

Programación Estática y Laboratorio web

Local Storage



Universidad Nacional del Comahue
Facultad de Informática



[Introducción]

LocalStorage y SessionStorage son propiedades del objeto window de javascript, que permiten almacenar datos en nuestro navegador web de manera muy similar a como lo hacen las cookies.

Ambos permiten guardar **pares de clave/valor** en el navegador.

Local Storage

Los datos guardados sobreviven por tiempo indefinido; sin importar que el navegador se cierre.

Session Storage

Los datos guardados sobreviven a una recarga de la página y se eliminan cuando el navegador se cierra.

[Introducción]

■ Características

- Permiten almacenar entre 5MB y 10MB de información (incluyendo texto y multimedia)
- La información está almacenada en la computadora del cliente y NO es enviada en cada petición del servidor, a diferencia de las cookies.
- Utilizan un número mínimo de peticiones al servidor para reducir el tráfico de la red
- Previenen pérdidas de información cuando se desconecta de la red.
- La información es guardada por dominio web (incluye todas las páginas del dominio)

[Ventajas/Desventajas]

- 10 mb para localStorage
- 5 mb para sessionStorage
- 4 kb para Cookies

■ Ventajas

- Entre 5 y 10 MB. Fácil de usar. No se necesita un web server o una base de datos.
- Muy buenos para “armar una idea” y testear rápidamente.

■ Limitaciones

- Solo se pueden guardar strings
- No son seguros. No guardar información privada o confidencial en localStorage!

[Como utilizar Local/Session Storage]

Ambos objetos proveen los mismos métodos y propiedades:

- **setItem** (clave, valor) – almacenar un par clave/valor.
- **getItem** (clave) – obtener el valor por medio de la clave.
- **removeItem** (clave) – eliminar la clave y su valor.
- **clear** () – borrar todo.
- **key** (índice) – obtener la clave de una posición dada.
- **length** – el número de ítems almacenados.

Es como una colección Map (**setItem/getItem/removeItem**), pero también permite el acceso a través de index con **key(index)**.

Como utilizar Local/Session Storage

- Si ejecutamos este Código:

```
1 localStorage.setItem('test', 1);
```

- Y luego cerramos el navegador o simplemente abrimos la misma página en otra ventana, podemos obtener el ítem que hemos guardado de este modo:

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

- Lo cual nos mostrará un mensaje con el valor 1

Como utilizar Local/Session Storage

■ Iterando sobre las claves

Los métodos proporcionan la funcionalidad get / set / remove. ¿Pero cómo conseguimos todas las claves o valores guardados?

Desafortunadamente, los objetos de almacenaje no son iterables.

Una opción es utilizar iteración sobre un array:

```
1 for(let i=0; i<localStorage.length; i++) {  
2   let key = localStorage.key(i);  
3   alert(`${key}: ${localStorage.getItem(key)}`);  
4 }
```

... O simplemente acceder a las claves “propias” con `Object.keys` y iterar sobre éstas si es necesario:

```
1 let keys = Object.keys(localStorage);  
2 for(let key of keys) {  
3   alert(`${key}: ${localStorage.getItem(key)}`);  
4 }
```

Como utilizar Local/Session Storage

■ Solo strings

Hay que tener en cuenta que tanto la clave como el valor deben ser **strings**. A pesar de eso, podemos utilizar JSON para almacenar objetos

```
1 localStorage.user = JSON.stringify({name: "John"});  
2  
3 // en algún momento más tarde  
4 let user = JSON.parse( localStorage.user );  
5 alert( user.name ); // John
```

¿Qué es JSON?

JSON es un formato de texto que forma parte del sistema de JavaScript y que se deriva de su sintaxis, pero no tiene como objetivo la creación de programas, sino el acceso, almacenamiento e intercambio de datos. Usualmente es conocido como una alternativa al lenguaje XML.

[Ejemplos]

- Cookies: EjemploJS05_Cookies.html
- Local Storage: EjemploJS05_LocalStorage.html
- Local Storage con arreglos: EjemploJS05_LocalStorageArreglos.html

[Referencias]

- <https://es.javascript.info/localstorage>
- <https://platzi.com/blog/local-storage-html5/>
- <https://www.tiny.cloud/blog/javascript-localstorage/>
- <https://blog.hubspot.es/website/que-es-json>