Window sessionStorage property

The read-only sessionStorage property accesses a session Storage object for the current origin. sessionStorage is similar to localStorage; the difference is that while data in localStorage doesn't expire, data in sessionStorage is cleared when the page session ends.

* Whenever a document is loaded in a particular tab in the browser, a unique page session gets created and assigned to that particular tab. That page session is valid only for that particular tab.
* A page session lasts as long as the tab or the browser is open, and survives over page reloads and restores.
* Opening a page in a new tab or window creates a new session with the value of the top-level browsing context, which differs from how session cookies work.
* Opening multiple tabs/windows with the same URL creates sessionStorage for each tab/window.
* Duplicating a tab copies the tab's sessionStorage into the new tab.
* Closing a tab/window ends the session and clears objects in sessionStorage.

Data stored in sessionStorage is specific to the protocol of the page. In particular, data stored by a script on a site accessed with HTTP (e.g., http://example.com) is put in a different sessionStorage object from the same site accessed with HTTPS (e.g., https://example.com).

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

**Value**

A Storage object which can be used to access the current origin's session 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.

**Examples**

Basic usage

JS

// Save data to sessionStorage

sessionStorage.setItem("key", "value");

// Get saved data from sessionStorage

let data = sessionStorage.getItem("key");

// Remove saved data from sessionStorage

sessionStorage.removeItem("key");

// Remove all saved data from sessionStorage

sessionStorage.clear();

**Saving text between refreshes**

The following example autosaves the contents of a text field, and if the browser is refreshed, restores the text field content so that no writing is lost.

JS

// Get the text field that we're going to track

let field = document.getElementById("field");

// See if we have an autosave value

// (this will only happen if the page is accidentally refreshed)

if (sessionStorage.getItem("autosave")) {

// Restore the contents of the text field

field.value = sessionStorage.getItem("autosave");

}

// Listen for changes in the text field

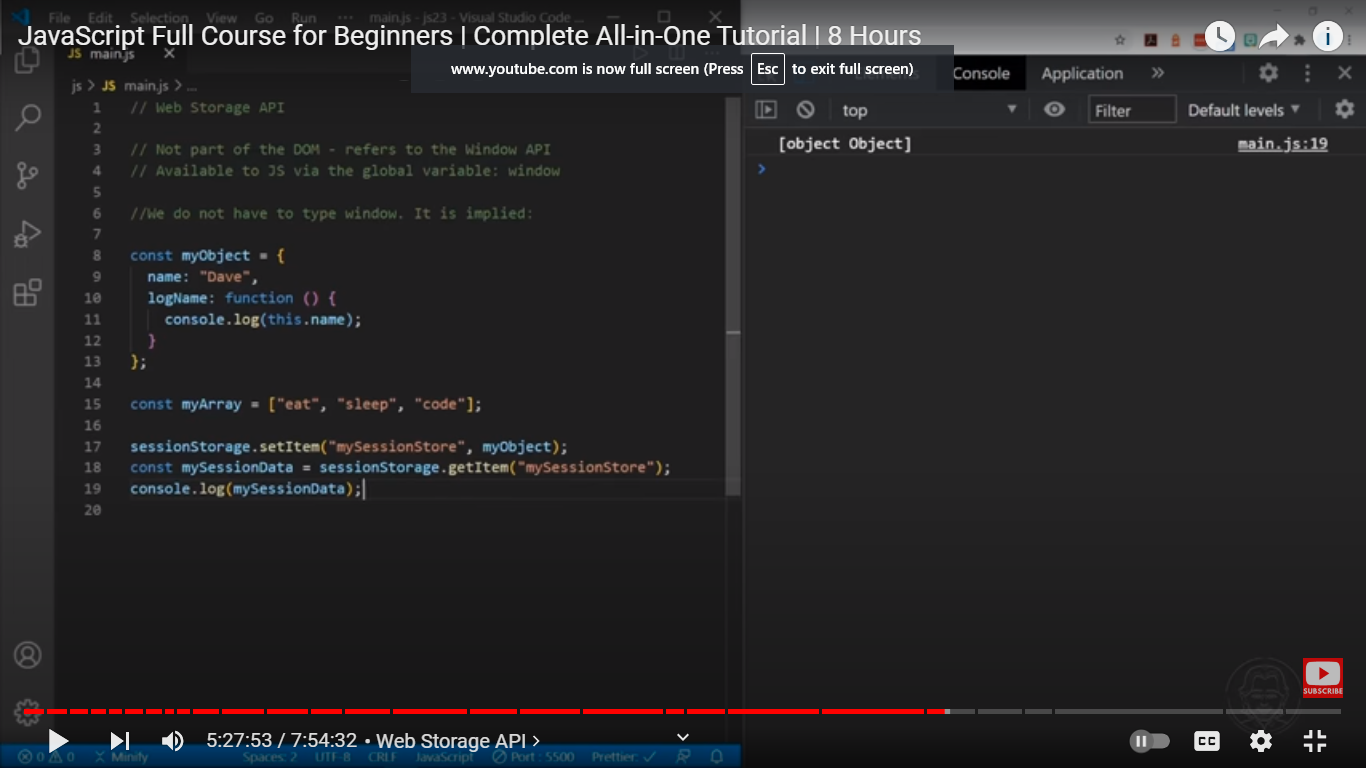
field.addEventListener("change", () => {

