Window localStorage property

The localStorage read-only property of the window interface allows you to access a Storage object for the Document's origin; the stored data is saved across browser sessions.

localStorage is similar to sessionStorage, except that while localStorage data has no expiration time, sessionStorage data gets cleared when the page session ends — that is, when the page is closed. (localStorage data for a document loaded in a "private browsing" or "incognito" session is cleared when the last "private" tab is closed.)

**Value**

A Storage object which can be used to access the current origin's local storage space.

Exceptions

SecurityError

Thrown in one of the following cases:

* The origin is not a valid scheme/host/port tuple. This can happen if the origin uses the file: or data: schemes, for example.
* The request violates a policy decision. For example, the user has configured the browsers to prevent the page from persisting data.

Note that if the user blocks cookies, browsers will probably interpret this as an instruction to prevent the page from persisting data.

**Description**

The keys and the values stored with localStorage are always in the UTF-16 string format, which uses two bytes per character. As with objects, integer keys are automatically converted to strings.

localStorage data is specific to the protocol of the document. In particular, for a site loaded over HTTP (e.g., http://example.com), localStorage returns a different object than localStorage for the corresponding site loaded over HTTPS (e.g., https://example.com).

For documents loaded from file: URLs (that is, files opened in the browser directly from the user's local filesystem, rather than being served from a web server) the requirements for localStorage behavior are undefined and may vary among different browsers.

In all current browsers, localStorage seems to return a different object for each file: URL. In other words, each file: URL seems to have its own unique local-storage area. But there are no guarantees about that behavior, so you shouldn't rely on it because, as mentioned above, the requirements for file: URLs remain undefined. So it's possible that browsers may change their file: URL handling for localStorage at any time. In fact some browsers have changed their handling for it over time.

**Examples**

The following snippet accesses the current domain's local Storage object and adds a data item to it using Storage.setItem().

JS

localStorage.setItem("myCat", "Tom");

The syntax for reading the localStorage item is as follows:

JS

const cat = localStorage.getItem("myCat");

The syntax for removing the localStorage item is as follows:

JS

localStorage.removeItem("myCat");

The syntax for removing all the localStorage items is as follows:

JS

localStorage.clear();

examples



