
Session 19

Mohamed Emary

May 31, 2024

1 Local Storage

Local storage is a way to store data in the browser (client-side storage). It is a key-value pair storage limited storage (5MB).

When we say it's Local Storage, it means it's local to the browser. It is not stored on the server. It is stored on the client's machine, so it is not shared with other users.

To see the local storage in the browser, open the developer tools and go to the Application tab. Then, click on Local Storage.

You can *only store strings* in the local storage.

To store a value in the local storage, you can use the `setItem()` method. The `setItem()` method takes two parameters: the key and the value.

```
1 | localStorage.setItem('name', 'Mohamed');
```

Keys are unique. If you set a value to a key that already exists, it will overwrite the old value.

```
1 | localStorage.setItem('name', 'Ahmed');
```

Now the value of the key `name` is `Ahmed`.

To get a value from the local storage, you can use the `getItem()` method. The `getItem()` method takes one parameter: the key.

```
1 | var name = localStorage.getItem('name');
2 | console.log(name); // Ahmed
```

To remove a value from the local storage, you can use the `removeItem()` method. The `removeItem()` method takes one parameter: the key.

```
1 | localStorage.removeItem('name');
2 | var name = localStorage.getItem('name');
3 | console.log(name); // null
```

To know how many items are stored in the local storage, you can use the `length` property.

3 STORING OBJECTS

To clear the local storage, you can use the `clear()` method. The `clear()` method takes no parameters.

```
1 localStorage.setItem('name', 'Mohamed');
2 localStorage.setItem('age', '25');
3 console.log(localStorage.length); // 2
4 localStorage.clear();
5 var name = localStorage.getItem('name');
6 var age = localStorage.getItem('age');
7 console.log(name); // null
8 console.log(age); // null
```

To know which key at a specific index, you can use the `key()` method. The `key()` method takes one parameter: the index.

```
1 localStorage.setItem('name', 'Mohamed');
2 localStorage.setItem('age', '25');
3 console.log(localStorage.key(0)); // name
4 console.log(localStorage.key(1)); // age
```

You shouldn't store sensitive data in the local storage because it's not secure. It's accessible by anyone who has access to the client's machine.

We don't get all data from backend some data that are not sensitive like language. can be stored in the local storage.

Local storage data are not removed even if you close the browser. It will be removed when you clear the local storage or when you delete the browser's data.

2 Session Storage

Session storage is similar to local storage, but it's for the session only which means it's removed when the session is ended like when you close the tab or the browser.

We have a method called `sessionStorage` that works the same as `localStorage` with the same methods and properties like:

- `setItem()`
- `removeItem()`
- `clear()`
- `getItem()`
- `length`
- `key()`

3 Storing Objects

As we mentioned before, you can only store strings in the local storage. If you want to store an object, you need to convert it to a string using `JSON.stringify()`.

```
1 var person = {
2   name: 'Mohamed',
3   age: 25
4 };
5
6 localStorage.setItem('person', JSON.stringify(person));
```

To get the object from the local storage, you need to parse the string using `JSON.parse()`.

3 STORING OBJECTS

```
1 var person = JSON.parse(localStorage.getItem('person'));
2 console.log(person.name); // Mohamed
3 console.log(person.age);  // 25
```

The same can be done with arrays:

```
1 var people = [
2   { name: 'Mohamed', age: 25 },
3   { name: 'Ahmed', age: 30 },
4   { name: 'Ali', age: 35 }
5 ];
6
7 localStorage.setItem('people', JSON.stringify(people));
8 var people = JSON.parse(localStorage.getItem('people'));
9 console.log(people[1]); // { name: 'Ahmed', age: 30 }
```