// And save the results into the session storage object

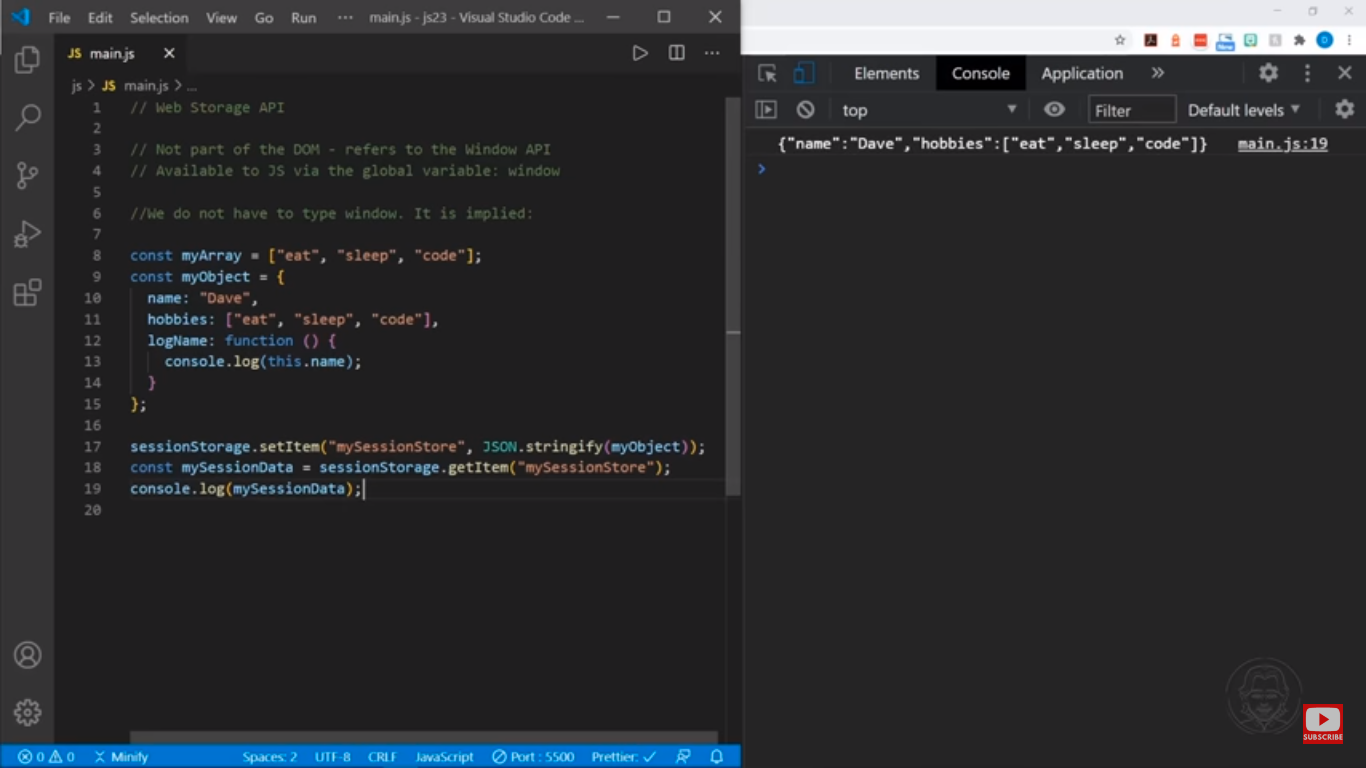
sessionStorage.setItem("autosave", field.value);

