

JAVASCRIPT

Local Storage vs Cookies: What's the Difference?

In this article, you'll learn about some of the main differences between local storage and cookies and which you should choose for your websites.



SHAHED NASSER

5 MAY 2022 • 3 MIN READ



Local Storage vs Cookies: What's the Difference?

blog.shahednasser.com



Adobe Creative Cloud for Teams starting at \$33.99 per month.

ADS VIA CARBON

TABLE OF CONTENTS

What are Cookies?

What is the Local Storage?

What are the Differences Between Cookies and Local Storage

Access on the Server

Data Size

Data Expiry

Better API

Which Should You Choose?

Conclusion

Cookies in JavaScript are used to read from and store small data related to the user on the browser.

Local Storage allows you to also read from and store data related to the user on the browser using JavaScript's [Storage API](#).

In this article, you'll learn about some of the main differences between local storage and cookies and which you should choose for your websites.

What are Cookies?

Cookies are pieces of data that are communicated between the server and the browser in the header of requests. This allows the server to learn more information about the user accessing the website but also gives you access in the browser to interact with that data.

authenticate the user in later requests and retrieve data or process actions based on who the user is.

Similarly, as a developer, you'll probably need access to that token to use it for future requests to send to your server.

Cookies on the client-side (the browser) are natively accessed with `document.cookies`. When you store cookies on the browser, you can make them specific to a path in a domain name and you can add an expiry date for them.

What is the Local Storage?

Local Storage is used to read and write data in the browser's storage. The data is persistent and has no expiry date. The data is stored in key-value pairs. The server has no access to the client's local storage.

An example of using local storage is storing the user's color mode preference (light or dark mode). This piece of information is generally not necessary for the server to know, and is used to ensure that the user's preference is reflected on the website.

Local storage is specific to the protocol of a domain. So, the information returned for a website with HTTP protocol is different than the information returned for that same website with HTTPS protocol.

Local storage is accessed using `window.localStorage`. It's a read-only property with methods like `getItem` and `setItem` to access and modify or add data to the local storage.

What are the Differences Between Cookies and Local Storage

cookies but not the data stored in the client's storage.

So, in use cases where it's important for the server to have access to a set of data, you should use cookies.

Data Size

There's a big difference between the size of data that cookies can hold and that of the local storage. For cookies, the maximum size is 4096 bytes, whereas for local storage it's 5MB.

For that reason, cookies should not be used to store large pieces of data. For example, if you want to store the user's details in the browser then it's best to store them in the local storage.

Data Expiry

Cookies have an expiry date. You can set it to a very far away date, but it still expires. On the other hand, data in the local storage does not expire and will be available any time the user opens your website. The only way to clear local storage is either manually with code or if the user clears their browser storage manually.

Better API

Unless you're using a JavaScript library to facilitate this, it can be a hassle to read or write cookies using `document.cookies` as there is no straight way to do it. Here's an example from [W3Schools](#) of the code you'd need to use to get and set a cookie by name:

```
function setCookie(cname, cvalue, exdays) {  
  const d = new Date();  
  d.setTime(d.getTime() + (exdays*24*60*60*1000));  
  let expires = "expires=" + d.toUTCString();
```

```
function getCookie(cname) {  
  let name = cname + "=";  
  let decodedCookie = decodeURIComponent(document.cookie);  
  let ca = decodedCookie.split(';');  
  for(let i = 0; i <ca.length; i++) {  
    let c = ca[i];  
    while (c.charAt(0) == ' ') {  
      c = c.substring(1);  
    }  
    if (c.indexOf(name) == 0) {  
      return c.substring(name.length, c.length);  
    }  
  }  
  return "";  
}
```

As for local storage, it implements the Web Storage API which contains easy-to-use methods to read and write data. Here's an example of setting data in the storage and reading them:

```
localStorage.setItem('name', 'Shahed');  
  
//later  
const name = localStorage.getItem('name');
```

Which Should You Choose?

From reading the differences and definitions this should be clear.

However, to summarize it and make it even clearer:

1. If you need to store data that is accessible for both the server and the client, use cookies. Otherwise, use local storage.
2. If you need to store larger data, use local storage.

stored on the client, use local storage.

Conclusion

If you're interested to learn more about cookies and local storage, check out MDN's documentation for [cookies](#) and [localStorage](#) for a more thorough explanation of each.

If you like my content and want to support me, please share this article on your social media or [Buy me a coffee!](#)

Subscribe to Newsletter

Subscribe to the newsletter to be notified of new tutorials and articles!

You can unsubscribe anytime.

MORE IN JAVASCRIPT

How to Get Elements From the DOM Using JavaScript

19 Sep 2022 – 4 min read

What are Events in JavaScript and How to Handle Them?

10 Aug 2022 – 6 min read

React (TanStack) Query Tutorial for Beginners

24 Jul 2022 – 13 min read

[See all 52 posts →](#)

[Shahed Nasser](#)[HOME](#)[TOPICS](#)[MY GUEST WRITINGS](#)[REVIEWS](#)[COLLABORATE](#)

How to Create and Deploy a Documentation Website With Docusaurus and Netlify

blog.shahednasser.com

JAVASCRIPT

How to Create and Deploy a Documentation Website With Docusaurus and Netlify

In this tutorial, you'll learn how to create and deploy documentation with Docusaurus and Netlify.



SHAHED NASSER

16 MAY 2022 • 9 MIN READ

How to use Transitions in React 18

blog.shahednasser.com

REACT

How to use Transitions in React 18

In this tutorial, you'll learn more about Transitions in React 18 and see them in action.



SHAHED NASSER

25 APR 2022 • 3 MIN READ

Shahed Nasser © 2022

[Latest Posts](#)

