

English ▼

Element.setAttribute()

Sets the value of an attribute on the specified element. If the attribute already exists, the value is updated; otherwise a new attribute is added with the specified name and value.

To get the current value of an attribute, use `getAttribute()`; to remove an attribute, call `removeAttribute()`.

Syntax

```
Element.setAttribute(name, value);
```

Parameters

name

A `DOMString` specifying the name of the attribute whose value is to be set. The attribute name is automatically converted to all lower-case when `setAttribute()` is called on an HTML element in an HTML document.

value

A `DOMString` containing the value to assign to the attribute. Any non-string value specified is converted automatically into a string.

Boolean attributes are considered to be `true` if they're present on the element at all, regardless of their actual `value`; as a rule, you should specify the empty string (`"`) in `value` (some people use the attribute's name; this works but is non-standard). See the [example](#) below for a practical demonstration.

Since the specified `value` gets converted into a string, specifying `null` doesn't necessarily do what you expect. Instead of removing the attribute or setting its value to be `null`, it instead

sets the attribute's value to the string `"null"`. If you wish to remove an attribute, call `removeAttribute()`.

Return value

`undefined`.

Exceptions

`InvalidCharacterError`

The specified attribute `name` contains one or more characters which are not valid in attribute names.

Example

In the following example, `setAttribute()` is used to set attributes on a `<button>`.

HTML

```
1 | <button>Hello World</button>
```

JavaScript

```
1 | var b = document.querySelector("button");
2 |
3 | b.setAttribute("name", "helloButton");
4 | b.setAttribute("disabled", "");
```

This demonstrates two things:

- The first call to `setAttribute()` above shows changing the `name` attribute's value to `"helloButton"`. You can see this using your browser's page inspector ([Chrome](#), [Edge](#), [Firefox](#), [Safari](#)).

- To set the value of a Boolean attribute, such as `disabled`, you can specify any value. An empty string or the name of the attribute are recommended values. All that matters is that if the attribute is present at all, *regardless of its actual value*, its value is considered to be `true`. The absence of the attribute means its value is `false`. By setting the value of the `disabled` attribute to the empty string (`""`), we are setting `disabled` to `true`, which results in the button being disabled.



DOM methods dealing with element's attributes:

Not namespace-aware, most commonly used methods	Namespace-aware variants (DOM Level 2)	DOM Level 1 methods for dealing with Attr nodes directly (seldom used)	DOM Level 2 namespace-aware methods for dealing with Attr nodes directly (seldom used)
<code>setAttribute</code> (DOM 1)	<code>setAttributeNS</code>	<code>setAttributeNode</code>	<code>setAttributeNodeNS</code>
<code>getAttribute</code> (DOM 1)	<code>getAttributeNS</code>	<code>getAttributeNode</code>	<code>getAttributeNodeNS</code>
<code>hasAttribute</code> (DOM 2)	<code>hasAttributeNS</code>	-	-
<code>removeAttribute</code> (DOM 1)	<code>removeAttributeNS</code>	<code>removeAttributeNode</code>	-

Specifications

Specification	Status	Comment
<code>DOM</code> The definition of 'setAttribute()' in that specification.	<div><div>LS</div>Living Standard</div>	

Browser compatibility

[Update compatibility data on GitHub](#)

setAttribute	
Chrome	1
Edge	12
Firefox	1
IE	5★
Opera	8
Safari	3
WebView Android	1
Chrome Android	18
Firefox Android	4
Opera Android	10.1
Safari iOS	1
Samsung Internet Android	1.0

What are we missing?



Full support

★

See implementation notes.

Gecko notes

Using `setAttribute()` to modify certain attributes, most notably `value` in XUL, works inconsistently, as the attribute specifies the default value. To access or modify the current values, you should use the properties. For example, use `Element.value` instead of `Element.setAttribute()`.

