERBAIJAN	AZ
BAHAMAS	BS
BAHRAIN	BH
BANGLADESH	BD
BARBADOS	BB

BELARUS	BY
BELGIUM	BE
BELIZE	BZ
BENIN	BJ
BERMUDA	BM
BHUTAN	BT
BOLIVIA	BO
BOSNIA AND HERZEGOVINA	BA
BOTSWANA	BW
BOUVET ISLAND	BV
BRAZIL	BR
BRITISH INDIAN OCEAN TERRITORY	IO
BRUNEI DARUSSALAM	BN
BULGARIA	BG
BURKINA FASO	BF
BURUNDI	BI
CAMBODIA	KH
CAMEROON	CM
CANADA	CA
CAPE VERDE	CV
CAYMAN ISLANDS	KY
CENTRAL AFRICAN REPUBLIC	CF
CHAD	TD
CHILE	CL
CHINA	CN
CHRISTMAS ISLAND	CX
COCOS (KEELING) ISLANDS	CC
COLOMBIA	CO
COMOROS	KM
CONGO	CG
CONGO, THE DEMOCRATIC REPUBLIC OF THE	CD
COOK ISLANDS	CK
COSTA RICA	CR
CÔTE D'IVOIRE	CI
CROATIA	HR
CUBA	CU
CYPRUS	CY
CZECH REPUBLIC	CZ
DENMARK	DK

DJIBOUTI	DJ
DOMINICA	DM
DOMINICAN REPUBLIC	DO
ECUADOR	EC
EGYPT	EG
EL SALVADOR	SV
EQUATORIAL GUINEA	GQ
ERITREA	ER
ESTONIA	EE
ETHIOPIA	ET
FALKLAND ISLANDS (MALVINAS)	FK
FAROE ISLANDS	FO
FIJI	FJ
FINLAND	FI
FRANCE	FR
FRENCH GUIANA	GF
FRENCH POLYNESIA	PF
FRENCH SOUTHERN TERRITORIES	TF
GABON	GA
GAMBIA	GM
GEORGIA	GE
GERMANY	DE
GHANA	GH
GIBRALTAR	GI
GREECE	GR
GREENLAND	GL
GRENADA	GD
GUADELOUPE	GP
GUAM	GU
GUATEMALA	GT
GUINEA	GN
GUINEA-BISSAU	GW
GUYANA	GY
HAITI	HT
HEARD ISLAND AND MCDONALD ISLANDS	HM
HONDURAS	HN
HONG KONG	HK
HUNGARY	HU
ICELAND	IS

INDIA	IN
INDONESIA	ID
IRAN, ISLAMIC REPUBLIC OF	IR
IRAQ	IQ
IRELAND	IE
ISRAEL	IL
ITALY	IT
JAMAICA	JM
JAPAN	JP
JORDAN	JO
KAZAKHSTAN	KZ
KENYA	KE
KIRIBATI	KI
KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF	KP
KOREA, REPUBLIC OF	KR
KUWAIT	KW
KYRGYZSTAN	KG
LAO PEOPLE'S DEMOCRATIC REPUBLIC (LAOS)	LA
LATVIA	LV
LEBANON	LB
LESOTHO	LS
LIBERIA	LR
LIBYAN ARAB JAMAHIRIYA	LY
LIECHTENSTEIN	LI
LITHUANIA	LT
LUXEMBOURG	LU
MACAO	MO
MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF	MK
MADAGASCAR	MG
MALAWI	MW
MALAYSIA	MY
MALDIVES	MV
MALI	ML
MALTA	MT
MARSHALL ISLANDS	MH
MARTINIQUE	MQ
MAURITANIA	MR
MAURITIUS	MU
MAYOTTE	YT

MEXICO	MX
MICRONESIA, FEDERATED STATES OF	FM
MOLDOVA, REPUBLIC OF	MD
MONACO	MC
MONGOLIA	MN
MONTENEGRO	ME
MONTSERRAT	MS
MOROCCO	MA
MOZAMBIQUE	MZ
MYANMAR	MM
NAMIBIA	NA
NAURU	NR
NEPAL	NP
NETHERLANDS	NL
NETHERLANDS ANTILLES	AN
NEW CALEDONIA	NC
NEW ZEALAND	NZ
NICARAGUA	NI
NIGER	NE
NIGERIA	NG
NIUE	NU
NORFOLK ISLAND	NF
NORTHERN MARIANA ISLANDS	MP
NORWAY	NO
OMAN	OM
PAKISTAN	PK
PALAU	PW
PALESTINIAN TERRITORY, OCCUPIED	PS
PANAMA	PA
PAPUA NEW GUINEA	PG
PARAGUAY	PY
PERU	PE
PHILIPPINES	PH
PITCAIRN	PN
POLAND	PL
PORTUGAL	PT
PUERTO RICO	PR
QATAR	QA
RÉUNION	RE

