

Javascript: Local Storage

Introducción

Bienvenido a un nuevo paso del curso de javascript. En los siguientes pasos nos enfocaremos en aprender qué es el método `localStorage` y para qué sirve. `localStorage` es una parte fundamental del almacenamiento web, permitiendo que los datos se guarden de manera local en el navegador del usuario.

Teoría de localStorage

`localStorage` es una forma de almacenar datos en el navegador del usuario de manera persistente. A diferencia de las cookies, los datos almacenados en `localStorage` no se envían al servidor con cada solicitud. Esto lo hace ideal para almacenar información que no necesita ser transmitida constantemente, como preferencias del usuario o datos de la sesión.

Características clave de localStorage:

- **Persistencia:** Los datos guardados en `localStorage` permanecen almacenados incluso después de cerrar el navegador.
- **Tipo de datos:** `localStorage` solo admite guardar cadenas de texto .
- **Capacidad de almacenamiento:** Generalmente, puede almacenar hasta 5MB de datos, lo cual es mucho más de lo que almacenan las cookies.
- **Acceso sencillo:** Los datos se recuperan fácilmente usando JavaScript, lo que lo hace muy conveniente para aplicaciones web.
- **Seguridad:** Los datos están almacenados localmente en el navegador del usuario y son específicos del dominio, lo que significa que solo los scripts del mismo dominio pueden acceder a ellos.

Cómo usar localStorage:

1. **Guardar datos:** Puedes guardar datos en `localStorage` usando `localStorage.setItem('clave', 'valor')`.
2. **Leer datos:** Para leer datos, usa `localStorage.getItem('clave')`.
3. **Eliminar datos:** Puedes eliminar un ítem específico con `localStorage.removeItem('clave')` o limpiar todo el almacenamiento local con `localStorage.clear()`.

Ejemplo práctico:

Unset

```
// Guardar datos

localStorage.setItem('usuario', 'Juan Perez');


// Obtener datos

const usuario = localStorage.getItem('usuario');

console.log(usuario); // Salida: Juan Perez


// Eliminar datos específicos

localStorage.removeItem('usuario');


// Limpiar todo el almacenamiento local

localStorage.clear();
```

Uso de JSON.stringify y JSON.parse

En la práctica, a menudo necesitamos almacenar objetos o arrays en `localStorage`. Sin embargo, `localStorage` solo puede almacenar cadenas de texto. Aquí es donde `JSON.stringify` y `JSON.parse` se vuelven útiles.

Almacenar objetos y arrays

`JSON.stringify` se utiliza para convertir objetos o arrays en una cadena de texto, lo que permite almacenarlos en `localStorage`. Por otro lado, `JSON.parse` se utiliza para convertir la cadena de texto almacenada, de vuelta en un objeto o array.

Ejemplo de uso:

Guardar un objeto:

Unset

```
const usuario = {  
  nombre: 'Juan',  
  edad: 30  
};  
  
// Convertir el objeto a una cadena de texto y almacenarlo  
localStorage.setItem('usuario', JSON.stringify(usuario));  
  
// Obtener y convertir la cadena de texto de vuelta a un objeto  
const usuarioGuardado =  
JSON.parse(localStorage.getItem('usuario'));
```

```
1. console.log(usuarioGuardado); // Salida: { nombre: 'Juan',  
    edad: 30 }
```

