**LocalStorage, sessionStorage**

Web storage objects localStorage and sessionStorage allow to save key/value pairs in the browser.

What’s interesting about them is that the data survives a page refresh (for sessionStorage) and even a full browser restart (for localStorage). We’ll see that very soon.

We already have cookies. Why additional objects?

* Unlike cookies, web storage objects are not sent to server with each request. Because of that, we can store much more. Most browsers allow at least 2 megabytes of data (or more) and have settings to configure that.
* The server can’t manipulate storage objects via HTTP headers, everything’s done in JavaScript.
* The storage is bound to the origin (domain/protocol/port triplet). That is, different protocols or subdomains infer different storage objects, they can’t access data from each other.

Both storage objects provide same methods and properties:

* setItem(key, value) – store key/value pair.
* getItem(key) – get the value by 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.

Let’s see how it works.

**localStorage demo**

The main features of localStorage are:

* Shared between all tabs and windows from the same origin.
* The data does not expire. It remains after the browser restart and even OS reboot.

For instance, if you run this code…

localStorage.setItem('test', 1);

…And close/open the browser or just open the same page in a different window, then you can get it like this:

alert( localStorage.getItem('test') ); // 1

We only have to be on the same domain/port/protocol, the url path can be different.

The localStorage is shared, so if we set the data in one window, the change becomes visible in the other one.