

JS

Most Asked Question ?

Web Storage API & Mechanisms



@code.singh



@sahil singh


 JS

Web Storage API

- The Web Storage API is a set of mechanisms that enable browsers to store key-value pairs. It is designed to be much more intuitive than using cookies.
- The key-value pairs represent storage objects, which are similar to objects except they remain intact during page loads, and are always strings.
- You can access these values like an object or using the `getItem()` method .

Two mechanisms :

- `sessionStorage`
 - `localStorage`
- 

JS sessionStorage

- **sessionStorage** maintains a separate storage area for each given origin that's available for the duration of the page session (as long as the browser is open, including page reloads and restores)
- Stores data only for a session, meaning that the data is stored until the browser (or tab) is closed.
- Data is never transferred to the server.
- Storage limit is larger than a cookie



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.
- Storage limit is the maximum amongst the two.

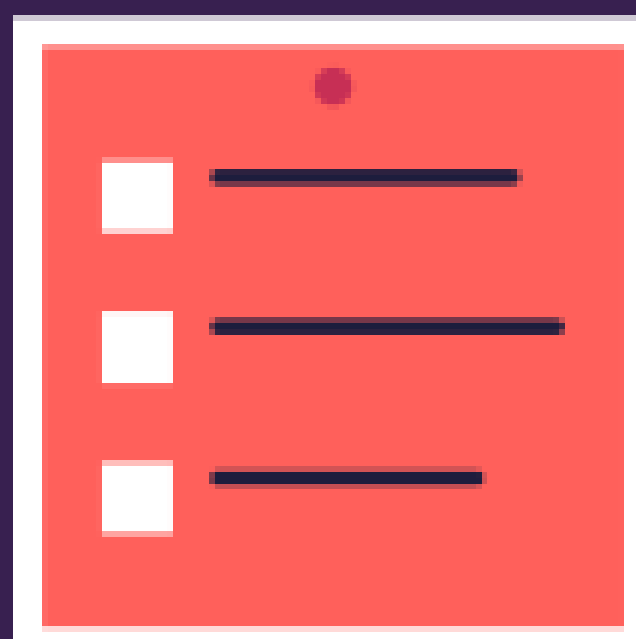
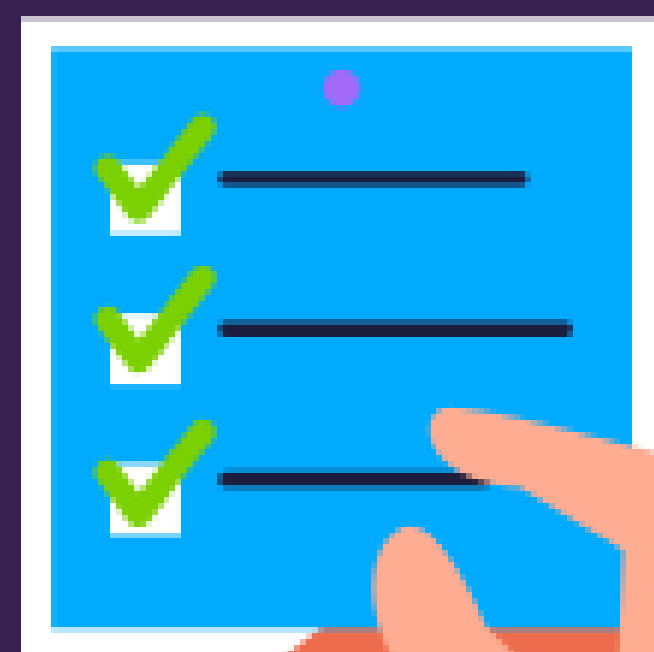
Syntax:

`window.localStorage`



JS localStorage Methods

- **setItem()**: Add key and 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



@code.singh



@sahil singh

`setItem()` ✨

- This method allows you to store values in the `localStorage` object.
- It takes two parameters: a key and a value. The key can be referenced later to fetch the value attached to it.

`localStorage.setItem(key, value);`

```
const person = {name: "sahil",  
location: "singh",}
```

```
window.localStorage.setItem('user',  
JSON.stringify(person));
```


JS getItem() ✨

- The **getItem()** method. getItem() allows you to access the data stored in the browser's localStorage object.
- getItem() accepts only one parameter, which is the key, and returns the value as a string.

```
let lastname = localStorage.getItem(key);
```

```
"{"name":"sahil","location":"singh"}"
```

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

JS `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 it exists. If there is no item associated with the given key, this method will do nothing.

`localStorage.removeItem(key);`



JS 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. It does not receive any parameters

localStorage.**clear()**;



JS key()

- The **key()** method comes in handy in situations where you need to loop through keys and allows you to pass a number or index to localStorage to retrieve the name of the key.

```
window.localStorage.key(index);
```



✦

We *daily* add this kind of posts
so, if you wanna to see more *content*
relate to *code*, then follow
@code.singh for more development
code ✦



...

SAHIL SINGH

 @code.singh

 @sahil singh