



Local Storage: A Complete Guide.

[Don't miss these.]

JS



@gulraizcodes



What the heck is LocalStorage?

localStorage is a property that allows JavaScript sites and apps to save key-value pairs in a web browser with no expiration date.

This means the data stored in the browser will persist even after the browser window is closed.

Web Storage API

The Web Storage API consists of two mechanisms:

- ★ sessionStorage
- ★ localStorage

sessionStorage VS localStorage

sessionStorage only maintains a storage area while the browser is open while on the other hand localStorage continues to store data after the browser is closed.

In a nutshell

sessionStorage expires, localStorage does not expire.



@gulraizcodes



Where is localStorage stored?

In Chrome, **web storage data** is saved in an **SQLite file** in a subfolder in the **user's profile**.

```
\AppData\Local\Google\Chrome\User Data\Default\Local Storage
```

How does localStorage work?

In order to use **localStorage**, there are **5 methods**.



`setItem()`

Add key & value to localStorage



`getItem()`

This is how you **get items** from localStorage.



`removeItem()`

Remove an item by key from localStorage.



`clear()`

Clear all localStorage.



`key()`

Passed a number to retrieve the key of a localStorage



@gulraizcodes



setItem():

As name suggests, this method allows you to store values in the localStorage object.

It takes two parameters.

★ a key

★ a value

The key can be referenced later to fetch the value attached to it.

```
window.localStorage.setItem('name', 'Gulraiz');
```

key

value

To store arrays or objects, you would have to convert them to strings.

To do this, we use the `JSON.stringify()` method before passing to `setItem()`.

(check next slide)

Note: localStorage can only store strings.



@gulraizcodes



```
const person = {  
  name: "gulraiz",  
  location: "Lagos",  
}  
  
window.localStorage.setItem('user',  
  JSON.stringify(person));
```

getItem():

In order to get **localStorage**, use the **getItem()** method. **getItem()** allows you to access the **data stored in browser's localStorage**.

getItem() accepts only one parameter, which is the **key**, and returns the **value as a string**.

To retrieve a user key:

```
window.localStorage.getItem('user');
```

This returns a string with value as:

```
"{"name":"Obaseki Nosa","location":"Lagos"}"
```



@gulraizcodes



To use this value, you would have to convert it back to an object.

To do this, we make use of `JSON.parse()` method, which converts a JSON string into a JS object.

```
JSON.parse(window.localStorage.getItem('user'));
```

`removeItem():`

To delete local storage sessions, use the `removeItem()` method.

When passed a key name, the `removeItem()` method removes that key from the storage. If there's no item associated w/ the given key, this method will do nothing.

```
window.localStorage.removeItem('name');
```



@gulraizcodes



clear():

Use the `clear()` method to delete all items in `localStorage`.

This method, when invoked, clears the entire storage of all records for that domain.

```
window.localStorage.clear();
```

key():

This method comes in handy in situations where you need to loop through keys and allows you to pass a number/index to `localStorage` to retrieve the name of the key.

```
var KeyName = window.localStorage.key(index);
```



@gulraizcodes



Limitations

Don't share sensitive user information in localStorage.

It is limited to 5MB across all the major browsers.

No data protection & can be accessed by any code on your web page.

It is synchronous, meaning each operation called would only execute one after other.



@gulraizcodes

