**What is a localStorage?**

*localStorage* is a type of web storage that allows JavaScript sites and apps to store and access data right in the browser with no expiration date. This means the data stored in the browser will persist even after the browser window has been closed. The stored data is saved across browser sessions.

it is a read only property of window object.it doesn’t get sent to server as it is stored locally in the web browser .

*Data in a localStorage object created in a "private browsing" or "incognito" session is cleared when the last "private" tab is closed.*

**Methods and Properties provided by Storage Object**

* ***setItem(key, value)*** – store key/value pair.

***it allows to add a key/value pair to the storage object***. if the key already exists the name value will overwrite the old value.

* ***getItem(key)*** – it returns the value of the item that is set with the given key.
* ***removeItem(key)*** – remove the key with its value.
* ***clear()*** – delete everything.
* ***key(index)*** – get the key on a given position.
* ***length*** – the number of stored items.

**First: setItem()**

Just as the name implies, this method allows you to store values in the localStorage object.

It takes two parameters: a *key* and a *value*. The key can be referenced later to fetch the value attached to it.

Where name is the key and Abhishek Srivastava is the value. Also note that localStorage can only store strings.

To store arrays or objects, you would have to convert them to strings.

To do this, we use the **JSON.stringify()** method before passing to setItem().

**Second: getItem()**

The getItem() method allows you to access the data stored in the browser’s localStorage object.

It accepts only one parameter which is the *key* and returns the *value* as a string.

To retrieve the user key stored above:

https://miro.medium.com/v2/resize:fit:700/1*ml04-BTOIsodKY-Kx5DTog.png

This returns a string with value as:

https://miro.medium.com/v2/resize:fit:700/1*VaDh6HYLJD73fa1M-0Ukig.png

To use this value, you would have to convert it back to an object.

To do this, we make use of the JSON.parse() method, which converts a JSON string into a JavaScript object.

https://miro.medium.com/v2/resize:fit:577/1*4rMKWV-4ZLnLyMJ4VwIRiQ.png

**Third:**removeItem**()**

When passed a key name, the *removeItem()*method will remove that key from the storage if it exists. If there is no item associated with the given key, this method will do nothing.

https://miro.medium.com/v2/resize:fit:577/1*kaLa6B5yYPGMINh0O9nhdw.png

Forth: clear()

This method, when invoked, clears the entire storage of all records for that domain. It does not receive any parameters.

https://miro.medium.com/v2/resize:fit:700/1*OJJek6dqrqoSzu0xLiMmAw.png

Fifth: key()

The *key* method comes in handy in situations where you need to loop through keys and allows you to pass a number or index to *localStorage* to retrieve the name of the key.

https://miro.medium.com/v2/resize:fit:700/1*y7aJuJsu597MgzBuSKUM-g.png

**Sixth: length**

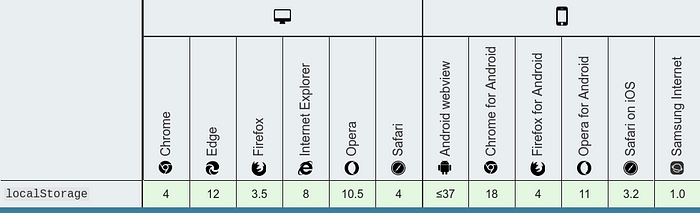
The length property returns the number of items stored in the browsers Storage Object, for this particular domain..

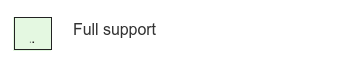
The length property belongs to the Storage Object, which can be either a localStorage object or a *sessionStorrage* object.

https://miro.medium.com/v2/resize:fit:648/1*h95_eG0gsu9vP9rwKxq4hg.png

*That’s it, now you would have good understanding of methods and properties provided by the Storage Object*

**Browser compatibility**





**localStorage limitations**

As easy as it is to use localStorage, it is also easy to misuse it. The following are limitations, and also ways to NOT use localStorage:

* Do *not store sensitive user information* in localStorage
* It is *not a substitute for a server based database* as information is only stored on the browser
* localStorage is limited to *5MB across all major browsers*
* localStorage is *quite insecure* as it has no form of data protection and can be accessed by any code on your web page
* localStorage is *synchronous*, meaning each operation called would only execute one after the other

Session Storage:- it a read only property of the window object. it stores data in a web browser specifically to the domain and protocol for a particular session.

it doesn’t get sent to the server.Data stored in sessionStorage gets cleared when the page session ends. A page session lasts for as long as the browser is open and survives over page reloads and restores.

Exception Handling:- an exception handling is a generalization of the concept of an error to include any unexpected condition encountered during exception.

Try

catch

finally

throw