More methods from localStorage/sessionStorage

This time we will look at another example that uses new methods from the API:

- `localStorage.setItem(...)`,
- `localStorage.getItem(...)`,
- `localStorage.removeItem(...)`,
- `localStorage.clear()`.

Browser's view of internal web storage implementation...
age
FirstName
lastName
nano robot controlled through JavaScript code.

GETTING/SETTING VALUES USING THE`GETITEM(KEY)` AND `SETITEM(KEY, VALUE)` METHODS

If you want to keep a simple counter of the number of times a given user has loaded your application, you can use the following code (just to show how to use `setItem/removeItem`methods):

```
var counter = localStorage.getItem("count") || 0;
counter++;
localStorage.setItem("count", counter);
```

As you can easily guess from the above, we use `var value = getItem(key)` to retrieve a key's value and `setItem(key, value)` to set it. This is similar to what we saw in the examples of the page above, except that this time:

- The key can contain spaces, for example we can write:`localStorage.setItem("Instructor's name", "Michel");` and `var name =` localStorage.getItem("Instructor's name");, while `var name = localStorage.Instructor's name;` will not work!

- In a loop or in an iterator, sometimes we need to set/get localStorage values using this syntax, for example:

```
var inputField =document.getElementById("firstName");
saveInputFieldValue(inputField);
...
function saveInputFieldValue(field) {
    localStorage.setItem(field.id, field.value);
}
```

DELETING A KEY WITH `REMOVEITEM(KEY)`, OR ALL KEYS WITH `CLEAR()`

Deleting a key can be performed through `removeItem()`. And if you wish to reset the entire store, simply call`localStorage.clear()`.

Note that it may be quite rare that you will want the entire store to be cleared by the user in production software (since that effectively deletes their entire data). However, it is a rather a common operation needed during development, since bugs may store faulty data the persistence of which can break your application, since the way you store data may evolve over time, or simply because you also need to test the experience of the user when first using the application.

One way of handling that is to add a user interface button that calls `clear()` when clicked, but you then you need not to forget to remove it when you ship! The recommended approach to use (whenever possible) is to simply open the dev. tool's console and type `localStorage.clear()` there — it's safer and works just as well.

ITERATING LOCAL STORES

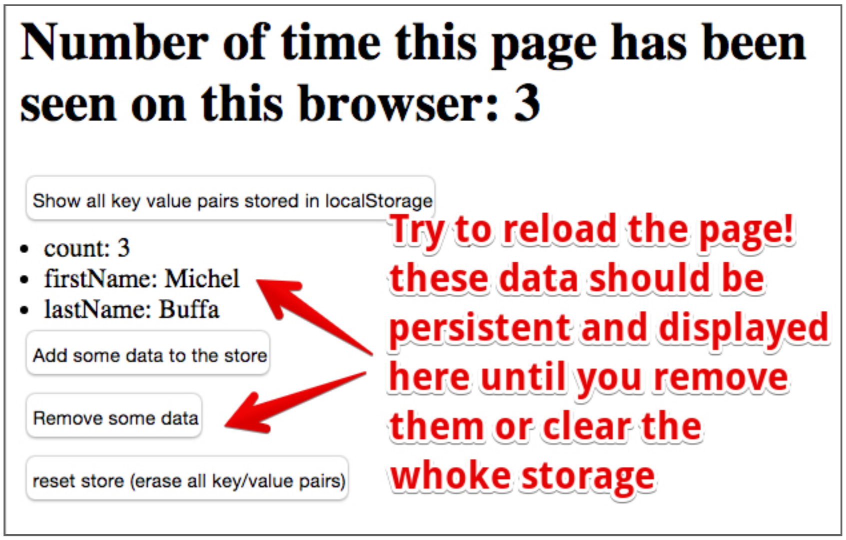
Local stores (`localStorage` or `sessionStorage`) can also be iterated through in order to list all the content that they contain. The order is not guaranteed, but this may be useful at times (if only for debugging purposes!). The following code lists everything in the current store:

```
for (var i = 0, n = localStorage.length; i < n; i++) {
    var k = localStorage.key(i);
    console.log(k + ": " + localStorage[k]); // get the ith value, the one with a key that is in the variable k.
}
```

Smart students will note something off in the example above: instead of calling `localStorage.getItem(k)`, we simply access `localStorage[k]`. Why? Because keys in the local store can also be accessed as if the store were a simple JavaScript object. So instead of`localStorage.getItem("foo")` and`localStorage.setItem("foo", "bar")`, one can write`localStorage.foo` and `localStorage.foo = "bar"`. Of course there are limitations to this mapping: any string can serve as a key, so that `localStorage.getItem("one two three")` works, whereas that string would not be a valid identifier after the dot (but it could still work as`localStorage["one two three"]`).

EXAMPLE THAT SHOWS ALL THE METHODS OF THE LOCAL STORAGE API IN ACTION

Online example at JS Bin, run it, then click on the first button to show all key/values in the `localStorage`. Open the URL in another tab, and see that the data is shared between tabs, as local stores are attached to an origin.



Number of time this page has been seen on this browser: 3

Show all key value pairs stored in localStorage

- count: 3
- firstName: Michel
- lastName: Buffa

Add some data to the store
Remove some data
reset store (erase all key/value pairs)

Try to reload the page! these data should be persistent and displayed here until you remove them or clear the whoke storage

Then click on the second button to add some data in the store, click on the third to remove some data. Finally, the last one clears the whole data store.

Source code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset=utf-8 />
<title>Example of localStorare API use</title>
 <script>
   // Using localStorage
   var counter = localStorage.getItem("count") || 0;
   counter++;
   localStorage.setItem("count", counter);
   function getCountValue() {
      // retrieve data

document.querySelector("#counter").innerHTML= localStorage.count;
   }
   function seeAllKeyValuePairsStored() {
      // clear list first
      document.querySelector('#list').innerHTML="";
      for (var i = 0, n = localStorage.length; i < n; i++) {
         var key = localStorage.key(i);
         var value = localStorage[key];
         console.log(key + ": " + value);
         var li = document.createElement('li');
         li.innerHTML = key + ": " + value;
         document.querySelector('#list').insertBefore(li,null);
      }
   }
   function resetStore() {
        // erase all key values from store
        localStorage.clear();
        // reset displayed list too
       document.querySelector('#list').innerHTML="";
   }
   function addSomeData() {
      // store data
      localStorage.lastName = "Buffa";
      localStorage.firstName = "Michel";
      // refresh display
      seeAllKeyValuePairsStored();
   }
   function removeSomeData() {
      // store data
      localStorage.removeItem("lastName");
      localStorage.removeItem("firstName");
      // refresh display
      seeAllKeyValuePairsStored();
   }
 </script>
</head>
<body onload="getCountValue()">
   <h1>Number of times this page has been seen on this browser: <span id="counter"></span></h1>
   <buttononclick="seeAllKeyValuePairsStored()">Show all key value pairs stored in localStorage</button><br/>
   <output id="list"></output>

   <button onclick="addSomeData()">Add some data to the store</button><br/>
   <button onclick="removeSomeData()">Remove some data</button><br/>
   <button onclick="resetStore()">reset store (erase all key/value pairs)</button>
</body>
</html>
```

You can check in the Chrome dev. tools user interface that the content of the `localStorage` changes as you click on the buttons.