

DESARROLLO WEB EN ENTORNO CLIENTE

^

PRACTICA COOKIES
JAVASCRIPT

ISAAC SOLER FORES

Parámetros

Veamos cada uno de los parámetros de una cookie en más detalle.

<nombre>=<valor>

<nombre> es el nombre (key) que identifica a la cookie y <valor> es su valor. El <nombre> es obligatorio para poder crear una cookie mientras que el valor es opcional (más adelante veremos como las cookies sin valor o con valor vacío son tratadas de forma diferente en PHP y JavaScript).

Como <nombre> y <valor> se puede utilizar cualquier valor arbitrario que deseemos, sólo hay que tener en cuenta que:

- <nombre>: es definido como un **token** y como tal solo puede contener caracteres alfanuméricos más!#\$%&'*+-.^_`|~. No puede contener ningún espacio, coma o punto y coma. Tampoco están permitidos los caracteres de control (\x00, \x1F más \x7F) y no se debería utilizar el signo = que es utilizado como separador.
- <valor> puede contener cualquier valor alfanumérico excepto espacios, comas, punto y coma, caracteres de control, barra invertida y comillas dobles. En caso de que sea necesario el uso de estos caracteres, el valor de la cookie **deberá ser codificado** (reemplazar esos caracteres por su código ASCII); en JavaScript lo podemos hacer con encodeURIComponent(); en PHP se podría hacer con urlenconde(); al leer el valor habría que descodificarlo con decodeURIComponent() o urldecode().

expires=<fecha> y max-age=<segundos>

Opcional. Ambos parámetros especifican el tiempo de validez de la cookie. expires establece una fecha, que ha de estar en formato GMT/UTC (por ejemplo, Mon, 03 Jul 2006 21:44:38 UTC). max-age establece una duración máxima en segundos (si segundos). Si se especifican ambos, max-age toma preferencia (max-age no es soportado por IE 8 e inferior).

Si no se especifica ninguno de los dos, la cookie sólo es válida para la sesión actual, lo que se conoce como **session cookie** (por ejemplo, hasta Cookies que el usuario cierre el navegador). Si max-age es cero la cookie se elimina, al igual que si expires es una fecha pasada.

path=<ruta>

Opcional. Establece la ruta para la cual la cookie es válida. Algo así como los «directorios» o «secciones» de la web. Por defecto, si no se especifica ningún valor, una cookie sólo es **válida para el path actual** (el directorio que contiene la página actual). Por ejemplo, si la cookie se establece en «<https://ejemplo.com/noticias/pagina.html>», la cookie será válida para cualquier otra página del directorio «<https://ejemplo.com/noticias/>». Utilizando el parámetro path podemos establecer un directorio diferente. Por ejemplo, «/» sería para todos los directorios del dominio, incluyendo el directorio raíz; «/blog» sería sólo para páginas bajo el directorio «<http://dominio.com/blog/>».

secure

Opcional. Este parámetro no tiene ningún valor. Si está presente la cookie sólo es válida para **conexiones encriptadas** (por ejemplo, mediante protocolo HTTPS).

HttpOnly

Opcional. Este parámetro no tiene ningún valor. Si está presente, la cookie solo es accesible mediante protocolo HTTP (o HTTPS). Estas cookies no pueden ser leídas ni creadas mediante otros protocolos y APIs, por ejemplo, JavaScript.

domain=<dominio>

Opcional. Por defecto, las cookies son **válidas sólo para el subdominio actual** en el que se crea la cookie (ten en cuenta que www.ejemplo.com se considera el subdominio www).

Esto quiere decir que si el atributo domain está vacío y estamos en ejemplo.com, la cookie es válida sólo para ejemplo.com, si estamos en www.ejemplo.com, la cookie será válida sólo para www.ejemplo.com, si estamos en sub.ejemplo.com, la cookie es válida sólo para sub.ejemplo.com, o si estamos en foo.sub.ejemplo.com, la cookie es válida sólo para foo.sub.ejemplo.com.

