



Cookies. Ejemplo avanzado

-
- C.F.G.S. DAW
 - 6 horas semanales
 - Mes aprox. de impartición: Ene
 - iPasen - cjaedia071@g.educaand.es

Carmelo José Jaén Díaz

Índice



Objetivo

Glosario

Interacción persona - ordenador

Objetivos

Características. Usable.

Características. Visual.

Características. Educativo y actualizado.

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



Cerramos el bloque de *cookies* con esta lección en la que vemos cómo trabajar con *cookies* de un modo avanzado, diseñando nuestras propias funciones para crearlas, modificarlas, obtener su valor y borrarlas, en todas ellas controlando la fecha de expiración.

¿Sabías que puedes eliminar una cookie simplemente estableciendo su fecha de expiración a cero?

COOKIES - EJEMPLO AVANZADO

Botones



En el ejemplo anterior, se implementaron una serie de botones asociados a una serie de eventos que interactuaban con unas *cookies*. En este caso, se mejora la eficiencia y escalabilidad de la implementación mediante una serie de funciones que nos servirán cada vez que trabajamos con *cookies*.

En primer lugar, vamos a implementar una serie de botones en nuestro código HTML:

```
<button type="button" id="verTodas">Ver todas las cookies</button>  
<button type="button" id="crearCookie">Crear cookie</button>  
<button type="button" id="modificarCookie">Modificar cookie</button>  
<button type="button" id="leerCookie">Leer cookie</button>  
<button type="button" id="borrarCookie">Borrar cookie</button>
```

COOKIES - EJEMPLO AVANZADO

setCookie()



Y a continuación, implementaremos una serie de funciones. Estas funciones son una modificación de las que podemos encontrar en [W3SCHOOLS](#).

Mediante la función `setCookie()` se crean y modifican las *cookies*.

- A la hora de crearla, el nombre de la *cookie* no existiría.
- Si la estamos modificando, reconoce que el nombre ya existe y cambia el valor y/o fecha de expiración.

La expiración de la *cookie* será un número de días pasado como argumento.

COOKIES - EJEMPLO AVANZADO

setCookie()



```
function setCookie(nombre, valor, diasExpiracion){  
    let d = new Date(); //Creamos un objeto con la fecha actual  
    //A esa fecha le sumamos los días de expiración  
    //Como a la fecha está expresada en milisegundos,  
    //convertimos esos días a milisegundos.  
    d.setTime(d.getTime()+diasExpiracion*24*60*60*1000);  
  
    //Creamos una variable con el string referente a la expiración  
    //Como necesitamos esa fecha en formato UTC, la formateamos  
    let expiracion = "expires="+d.toUTCString();  
  
    //Por ultimo, creamos la cookie con los argumentos pasados  
    document.cookie = nombre+"="+valor+";"+expiracion+";path=/";  
}
```


COOKIES - EJEMPLO AVANZADO

getCookie()



Mediante la función `getCookie()` obtenemos los datos de una *cookie* pasada como argumento.

```
function getCookie(nombre){  
    //Añadimos el ; al nombre de la cookie  
    let nom = nombre+"=";  
  
    //Almacenamos en un array todas las cookies de nuestra aplicación web  
    //separadas por ;  
    //Los elementos de ese array serán un string del tipo nombre=valor;  
    let array = document.cookie.split(";");
```

COOKIES - EJEMPLO AVANZADO

getCookie()



Mediante la función `getCookie()` obtenemos los datos de una *cookie* pasada como argumento.

```
//Recorremos el array
for (let i=0; i<array.length; i++){
    //Almacenamos en una variable cada uno de los elementos del array
    let c = array[i];
    //Recorremos el elemento, un string del tipo nombre=valor; mientras
    //el caracter sea un espacio
    while (c.charAt(0)==" "){
        //Sobreescribimos el string c, con todo lo que haya a partir del
        //segundo caracter incluido (es decir omite el espacio y se queda
        //con todo lo demás)
        c = c.substring(1);
    }
}
```

COOKIES - EJEMPLO AVANZADO

getCookie()



Mediante la función `getCookie()` obtenemos los datos de una *cookie* pasada como argumento.

```
//Ahora extraemos la parte del final de la cadena  
//En c, tendremos un string del tipo nombre=valor  
//A través de indexOf se comprueba si en la primera posición  
//se encuentra lo que tenemos guardado en la variable nombre  
//recibida como parámetro  
if (c.indexOf(nomb) == 0){  
    //en caso de que así sea, devolvemos el texto valor  
    //es decir lo que hay después del igual  
    //parámetro desde substring: la longitud del nombre con el =  
    //parámetro hasta substring: la longitud de c, es decir valor  
    return c.substring(nom.length, c.length);  
}
```

COOKIES - EJEMPLO AVANZADO

getCookie()



