



UNIVERSIDAD AUTÓNOMA DE ENCARNACIÓN
FACULTAD DE CIENCIAS, ARTE Y TECNOLOGÍA

TRABAJO DE INVESTIGACION JQUERY VS JAVASCRIPT VANILLA

Diseño Y Programación Web II

Análisis De Sistemas.

Prof. Osvaldo Enrique Micniuk

Franco Vergara

Kevin Nieto Vera

Alejandro Dutil

Katsuo Kumagai

David Rodriguez

CURSO: 2° año

Encarnación – Paraguay

Abril – 2023

Introducción

Estos son dos aspectos fundamentales del desarrollo web moderno: el uso de Local Storage en HTML5 y la manipulación de datos en formato JSON en aplicaciones web. Ambos temas son vitales para entender cómo almacenar, manipular y transmitir datos de manera eficiente y segura en entornos web.

Local Storage de HTML5

El Local Storage de HTML5 nos permite almacenar datos de forma persistente en el navegador del usuario, lo que resulta útil para mantener preferencias de usuario, cachear datos y mejorar la experiencia general de la aplicación. Sin embargo, es crucial comprender sus limitaciones, buenas prácticas y cómo garantizar la seguridad de la información almacenada.

Almacenamiento de datos y seguridad de Local Storage

LocalStorage es un sistema de almacenamiento de **clave-valor** basado en cadenas de texto. Almacena los datos de forma persistente en el navegador del usuario incluso después de que se cierre la ventana del navegador. LocalStorage, aunque ampliamente utilizado, tiene una capacidad de almacenamiento limitada (**normalmente 5 MB por dominio**). Este límite puede ser un obstáculo significativo cuando se trata de aplicaciones más robustas que manejan grandes cantidades de datos. LocalStorage **no proporciona una capa robusta de seguridad**. Los datos almacenados en LocalStorage son accesibles fácilmente a través de scripts, lo que podría ser un **riesgo de seguridad** significativo para aplicaciones que manejan información confidencial.

Buenas prácticas con Local Storage

- Limitar el uso de Local Storage para datos pequeños y no críticos.
- Evitar almacenar información sensible como contraseñas o información de sesión.
- Realizar una limpieza periódica del almacenamiento para eliminar datos obsoletos y liberar espacio.
- Implementar manejo de errores para gestionar posibles excepciones al acceder al almacenamiento.

Casos de uso del Local Storage

Local Storage es más útil en escenarios donde se necesite almacenar datos de forma persistente en el dispositivo del usuario y donde la aplicación pueda beneficiarse de recordar cierta información entre sesiones o páginas.

- **Preferencias del usuario:** muchas aplicaciones web utilizan el Local Storage para almacenar las preferencias del usuario, como la configuración de idioma, el tema preferido, la configuración de notificaciones, etc.
- **Caché de datos:** datos en caché para mejorar el rendimiento de la aplicación, como datos estáticos que no cambian con frecuencia para evitar llamadas innecesarias al servidor.

JSON

¿Qué es JSON?

JSON (**JavaScript Object Notation**) es un formato ligero de intercambio de datos que se utiliza para transmitir datos estructurados entre un servidor y un cliente. Es

importante en el desarrollo web moderno porque es **fácil de leer y escribir** para los humanos, además de ser **fácilmente parseable** por las máquinas. JSON se ha convertido en el formato estándar para la transmisión de datos en aplicaciones web y es ampliamente compatible con la mayoría de los lenguajes de programación.

Serialización y deserialización en JavaScript

La **serialización** es el proceso de convertir un objeto JavaScript en una cadena de texto JSON, mientras que la **deserialización** es el proceso inverso, es decir, convertir una cadena de texto JSON en un objeto JavaScript. En JavaScript, se utiliza el método `JSON.stringify()` para **serializar** un objeto y `JSON.parse()` para **deserializar** una cadena JSON en un objeto.

Buenas prácticas y seguridad de datos

- Validar los datos antes de serializarlos o deserializarlos para **garantizar su integridad**.
- Evitar la inclusión de funciones en objetos JSON para **prevenir ataques** de ejecución de código no deseado.
- Utilizar métodos de serialización y deserialización proporcionados por la plataforma (como `JSON.stringify()` y `JSON.parse()`) en lugar de implementar soluciones personalizadas.

Acciones realiza cada ejemplo de código

1)

```
document.getElementById('miElemento').textContent = 'Nuevo Texto';
```

Este código selecciona un elemento del DOM con el id 'miElemento' y cambia su contenido de texto por 'Nuevo Texto'.

2)

```
const nuevoElemento = document.createElement('div');  
nuevoElemento.textContent = 'Soy un nuevo elemento';  
document.body.appendChild(nuevoElemento);
```

Aquí se crea un nuevo elemento div, se le asigna el texto interno 'Soy un nuevo elemento' y luego se agrega al final del cuerpo del documento (DOM).

3)

```
const elementoParaEliminar =  
document.getElementById('elementoEliminar');  
elementoParaEliminar.parentNode.removeChild(elementoParaEliminar);
```

Este código selecciona un elemento del DOM con el id 'elementoEliminar' y luego elimina ese elemento y su respectivo nodo padre del DOM.

4)

```
const miObjeto = { nombre: 'Juan', edad: 30 };  
localStorage.setItem('usuario', JSON.stringify(miObjeto));
```

Aquí se guarda un objeto JavaScript miObjeto en el almacenamiento local del navegador bajo la clave 'usuario'. Antes de guardarlo, se convierte en una cadena JSON utilizando JSON.stringify().

5)

```
const usuario = JSON.parse(localStorage.getItem('usuario'));  
console.log(usuario.nombre);
```

Este código obtiene la cadena JSON del almacenamiento local con la clave 'usuario', la convierte de nuevo en un objeto JavaScript usando JSON.parse() y luego accede a la propiedad 'nombre' del objeto y la imprime en la consola. En este caso, imprimirá 'Juan'.

Bibliografía

<https://atacomsian.com/blog/what-is-json>

<https://blog.hubspot.es/website/que-es-json>

<https://blog.hubspot.es/website/que-es-json>

<https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON>

<https://techieshield.com/serializar-y-deserializar-objetos-en-javascript/>

<https://json.org/>

<https://json.org/json-fr.html>

<https://platzi.com/blog/local-storage-html5/>