



APRENDERAPROGRAMAR.COM

COOKIES JAVASCRIPT:  
¿QUÉ SON? FECHA  
EXPIRES, MAX AGE. PATH,  
DOMAIN, SECURE. CREAR,  
LEER, BORRAR, AÑADIR.  
DOCUMENT.COOKIE .  
EJEMPLOS (CU01188E)

Sección: Cursos

Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

**Resumen:** Entrega nº88 del Tutorial básico "JavaScript desde cero".

Autor: César Krall

## COOKIES CON JAVASCRIPT

Los programadores JavaScript se enfrentan con frecuencia a la aparición de errores durante la ejecución del código en los navegadores. Al igual que otros lenguajes de programación, JavaScript permite la captura de esos errores con instrucciones que permiten identificar y tratar dichos errores.



## TRASLADAR INFORMACIÓN DURANTE LA NAVEGACIÓN

Hemos visto muchos ejemplos donde trabajamos sobre un documento HTML, pero obviamente durante el desarrollo de aplicaciones reales el usuario visitará numerosos documentos HTML que se corresponderán con diferentes urls durante una navegación a través de una página web o una aplicación web.

Para situarnos, procede a hacer lo siguiente: crea un archivo html con el siguiente contenido y guárdalo con el nombre de archivo inicio.html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">
<script type="text/javascript">
var miVariableGlobal = 8;
window.onload =
alert ('En esta url mi variable global vale: ' + miVariableGlobal);
</script>
</head>
<body><div id="cabecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos JavaScript</h3></div>
<div style="color:blue;" id="pulsador" ><a href="item1.html"> Pulse aquí para visitar la sección "Animales salvajes"</a></div>
</body>
</html>
```

Seguidamente crea un documento HTML con el siguiente contenido y guárdalo con el nombre de archivo item1.html:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">
<script type="text/javascript">
window.onload = alert ('En esta url mi variable global vale: ' + miVariableGlobal);
</script>
</head>
<body><div id="cabecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos JavaScript</h3></div>
<div style="color:blue;" id="pulsador" ><a href="inicio.html"> Pulse aquí para volver</a></div>
</body>
</html>
```

Comprueba que al cargar la página inicio.html se muestra por pantalla el mensaje << En esta url mi variable global vale: 8>>. Sin embargo, cuando hacemos click en el enlace y llegamos al archivo item1.html no se muestra nada. De hecho, si desplegamos la consola del navegador observaremos que aparece un error del tipo << ReferenceError: miVariableGlobal is not defined>>. Lo que esto nos deja claro es que las variables JavaScript tienen una vida limitada al propio documento HTML en el que van insertas y su información desaparece cuando se carga un nuevo documento HTML.

Cargar un archivo común donde definamos unas variables nos puede servir para disponer de dichas variables con su valor inicial en diferentes urls, pero ¿qué ocurre si el valor de esa variable se modifica y continúa la navegación? Que no dispondremos de información sobre dicha modificación en el resto de urls, ya que “se perderán”.

Lo que vamos buscando es la persistencia de la información, es decir, que no desaparezca la información cuando cambiamos de url. Esto hay diferentes maneras de abordarlo: se puede transmitir información almacenándola en base de datos y recuperándola posteriormente, también usando formularios para enviar información, y de otras maneras. En estos ejemplos el trabajo (base de datos, recuperación de información de formularios) se basaría principalmente en lenguajes que trabajan del lado del servidor. Pero nosotros vamos buscando una forma de trabajar del lado del cliente, en este caso con JavaScript.

## CONCEPTO DE COOKIE

Las cookies fueron creadas por trabajadores de la empresa Netscape como forma de dar una respuesta sencilla a la necesidad de almacenar información relacionada con la identificación de usuarios y acciones que un usuario desarrolla durante la navegación. Por ejemplo se planteaba la siguiente cuestión: si un usuario accede a una tienda web y queremos que pueda ir agregando productos a un carrito de compras, ¿cómo saber qué productos ha almacenado cuando cambia de url? Sin el uso de alguna forma de dotar de persistencia a la información, la información se perdía. Y cierta información sería muy costoso manejarla con herramientas como bases de datos o formularios.

