



LocalStorage



Las cookies eran bastante mejorables así que se decidieron a incluir en HTML5 `localStorage` y `sessionStorage` que podríamos decir que es la evolución de las cookies.

Las características tanto para `localStorage` como para `sessionStorage` son:

La capacidad es de 5Mb mientras que la de las cookies era de tan solo 4Kb.

La información no es incluida en cada petición al servidor como sucede con las cookies, sino que solo se obtiene cuando se quiere usar.

La información es almacenada en pares clave-valor por lo que se puede usar como si se tratase de variables.

Una página web únicamente puede acceder a la información que ha almacenado ella por lo que es más seguro.

La diferencia entre `localStorage` y `sessionStorage` es que la información almacenada con `localStorage` se guarda indefinidamente hasta que sea eliminada mediante código o bien borrada desde el navegador y con `sessionStorage` la información se guarda hasta que se cierra el navegador. Por lo tanto como la única diferencia entre ambas es el tiempo, en los ejemplos aunque voy a usar `localStorage` el código para usar `sessionStorage` sería idéntico (cambiando `localStorage` por `sessionStorage`).

El uso de `sessionStorage` o `localStorage` dependerá de que es lo que se quiere almacenar y con que fin pero el funcionamiento como decía es análogo.

Docs_ Generalidades Curiosidades

Vamos a ver un ejemplo de localStorage en el que mediante un «formulario» solicitamos el nombre y la password y los almacenamos y después de haberlos almacenado una vez si pulsas en el botón Recuperar datos obtendrás los datos que habías puesto aunque hayas cerrado el navegador si lo introduces los datos los podrás recuperar siempre que quieras.

El código HTML para el ejemplo:

```
<label for="nombre">Nombre:</label>
<input type="text" id="nombre" />

<label for="password">Password:</label>
<input type="text" id="password" />

<input type="button" onclick="guardarDatos()" value="Guardar datos" />

<p id="datos">Pulsa para ver tu nombre y tu password.</p>
<input type="button" onclick="recuperarDatos()" value="Recuperar datos" />
```

Y las 2 funciones para guardar y recuperar los datos que creo que sobran las explicaciones localStorage.nombre y se usa como una variable más. También están los métodos localStorage.getItem(«clave») y localStorage.setItem(«clave»,»valor») aunque el resultado es el mismo.

```
function guardarDatos() {
    localStorage.nombre = document.getElementById("nombre").value;
    localStorage.password = document.getElementById("password").value;
}

function recuperarDatos() {
    if ((localStorage.nombre != undefined) && (localStorage.password != undefined)) {
        document.getElementById("datos").innerHTML = "Nombre: " +
        localStorage.nombre + " Password: " + localStorage.password;
    } else {
        document.getElementById("datos").innerHTML = "No has introducido tu
        nombre y tu password";
    }
}
```

Docs_ Generalidades Curiosidades

Como supongo que has imaginado recuperar los datos para mostrarlos pues tampoco sirve de mucho pero con esto por ejemplo podríamos mantener logeados a los usuarios de nuestra web que quisiesen un login automático simplemente comprobando en la función que compruebe si tiene la sesión abierta si existen el nombre y el password y si existen hacer el logeo automáticamente sin molestar al pobre usuario que no tiene ganas de escribir.

Vamos a ver otro ejemplo para ver como eliminar los datos almacenados con un ejemplo de un contador de clicks.

```
<input type="button" onclick="contadorClicks()" value="Haz Click" />
<input type="text" id="contador" onload="contadorClicks()" />
<input type="button" onclick="resetearClicks()" value="Resetear contador" />
```

La función contadorClicks no tiene nada nuevo respecto al ejemplo anterior, únicamente hay que comprobar si no existe para iniciarlo para que no de un error al convertirlo a entero para hacer la suma.

Y en la función resetearClicks puedes ver como se elimina un dato con removeItem(«clave»), en el caso de querer eliminar todos los datos creados por la página se puede usar la localStorage.clear()

```
function contadorClicks() {
    // Si existe contador se incrementa y sino se inicializa
    if (localStorage.contador) {
        // Se convierte el contador a entero porque por defecto es una
        cadena
        localStorage.contador = parseInt(localStorage.contador) + 1;
    } else {
        localStorage.contador = 1;
    }
    document.getElementById("contador").value = localStorage.contador + "
Clicks";
}

function resetearClicks() {
    localStorage.removeItem("contador");
}
```

Docs_ Generalidades Curiosidades

Los dos ejemplos anteriores son muy sencillos pero ciertamente es que el uso de localStorage es muy simple con lo que creo que no hay motivos para seguir usando cookies teniendo localStorage porque tienes más espacio aunque tampoco es cuestión de estar almacenando cualquier cosa por poder hacerlo, además es más sencillo y más eficiente porque las cookies se envían con cada vez que se carga una pagina y de esta otra forma solo se envían los datos cuando se necesitan (cuando se piden) y además es más seguro.