

LocalStorage & SessionStorage



Jacob Hornbeck



What is LocalStorage?

`LocalStorage` is a semi-temporary storage API that is available in nearly every version of all browsers.

The versions where each browser began using `localStorage` is pretty low (meaning the newer versions are most likely to be compatible).

`LocalStorage` is stored somewhere on your computer. The precise location depends on the OS and the browser. Whatever you store in `localStorage`, can be accessed as long as it hasn't been removed.

What is SessionStorage?

`SessionStorage` is actually very similar to `localStorage`, but `sessionStorage` is temporary. Your browser will clear it when it closes.

With a little bit of testing I found out that `sessionStorage` can only be accessed in the exact tab it was created, even if it is the same url.

Similar to `localStorage`, `sessionStorage` is also stored on your computer. The exact location varies from OSs and browsers.

How to use LocalStorage

There are three fundamental methods:

1. `.setItem(key,value)`
2. `.getItem(key)`
3. `.removeItem(key)`

These three methods are pretty much all you need to know, to use `localStorage`.

The only other method that you may need to know is `.clear()`

Examples:

```
localStorage.setItem('first','Hello World!')
```

That line of code would store the value "Hello World!" into our "variable" "first".

We could then access it by doing:

```
localStorage.getItem('first')
```

This would return the value "Hello World!".

If we wanted to remove it we could do:

```
localStorage.removeItem('first')
```

How to use SessionStorage

Same thing, three main methods:

1. `.setItem(key,value)`
2. `.getItem(key)`
3. `.removeItem(key)`

These three methods allow you to

The `.clear()` method also works with `sessionStorage`

Examples:

The only difference with the code you would use is `"sessionStorage"`

```
sessionStorage.setItem('first','Hello World!')
sessionStorage.getItem('first')
sessionStorage.removeItem('first')
```

Each line of code would do the same thing as the code for `localStorage`

Web Storage Methods

`localStorage` & `sessionStorage`

Note: The following Web Storage methods work for both `localStorage` and `sessionStorage`. So in my examples if I only reference one in my examples, just know it works for both.



The `.clear()` method

We are kind of going backwards with the different methods, but first we will discuss the `.clear()` method.

This is the thing that I had mentioned saying that I wanted to clear up some common mistakes on the use of it... but all I did was figure out that I was wrong.

I had been thinking that you should never use `.clear()` because it would clear all the `localStorage` that is saved... this method will only clear the `localStorage` that is stored for that domain (just like you can only access the `localStorage` when on the same domain as it was created. So you don't have to be as careful as I was thinking...

The `.setItem()` method

This method is probably *the* most important Web Storage method. Without it, there wouldn't be any reason for the others.

`localStorage.setItem()` and `sessionStorage.setItem()` are not the only way of storing values in session and `localStorage`. On the right we see a few ways that work to set values.

Because `localStorage` (and `sessionStorage`) is in JavaScript Object Notation (JSON), we can access the keys and values the same way.

```
localStorage.first = "Hello World!"  
localStorage['first'] = "Hello World!"  
localStorage.setItem('first', 'Hello World')
```

All of the above work, but MDN says that it is best to use the last one, to "prevent the pitfalls..."

The `.getItem()` method

This method is probably the second most important Web Storage method. Since the stored value isn't very helpful if we can't access it.

Just like with `.setItem()` we don't only have to use that method, though it is better to.

On the right, you can see our examples.

`localStorage` hasn't changed, so we can still access the values using JSON access methods (as seen below).

```
localStorage.first  
localStorage['first']  
localStorage.setItem('first')  
// All of the above give us "Hello World!"
```

Just remember that the `.setItem()` is the safest and most reliable to use.

Other Web Storage Options

The methods I showed in the previous slides are just the main methods, there are a few other options that Web Storage API offers us:

- `.key(index)`
- `.length`

The `.key()` method allows you to access the value of the key at a certain index. We could do `localStorage.key(0)` and we would get "first"

The `.length` method returns the number of values in `localStorage` or `sessionStorage`. Because we have only added one value in, if we do `localStorage.length` we would get 1.

Examples




`localStorage` & `sessionStorage`

Note: All the examples I will show are on my GitHub Pages and I will provide the link to this and all my other sources.

References

1. Title Page
2. What is LocalStorage?
3. What is SessionStorage?
4. How to use LocalStorage
5. How to use SessionStorage
6. Section Heading
7. The `.clear()` method
8. The `.setItem()` method
9. The `.getItem()` method
10. Other Web Storage Options
11. Examples

1. None
2. MDN, StackOverflow
 - a. [Window.localStorage \(Browser Compatibility\)](#)
 - b. [Local storage - Where the sessionStorage and localStorage stored?](#)
3. MDN, Stack Overflow
 - a. [Window.sessionStorage \(Browser Compatibility\)](#)
 - b. [Local storage - Where the sessionStorage and localStorage stored?](#)
- 4-10. MDN
 - a. [Using the Web Storage API](#)
5. None
11.  [Presentation Examples](#)

Thanks!

If you have any questions please let me know, and I will see what I can do to answer your questions!