CodeWithHarry

HTML CSS JS C C++ JAVA PYTHON PHP

Local & Session storage in JavaScript | JavaScript Tutorial In Hindi #20

Overview Q&A Downloads Announcements

Local & Session storage in JavaScript | JavaScript Tutorial In Hindi #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 right-clicking on the webpage and then selecting option applications. Here we can see the option of local and session storage. Let's talk about all these web storage in detail.

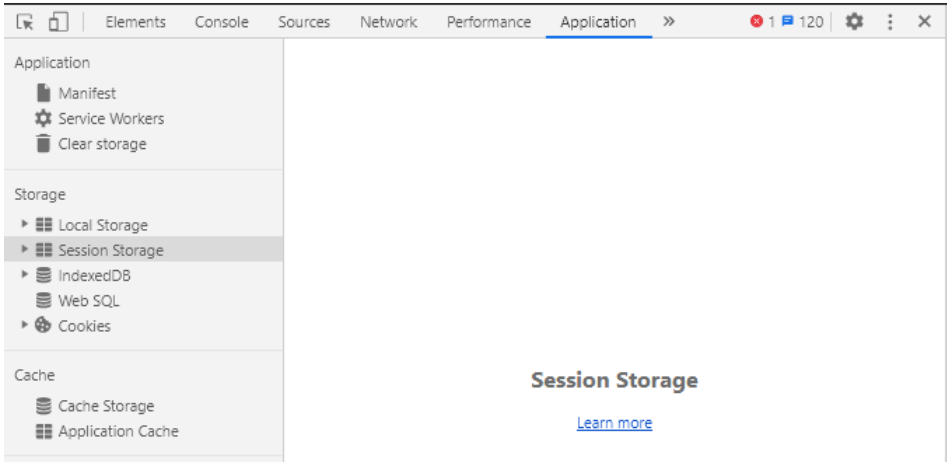


Figure1: Inspect->Application

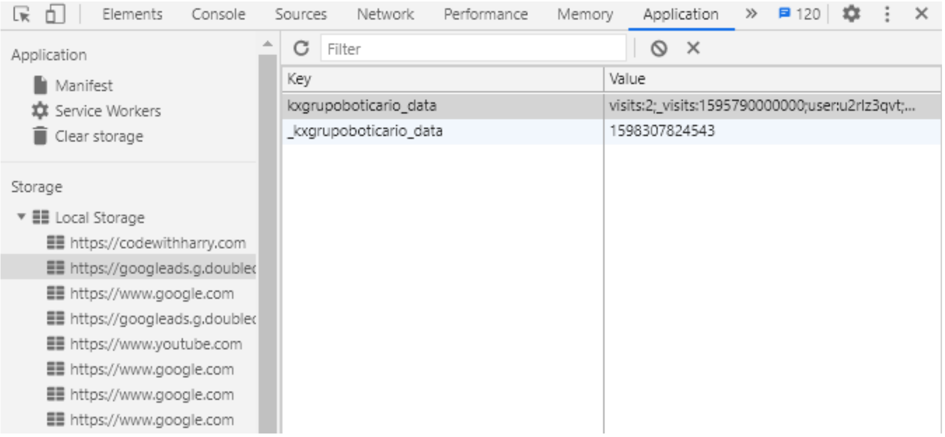
HTML Web Storage Objects:-

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

- **window.localStorage** – It stores the data with no expiration date
- **window.sessionStorage** -It stores the data for one session. That means the data is lost when the browser tab is closed.

LocalStorage:-

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 can clear the browser data to erase all localStorage data.