Guardar un array

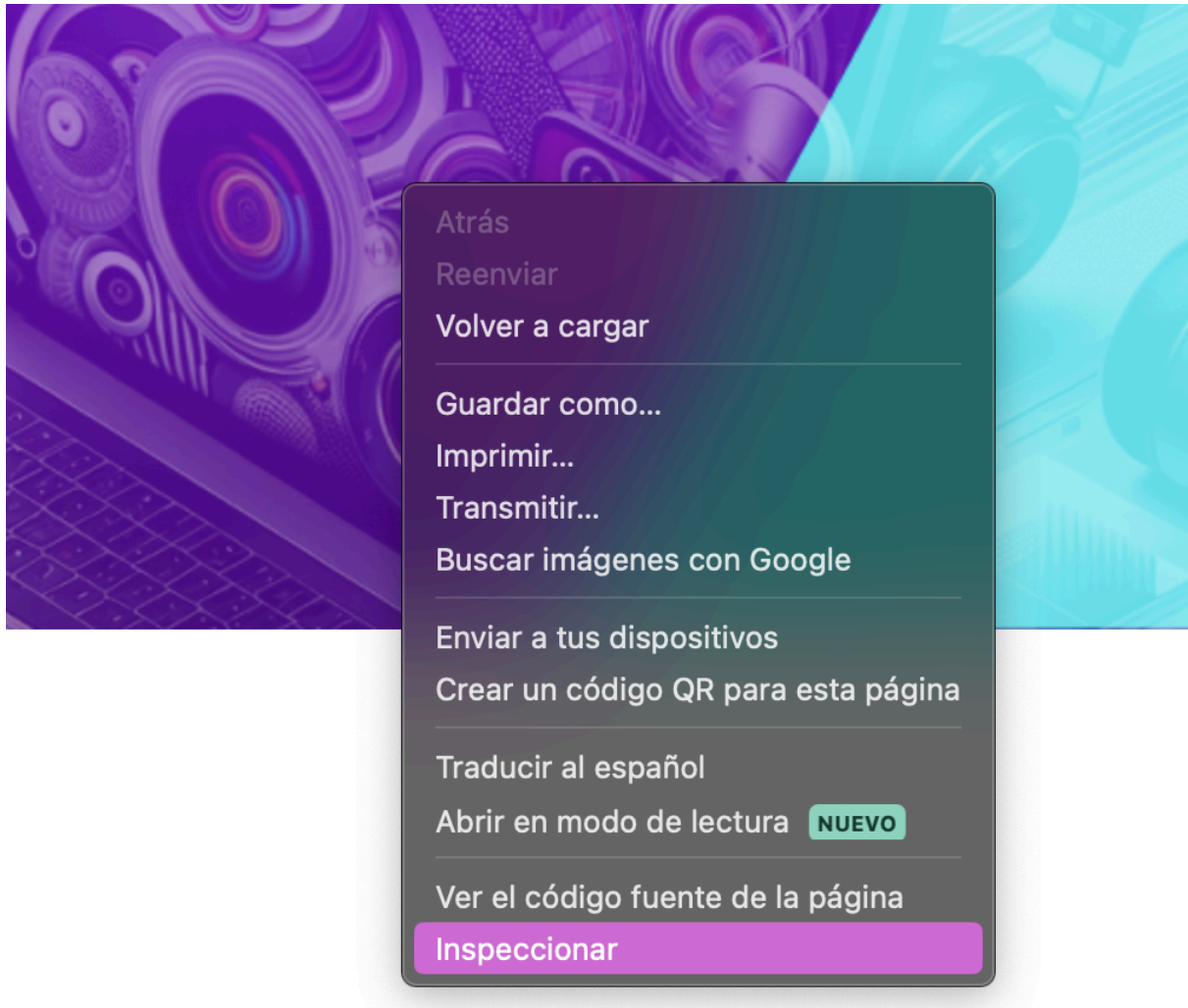
Unset

```
const frutas = ['manzana', 'banana', 'naranja'];  
  
// Convertir el array a una cadena de texto y almacenarlo  
localStorage.setItem('frutas', JSON.stringify(frutas));  
  
// Obtener y convertir la cadena de texto de vuelta a un array  
const          frutasGuardadas          =  
JSON.parse(localStorage.getItem('frutas'));  
  
2. console.log(frutasGuardadas); // Salida: ['manzana',  
    'banana', 'naranja']
```

