

## ¿Qué es el Almacenamiento Local o el Local Storage ?

El almacenamiento local es una característica de los navegadores web que permite a los desarrolladores guardar datos en el navegador del usuario. Es parte de la API de almacenamiento web, junto con el almacenamiento de sesiones.

El almacenamiento local funciona **aceptando datos en pares clave-valor. Conserva los datos** incluso cuando el usuario actualiza la página o cierra la pestaña o el navegador.

## Cómo almacenar datos en almacenamiento local

Para **almacenar datos** en el almacenamiento local, utiliza el **setItem()** método. Este método toma dos argumentos, una clave y un valor.

*localStorage.setItem(key, value)*

## Cómo leer datos del almacenamiento local

Para recuperar y utilizar datos del almacenamiento local, utilice el **getItem()** método. Este método toma una clave como argumento.

*localStorage.getItem(key)*

**Si la clave dada existe** en el almacenamiento local, el método **devuelve el valor** de esa clave. **Si no es así, el método devuelve null.**

## Cómo almacenar y leer valores de datos complejos en el almacenamiento local

El almacenamiento local **solo puede almacenar cadenas**. Esto significa que si necesita almacenar valores como objetos o matrices, primero debe obtener una representación de cadena del valor. Esto lo haces usando el **JSON.stringify()** método.

Ejemplo:

```
const userObj = {  
  username : "Maria",  
  email: "maria@mail.com"  
}
```

```
localStorage.setItem('user', JSON.stringify(userObj))
```

El **JSON.stringify()** método **convierte el userObj que es un objeto en una representación de cadena** antes de enviarlo al almacenamiento local.

Ahora, cuando desee **recuperar los datos** del almacenamiento local, también **deberá cambiar su representación de cadena al formato original**. Y lo haces usando el **JSON.parse()** método.

Ejemplo:

```
const storedUserData = localStorage.getItem('user')
```

```
if (storedUserData) {  
  const userData = JSON.parse(storedUserData)  
  // You can use userData here...  
} else {  
  console.log('User data not found in local storage')  
}
```

En el ejemplo anterior, primero verificamos si hay datos para el 'usuario' en el almacenamiento local antes de usar el **JSON.parse()** método. Esto es

importante porque si no existe en el almacenamiento local, **JSON.parse()** se aplicará **a un null valor** (lo que **resultará en un error**).

## **Cómo eliminar datos del almacenamiento local**

Hay dos métodos disponibles para eliminar datos del almacenamiento local. Uno es el **removeItem()** método y el otro es el **clear()** método.

Utilice este **removeItem() método cuando desee eliminar un solo elemento** del almacenamiento local. El método toma una clave como argumento y elimina el par clave-valor correspondiente del almacenamiento local.

*localStorage.removeItem(key)*

Pero, ¿qué pasa si, en lugar de eliminar un único par clave-valor, desea borrar todos los datos del almacenamiento local? Bueno, el almacenamiento local tiene un método para eso: el **clear()** método.

*localStorage.clear()*

**El clear() método elimina todos los pares clave-valor** en el almacenamiento local del dominio actual.