La forma de solucionar esto inventar las cookies. Las cookies son información que se almacena en el navegador de forma persistente, es decir, que no desaparece cuando el usuario navega a través de diferentes urls. Además, las cookies son enviadas al servidor cuando el usuario hace una petición, de modo que el servidor puede conocer esa información y actuar en consecuencia.

Las cookies pueden ser creadas de diferentes maneras, por ejemplo:

a) Ser creadas por el servidor, y enviarlas al navegador para que las almacene. Supongamos que entramos en una página web como los foros de aprenderaprogramar.com. En ese momento somos usuarios anónimos, pero una vez introduzcamos nuestro nombre de usuario y password (por ejemplo supongamos que somos el usuario Albert Einstein), el servidor envía una cookie al navegador que podría ser:

session\_id\_foros\_apr = 6n4465736gf9863b52e641757fa0b7db, donde sesión\_id\_foros\_apr es el nombre la cookie y la cadena de letras y números el valor que tiene la cookie, lo que permitirá saber al servidor que quien hace una petición es la misma persona (el mismo computador cliente) que había hecho una petición anteriormente. La comprobación de que sea el mismo computador cliente se basa en comparar el valor de esa cadena larga: si coincide, es el mismo usuario, si es diferente, es otro

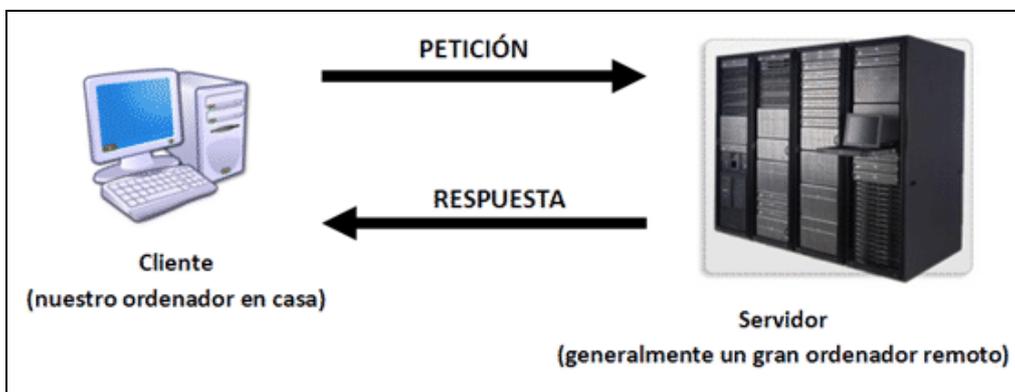
usuario. Un usuario no puede conocer el valor de la cookie que tiene otro debido a que existen prácticamente infinitas posibilidades de combinación de letras y números. Si se usan cadenas arbitrariamente largas, es prácticamente imposible averiguar por casualidad o mediante pruebas el contenido de una cookie.

Ahora cada vez que cambiemos de url esta cookie es enviada al servidor. El servidor lee la cookie y comprueba: ¿esta cookie pertenece al usuario Albert Einstein!

Aunque el usuario haya cambiado de url, la cookie informa en cada momento que él es Albert Einstein, con lo cual no hace falta estar introduciendo en cada url que se visite el nombre de usuario y password.

b) Ser creadas mediante JavaScript en el navegador, almacenarse en el navegador y enviarse posteriormente al servidor en cada comunicación que tenga lugar.

Simplificadamente podríamos decir que navegar sin cookies sería algo similar a lo que se muestra en este esquema, donde cada petición al servidor equivale a una "petición nueva":



Mientras que navegar con cookies sería algo similar a lo que se muestra en este esquema, donde el navegador está almacenando información en un almacén de cookies y con cada petición que hace esas cookies son transmitidas (y además pueden irse añadiendo más cookies a las ya existentes previamente):



Resumiendo: las cookies son datos que se almacenan en el navegador del usuario.

