



History API

The DOM [Window](#) object provides access to the browser's session history (not to be confused for [WebExtensions history](#)) through the [history](#) object. It exposes useful methods and properties that let you navigate back and forth through the user's history, and manipulate the contents of the history stack.

Concepts and usage

Moving backward and forward through the user's history is done using the [back\(\)](#), [forward\(\)](#), and [go\(\)](#) methods.

Moving forward and backward

To move backward through history:

```
window.history.back()
```

This acts exactly as if the user clicked on the **Back** button in their browser toolbar.

Similarly, you can move forward (as if the user clicked the **Forward** button), like this:

```
window.history.forward()
```

Moving to a specific point in history

You can use the [go\(\)](#) method to load a specific page from session history, identified by its relative position to the current page. (The current page's relative position is 0.)

To move back one page (the equivalent of calling [back\(\)](#)):

```
window.history.go(-1)
```

To move forward a page, just like calling [forward\(\)](#):

to move forward a page, just like calling `forward()`.

```
window.history.go(1)
```

Similarly, you can move forward 2 pages by passing `2`, and so forth.

Another use for the `go()` method is to refresh the current page by either passing `0`, or by invoking it without an argument:

```
// The following statements
// both have the effect of
// refreshing the page
window.history.go(0)
window.history.go()
```

You can determine the number of pages in the history stack by looking at the value of the `length` property:

```
let numberOfEntries = window.history.length
```

Interfaces

History

Allows manipulation of the browser *session history* (that is, the pages visited in the tab or frame that the current page is loaded in).

Examples

The following example assigns a listener to the `onpopstate` property. And then illustrates some of the methods of the history object to add, replace, and move within the browser history for the current tab.

```
window.onpopstate = function(event) {
  alert(`location: ${document.location}, state: ${JSON.stringify(event.state)}`)
}

history.pushState({page: 1}, "title 1", "?page=1")
history.pushState({page: 2}, "title 2", "?page=2")
history.replaceState({page: 3}, "title 3", "?page=3")
history.back() // alerts "location: http://example.com/example.html?page=1, state: null"
history.back() // alerts "location: http://example.com/example.html, state: null"
```

Specifications

Specification	Status	Comment
HTML Living Standard The definition of 'History' in that specification.	Living Standard	No change from HTML5 .
HTML5 The definition of 'History' in that specification.	Recommendation	Initial definition.

Browser compatibility

[Report problems with this compatibility data on GitHub](#)

History	
Chrome	1
Edge	12
Firefox	1
Internet Explorer	10
Opera	3
Safari	1
WebView Android	1
Chrome Android	18
Firefox for Android	4
Opera Android	10.1
Safari on iOS	1
Samsung Internet	1.0