ROMANIA	RO
RUSSIAN FEDERATION	RU
RWANDA	RW
SAINT HELENA	SH
SAINT KITTS AND NEVIS	KN
SAINT LUCIA	LC
SAINT PIERRE AND MIQUELON	PM
SAINT VINCENT AND THE GRENADINES	VC
SAMOA	WS
SAN MARINO	SM
SAO TOME AND PRINCIPE	ST
SAUDI ARABIA	SA
SENEGAL	SN
SERBIA	RS
SEYCHELLES	SC
SIERRA LEONE	SL
SINGAPORE	SG
SLOVAKIA	SK
SLOVENIA	SI
SOLOMON ISLANDS	SB
SOMALIA	SO
SOUTH AFRICA	ZA
SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS	GS
SPAIN	ES
SRI LANKA	LK
SUDAN	SD
SURINAME	SR
SVALBARD AND JAN MAYEN	SJ
SWAZILAND	SZ
SWEDEN	SE
SWITZERLAND	CH
SYRIAN ARAB REPUBLIC	SY
TAIWAN	TW
TAJIKISTAN	TJ
TANZANIA, UNITED REPUBLIC OF	TZ
THAILAND	TH
TIMOR-LESTE	TL
TOGO	TG
TOKELAU	TK

TONGA		TO
TRINIDAD AND TOBAGO		TT
TUNISIA		TN
TURKEY		TR
TURKMENISTAN		TM
TURKS AND CAICOS ISLANDS		TC
TUVALU		TV
UGANDA		UG
UKRAINE		UA
UNITED ARAB EMIRATES		AE
UNITED KINGDOM		GB
UNITED STATES		US
UNITED STATES MINOR OUTLYING ISLANDS		UM
URUGUAY		UY
UZBEKISTAN		UZ
VANUATU		VU
VENEZUELA		VE
VIET NAM		VN
VIRGIN ISLANDS, BRITISH		VG
VIRGIN ISLANDS, U.S.		VI
WALLIS AND FUTUNA		WF
WESTERN SAHARA		EH
YEMEN		YE
ZAMBIA		ZM
ZIMBABWE		ZW

HTTP Status Messages

When a browser requests a service from a web server, an error might occur.

This is a list of HTTP status messages that might be returned:

1xx: Information

Message:	Description:
100 Continue	The server has received the request headers, and the client should proceed to send the request body
101 Switching Protocols	The requester has asked the server to switch protocols
103 Checkpoint	Used in the resumable requests proposal to resume aborted PUT or POST requests

2xx: Successful

Message:	Description:
200 OK	The request is OK (this is the standard response for successful HTTP requests)
201 Created	The request has been fulfilled, and a new resource is created
202 Accepted	The request has been accepted for processing, but the processing has not been completed
203 Non-Authoritative Information	The request has been successfully processed, but is returning information that may be from another source
204 No Content	The request has been successfully processed, but is not returning any content
205 Reset Content	The request has been successfully processed, but is not returning any content, and requires that the requester reset the document view
206 Partial Content	The server is delivering only part of the resource due to a range header sent by the client

This is not in used

3xx: Redirection

Message:	Description:

300 Multiple Choices	A link list. The user can select a link and go to that location. Maximum five addresses
301 Moved Permanently	The requested page has moved to a new URL
302 Found	The requested page has moved temporarily to a new URL
303 See Other	The requested page can be found under a different URL
304 Not Modified	Indicates the requested page has not been modified since last requested
306 Switch Proxy	*No longer used*
307 Temporary Redirect	The requested page has moved temporarily to a new URL
308 Resume Incomplete	Used in the resumable requests proposal to resume aborted PUT or POST requests