});

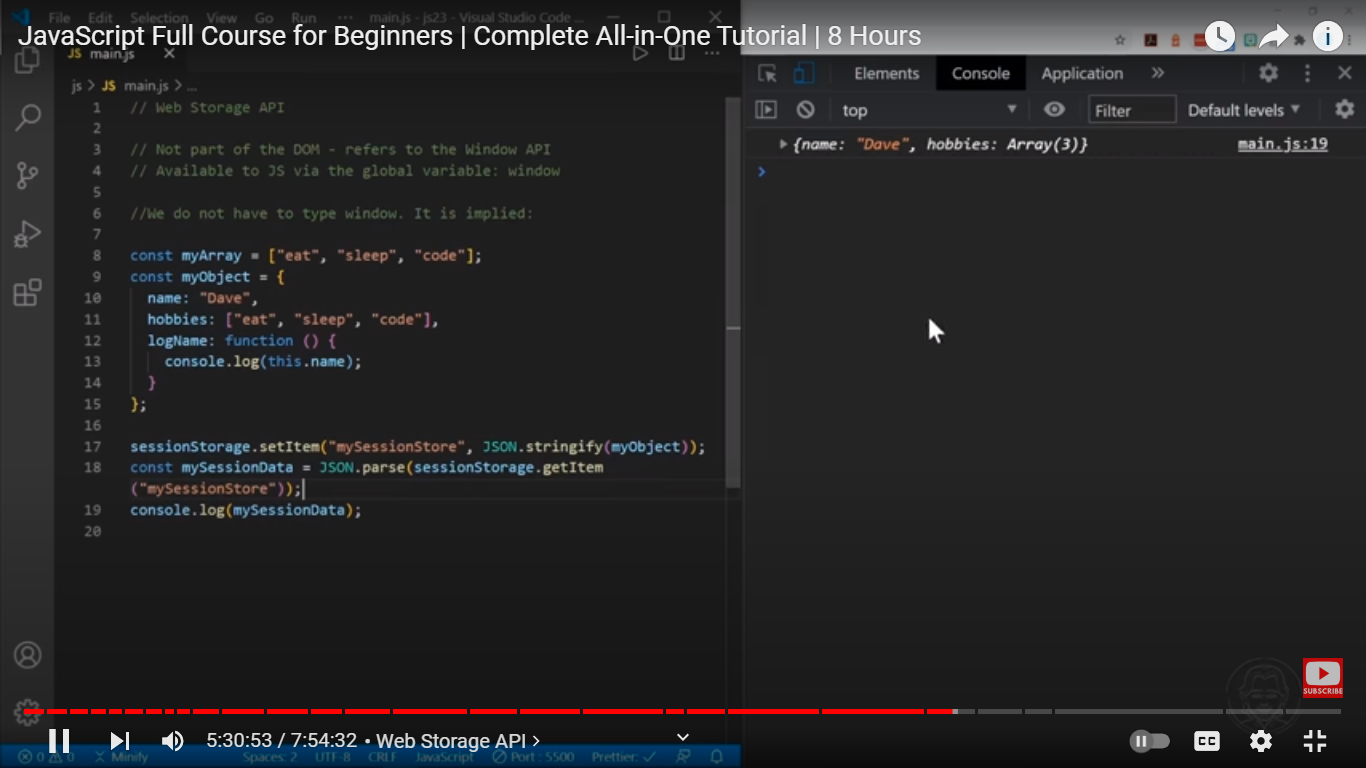
Session storage using object



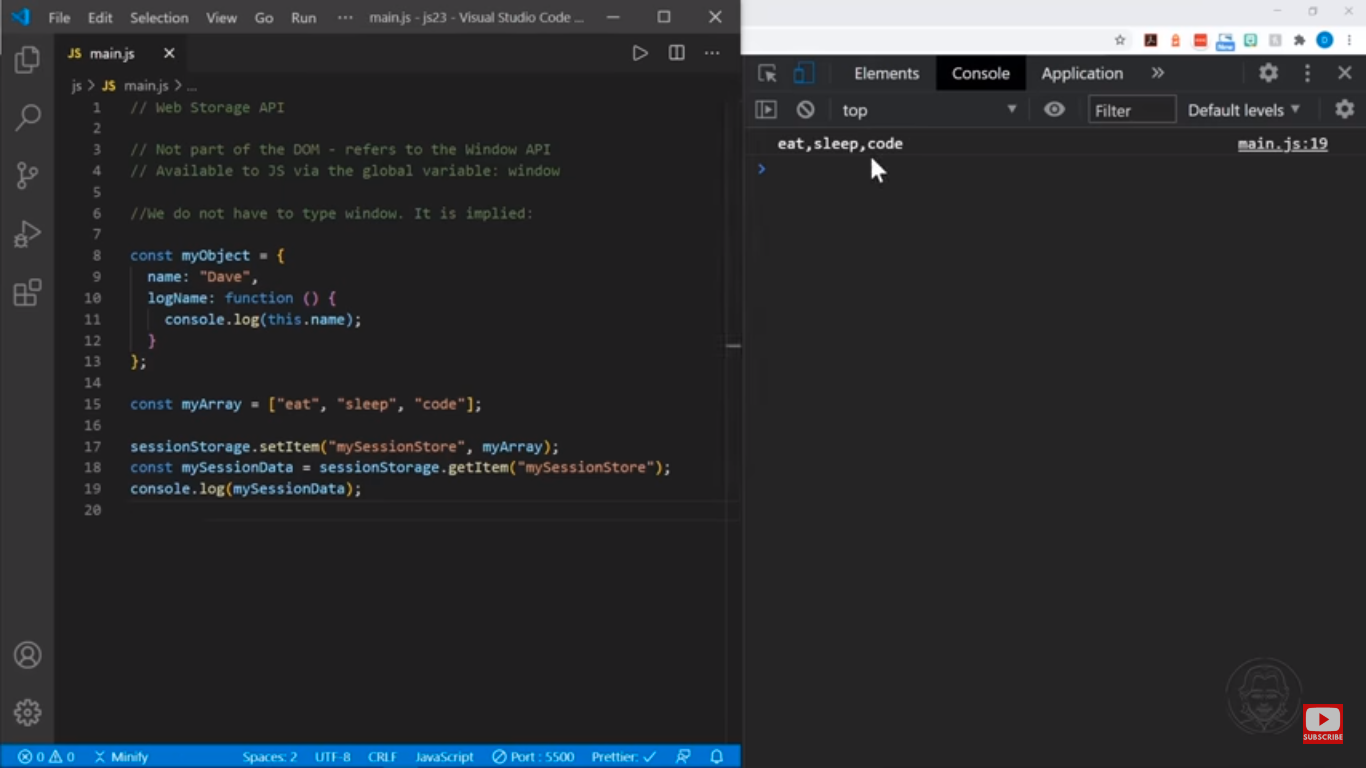
JSON.stringify(object) object need to be converted into string



JSON.parse(object): converting string JSON into JS object



Session storage using array



Session storage using json.stringify Array