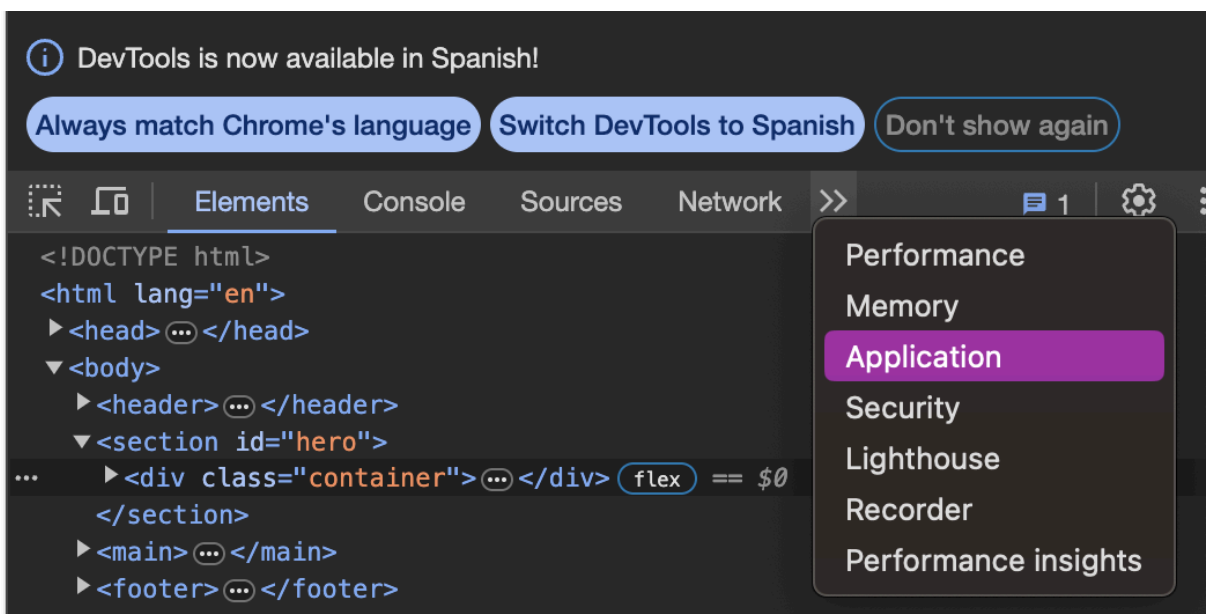
Acceder visualmente al LocalStorage

Se puede ver el `localStorage` en tiempo real, como así también modificar y eliminar registros del mismo.

Para ello, tendremos que abrir la consola de desarrolladores con `click derecho` y seleccionar la opción `Inspeccionar`.

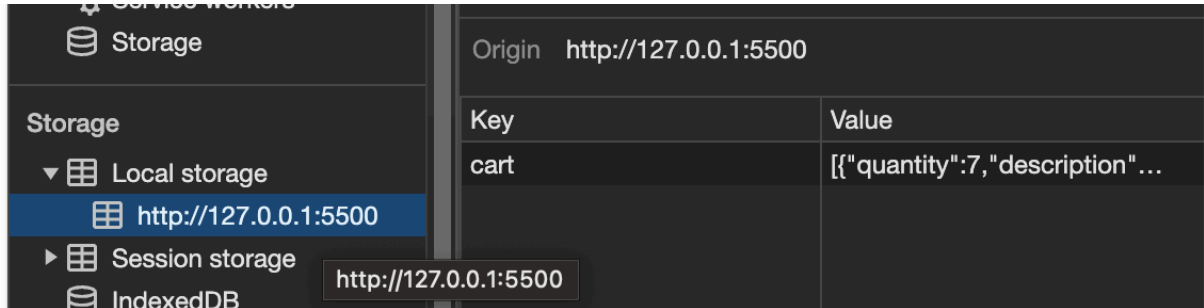


Se nos abrirá una solapa donde tendremos varias sub-pestañas, la que nos importa es **Aplicación**.



Para seleccionarla, tendremos que apretar el símbolo >> para desplegar todas las opciones.

Una vez clickeada esa opción, nos dirigiremos al apartado **Local storage**



Ahí podrás ver todos los elementos guardados en el **localStorage**. Si haces **click derecho** en cada ítem, podrás modificarlo o eliminarlo.

Si haces **doble click** en cada ítem, podrás modificarlo.

http://127.0.0.1:5500

Origin http://127.0.0.1:5500

Key	Value
cart	":7,"description"...

Header Options >

Refresh

Edit "Value"

Delete

▼ [{quantity: 7,...}]
▶ 0: {quantity: 7,...}

Abajo, podrás notar que el objeto `cart` se representa como un objeto JavaScript.

Más info en [LocalStorage](#), [JSON.stringify](#) y [JSON.parse](#)