



# Main three mechanism for storing Data in JavaScript

Cookies

Local  
Storage

Session  
Storage

VS



VS



SWIPE >>



# Local Storage:

- **Purpose:** Local storage is designed for persistently storing larger amounts of data in the web browser. This data persists even after the browser is closed and reopened.
- **Capacity:** Typically, local storage allows storing more data than cookies.
- **Scope:** Data stored in local storage is scoped to the origin (the combination of protocol, domain, and port). This means that another origin cannot access data stored in local storage for one origin.
- **API:** Accessible via the localStorage object in JavaScript.

SWIPE >>



# Local Storage Example:

```
// Storing data in local storage
localStorage.setItem('username', 'John');

// Retrieving data from local storage
const username = localStorage.getItem('username');
```

SWIPE >>



# Cookies

- **Purpose:** Cookies are small pieces of data sent from a website and stored on the user's device by the web browser. They are primarily used for maintaining session state, user authentication, and tracking user activity.
- **Capacity:** Cookies have size limitations (usually a few kilobytes).
- **Scope:** Cookies are scoped to the domain and path of the website that sets them. Some cookies can also be set to be accessible across subdomains.
- **API:** Cookies can be manipulated using JavaScript or server-side code. JavaScript libraries like js-cookie provide convenient methods for cookie management.

SWIPE >>



# Example using js-cookie library:

```
import Cookies from 'js-cookie';  
  
// Setting a cookie  
Cookies.set('username', 'John');  
  
// Retrieving a cookie  
const username = Cookies.get('username');
```

SWIPE >>



# Session Storage

- **Purpose:** Session storage is similar to local storage but scoped to the current browser tab. Data stored in session storage persists only for the duration of the page session. Once the tab is closed, the data is cleared.
- **Capacity:** Similar to local storage, session storage also allows storing larger amounts of data compared to cookies.
- **Scope:** Data stored in session storage is scoped to the origin and persists only for the duration of the page session.
- **API:** Accessible via the sessionStorage object in JavaScript.

SWIPE >>

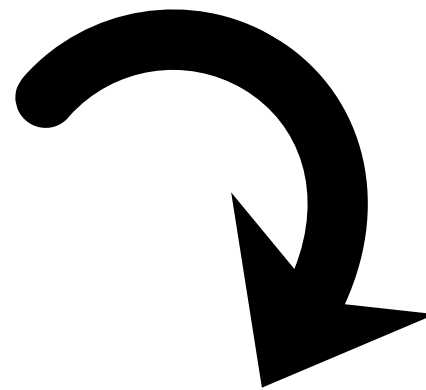


# Session Storage Example

```
// Storing data in session storage  
sessionStorage.setItem('username', 'John');  
  
// Retrieving data from session storage  
const username = sessionStorage.getItem('username');
```

SWIPE >>

**For More  
Knowledge, Follow  
Us on Instagram**



**@codehab3**

Code In Web3