



Carmelo José Jaén Díaz



# Cookies. Ejemplo sencillo

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

# 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

---



En la lección anterior vimos una pequeña introducción sobre las cookies: qué son, cómo crearlas, modificarlas y borrarlas.

Esta vez vamos a ver un ejemplo sencillo en el que, mediante una serie de botones asociados a una serie de eventos, crearemos y borraremos cookies y comprobaremos cómo las guarda nuestro navegador.

Para crear, leer y borrar cookies únicamente debemos acceder a la propiedad `document.property`.

# COOKIES - EJEMPLO SENCILLO

## Botones

---



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="crearCookie1">Crear cookie 1</button>  
<button type="button" id="crearCookie2">Crear cookie 2</button>  
<button type="button" id="borrarCookie1">Borrar cookie 1</button>  
<button type="button" id="borrarCookie2">Borrar cookie 2</button>
```

# COOKIES - EJEMPLO SENCILLO

## Eventos

---



Y a continuación, implementaremos una serie de funciones que serán invocadas mediante eventos asociados a los botones anteriores.

La lógica de dichas funciones es bastante sencilla y por ahora, no tiene una sintaxis óptima debido a su posterior reutilización en un ejemplo más avanzado.

```
document.getElementById("verTodas").addEventListener("click", verTodas);
document.getElementById("crearCookie1").addEventListener("click", crearCookie1);
document.getElementById("crearCookie2").addEventListener("click", crearCookie2);
document.getElementById("borrarCookie1").addEventListener("click", borrarCookie1);
document.getElementById("borrarCookie2").addEventListener("click", borrarCookie2);
```

# COOKIES - EJEMPLO SENCILLO

## Eventos

---



Como se puede observar esta implementación es de lejos la más eficiente, ya que para cada *cookie* debemos escribir una sentencia accediendo al elemento y asociándole una funcionalidad mediante un evento. Si tuviéramos muchas cookies esto no es escalable.

Por tanto vamos a integrarlas, obteniendo tres funciones `verTodas()`, `crearCookie()`, `borrarCookie()`.

```
document.getElementById("verTodas").addEventListener("click", verTodas);
document.getElementById("crearCookie1").addEventListener("click", crearCookie);
document.getElementById("crearCookie2").addEventListener("click", crearCookie);
document.getElementById("borrarCookie1").addEventListener("click", borrarCookie);
document.getElementById("borrarCookie2").addEventListener("click", borrarCookie);
```



# COOKIES - EJEMPLO SENCILLO

## verTodas()



A continuación, desarrollamos cada una de las funciones asociadas a los eventos anteriores.

```
function verTodas(){  
    alert(document.cookie);  
    //Recuerda que muestra todas las cookies separadas por ;  
}
```

# COOKIES - EJEMPLO SENCILLO

## crearCookie()



Recuerda que no podemos implementar ningún parámetro, debido a que si definimos la función en la sentencia del evento con `()`, esta se invocaría automáticamente y solo queremos que lo haga cuando ocurra el evento.

Para saber qué botón está invocando el evento, debemos utilizar el objeto Evento creado automáticamente.

```
function crearCookie(e){  
    //Para asegurar la compatibilidad con navegadores Microsoft previos IE9  
    if(!e) e = window.event;  
    //Si el elemento que ha invocado tiene la id "crearCookie1"  
    if (e.target.id=="crearCookie1")  
        document.cookie ="docente = Carmelo";  
    else if (e.target.id=="crearCookie2")  
        document.cookie ="especialidad = SAI";}
```

No utilizamos dos `if`, para evitar que compruebe todos los casos, con `else if` no entra si el `if` se cumple.

# COOKIES - EJEMPLO SENCILLO

## *borrarCookie()*



Recuerda que para borrar una cookie, basta con agregarle una fecha de expiración previa a la actual.

```
function borrarCookie(e){  
    //Para asegurar la compatibilidad con navegadores Microsoft previos IE9  
    if(!e) e = window.event;  
    //Si el elemento que ha invocado tiene la id "crearCookie1"  
    if (e.target.id=="borrarCookie1")  
        document.cookie ="docente =;expires=Thu, 01 Jan 1970 00:00:00 UTC;";  
    else if (e.target.id=="borrarCookie2")  
        document.cookie ="especialidad =;expires=Thu, 01 Jan 1970 00:00:00 UTC;";  
}
```

# COOKIES - EJEMPLO SENCILLO

## 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/RNbdWwa>

