

HTML attribute reference

Elements in HTML have **attributes**; these are additional values that configure the elements or adjust their behavior in various ways to meet the criteria the users want.

Attribute list

Attribute Name	Elements	Description
accept	<form> , <input>	List of types the server accepts, typically a file type.
accept-charset	<form>	List of supported charsets.
accesskey	Global attribute	Keyboard shortcut to activate or add focus to the element.
action	<form>	The URI of a program that processes the information submitted via the form.
align	<caption> , <col> , <colgroup> , <hr> , <iframe> , , <table> , <tbody> , <td> , <tfoot> , <th> , <thead> , <tr>	Specifies the horizontal alignment of the element.
allow	<iframe>	Specifies a feature-policy for the iframe.
alt	<area> , , <input>	Alternative text in case an image can't be displayed.
as	<link>	Specifies the type of content being loaded

Attribute Name	Elements	Description
		by the link.
async	<script>	Executes the script asynchronously.
autocapitalize	Global attribute	Sets whether input is automatically capitalized when entered by user
autocomplete	<form> , <input> , <select> , <textarea>	Indicates whether controls in this form can by default have their values automatically completed by the browser.
autoplay	<audio> , <video>	The audio or video should play as soon as possible.
background	<body> , <table> , <td> , <th>	Specifies the URL of an image file. Note: Although browsers and email clients may still support this attribute, it is obsolete. Use CSS background-image instead.
bgcolor	<body> , <col> , <colgroup> , <marquee> , <table> , <tbody> , <tfoot> , <td> , <th> , <tr>	Background color of the element. Note: This is a legacy attribute. Please use the CSS background-color property instead.
border	 , <object> , <table>	The border width.

Attribute Name	Elements	Description
		<p>Note: This is a legacy attribute. Please use the CSS border property instead.</p>
capture	<input>	From the Media Capture specification , specifies a new file can be captured.
charset	<meta>	Declares the character encoding of the page or script.
checked	<input>	Indicates whether the element should be checked on page load.
cite	<blockquote> , , <ins> , <q>	Contains a URI which points to the source of the quote or change.
class	Global attribute	Often used with CSS to style elements with common properties.
color	 , <hr>	<p>This attribute sets the text color using either a named color or a color specified in the hexadecimal #RRGGBB format.</p> <p>Note: This is a legacy attribute. Please use the CSS color property instead.</p>
cols	<textarea>	Defines the number of columns in a textarea.
colspan	<td> , <th>	The colspan attribute defines the number of columns a cell should span.

Attribute Name	Elements	Description
content	<meta>	A value associated with <code>http-equiv</code> or <code>name</code> depending on the context.
contenteditable	Global attribute	Indicates whether the element's content is editable.
controls	<audio> , <video>	Indicates whether the browser should show playback controls to the user.
coords	<area>	A set of values specifying the coordinates of the hot-spot region.
crossorigin	<audio> , , <link> , <script> , <video>	How the element handles cross-origin requests
csp	<iframe>	Specifies the Content Security Policy that an embedded document must agree to enforce upon itself.
data	<object>	Specifies the URL of the resource.
data-*	Global attribute	Lets you attach custom attributes to an HTML element.
datetime	 , <ins> , <time>	Indicates the date and time associated with the element.
decoding		Indicates the preferred method to decode the image.
default	<track>	Indicates that the track should be enabled unless the user's preferences indicate something different.
defer	<script>	Indicates that the script should be executed after the page has been parsed.

Attribute Name	Elements	Description
dir	Global attribute	Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
dirname	<input> , <textarea>	
disabled	<button> , <fieldset> , <input> , <optgroup> , <option> , <select> , <textarea>	Indicates whether the user can interact with the element.
download	<a> , <area>	Indicates that the hyperlink is to be used for downloading a resource.
draggable	Global attribute	Defines whether the element can be dragged.
enctype	<form>	Defines the content type of the form data when the method is POST.
enterkeyhint	<textarea> , contenteditable	The enterkeyhint specifies what action label (or icon) to present for the enter key on virtual keyboards. The attribute can be used with form controls (such as the value of <code>textarea</code> elements), or in elements in an editing host (e.g., using <code>contenteditable</code> attribute).
for	<label> , <output>	Describes elements which belongs to this one.
form	<button> , <fieldset> , <input> , <label> , <meter> , <object> , <output> , <progress> , <select> , <textarea>	Indicates the form that is the owner of the element.

