



JS

UT8. ALMACENAMIENTO EN EL LADO DEL CLIENTE

Desarrollo Web en Entorno
Cliente
2ºDAW

1. INTRODUCCIÓN

Los navegadores web modernos admiten varias formas para que los sitios web almacenen datos en la computadora del usuario, con el permiso del usuario, y luego los recuperen cuando sea necesario

Usos:

- Personalizar las preferencias del sitio
 - Persistencia de la actividad anterior del sitio
 - Guardar datos de forma local para que sea más rápido o incluso usar sin conexión a red.
 - Guardar documentos generados por aplicaciones web localmente para usarlos sin conexión
-
- https://developer.mozilla.org/es/docs/Learn/JavaScript/Client-side_web_APIs/Client-side_storage

2. FORMAS DE ALMACENAMIENTO

Vieja escuela: Cookies

Nueva escuela:

- API de almacenamiento web
- API IndexedDB

El Futuro: API Cache

3. COOKIES



Es una pequeña parte de información guardada en el cliente. Se usan para persistencia de datos en el cliente. Sobre todo para:

- Personalización: Preferencias de usuario, temas y otras configuraciones

- Rastreo: Guardar y analizar el comportamiento del usuario

Hoy en día es preferible usar los métodos más modernos mencionados anteriormente.

<https://developer.mozilla.org/es/docs/Web/HTTP/Cookies>

3. TIPOS DE COOKIES



De sesión.

Se elimina cuando el cliente se cierra

Permanentes

Expiran en una fecha específica

<https://developer.mozilla.org/es/docs/Web/HTTP/Cookies>

3.1 ESCRIBIR COOKIES

Expires y **path** son opcionales.

Si no ponemos fecha de expiración la cookie será de sesión, y si ponemos tendrá una fecha de caducidad.

La ruta por defecto será /, alternatively, si queremos que la cookie sea accesible solo en un cierto subdirectorío tenemos que reescribir

```
var nuevaCookie = nombre_cookie + "=" + valorCookie +  
                    "; expires=" + expira +  
                    "; path=" + ruta;  
document.cookie = nuevaCookie;
```

<http://w3.unpocodetodo.info/jsblog/cookies.php>

https://aprende-web.net/javascript/js14_1.php

3.1 ESCRIBIR COOKIES

```
function crearCookie(nombre, valorCookie, dias) {  
    let ruta = "/";  
    let nuevaCookie  
    if (dias) {  
        // el argumento dias es opcional  
        // si no especificamos la data cuando expira, se considera que la cookie dura solo una sesion y es c  
        let data = new Date();  
        // establece la data cuando la cookie expira en milisegundos  
        data.setTime(data.getTime() + dias * 24 * 60 * 60 * 1000);  
        //El método toUTCString() convierte la data en una cadena de texto (string), utilizando la zona hora  
        let expira = data.toUTCString();  
        // "/" representa el directorio raíz. 0 sea: la cookie es accesible en todo el dominio web si var r  
        nuevaCookie = nombre + "=" + valorCookie + "; expires=" + expira + "; path=" + ruta;  
    } else {  
        nuevaCookie = nombre + "=" + valorCookie + "; path=" + ruta  
    }  
    document.cookie = nuevaCookie;  
}
```

3.2 LEER COOKIES

Las cookies se almacenan todas en una sola cadena, así que tenemos que extraer el valor de la que nos interese. Podemos programar una función que nos extraiga el valor de una cookie cualquiera

```
function leerCookie(nombre) {  
    const lista = document.cookie.split(";");  
    for (i in lista) {  
        let busca = lista[i].search(nombre);  
        if (busca > -1) { micookie = lista[i] }  
    }  
    let igual = micookie.indexOf("=");  
    let valor = micookie.substring(igual + 1);  
    return valor;  
}
```

https://aprende-web.net/javascript/js14_2.php

3.3 BORRAR UNA COOKIE

Para borrar una cookie simplemente creamos de nuevo la misma cookie, cuyo valor esta vez es una cadena vacía de texto (""), y que expire antes del momento actual: por ejemplo ayer

```
function borrarCookie(nombre) {  
    crearCookie(nombre, "", -1);  
}
```

4. API DE ALMACENAMIENTO WEB

Todos tus datos de almacenamiento web están contenidos en dos estructuras similares a objetos dentro del navegador:

[sessionStorage](#) y [localStorage](#).

El primero conserva los datos mientras el navegador está abierto (los datos se pierden cuando se cierra el navegador) y el segundo conserva los datos incluso después de que el navegador se cierra y luego se vuelve a abrir. Usaremos el segundo, ya que generalmente es más útil.

4.1 API WEB STORAGE. MÉTODOS.

```
localStorage.setItem('nombre', 'Chris');
```

```
let miNombre = localStorage.getItem('nombre');  
miNombre
```

```
localStorage.removeItem('nombre');  
let miNombre = localStorage.getItem('nombre');  
miNombre
```

5. API INDEXEDDB

Es un sistema de base de datos completo disponible en el navegador en el que puedes almacenar datos complejos relacionados, tipos de los cuales no se limitan a valores simples como cadenas o números. Puedes almacenar videos, imágenes y casi cualquier otra cosa en una instancia de IndexedDB.

Sin embargo, esto tiene un costo: IndexedDB es mucho más complejo de usar que la API de almacenamiento web.

https://developer.mozilla.org/es/docs/Learn/JavaScript/Client-side_web_APIs/Client-side_storage