**[Introduction](https://www.digitalocean.com/community/tutorials/js-introduction-localstorage-sessionstorage" \l "introduction)**

The localStorage and sessionStorage objects, part of the web storage API, are two great tools for saving key/value pairs locally. Using localStorage and sessionStorage for storage is an alternative to using cookies and there are some advantages:

* The data is saved locally only and can’t be read by the server, which eliminates the security issue that cookies present.
* It allows for much more data to be saved (10mb for most browsers).
* The syntax is straightforward.

It’s also supported in all modern browsers, so you can use it today without an issue. Cookies are still useful, especially when it comes to authentication, but there are times when using localStorage or sessionStorage may be a better alternative.

[**Prerequisites**](https://www.digitalocean.com/community/tutorials/js-introduction-localstorage-sessionstorage#prerequisites)

To complete this tutorial, you will need the following:

* The latest version of Node installed on your machine. To install Node, follow the steps outlined in this [How To Install Node.js](https://www.digitalocean.com/community/tutorial_collections/how-to-install-node-js) tutorial.
* A basic understanding of coding in JavaScript, which you can find in this [How to Code in JavaScript](https://www.digitalocean.com/community/tutorial_series/how-to-code-in-javascript) series.

[**Step 1 — Understanding localStorage vs sessionStorage**](https://www.digitalocean.com/community/tutorials/js-introduction-localstorage-sessionstorage#step-1-understanding-localstorage-vs-sessionstorage)

localStorage and sessionStorage are almost identical and have the same API. The difference is that with sessionStorage, the data is persisted only until the window or tab is closed. With localStorage, the data is persisted until the user manually clears the browser cache or until your web app clears the data. This tutorial features localStorage, but the syntax for sessionStorage is the same.

With this knowledge, you can now create, read, and update key/value pairs in localStorage.

[**Step 2 — Creating, Reading, and Updating Entries**](https://www.digitalocean.com/community/tutorials/js-introduction-localstorage-sessionstorage#step-2-creating-reading-and-updating-entries)

You can create entries for the localStorage object by using the setItem() method. The setItem() method takes two arguments, the key and corresponding value:

let key = 'Item 1';

localStorage.setItem(key, 'Value');

Copy

To read entries, use the getItem() method. The getItem() method takes one argument which must be the key. This function will return the corresponding value as a string:

let myItem = localStorage.getItem(key);

Copy

This code sets myItem equal to 'Value', which is the corresponding value for key.

Updating an entry is done with the setItem() method. Again, it takes two arguments. The key argument will be an existing key while the value argument will be a new value:

localStorage.setItem(key, 'New Value');

Copy

Now, the localStorage value for key is 'New Value' instead of 'Value'.

You can create, read, and update entries in the localStorage object. You can also delete individual entries and clear all entries in localStorage.

[**Step 3 — Deleting and Clearing Entries**](https://www.digitalocean.com/community/tutorials/js-introduction-localstorage-sessionstorage#step-3-deleting-and-clearing-entries)

You can delete an entry with the removeItem() method. The removeItem() method takes one argument which will be a key of the localStorage object:

localStorage.removeItem(key);

Copy

You can also clear all items in localStorage. This can be done with the clear() method:

Here’s how to clear everything that’s stored in localStorage:

localStorage.clear();

Copy

These methods give you more the ability to remove and clear items from localStorage quickly. There are some limits to localStorage, though. Both localStorage and sessionStorage can only store strings. To work around this, you will have to use JSON methods.

[**Step 4 — Storing Non-String values with JSON**](https://www.digitalocean.com/community/tutorials/js-introduction-localstorage-sessionstorage#step-4-storing-non-string-values-with-json)

localStorage can only store string values. If you want to store objects or arrays as values in localStorage, you can use JSON.stringify() to convert them into strings. When creating or updating key/value pairs in localStorage, use JSON.stringify() with the object or array as the argument:

let myObj = { name: 'Skip', breed: 'Labrador' };

localStorage.setItem(key, JSON.stringify(myObj));

Copy

Although myObj is an object, JSON.stringify(myObj) converts it into a string. So myObj is now a valid localStorage value.

To read and return stringified values, use the JSON.parse() method. The JSON.parse() method takes JSON strings and converts them into JavaScript objects. JSON.parse() takes .getItem() as it’s argument:

let item = JSON.parse(localStorage.getItem(key));

Copy

Now item is set equal to the value of key. In the previous code snippet, the value of key was set to the stringified version of myObj. Using JSON.parse converts myObj back into an object. So item is set equal to myObj as an object, not a string.

Being able to convert arrays and objects into strings using JSON.stringify and convert them back using JSON.parse is very useful. You will also need to know how to check if localStorage is empty or not.

[**Step 5 — Checking for Items**](https://www.digitalocean.com/community/tutorials/js-introduction-localstorage-sessionstorage#step-5-checking-for-items)

In this step, you will test for the presence of items in the localStorage. You can use if statements to check if localStorage is holding items or if it is empty. To do this, check the length of localStorage:

if (localStorage.length > 0) {

// ...

}

Copy

If localStorage.length is greater than 0, then there are items within the localStorage object. Include an else statement in case there are no items in localStorage:

if (localStorage.length > 0) {

// ...

} else {

// ...

}

Copy

You can include code to be applied in the if and else statements. You may want to iterate over items in localStorage.

[**Step 6 — Iterating Over Items**](https://www.digitalocean.com/community/tutorials/js-introduction-localstorage-sessionstorage#step-6-iterating-over-items)

The localStorage and sessionStorage objects don’t support the forEach method. To iterate over the items in localStorage, use a for loop:

for (let i = 0; i < localStorage.length; i++){

}

Copy

The key() method takes an integer as an argument and returns the corresponding key. With a for loop, pass i in as the integer for key():

for (let i = 0; i < localStorage.length; i++){

let key = localStorage.key(i);

}

Copy

Use the getItem method to return the corresponding value for key:

for (let i = 0; i < localStorage.length; i++){

let key = localStorage.key(i);

let value = localStorage.getItem(key);

}

Copy

Create a console.log statement to print both key and value to the screen:

for (let i = 0; i < localStorage.length; i++){

let key = localStorage.key(i);

let value = localStorage.getItem(key);

console.log(key, value);

}

Copy

The key() is very useful for iterating through localStorage using for loops. Not all browsers support localStorage.

[**Step 7 — Checking for Support**](https://www.digitalocean.com/community/tutorials/js-introduction-localstorage-sessionstorage#step-7-checking-for-support)

You can test for localStorage support by checking if it’s available on the window object using an if statement:

if (window.localStorage) {

// localStorage supported

}

Copy

You can also use the [Can I use…](https://caniuse.com/namevalue-storage) website to check for localStorage support across major browsers.

[**Conclusion**](https://www.digitalocean.com/community/tutorials/js-introduction-localstorage-sessionstorage#conclusion)

You can use the localStorage or sessionStorage objects to store key/value pairs. There are methods that enable you to interact with the items within localStorage. With this tutorial, you created, removed, and updated entries within localStorage. You also converted array and object data into strings and back using JSON methods.

At times, it may be best to use cookies instead of localStorage or session. [What Are Cookies & How to Work With Them Using JavaScript](https://www.digitalocean.com/community/tutorials/js-what-are-cookies) thoroughly covers this topic and is a great next step.