Attribute Name	Elements	Description
<u>formaction</u>	<input> , <button>	Indicates the action of the element, overriding the action defined in the <form> .
<u>formenctype</u>	<button> , <input>	If the button/input is a submit button (e.g. type="submit"), this attribute sets the encoding type to use during form submission. If this attribute is specified, it overrides the enctype attribute of the button's form owner.
<u>formmethod</u>	<button> , <input>	If the button/input is a submit button (e.g. type="submit"), this attribute sets the submission method to use during form submission (GET , POST , etc.). If this attribute is specified, it overrides the method attribute of the button's form owner.
<u>formnovalidate</u>	<button> , <input>	If the button/input is a submit button (e.g. type="submit"), this boolean attribute specifies that the form is not to be validated when it is submitted. If this attribute is specified, it overrides the novalidate attribute of the button's form owner.
<u>formtarget</u>	<button> , <input>	If the button/input is a submit button (e.g. type="submit"), this attribute specifies the browsing context (for example, tab, window, or inline frame) in which to display the response that is received after submitting the form. If this attribute is specified, it overrides the target attribute of the button's form owner.
<u>headers</u>	<td> , <th>	IDs of the <th> elements which applies to

Attribute Name	Elements	Description
		this element.
height	<canvas> , <embed> , <iframe> , , <input> , <object> , <video>	Specifies the height of elements listed here. For all other elements, use the CSS height property. Note: In some instances, such as <div> , this is a legacy attribute, in which case the CSS height property should be used instead.
hidden	Global attribute	Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
high	<meter>	Indicates the lower bound of the upper range.
href	<a> , <area> , <base> , <link>	The URL of a linked resource.
hreflang	<a> , <link>	Specifies the language of the linked resource.
http-equiv	<meta>	Defines a pragma directive.
id	Global attribute	Often used with CSS to style a specific element. The value of this attribute must be unique.
integrity	<link> , <script>	Specifies a Subresource Integrity value that

Attribute Name	Elements	Description
		allows browsers to verify what they fetch.
intrinsicsize		This attribute tells the browser to ignore the actual intrinsic size of the image and pretend it's the size specified in the attribute.
inputmode	<textarea> , contenteditable	Provides a hint as to the type of data that might be entered by the user while editing the element or its contents. The attribute can be used with form controls (such as the value of <code>textarea</code> elements), or in elements in an editing host (e.g., using <code>contenteditable</code> attribute).
ismap		Indicates that the image is part of a server-side image map.
itemprop	Global attribute	
kind	<track>	Specifies the kind of text track.
label	<optgroup> , <option> , <track>	Specifies a user-readable title of the element.
lang	Global attribute	Defines the language used in the element.
language	<script>	Defines the script language used in the element.
loading	 , <iframe>	Indicates if the element should be loaded lazily (<code>loading="lazy"</code>) or loaded immediately (<code>loading="eager"</code>).
list	<input>	Identifies a list of pre-defined options to suggest to the user.

Attribute Name	Elements	Description
<u>loop</u>	<u><audio></u> , <u><marquee></u> , <u><video></u>	Indicates whether the media should start playing from the start when it's finished.
<u>low</u>	<u><meter></u>	Indicates the upper bound of the lower range.
<u>max</u>	<u><input></u> , <u><meter></u> , <u><progress></u>	Indicates the maximum value allowed.
<u>maxlength</u>	<u><input></u> , <u><textarea></u>	Defines the maximum number of characters allowed in the element.
<u>minlength</u>	<u><input></u> , <u><textarea></u>	Defines the minimum number of characters allowed in the element.
<u>media</u>	<u><a></u> , <u><area></u> , <u><link></u> , <u><source></u> , <u><style></u>	Specifies a hint of the media for which the linked resource was designed.
<u>method</u>	<u><form></u>	Defines which <u>HTTP</u> method to use when submitting the form. Can be GET (default) or POST .
<u>min</u>	<u><input></u> , <u><meter></u>	Indicates the minimum value allowed.
<u>multiple</u>	<u><input></u> , <u><select></u>	Indicates whether multiple values can be entered in an input of the type email or file .
<u>muted</u>	<u><audio></u> , <u><video></u>	Indicates whether the audio will be initially silenced on page load.
<u>name</u>	<u><button></u> , <u><form></u> , <u><fieldset></u> , <u><iframe></u> , <u><input></u> , <u><object></u> , <u><output></u> , <u><select></u> ,	Name of the element. For example used by the server to identify the fields in form submits.