4xx: Client Error

Message:	Description:
400 Bad Request	The request cannot be fulfilled due to bad syntax
401 Unauthorized	The request was a legal request, but the server is refusing to respond to it. For use when authentication is possible but has failed or not yet been provided
402 Payment Required	*Reserved for future use*
403 Forbidden	The request was a legal request, but the server is refusing to respond to it
404 Not Found	The requested page could not be found but may be available again in the future
405 Method Not Allowed	A request was made of a page using a request method not supported by that page
406 Not Acceptable	The server can only generate a response that is not accepted by the client
407 Proxy Authentication Required	The client must first authenticate itself with the proxy
408 Request Timeout	The server timed out waiting for the request
409 Conflict	The request could not be completed because of a conflict in the request
410 Gone	The requested page is no longer available
411 Length Required	The "Content-Length" is not defined. The server will not accept the request without it
412 Precondition Failed	The precondition given in the request evaluated to false by the server
413 Request Entity Too Large	The server will not accept the request, because the request entity is too large

414 Request-URI Too Long	The server will not accept the request, because the URL is too long. Occurs when you convert a POST request to a GET request with a long query information
415 Unsupported Media Type	The server will not accept the request, because the media type is not supported
416 Requested Range Not Satisfiable	The client has asked for a portion of the file, but the server cannot supply that portion
417 Expectation Failed	The server cannot meet the requirements of the Expect request-header field

5xx: Server Error

Message:	Description:
500 Internal Server Error	A generic error message, given when no more specific message is suitable
501 Not Implemented	The server either does not recognize the request method, or it lacks the ability to fulfill the request
502 Bad Gateway	The server was acting as a gateway or proxy and received an invalid response from the upstream server
503 Service Unavailable	The server is currently unavailable (overloaded or down)
504 Gateway Timeout	The server was acting as a gateway or proxy and did not receive a timely response from the upstream server
505 HTTP Version Not Supported	The server does not support the HTTP protocol version used in the request
511 Network Authentication Required	The client needs to authenticate to gain network access

HTTP Request Methods

The two most common HTTP methods are: GET and POST.

What is HTTP?

The Hypertext Transfer Protocol (HTTP) is designed to enable communications between clients and servers.

HTTP works as a request-response protocol between a client and server.

A web browser may be the client, and an application on a computer that hosts a web site may be the server.

Example: A client (browser) submits an HTTP request to the server; then the server returns a response to the client. The response contains status information about the request and may also contain the requested content.

HTTP Methods

- **GET**
- **POST**
- **PUT**
- **HEAD**
- **DELETE**
- **PATCH**
- **OPTIONS**

The GET Method

GET is used to request data from a specified resource.

GET is one of the most common HTTP methods.

Note that the query string (name/value pairs) is sent in the URL of a GET request:

/test/demo_form.php?name1=value1&name2=value2

Some other notes on GET requests:

- GET requests can be cached
- GET requests remain in the browser history
- GET requests can be bookmarked
- GET requests should never be used when dealing with sensitive data
- GET requests have length restrictions
- GET requests is only used to request data (not modify)

The POST Method

POST is used to send data to a server to create/update a resource.

The data sent to the server with POST is stored in the request body of the HTTP request:

POST /test/demo_form.php HTTP/1.1
Host: w3schools.com
name1=value1&name2=value2

POST is one of the most common HTTP methods.

Some other notes on POST requests:

- POST requests are never cached
- POST requests do not remain in the browser history
- POST requests cannot be bookmarked
- POST requests have no restrictions on data length

The PUT Method

PUT is used to send data to a server to create/update a resource.

The difference between POST and PUT is that PUT requests are idempotent. That is, calling the same PUT request multiple times will always produce the same result. In contrast, calling a POST request repeatedly have side effects of creating the same resource multiple times.

The HEAD Method

HEAD is almost identical to GET, but without the response body.

In other words, if GET /users returns a list of users, then HEAD /users will make the same request but will not return the list of users.

HEAD requests are useful for checking what a GET request will return before actually making a GET request - like before downloading a large file or response body.

The DELETE Method

The DELETE method deletes the specified resource.

The OPTIONS Method

The OPTIONS method describes the communication options for the target resource.

Compare GET vs. POST

The following table compares the two HTTP methods: GET and POST.

	GET	POST
BACK button/Reload	Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
Bookmarked	Can be bookmarked	Cannot be bookmarked
Cached	Can be cached	Not cached
Encoding type	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data
History	Parameters remain in browser	Parameters are not saved in

	history	browser history
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed
Security	GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

Keyboard Shortcuts

Save time by using keyboard shortcuts.

