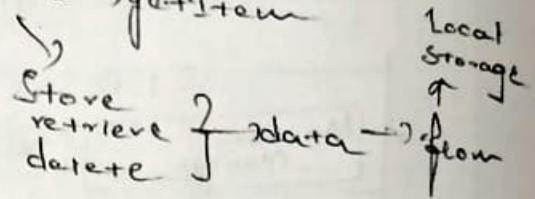


iii) Local Storage

- It is way -> web developers -> Store data on user browser
 - using key value pairs
- Data persists even after -> user -> turn off computer
 - close browser
- Perfect for
 - User Preferences
 - Form Input
 - Session Data
- All done on User's Machine no need -> Server side Database

How to use?

- Use LocalStorage object.
 - methods -> removeItem, setItem, getItem
- Limit is 5-10MB
 - Depend on Browser
- setItem
 - Set Item in local Storage



<script>

```
let p1 = document.getElementById("p1");
let p2 = document.getElementById("p2");
```

```
// Set an item in local storage
localStorage.setItem('username', 'Manu');
```

```
// Retrieve item from local storage
```

```
const StoreItem = localStorage.getItem('username');
p1.innerHTML = StoreItem;
console.log(StoreItem); // Manu
```

```
// Remove an item from local storage
```

```
localStorage.removeItem('username');
```

```
const removedItem = localStorage.getItem('username');
if (removedItem === null) {
  p2.innerHTML = 'The Item has been removed';
  console.log(removedItem);
}
```

Console

Manu
null

O/P

Manu
The Item
has been
removed

1) Session Storage

It stores data only for a session. (Data stored

in Session Storage

deleted

when browser closed

When open page in New Tab or window } Browser create New Session

It open multiple Tabs or windows with same URL } Browser create

Tab or window cache for Session Storage ← Separate

Wish store object

→ List

→ Arrays

convert

String

using

`JSON.stringify()`

Access Session Storage via property of window object.

`window.sessionStorage;`

`Storage {length: 0}`

`window.localStorage;`

`Storage {clickcount: 4, length: 1}`

Provides properties and methods

`setItem(key, value)`
`getItem(key)`
`removeItem(key)`
`keyIndex()`

`clear()` → clear all values
`length()`

get name of value pairs from session storage

from specific location value → Retrieve

LocalStorage Vs SessionStorage

localStorage:

- Data stored in localStorage **persists** even **after** the **browser is closed** and reopened.
- The data is **available** **across** **all** **tabs** and **windows** of the **same** **origin** (i.e., protocol, domain, and port).
- The data can be **accessed** by any JavaScript **code** on the **same** **domain**.

sessionStorage:

- Data is only **persists** for the **duration** of the **user's session** on a specific tab.
- The data is **only** **available** to the **page** that **created** it and is **not** **shared** across tabs or windows of the **same** **origin**.
- The data can be **accessed** by any JavaScript **code** running on the **same** **page** that created it.