

Local & Session storage :

20

- > Web applications can store data locally within the user's browser with web storage. Before HTML5, application data stored in cookies, included in every server request. But with the advent of HTML5, we have got various options to store information on the client browser. Previously we were having only cookies, which were very restrictive, and the size of the cookies was very small. But now the web storage is more secure, and large amounts of data can be stored locally, without affecting website performance. We have local storage and session storage. We can access the web storage by opening inspect element on that web page and then selecting option applications. Here, we can see the option of local and session storage.

HTML Web Storage Objects :

-> HTML web storage provides us with two objects for storing data:

- > WINDOW.LOCALSTORAGE -> It stores the data with no expiration date.

Date _____

→ `WINDOW.SESSIONSTORAGE` → It stores the data for one session. That means the data is lost when the browser tab is closed.

→ Local storage:

→ The way to store data on the client's computer is by local storage. The local storage allows us to save the key/value pairs in a web browser, and it stores data with no expiration date. We can access local storage via JavaScript and HTML5. However, the user also can clear the browser data to erase all local storage data.

→ Session storage:

→ The session storage is used to store data only for a session, meaning that it is stored until the browser (or tab) is closed. Remember that, in session storage, the data is never transferred to the server and can only be read on the client-side. The storage limit is b/w 5-10 MB. By opening multiple windows or tabs with the same URL, creates session storage for each tab or window.

→ Both storage objects provide the same methods and properties like `GETITEM`, `GETITEM`, `REMOVEITEM`, and `CLEAR`.

The examples in this ~~lab~~ here are of LOCALSTORAGE, but the same syntax works for SESSIONSTORAGE.

→ Creating Entries: We can create the key/value pair entries with LOCALSTORAGE.SETITEM, providing a key and a value. Ex:

```
LE? KEY = 'ENTRY 1';  
LOCALSTORAGE.SETITEM('KEY', 'VALUE');
```

→ Reading Entries: We can read entries with LOCALSTORAGE.GETITEM. Ex:

```
LE? MYITEM = LOCALSTORAGE.GETITEM(KEY);
```

→ Updating Entries: We can update an entry just as we create a new one with SETITEM, but with a key that already exists. Ex:

```
LOCALSTORAGE.SETITEM(KEY, 'NEW VALUE');
```

→ Deleting Entries: We can delete an entry with the REMOVEITEM method. Ex:

```
LOCALSTORAGE.REMOVEITEM(KEY);
```

→ Clearing Everything: We can clear

everything that is stored in LOCALSTORAGE.

Ex:

```
LOCALSTORAGE.CLEAR();
```

→ storing JSON objects: Only strings can be stored with LOCALSTORAGE or SESSION STORAGE, but we can use JSON.stringify to store more complex objects and JSON.parse to read them. Ex:

// Create item:

```
LET MYOBJ = { NAME: 'HARRY', LANGUAGE: 'JAVASCRIPT' };
```

```
LOCALSTORAGE.SETITEM(KEY, JSON.stringify(MYOBJ));
```

// Read item:

```
LET ITEM = JSON.parse(LOCALSTORAGE.GETITEM(KEY));
```