Pero si el atributo domain no está vacío, podemos especificar otros subdominios para los que la cookie es válida. Por ejemplo, domain=sub.ejemplo.com creará una cookie válida sólo para sub.dominio.com; si estamos en otro subdominio, por ejemplo foo.ejemplo.com, e intentamos leer esa cookie, no podremos, ya que sólo era válida para sub.ejemplo.com.

También podemos hacer que una cookie sea **válida para todo el dominio y sus subdominios** estableciendo domain=.ejemplo.com (nota el punto Cookies delante del nombre del dominio). La especificación RFC 6265 antes mencionada, dice que **el primer punto será ignorado**, es decir, **domain=.ejemplo.com y domain=ejemplo.com tienen el mismo efecto y crean una cookie válida para cualquier subdominio**.

Lo anterior tiene una importante implicación. Si estamos en ejemplo.com y queremos una cookie válida sólo para ejemplo.com, la única opción es dejar el atributo domain vacío ya que si establecemos domain=ejemplo.com hacemos la cookie válida para todos los subdominios, no sólo para ejemplo.com.

Sin embargo, la especificación [RFC 2965](#) sobre la cabecera Set-Cookie2, **requiere el punto precedente** para crear una cookie válida para todos los subdominios. Además, en la especificación RFC 2109 ya obsoleta, domain=.ejemplo.com y domain=ejemplo.com no eran tratados como lo mismo.

Por estos motivos, en mi opinión, **es recomendable utilizar siempre el punto precedente si queremos crear una cookie válida para todos los subdominios** ya que garantiza el mismo comportamiento en implementaciones antiguas que puedan surgir especificaciones obsoletas, pero también en implementaciones modernas.

Por motivos de seguridad, **no se permite crear cookies para dominios diferentes** al que crea la cookie (**same-origin policy**).

- **Ejercicio 1**

Crearemos un pagina web con 5 botones, que harán las siguientes funciones:

Botón 1 → Crear cookie 1

Pulsaremos el botón y se creará la cookie 1, con el nombre y el valor preestablecidos en el script. Aparecerá un alert diciéndonos que la cookie se ha creado satisfactoriamente.

Botón 2 → Crear cookie 2

Pulsaremos el botón y se creará la cookie 2, con el nombre y el valor preestablecidos en el script. Aparecerá un alert diciéndonos que la cookie se ha creado satisfactoriamente.

Botón 3 → Mostrar cookies

Pulsaremos el botón y aparecerá un alert con todas las cookies creadas hasta el momento.

Botón 4 → Borrar cookie 1

Pulsaremos el botón y se borrará la cookie 1 . Aparecerá un alert diciéndonos que la cookie se ha eliminado satisfactoriamente.

Botón 5 → Borrar cookie 2

Pulsaremos el botón y se borrará la cookie 2 . Aparecerá un alert diciéndonos que la cookie se ha eliminado satisfactoriamente.

- **Ejercicio 2**

Vamos a mejorar el ejercicio 1, crearemos un página web con 5 botones, que harán las siguientes funciones:

Botón 1 → Crear cookie

Pulsaremos el botón y el script nos va a pedir que introduzcamos el nombre de la cookie, pulsaremos aceptar y seguidamente nos solicitará que introduzcamos el valor de la cookie, finalmente nos va a pedir los días necesarios para que expire la cookie. Aparecerá un alert diciéndonos que la cookie se ha creado satisfactoriamente.

Botón 2 → Modificar cookie

Pulsaremos el botón y el script nos va a pedir que introduzcamos el nombre de la cookie a modificar, seguidamente nos va a pedir que introduzcamos el nuevo valor de la cookie y el tiempo de expiración. Aparecerá un alert diciéndonos que la cookie se ha modificado satisfactoriamente.

Botón 3 → Mostrar cookies

Pulsaremos el botón y aparecerá un alert con todas las cookies creadas hasta el momento.

Botón 4 → Leer cookie

Pulsaremos el botón y nos pedirá que introduzcamos el nombre de la cookie para ver su valor. Aparecerá un alert con el valor de la cookie.

Botón 5 → Borrar cookie

Pulsaremos el botón y nos pedirá que introduzcamos el nombre de la cookie a eliminar. Se eliminará la cookie y aparecerá un alert diciéndonos que se ha eliminado satisfactoriamente.