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	Defines a 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`	`<applet>`, `<caption>`, `<col>`, `<colgroup>`, `<hr>`, `<iframe>`, `<img>`, `<table>`, `<tbody>`, `<td>`, `<tfoot>` , `<th>`, `<thead>`, `<tr>`	Specifies the horizontal alignment of the element.
`allow`	`<iframe>`	Specifies a feature-policy for the iframe.

Attribute Name	Elements	Description
`alt`	`<applet>`, `<area>`, `<img>`, `<input>`	Alternative text in case an image can't be displayed.
`async`	`<script>`	Indicates that the script should be executed asynchronously.
`autocapitalize`	Global attribute	Controls whether and how text input is automatically capitalized as it is entered/edited by the user.
`autocomplete`	`<form>`, `<input>`, `<textarea>`	Indicates whether controls in this form can by default have their values automatically completed by the browser.
`autofocus`	`<button>`, `<input>`, `<keygen>`, `<select>`, `<textarea>`	The element should be automatically focused after the page loaded.
`autoplay`	`<audio>`, `<video>`	The audio or video should play as soon as possible.
`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`	`<img>`, `<object>`, `<table>`	The border width. **Note:** This is a legacy attribute. Please use the CSS `border` property instead.
`buffered`	`<audio>`, `<video>`	Contains the time range of already buffered media.
`challenge`	`<keygen>`	A challenge string that is submitted along with the public key.
`charset`	`<meta>`, `<script>`	Declares the character encoding of the page or script.

Attribute Name	Elements	Description
checked	<command>, <input>	Indicates whether the element should be checked on page load.
cite	<blockquote>, <del>, <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.
code	<applet>	Specifies the URL of the applet's class file to be loaded and executed.
codebase	<applet>	This attribute gives the absolute or relative URL of the directory where applets' .class files referenced by the code attribute are stored.
color	<basefont>, <font>, <hr>	This attribute sets the text color using either a named color or a color specified in the hexadecimal #RRGGBB format. **Note:** This is a legacy attribute. Please use the CSS color property instead.
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.
content	<meta>	A value associated with http-equiv or name depending on the context.
contenteditable	Global attribute	Indicates whether the element's content is editable.
contextmenu	Global attribute	Defines the ID of a <menu> element which will serve as the element's context menu.
controls	<audio>, <video>	Indicates whether the browser should show playback controls to the user.

Attribute Name	Elements	Description
coords	<area>	A set of values specifying the coordinates of the hot-spot region.
crossorigin	<audio>, <img>, <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	<del>, <ins>, <time>	Indicates the date and time associated with the element.
decoding	<img>	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.
dir	Global attribute	Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
dirname	<input>, <textarea>	
disabled	<button>, <command>, <fieldset>, <input>, <keygen>, <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.

Attribute Name	Elements	Description
`draggable`	Global attribute	Defines whether the element can be dragged.
`dropzone`	Global attribute	Indicates that the element accept the dropping of content on it.
`enctype`	`<form>`	Defines the content type of the form date when the `method` is POST.
`for`	`<label>`, `<output>`	Describes elements which belongs to this one.
`form`	`<button>`, `<fieldset>`, `<input>`, `<keygen>`, `<label>`, `<meter>`, `<object>`, `<output>`, `<progress>`, `<select>`, `<textarea>`	Indicates the form that is the owner of the element.
`formaction`	`<input>`, `<button>`	Indicates the action of the element, overriding the action defined in the `<form>`.
`headers`	`<td>`, `<th>`	IDs of the `<th>` elements which applies to this element.
`height`	`<canvas>`, `<embed>`, `<iframe>`, `<img>`, `<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.

