

# HTML Web Storage API

Web hosting is more secure, and large amounts of data can be stored locally without impacting website performance.

HTML web storage is better than cookies .

## What is HTML Web Storage?

With Web Storage, web applications can store data locally within the user's browser.

Before HTML5 , application data had to be stored in cookies, which were included in every server request. Web hosting is more secure, and large amounts of data can be stored locally without impacting website performance.

Unlike cookies, the storage limit is much larger (at least 5MB) and the information is never transferred to the server.

Web hosting is origin-based (by domain and protocol). All pages from a single source can store and access the same data.

## Browser support

The numbers in the table indicate the first browser version to fully support Web Storage.

API	Google Chrome	MS Edge	Firefox	Safari	Opera
Web Storage	4.0	8.0	3.5	4.0	11.5

## HTML Web Storage Objects

HTML Web Storage provides two objects for storing data on the client side:

1. `window.localStorage`- storing data without an expiration date
2. `window.sessionStorage`- Stores data for a session (data is lost when closing the browser tab)

Before using web hosting, check your browser's support for `localStorage` and `sessionStorage`:

```
if (typeof(Storage) !== "undefined") { // Code for localStorage/sessionStorage.
```

# localStorage object

This object `localStorage` stores data without an expiration date. The data will not be deleted when the browser is closed and will be available the following day, week, or year.

For example:

```
// Store localStorage.setItem("lastname", "Smith"); // Retrieve document.getElementById("result").value = localStorage.getItem("lastname");
```

The example is explained as follows:

1. Create a name/value pair for `localStorage` with **name="lastname"** and **value="Smith"**
2. Get the value of **"lastname"** and insert it into the element with **id="result"**

The example above could also be written like this:

```
// Store localStorage.lastname = "Smith"; // Retrieve document.getElementById("result").value = localStorage.lastname;
```

The syntax for deleting the `localStorage` entry **"lastname"** is as follows:

```
localStorage.removeItem("lastname");
```

**Note** : Name/value pairs are always stored as strings. Remember to convert them to other formats when necessary!

The following example counts the number of times a user has clicked a button. In this code, the string value is converted into a number so that the counter can be incremented:

```
if (localStorage.clickcount) { localStorage.clickcount = Number(localStorage.clickcount) + 1; } else { localStorage.clickcount = 0; }
```

# sessionStorage object

This object `sessionStorage` is equivalent to the `localStorage` object, except that it stores data only for a single session. The data is deleted when the user closes a particular browser tab.

The following example counts the number of times a user has clicked a button in the current session:

```
if (sessionStorage.clickcount) { sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1; } else { sessionStorage.clickcount = 0; }
```

You finished reading the article "**HTML Web Storage API**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.