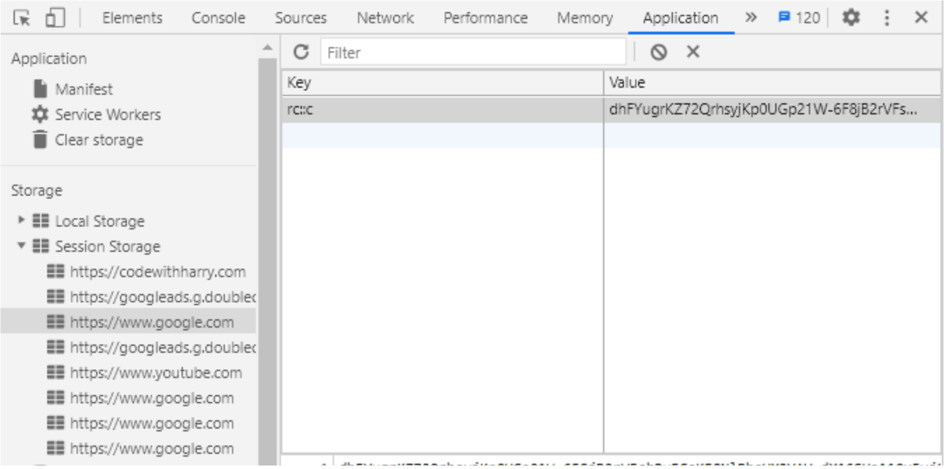


Key	Value
kxgrupoboticario_data	visits:2;_visits:1595790000000;user:u2rlz3qvt;...
_kxgrupoboticario_data	1598307824543

Figure 2: Local Storage

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 between 5-10MB. By opening multiple windows or tabs with the same URL creates sessionStorage for each tab or window.



Key	Value
rc::c	dhFYugrKZ72QrhsyjKp0UGp21W-6F8jB2rVFs...

Figure3: Session storage

Both storage objects provide the same methods and properties like setItem, getItem, removeItem, and clear. The examples in this tutorial are for 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. Here is an example:

```
let key = 'Entry_1';
localStorage.setItem(key, 'Value');
```

- **Reading Entries:-** We can read entries with localStorage.getItem. Here is an example:

```
let 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. Here is an example:

```
localStorage.setItem(key, 'New Value');
```

- **Deleting Entries:-** We can delete an entry with the removeItem method. Here is an example:

```
localStorage.removeItem(key);
```

- **Clearing Everything:-** We can clear everything that's stored in localStorage. Here is an example:

```
localStorage.clear();
```

- **Storing JSON Objects:-** Only strings can be stored with localStorage or sessionStorage, but we can use **JSON.stringify** to store more complex objects and **JSON.parse** to read them. Here is an example:

```
// Create item:
let myObj = { name: 'Harry', language: 'JavaScript' };
localStorage.setItem(key, JSON.stringify(myObj));
// Read item:
let item = JSON.parse(localStorage.getItem(key));
```

Website.html code as described/written in the video

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
</head>
<body>
    <div class="container">
        <h1 id="heading"> Welcome to Code With Harry</h1>
        <div id="myfirst" class="child red good" id="first">child 1

            <ul class="this">
                <li class="childul">this</li>
                <li class="childul">is</li>
                <li class="childul">a</li>
                <li class="childul">list </li>
                <li class="childul">of my dreams</li>
            </ul>
        </div>
        <div class="child">child 2</div>
        <div class="child red">child 3</div>
        <div class="child">child 4</div>
        <form action="none.html" method="post">
            <a href="//codewithharry.com">Go to Code With Harry</a>
            <br>
            <br>
            Search this website: <input type="text" name="Hello" id="">
            <input type="button" value="submit">
        </form>
    </div>
    <br>
    <div class="no">this is a dummy div1</div>
    <div class="no">this is a dummy div2</div>
    <div class="no">this is a dummy div3</div>
</body>
```

```
<!-- <script src="js/tut12.js"></script> -->
<!-- <script src="js/tut14.js"></script> -->
<script src="js/tut15.js"></script>
</html>
```

JavaScript code as described/written in the video

```
console.log('This is tut 20');
let impArray = ['adrak', 'pyaz', 'bhindi'];

// // Add a key-value pair inside local Storage
// localStorage.setItem('Name', 'Harry');
// localStorage.setItem('Name2', 'Rohan');
// localStorage.setItem('Sabzi', JSON.stringify(impArray));

// Clears the entire local storage
// localStorage.clear();

// Clear a particular key-value pair
// localStorage.removeItem('Name2');

// Retrieve an item from the local Storage
let name = localStorage.getItem('Name');
name = JSON.parse(localStorage.getItem('Sabzi'));
console.log(name)

// sessionStorage.setItem('sessionName', 'sHarry');
// sessionStorage.setItem('sessionName2', 'sRohan');
// sessionStorage.setItem('sessionSabzi', JSON.stringify(impArray));
```

Previous Next

CodeWithHarry