Related Topics

Document Object Model

Element

▼ Properties

- attributes
- childElementCount
- children
- classList
- className
- clientHeight
- clientLeft
- clientTop
- clientWidth
- ⚠️ currentStyle
- firstElementChild
- id
- innerHTML
- lastElementChild
- localName
- namespaceURI
- nextElementSibling
- onfullscreenchange
- onfullscreenerror
- ⚠️ openOrClosedShadowRoot
- outerHTML
- part
- prefix
- previousElementSibling
- ⚠️ runtimeStyle
- scrollHeight
- scrollLeft

⚠️ scrollLeftMax

scrollTop

⚠️ scrollTopMax

scrollWidth

shadowRoot

slot

⚠️ 🗑️ tabStop

tagName

▼ Methods

after()

animate()

append()

attachShadow()

before()

closest()

🧪 computedStyleMap()

⚠️ 🗑️ createShadowRoot()

🧪 getAnimations()

getAttribute()

getAttributeNames()

getAttributeNode()

getAttributeNodeNS()

getAttributeNS()

getBoundingClientRect()

getClientRects()

getElementsByClassName()

getElementsByTagName()

getElementsByTagNameNS()

hasAttribute()

hasAttributeNS()

hasAttributes()

hasPointerCapture()

insertAdjacentElement()

`insertAdjacentHTML()`

`insertAdjacentText()`

`matches()`

 `msZoomTo()`

`prepend()`

`querySelector()`

`querySelector()`

`querySelectorAll()`

`querySelectorAll()`

`releasePointerCapture()`

 `remove()`

`removeAttribute()`

`removeAttributeNode()`

`removeAttributeNS()`

`replaceChildren()`

`replaceWith()`

`requestFullscreen()`

 `requestPointerLock()`

`scroll()`

`scrollBy()`

`scrollIntoView()`

 `scrollIntoViewIfNeeded()`

`scrollTo()`

`setAttribute()`

`setAttributeNode()`

`setAttributeNodeNS()`

`setAttributeNS()`

 `setCapture()`

`setPointerCapture()`

`toggleAttribute()`

▼ Events

 `afterscriptexecute`

`auxclick`

blur

click

compositionend

compositionstart

compositionupdate

contextmenu

copy

cut

dblclick

 DOMActivate

 DOMMouseScroll

error

focus

focusin

focusout

fullscreenchange

fullscreenerror

 gesturechange

 gestureend

 gesturestart

keydown

keypress

keyup

mousedown

mouseenter

mouseleave

mousemove

mouseout

mouseover

mouseup

 mousewheel

 MozMousePixelScroll

 msContentZoom

 MSGestureChange

- ⚠️ `MSGestureEnd`
- ⚠️ `MSGestureHold`
- ⚠️ `MSGestureStart`
- ⚠️ `MSGestureTap`
- ⚠️ `MSInertiaStart`
- ⚠️ `MSManipulationStateChanged`

`overflow`

`paste`

`scroll`

`select`

🗨️ `show`

`touchcancel`

`touchend`

`touchmove`

`touchstart`

⚠️ `underflow`

⚠️ `webkitmouseforcechanged`

`webkitmouseforcedown`

⚠️ `webkitmouseforceup`

⚠️ `webkitmouseforcewillbegin`

`wheel`

▼ Inheritance:

`Node`

`EventTarget`

▼ Related pages for DOM

`AbortController`

`AbortSignal`

`AbstractRange`

`Attr`

`ByteString`

`CDATASection`

CSSPrimitiveValue

CSSValue

CSSValueList

CharacterData

ChildNode

Comment

CustomEvent

DOMConfiguration

DOMError

DOMErrorHandler

DOMException

DOMImplementation

DOMImplementationList

DOMImplementationRegistry

DOMImplementationSource

DOMLocator

DOMObject

DOMParser

DOMPoint

DOMPointInit

DOMPointReadOnly

DOMRect

DOMString

DOMTimeStamp

DOMTokenList

DOMUserData

Document

DocumentFragment

DocumentType

ElementTraversal

Entity

EntityReference

Event

EventTarget

HTMLCollection

MutationObserver

Node

NodeFilter

NodeIterator

NodeList

NonDocumentTypeChildNode

ProcessingInstruction

PromiseResolver

Range

StaticRange

Text

TextDecoder

TextEncoder

TimeRanges

TreeWalker

TypeInfo

USVString

UserDataHandler

XMLDocument