Attribute Name	Elements	Description
	<textarea> , <map> , <meta> , <param>	
novalidate	<form>	This attribute indicates that the form shouldn't be validated when submitted.
open	<details> , <dialog>	Indicates whether the contents are currently visible (in the case of a <code><details></code> element) or whether the dialog is active and can be interacted with (in the case of a <code><dialog></code> element).
optimum	<meter>	Indicates the optimal numeric value.
pattern	<input>	Defines a regular expression which the element's value will be validated against.
ping	<a> , <area>	The <code>ping</code> attribute specifies a space-separated list of URLs to be notified if a user follows the hyperlink.
placeholder	<input> , <textarea>	Provides a hint to the user of what can be entered in the field.
playsinline	<video>	A Boolean attribute indicating that the video is to be played "inline"; that is, within the element's playback area. Note that the absence of this attribute does not imply that the video will always be played in fullscreen.
poster	<video>	A URL indicating a poster frame to show until the user plays or seeks.
preload	<audio> , <video>	Indicates whether the whole resource, parts of it or nothing should be preloaded.

Attribute Name	Elements	Description
readonly	<input> , <textarea>	Indicates whether the element can be edited.
referrerpolicy	<a> , <area> , <iframe> , , <link> , <script>	Specifies which referrer is sent when fetching the resource.
rel	<a> , <area> , <link>	Specifies the relationship of the target object to the link object.
required	<input> , <select> , <textarea>	Indicates whether this element is required to fill out or not.
reversed		Indicates whether the list should be displayed in a descending order instead of an ascending order.
role	Global attribute	Defines an explicit role for an element for use by assistive technologies.
rows	<textarea>	Defines the number of rows in a text area.
rowspan	<td> , <th>	Defines the number of rows a table cell should span over.
sandbox	<iframe>	Stops a document loaded in an iframe from using certain features (such as submitting forms or opening new windows).
scope	<th>	Defines the cells that the header test (defined in the <code>th</code> element) relates to.
scoped	<style>	
selected	<option>	Defines a value which will be selected on page load.
shape	<a> , <area>	

Attribute Name	Elements	Description
size	<input> , <select>	Defines the width of the element (in pixels). If the element's <code>type</code> attribute is <code>text</code> or <code>password</code> then it's the number of characters.
sizes	<link> , , <source>	
slot	Global attribute	Assigns a slot in a shadow DOM shadow tree to an element.
span	<col> , <colgroup>	
spellcheck	Global attribute	Indicates whether spell checking is allowed for the element.
src	<audio> , <embed> , <iframe> , , <input> , <script> , <source> , <track> , <video>	The URL of the embeddable content.
srcdoc	<iframe>	
srclang	<track>	
srcset	 , <source>	One or more responsive image candidates.
start		Defines the first number if other than 1.
step	<input>	
style	Global attribute	Defines CSS styles which will override styles previously set.
summary	<table>	
tabindex	Global attribute	Overrides the browser's default tab order and follows the one specified instead.

Attribute Name	Elements	Description
<code>target</code>	<a> , <area> , <base> , <form>	Specifies where to open the linked document (in the case of an <code><a></code> element) or where to display the response received (in the case of a <code><form></code> element)
title	Global attribute	Text to be displayed in a tooltip when hovering over the element.
translate	Global attribute	Specify whether an element's attribute values and the values of its Text node children are to be translated when the page is localized, or whether to leave them unchanged.
<code>type</code>	<button> , <input> , <embed> , <object> , , <script> , <source> , <style> , <menu> , <link>	Defines the type of the element.
<code>usemap</code>	 , <input> , <object>	
<code>value</code>	<button> , <data> , <input> , , <meter> , <option> , <progress> , <param>	Defines a default value which will be displayed in the element on page load.
<code>width</code>	<canvas> , <embed> , <iframe> , , <input> , <object> , <video>	For the elements listed here, this establishes the element's width. Note: For all other instances, such as <code><div></code> , this is a legacy attribute,

Attribute Name	Elements	Description
		in which case the CSS width property should be used instead.
wrap	<textarea>	Indicates whether the text should be wrapped.