Mediante la función `getCookie()` obtenemos los datos de una *cookie* pasada como argumento.

```
//En caso de no encontrar una cookie con el dato recibido como parámetro  
//nombre, devolveremos vacío o null  
return null;  
}
```

COOKIES - *setCookie()* y *getCookie()*

Visualización del ejemplo



En el siguiente ejemplo, se han incluido sentencias de visualización para mejorar la comprensión de la funciones anteriores.

```
function setCookie(nombre, valor, diasExpiracion) {  
    let d = new Date();  
    d.setTime(d.getTime() + diasExpiracion * 24 * 60 * 60 * 1000);  
    let expiracion = "expires=" + d.toUTCString();  
    document.cookie = nombre + "=" + valor + ";" + expiracion + ";path=/";  
}
```

COOKIES - *setCookie()* y *getCookie()*

Visualización del ejemplo



```
function getCookie(nombre) {  
    let nom = nombre + "=";  
    let array = document.cookie.split(";");  
    console.log("Existen estas cookies: " + array);  
    for (let i = 0; i < array.length; i++) {  
        let c = array[i];  
        console.log("La cookie " + i + " es: " + array[i]);  
        while (c.charAt(0) == " ") {  
            c = c.substring(1);  
            console.log("El texto nombre=valor de la cookie " + i + " es: " + c);  
        }  
    }  
}
```

COOKIES - *setCookie()* y *getCookie()*

Visualización del ejemplo



```
if (c.indexOf(nombre) == 0) {  
    console.log("El dato de la cookie " + nombre + " es " +  
        c.substring(nom.length, c.length));  
    return c.substring(nom.length, c.length);  
}  
}  
return null;  
}  
  
setCookie("docente", "carmelo", 13);  
setCookie("especialidad", "SAI", 10);  
alert(getCookie("docente"));  
alert(getCookie("pepe"));
```

COOKIES - EJEMPLO AVANZADO

deleteCookie()



Mediante la función `deleteCookie()` modificamos los datos de una *cookie* pasada como argumento, con el fin de borrarla.

```
function deleteCookie(nombre){  
    //Pasamos un valor vacío  
    //Establecemos la fecha de expiración con el valor más pequeño posible  
    setCookie(nombre, "", 0);  
}
```


COOKIES - EJEMPLO AVANZADO

Eventos



Una vez implementadas las funcionalidades, se asocian a los botones anteriormente descritos mediante una serie de eventos. Para ello:

```
document.getElementById("verTodas").addEventListener("click", verCookies);  
document.getElementById("crearCookie").addEventListener("click", crearModifCookie);  
document.getElementById("modificarCookie").addEventListener("click", crearModifCookie);  
document.getElementById("leerCookie").addEventListener("click", leerCookie);  
document.getElementById("borrarCookie").addEventListener("click", borrarCookie);
```

Nota: los botones «crearCookie» y «modificarCookie» hacen uso de la misma funcionalidad `setCookie`, por lo que llaman a la misma función.

COOKIES - EJEMPLO AVANZADO

verCookies()



Por último, implementamos los métodos invocados por los botones, que hacen uso de las funciones anteriormente descritas.

```
//Mostramos todas las cookies que se encuentran actualmente  
function verCookies(){  
    alert("Cookies actuales:\n"+document.cookie);  
}
```

COOKIES - EJEMPLO AVANZADO

crearModifCookie()



```
function crearModifCookie(){  
    let nombre = prompt("Introduzca el nombre de la cookie");  
    let valor = prompt("Introduzca su valor");  
    let expiracion = parseInt(prompt("Introduzca el número de días para que  
    expire la cookie"));  
    setCookie(nombre, valor, expiracion);  
    verCookies();  
}
```

COOKIES - EJEMPLO AVANZADO

leerCookie()



```
function leerCookie(){  
    let nombre = prompt("Introduzca el nombre de la cookie a consultar");  
    alert("El valor de la cookie " + nombre + " es: "+getCookie(nombre));  
}
```

COOKIES - EJEMPLO AVANZADO

borrarCookie()



```
function borrarCookie(){  
    let nombre = prompt("Introduzca el nombre de la cookie a borrar");  
    deleteCookie(nombre);  
    verCookies();  
}
```

COOKIES - EJEMPLO AVANZADO

Finalidad didáctica



La finalidad de esta forma de implementación, es que las funciones `setCookie()`, `getCookie()` y `deleteCookie()`, puedan ser implementadas en un fichero JavaScript externo y así ser reutilizadas dependiendo del contexto de nuestra aplicación web.

En este caso, las funciones intermedias `verCookies()`, `crearModifCookie()`, `leerCookie()` y `borrarCookie()` hacen uso de las anteriores para ejemplificar su uso.

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/yyBrKOp>