Attribute Name	Elements	Description
`hreflang`	`<a>`, `<area>`, `<link>`	Specifies the language of the linked resource.
`http-equiv`	`<meta>`	Defines a pragma directive.
`icon`	`<command>`	Specifies a picture which represents the command.
`id`	Global attribute	Often used with CSS to style a specific element. The value of this attribute must be unique.
`importance`	`<iframe>`, `<img>`, `<link>`, `<script>`	Indicates the relative fetch priority for the resource.
`integrity`	`<link>`, `<script>`	Security Feature that allows browsers to verify what they fetch.
`ismap`	`<img>`	Indicates that the image is part of a server-side image map.
`itemprop`	Global attribute	
`keytype`	`<keygen>`	Specifies the type of key generated.
`kind`	`<track>`	Specifies the kind of text track.
`label`	`<track>`	Specifies a user-readable title of the text track.
`lang`	Global attribute	Defines the language used in the element.
`language`	`<script>`	Defines the script language used in the element.
`lazyload`	`<img>`, `<iframe>`	Indicates if the element should be loaded lazily.
`list`	`<input>`	Identifies a list of pre-defined options to suggest to the user.
`loop`	`<audio>`, `<bgsound>`, `<marquee>`, `<video>`	Indicates whether the media should start playing from the start when it's finished.

Attribute Name	Elements	Description
low	<meter>	Indicates the upper bound of the lower range.
manifest	<html>	Specifies the URL of the document's cache manifest.
max	<input>, <meter>, <progress>	Indicates the maximum value allowed.
maxlength	<input>, <textarea>	Defines the maximum number of characters allowed in the element.
minlength	<input>, <textarea>	Defines the minimum number of characters allowed in the element.
media	<a>, <area>, <link>, <source>, <style>	Specifies a hint of the media for which the linked resource was designed.
method	<form>	Defines which HTTP method to use when submitting the form. Can be GET (default) or POST.
min	<input>, <meter>	Indicates the minimum value allowed.
multiple	<input>, <select>	Indicates whether multiple values can be entered in an input of the type email or file.
muted	<audio>, <video>	Indicates whether the audio will be initially silenced on page load.
name	<button>, <form>, <fieldset>, <iframe>, <input>, <keygen>, <object>, <output>, <select>, <textarea>, <map>, <meta>, <param>	Name of the element. For example used by the server to identify the fields in form submits.
novalidate	<form>	This attribute indicates that the form shouldn't be validated when submitted.
open	<details>	Indicates whether the details will be shown on page load.
optimum	<meter>	Indicates the optimal numeric value.

Attribute Name	Elements	Description
pattern	<input>	Defines a regular expression which the element's value will be validated against.
ping	<a>, <area>	
placeholder	<input>, <textarea>	Provides a hint to the user of what can be entered in the field.
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.
radiogroup	<command>	
readonly	<input>, <textarea>	Indicates whether the element can be edited.
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 a ascending.
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 th element) relates to.
scoped	<style>	

Attribute Name	Elements	Description
`selected`	`<option>`	Defines a value which will be selected on page load.
`shape`	`<a>`, `<area>`	
`size`	`<input>`, `<select>`	Defines the width of the element (in pixels). If the element's `type` attribute is `text` or `password` then it's the number of characters.
`sizes`	`<link>`, `<img>`, `<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>`, `<img>`, `<input>`, `<script>`, `<source>`, `<track>`, `<video>`	The URL of the embeddable content.
`srcdoc`	`<iframe>`	
`srclang`	`<track>`	
`srcset`	`<img>`, `<source>`	One or more responsive image candidates.
`start`	`<ol>`	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.
`target`	`<a>`, `<area>`, `<base>`, `<form>`	
`title`	Global attribute	Text to be displayed in a tooltip when hovering over the element.

Attribute Name	Elements	Description
`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.
`type`	`<button>`, `<input>`, `<command>`, `<embed>`, `<object>`, `<script>`, `<source>`, `<style>`, `<menu>`	Defines the type of the element.
`usemap`	`<img>`, `<input>`, `<object>`	
`value`	`<button>`, `<option>`, `<input>`, `<li>`, `<meter>`, `<progress>`, `<param>`	Defines a default value which will be displayed in the element on page load.
`width`	`<canvas>`, `<embed>`, `<iframe>`, `<img>`, `<input>`, `<object>`, `<video>`	For the elements listed here, this establishes the element's width. **Note:** For all other instances, such as `<div>`, this is a legacy attribute, 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 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.

See also

- HTML elements