Keyboard Shortcuts For Windows and Mac

Keyboard shortcuts are often used in modern operating systems and computer software programs.

Using keyboard shortcuts could save you a lot of time.

Basic Shortcuts

Description	Windows	Mac OS
Edit menu	Alt + E	Ctrl + F2 + F
File menu	Alt + F	Ctrl + F2 + E
View menu	Alt + V	Ctrl + F2 + V
Select all text	Ctrl + A	Cmd + A
Copy text	Ctrl + C	Cmd + C
Find text	Ctrl + F	Cmd + F
Find and replace text	Ctrl + H	Cmd + F
New Document	Ctrl + N	Cmd + N
Open a file	Ctrl + O	Cmd + O
Print options	Ctrl + P	Cmd + P
Save file	Ctrl + S	Cmd + S

Paste text	Ctrl + V	Cmd + V
Cut text	Ctrl + X	Cmd + X
Redo text	Ctrl + Y	Shift + Cmd + Z
Undo text	Ctrl + Z	Cmd + Z

Text Editing

Description	Windows	Mac OS
Cursor Movement		
Go to the right or to the beginning of next line break	Right Arrow	Right Arrow
Go to the left or to the end of previous line break	Left Arrow	Left Arrow
Go up one row	Up Arrow	Up Arrow
Go down one row	Down Arrow	Down Arrow
Go to the beginning of the current line	Home	Cmd + Left Arrow
Go to the end of the current line	End	Cmd + Right Arrow
Go to the beginning of the document	Ctrl + Home	Cmd + Up Arrow
Go to the end of the document	Ctrl + End	Cmd + Down Arrow
Move up one frame	Page Up	Fn + Up Arrow
Move down one frame	Page Down	Fn + Down Arrow
Go to beginning of previous word	Ctrl + Left Arrow	Option + Left Arrow

Go to beginning of next word	Ctrl + Right Arrow	Option + Right Arrow
Go to beginning of line break	Ctrl + Up Arrow	Cmd + Left Arrow
Go to end of line break	Ctrl + Down Arrow	Cmd + Right Arrow

Text Selection

Select characters to the left	Shift + Left Arrow	Shift + Left Arrow
Select characters to the right	Shift + Right Arrow	Shift + Right Arrow
Select lines upwards	Shift + Up Arrow	Shift + Up Arrow
Select lines downwards	Shift + Down Arrow	Shift + Down Arrow
Select words to the left	Shift + Ctrl + Left	Shift + Opt + Left
Select words to the right	Shift + Ctrl + Right	Shift + Opt + Right
Select paragraphs to the left	Shift + Ctrl + Up	Shift + Opt + Up
Select paragraphs to the right	Shift + Ctrl + Down	Shift + Opt + Down
Select text between the cursor and the beginning of the current line	Shift + Home	Cmd + Shift + Left Arrow
Select text between the cursor and the end of the current line	Shift + End	Cmd + Shift + Right Arrow
Select text between the cursor and the beginning of the document	Shift + Ctrl + Home	Cmd + Shift + Up Arrow or Cmd + Shift + Fn + Left Arrow
Select text between the cursor and the end of the document	Shift + Ctrl + End	Cmd + Shift + Down Arrow or Cmd + Shift + Fn + Right Arrow
Select one frame at a time of text above the cursor	Shift + Page Up	Shift + Fn + Up Arrow

Select one frame at a time of text below the cursor	Shift + Page Down	Shift + Fn + Down Arrow
Select all text	Ctrl + A	Cmd + A
Find text	Ctrl + F	Cmd + F

Text Formatting

Make selected text bold	Ctrl + B	Cmd + B
Make selected text italic	Ctrl + I	Cmd + I
Underline selected text	Ctrl + U	Cmd + U
Make selected text superscript	Ctrl + Shift + =	Cmd + Shift + =
Make selected text subscript	Ctrl + =	Cmd + =

Text Editing

Delete characters to the left	Backspace	Backspace
Delete characters to the right	Delete	Fn + Backspace
Delete words to the right	Ctrl + Del	Cmd + Backspace
Delete words to the left	Ctrl + Backspace	Cmd + Fn + Backspace
Indent	Tab	Tab
Outdent	Shift + Tab	Shift + Tab
Copy text	Ctrl + C	Cmd + C
Find and replace text	Ctrl + H	Cmd + F

