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

```
<button>Hello World</button>
```

JavaScript

```
var b = document.querySelector("button");

b.setAttribute("name", "helloButton");
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)
`setAttribute` (DOM 1)	`setAttributeNS`	`setAttributeNode`	`setAttributeNodeNS`
`getAttribute` (DOM 1)	`getAttributeNS`	`getAttributeNode`	`getAttributeNodeNS`
`hasAttribute` (DOM 2)	`hasAttributeNS`	-	-
`removeAttribute` (DOM 1)	`removeAttributeNS`	`removeAttributeNode`	-

Specifications

Specification	Status	Comment
DOM The definition of 'setAttribute()' in that specification.	Living Standard	

Browser compatibility

The compatibility table on this page is generated from structured data. If you'd like to contribute to the data, please check out https://github.com/mdn/browser-compat-data and send us a pull request.

Update compatibility data on GitHub

	Desktop						Mobile					
	Chrome	**Edge**	**Firefox**	**Internet Explorer**	**Opera**	**Safari**	**Android webview**	**Chrome for Android**	**Firefox for Android**	**Opera for Android**	**Safari on iOS**	**Samsung Internet**
`setAttribute`	Chrome Full support 1	Edge Full support 12	Firefox Full support 1	IE Full support 5 Notes Full support 5 Notes Notes In Internet Explorer 7 and earlier, `setAttribute` doesn't set styles and removes events when you try to set them.	Opera Full support 8	Safari Full support 3	WebView Android Full support 1	Chrome Android Full support 18	Firefox Android Full support 4	Opera Android Full support 10.1	Safari iOS Full support 1	Samsung Internet Android Full support 1.0

What happens next?

Our team will review your report. Once we verify the information you have supplied we will update this browser compatability table accordingly.

Can I keep track of my report?

You can join the GitHub repository to see updates and commits for this table data:

https://github.com/mdn/browser-compat-data

Our goal is to provide accurate, real values for all our compatibility data tables. Notifying MDN of inaccurate data or supplying new data pushes us further towards our goal of providing 100% real values to the developer community.
Thank you for helping.

Please select the browser or browsers which are affected.

Briefly outline the issue you are highlighting. Minimum 10 and maximum 1,000 characters.

Browser documentation and release notes are good supporting items to accompany your message. A demo hosted on services like Codepen or JSBin are perfect for providing real examples of your findings.

Connection error:Sorry, we can’t seem to reach the server. We are working to fix the problem. Please try again later.

Legend

Full support
    Full support
See implementation notes.
    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()`.