Web Storage. Ejemplo de uso

- C.F.G.S. DAW
- 6 horas semanales
- Mes aprox. de impartición: Ene

Carmelo José Jaén Díaz

- iPasen - cjaedia071@g.educaand.es

Índice

OBJETIVO

- Descubrir elementos propios de JavaScript como los eventos
- Registrar en el front-end información de navegación, usuario, etc., mediante cookies o mediante el almacenamiento local que ofrece el estándar HTML5.

GLOSARIO

Expresión regular. Secuencia de caracteres que forman un patrón determinado, generalmente un patrón de búsqueda.

Listener. Código responsable de controlar eventos. Están a la escucha (de ahí su nombre) y, cuando ocurre un evento, se ejecuta el código que el programador haya implementado.

INTRODUCCIÓN

En esta lección trabajaremos con la **API de Almacenamiento Web**, es decir, con **Web Storage.**

Anteriormente, vimos cómo comprobar **si el navegador soportaba Web Storage** y cuáles eran los **métodos** que nos permitían trabajar con este tipo de elementos de almacenamiento web.

Ahora, por el contrario, **pondremos en práctica lo aprendido** con un ejemplo en el que se controla que un usuario haya accedido a nuestra página llevando un control de las veces que realiza esta acción.

INTRODUCCIÓN

Para ello, crearemos una serie de funciones que interactuarán con las variables de almacenamiento web:

- En primer lugar, **comprobaremos si el navegador soporta Web Storage** y mostraremos un mensaje con el nombre del usuario si este se ha *logueado* en la página, diferenciando si es su **primera visita** o ya ha entrado en **más ocasiones**.
- Utilizaremos una función para llevar el **control de las veces que el usuario ha accedido** a la página.
- Crearemos dos funciones para **incrementar y decrementar** el valor del contador anterior.
- Finalmente crearemos otra función para hacer *logout* y **restablecer los valores** anteriores.

EJEMPLO DE USO HTML

En primer lugar, se implementan en nuestra aplicación web:

- Tres botones, asociados mediante eventos a la funcionalidad de la aplicación web.

```
<button type="button" id="incrementar">Incrementar</button>
<button type="button" id="decrementar">Decrementar</button>
<button type="button" id="logout">Log out</button>
```

- Dos párrafos donde mostrar la información al usuario.

```
<p id="saludo"></p>
<p id="contador"></p>
```

EJEMPLO DE USO
Funcionalidad - Bienvenida al usuario

En primera instancia, se comprueba si el navegador soporta *Web Storage*:

```
if (typeof(Storage) !== "undefined") {
    alert("El navegador soporta WebStorage");
    //El código desarrollado a continuación va aquí
}
else {
    alert ("El navegador no soporta Web Storage");
}
```

EJEMPLO DE USO
Funcionalidad - Bienvenida al usuario

Si el navegador soporta *Web Storage*, vamos a comprobar si el usuario se ha *logueado* en la página web o no.

- Para ello, la primera vez que el usuario ingrese en la página (en esta situación, se asume que el usuario no está *logueado*) se le pregunta cuál es su nombre de usuario.

- A partir de entonces, cada vez que entre en la página se va a recordar que es él, independientemente de si cerramos o no la pestaña del navegador, es decir, empleamos `localStorage`. Para que se mantenga la información hasta que el usuario decida borrar la información (haciendo uso del botón «Log out»).

EJEMPLO DE USO
Funcionalidad - Bienvenida al usuario

Para comprobar si el usuario se ha logueado en la página web o no, comprobamos si el usuario existe en `localStorage`.

```
if (localStorage.getItem("usuario")!=null){
    //Si existe, modificamos el párrafo con id="saludo"
    document.getElementById("saludo").innerHTML="¡Bienvenido/a de nuevo,
"+localStorage.usuario+"!";
}
```

EJEMPLO DE USO
Funcionalidad - Bienvenida al usuario

```
else{
    //Si no existe, almacenamos el valor de usuario
    localStorage.setItem("usuario",prompt("¿Cómo te llamas?"));
    //y modificamos el párrafo con id="saludo"
    document.getElementById("saludo").innerHTML="¡Tu primera visita, "+
    localStorage.usuario+"!";
}
```

EJEMPLO DE USO
Funcionalidad - Contador

A continuación, mediante los botones «Incrementar» y «Decrementar», modificaremos el contenido del párrafo con `id="contador"`.

En primer lugar, tenemos que saber si contador tiene algún valor previo. La información del contador se perderá si se cierra la pestaña, por lo que emplearemos `sessionStorage`.

```
//Si no existe el elemento contador, lo creamos con valor 0
if(!sessionStorage.getItem("contador"))
    //Recuerda que los elementos son de tipo string
    sessionStorage.setItem("contador","0");
```

EJEMPLO DE USO
Funcionalidad - Contador

En caso de que exista el elemento, modificaremos el contenido del párrafo mencionado.

```
document.getElementById("contador").innerHTML="Contador:
"+sessionStorage.getItem("contador");
```

EJEMPLO DE USO
Funcionalidad - Botones

Asociamos la funcionalidad a los botones mediante eventos:

```
document.getElementById("incrementar").addEventListener("click",incrementar);
document.getElementById("decrementar").addEventListener("click",decrementar);
document.getElementById("logout").addEventListener("click",logout);
```

EJEMPLO DE USO
Funcionalidad - incrementar()

Se parte de que el elemento contador ya contiene un valor, 0 la primera vez que se entra a la página web, o uno establecido por los botones.

```
function incrementar(){
//Parseamos el string a número mediante Number
sessionStorage.setItem("contador",Number(sessionStorage.getItem("contador"))+1);
//Sobreescribimos el contenido de la etiqueta
document.getElementById("contador").innerHTML = "Contador: "+
sessionStorage.getItem("contador");
    }
```

EJEMPLO DE USO
Funcionalidad - decrementar()

Se parte de que el elemento contador ya contiene un valor, 0 la primera vez que se entra a la página web, o uno establecido por los botones.

```
function decrementar(){
//Parseamos el string a número mediante Number
sessionStorage.setItem("contador",Number(sessionStorage.getItem("contador"))-1);
//Sobreescribimos el contenido de la etiqueta
document.getElementById("contador").innerHTML = "Contador: "+
sessionStorage.getItem("contador");
    }
```

EJEMPLO DE USO
Funcionalidad - logout()

La función `logout()` borra la información del elemento `localStorage` **`usuario`**.

```
function logout(){
    //Informamos del cambio para visualizarlo
    alert("Se ha cerrado la sesión de "+localStorage.getItem("usuario"));

    localStorage.removeItem("usuario");
    //localStorage.clear(); Borraría todos los elementos localStorage, por lo que
    las sessionStorage, elementos contador no se borrarían

    document.getElementById("saludo").innerHTML = ""; //Actualizamos el párrafo
     }
```

COOKIES - EJEMPLO AVANZADO
Visualización del ejemplo

La salida del código anterior puede verse en la imagen adjunta o en el siguiente link:

https://codepen.io/Carmelo-Jos-Ja-n-D-az/pen/WbeBwjN

Nota: *la información de sessionStorage se mantiene en la misma pestaña pero no entre pestañas diferentes.*