

# LocalStorage In Javascript



# JavaScript localStorage

HTML5 specification introduces the `localStorage` as a way to store data with no expiration date in the web browsers.

- In other words, the data stored in the browsers will persist even after you close the browser windows.
- The data stored in the `localStorage` is bound to an origin. It means that the `localStorage` is unique per `protocol://host:port`.

# Accessing localStorage

You can **access** the **localStorage** via the property of the **window** object:

```
window.localStorage
```

- When you type the **upper code** in the Console.
- ... you'll see the **following object**:

```
Storage {length: 0}
```

- **getItem(key)**

- Used to get the value by a key.

```
localStorage.getItem('theme'); // 'dark'
```

- **length**

- Used to get the number of name-value pairs.

```
console.log(localStorage.length); // 1
```



# localStorage Methods

localStorage provide some methods and properties.

- **setItem(key, value)**

- Used to store a name-value pair in the localStorage.

```
localStorage.setItem('theme', 'dark');
```

- **removeItem(key)**

- Used to remove a name-value pair by a key.

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

# Loop over keys

The following stores three name-value pairs to the `localStorage`:

```
localStorage.setItem('theme', 'light');  
localStorage.setItem('backgroundColor', 'white');  
localStorage.setItem('color', '#111');
```

- To iterate over name-value pairs, you use the `Object.keys()`.

```
let keys = Object.keys(localStorage);  
for (let key of keys) {  
  console.log(`${key}: ${localStorage.getItem(key)}`);  
}
```

*// Output*

```
color: #111  
theme: light  
backgroundColor: white
```