

# Local/Session Storage

WebAPI | In Detail

A yellow square containing the letters 'JS' in a bold, black, sans-serif font, representing the JavaScript logo.

JS

Mallikarjun | @CodeBustler



# What is **localStorage** ?

localStorage is a **Web API** provided by modern browsers that allows web applications to **store key-value pairs locally on the user's computer.**

And you can **retrieve them** even after the browser has been **closed and reopened.**

---

There are few **methods & properties** are available to store and retrieve the data from local storage

# localStorage **Methods**

- **setItem(key, value)**
  - stores a key-value pairs
- **getItem(key)**
  - retrieves the value associated with a key in localStorage.
- **removeItem(key)**
  - removes the key-value pair associated with a key
- **clear()**
  - removes all key-value pairs from localStorage.

# localStorage **Properties**

- **length**

- returns the number of key-value pairs stored in localStorage.

- **key(n)**

- returns the name of the key at the **given index** in localStorage.
- **Example** : localStorage.**key(0)** returns the name of the **first key**, localStorage.**key(1)** returns the name of the **second key**, and so on.

# What is **sessionStorage**?

sessionStorage is a **web API** allows web applications to **store key-value** pairs in a **similar way to localStorage**.

The **difference is** that sessionStorage stores the data **only for the duration of a single browsing session**, while localStorage stores data indefinitely or until it is manually cleared.

---

This Web API makes it **ideal for storing temporary data** that needs to **persist** within the **current browsing session**.

# Difference Local/Session

Feature	localStorage	sessionStorage
Data Persistence	Data persists until explicitly cleared by the user	Data persists only for the current browsing session
Shared Across Tabs	Data is shared across all tabs and windows for the origin	Data is limited to the current tab or window
Storage Capacity	Can store larger amounts of data (up to 5-10 MB)	Can store smaller amounts of data (up to 5-10 MB)
Accessibility	Can be accessed by any script running on the same origin	Only accessible within the current tab or window
Use Cases	Good for storing persistent data, such as user settings	Good for storing temporary data, such as user state for an app
API Methods	<code>`setItem()`</code> , <code>`getItem()`</code> , <code>`removeItem()`</code> , <code>`clear()`</code>	<code>`setItem()`</code> , <code>`getItem()`</code> , <code>`removeItem()`</code> , <code>`clear()`</code>







**@CodeBustler**

Mallikarjun | Web Developer

**Follow For More !**

And i need dopamine ⚡