Paste text	Ctrl + V	Cmd + V
Cut text	Ctrl + X	Cmd + X
Redo text	Ctrl + Y	Shift + Cmd + Z
Undo text	Ctrl + Z	Cmd + Z

Web Browsers

Description	Windows	Mac OS
Navigation		
Scroll down a frame	Space or Page Down	Space or Fn + Down Arrow
Scroll up a frame	Shift + Space or Page Up	Shift + Space or Fn + Up Arrow
Go to bottom of the page	End	Cmd + Down Arrow
Go to top of the page	Home	Cmd + Up Arrow
Go back	Alt + Left Arrow or Backspace	Cmd + Left Arrow
Go forward	Alt + Right Arrow or Shift + Backspace	Cmd + Right Arrow
Refresh a webpage	F5	Cmd + R
Refresh a webpage (no cache)	Ctrl + F5	Cmd + Shift + R
Stop	Esc	Esc
Toggle full-screen	F11	Cmd + Shift + F

Zoom in	Ctrl + +	Cmd + +
Zoom out	Ctrl + -	Cmd + -
Zoom 100% (default)	Ctrl + 0	Cmd + 0
Open homepage	Alt + Home	Option + Home or Option + Fn + Left Arrow
Find text	Ctrl + F	Cmd + F

Tab / Window Management

Open a new tab	Ctrl + T	Cmd + T
Close current tab	Ctrl + W	Cmd + W
Close all tabs	Ctrl + Shift + W	Cmd + Q
Close all tabs except the current tab	Ctrl + Alt + F4	Cmd + Opt + W
Go to next tab	Ctrl + Tab	Control + Tab or Cmd + Shift + Right Arrow
Go to previous tab	Ctrl + Shift + Tab	Shift + Control + Tab or Cmd + Shift Left Arrow
Go to a specific tab number	Ctrl + 1-8	Cmd + 1-8
Go to the last tab	Ctrl + 9	Cmd + 9
Reopen the last closed tab	Ctrl + Shift + T	Cmd + Shift + T
Open a new window	Ctrl + N	Cmd + N

Close current window	Alt + F4	Cmd + W
Go to next window	Alt + Tab	Cmd + Tab
Go to previous window	Alt + Shift + Tab	Cmd + Shift + Tab
Reopen the last closed window	Ctrl + Shift + N	
Open links in a new tab in the background	Ctrl + Click	Cmd + Click
Open links in a new tab in the foreground	Ctrl + Shift + Click	Cmd + Shift + Click
Print current webpage	Ctrl + P	Cmd + P
Save current webpage	Ctrl + S	Cmd + S

Address Bar

Cycle between toolbar, search bar, and page elements	Tab	Tab
Go to browser's address bar	Ctrl + L or Alt + D	Cmd + L
Focus and select the browser's search bar	Ctrl + E	Cmd + E / Cmd + K
Open the address bar location in a new tab	Alt + Enter	Opt + Enter
Display a list of previously typed addresses	F4	
Add "www." to the beginning and ".com" to the end of the text typed in the address bar (e.g., type "w3schools" and press Ctrl + Enter to open "www.w3schools.com")	Ctrl + Enter	Cmd + Enter or Control + Enter

Bookmarks

Open the bookmarks menu	Ctrl + B	Cmd + B
Add bookmark for current page	Ctrl + D	Cmd + Opt + B or

		Cmd + Shift + B
Open browsing history	Ctrl + H	Cmd + Shift + H or Cmd + Y
Open download history	Ctrl + J	Cmd + J or Cmd + Shift + J

Screenshots

Description	Windows	Mac OS
Save screenshot of the whole screen as file		Cmd + Shift + 3
Copy screenshot of the whole screen to the clipboard	PrtScr (Print Screen) or Ctrl + PrtScr	Cmd + Ctrl + Shift + 3
Save screenshot of window as file		Cmd + Shift + 4, then Space
Copy screenshot of window to the clipboard	Alt + PrtScr	Cmd + Ctrl + Shift + 4, then Space
Copy screenshot of wanted area to the clipboard		Cmd + Ctrl + Shift + 4
Save screenshot of wanted area as file **Note:** Due to different keyboard setups, some shortcuts may not be compatible for all users		Cmd + Shift + 4