Content versus IDL attributes

In HTML, most attributes have two faces: the **content attribute** and the **IDL (Interface Definition Language) attribute**.

The content attribute is the attribute as you set it from the content (the HTML code) and you can set it or get it via [element.setAttribute\(\)](#) or [element.getAttribute\(\)](#). The content attribute is always a string even when the expected value should be an integer. For example, to set an [<input>](#) element's `maxLength` to 42 using the content attribute, you have to call `setAttribute("maxLength", "42")` on that element.

The IDL attribute is also known as a JavaScript property. These are the attributes you can read or set using JavaScript properties like `element.foo`. The IDL attribute is always going to use (but might transform) the underlying content attribute to return a value when you get it and is going to save something in the content attribute when you set it. In other words, the IDL attributes, in essence, reflect the content attributes.

Most of the time, IDL attributes will return their values as they are really used. For example, the default `type` for [<input>](#) elements is "text", so if you set `input.type="foobar"`, the `<input>` element will be of type text (in the appearance and the behavior) but the "type" content attribute's value will be "foobar". However, the `type` IDL attribute will return the string "text".

IDL attributes are not always strings; for example, `input.maxLength` is a number (a signed long). When using IDL attributes, you read or set values of the desired type, so `input.maxLength` is always going to return a number and when you set `input.maxLength`, it wants a number. If you pass another type, it is automatically converted to a number as specified by the standard JavaScript rules for type conversion.

IDL attributes can [reflect other types](#) such as unsigned long, URLs, booleans, etc. Unfortunately, there are no clear rules and the way IDL attributes behave in conjunction with their corresponding content attributes depends on the attribute. Most of the time, it will follow [the rules laid out in the specification](#), but sometimes it doesn't. HTML specifications try to make this as developer-friendly as possible, but for various reasons (mostly historical), some attributes behave oddly (`select.size`, for example) and you should read the specifications to understand how exactly they behave.

Boolean Attributes

Some content attributes (e.g. `required`, `readonly`, `disabled`) are called [boolean attributes](#). If a boolean attribute is present, its value is **true**, and if it's absent, its value is **false**.

HTML defines restrictions on the allowed values of boolean attributes: If the attribute is present, its value must either be the empty string (equivalently, the attribute may have an unassigned value), or a value that is an ASCII case-insensitive match for the attribute's canonical name, with no leading or trailing whitespace. The following examples are valid ways to mark up a boolean attribute:

HTML

```
<div itemscope>This is valid HTML but invalid XML.</div>
<div itemscope=itemscope>This is also valid HTML but invalid XML.</div>
<div itemscope="">This is valid HTML and also valid XML.</div>
<div itemscope="itemscope">
  This is also valid HTML and XML, but perhaps a bit verbose.
</div>
```

To be clear, the values "true" and "false" are not allowed on boolean attributes. To represent a false value, the attribute has to be omitted altogether. This restriction clears up some common misunderstandings: With `checked="false"` for example, the element's `checked` attribute would be interpreted as **true** because the attribute is present.

Event handler attributes

Warning: The use of event handler content attributes is discouraged. The mix of HTML and JavaScript often produces unmaintainable code, and the execution of event handler attributes may also be blocked by content security policies.

In addition to the attributes listed in the table above, global [event handlers](#) — such as [onclick](#) — can also be specified as [content attributes](#) on all elements.

All event handler attributes accept a string. The string will be used to synthesize a [JavaScript function](#) like `function name(*args*) {body}`, where `name` is the attribute's name, and `body` is the attribute's value. The handler receives the same parameters as its JavaScript event handler counterpart — most handlers receive only one `event` parameter, while `onerror` receives five: `event`, `source`, `lineno`, `colno`, `error`. This means you can, in general, use the `event` variable within the attribute.

HTML

```
<div onclick="console.log(event)">Click me!</div>
<!-- The synthesized handler has a name; you can reference itself -->
<div onclick="console.log(onclick)">Click me!</div>
```

See also

- [HTML elements](#)

Help improve MDN



Was this page helpful to you?

[Learn how to contribute.](#)

This page was last modified on Jun 4, 2024 by [MDN contributors](#).