Las cookies pueden presentar algunos problemas de seguridad, pero en general se considera que si se utilizan de forma adecuada son un mecanismo que resulta práctico para realizar muchas de las tareas que normalmente se requieren en la navegación web.

## TIEMPO DE VIDA DE LAS COOKIES

Las cookies son datos temporales, es decir, su intención no es almacenarse “para siempre”, sino almacenarse por un tiempo para facilitar la navegación. Una cookie puede tener asociada una fecha de borrado o expiración, en cuyo caso permanecerá en el navegador del usuario hasta que llegue dicha fecha (a no ser que el usuario decida hacer un borrado de cookies). En este caso, puede ocurrir que el usuario cierre el navegador y lo abra al cabo de unas horas o días y la información en forma de cookies siga estando ahí. Las cookies con fecha de borrado se suelen llamar cookies persistentes, porque no se destruyen excepto cuando llega la fecha de expiración.

Otras cookies no tienen fecha de borrado o expiración, o si la tienen es muy corta (pongamos que una hora de duración). Si la cookie no tiene fecha de borrado, se destruye cuando se cierra el navegador.

Tener en cuenta que los navegadores pueden almacenar información de otras maneras además de como cookies (por ejemplo como opciones de configuración, perfiles de usuario, contraseñas, etc.).

Tener en cuenta también que las cookies (al igual que JavaScript) pueden desactivarse en los navegadores. La mayoría de los usuarios navegan con cookies activadas (al igual que con JavaScript activado), pero teóricamente un usuario puede deshabilitarlas.

## CONTENIDOS DE LAS COOKIES

Las cookies podemos verlas como pequeños ficheros de texto que se almacenan en el navegador, cuyo contenido es el siguiente:

1. Un par nombre – valor que define la cookie. Por ejemplo el nombre puede ser `user_name_foros_apr` y el valor `6n4465736gf9863b52e641757fa0b7db`
2. Una fecha de caducidad (en algunos casos, estará indefinida, con lo cual la cookie será borrada cuando se cierre el navegador). La fecha se expresa en tiempo UTC (ver entregas del curso donde se explican los formatos de tiempo) o como un número de segundos desde el momento actual.
3. El dominio y ruta del servidor donde la cookie fue definida, lo que permite que si existieran dos cookies con el mismo nombre se pudiera saber qué cookie corresponde a cada url que se visita. No está permitido falsear dominios, es decir, si el dominio desde el que se establece una cookie es `aprenderaprogramar.com` no se podría poner como información asociada a la cookie que el dominio es `microsoft.com`, sino que la cookie únicamente puede ir asociada a `aprenderaprogramar.com`. Esto permite que si se está navegando por un sitio, no haya necesidad de enviar todas las cookies almacenadas en el navegador al servidor de ese sitio, sino únicamente las cookies relacionadas con ese sitio.

La ruta permite especificar un directorio específico al cual se deba considerar asociada la cookie. De este modo, el navegador no tendría que enviar la cookie a todas las páginas de un dominio, sino solo a las páginas concretas cuya ruta esté definida. Normalmente la ruta definida es simplemente << / >> lo que significa que la cookie es válida en todo el dominio.

## COOKIES CON JAVASCRIPT: DOCUMENT.COOKIE

Las cookies podemos verlas como pequeños ficheros de texto que se almacenan en el navegador.

Para crear una cookie con JavaScript podemos usar una sintaxis como esta:

```
document.cookie =  
'nombreCookie=valorCookie;  
expires=fechaDeExpiración;  
path=rutaParaLaCookie';
```

Podemos dejar sin especificar expires (la cookie se borrará al cerrar el navegador) y path (la cookie quedará asociada al dominio), con lo que la definición quedaría:

```
document.cookie = 'nombreCookie = valorCookie;';
```

También se pueden indicar como parámetros de la cookie domain (dominio) y secure (en este caso a secure no se le asigna ningún valor, basta con incluir la palabra. Si se incluye, significa que la cookie sólo se enviará si la conexión se está realizando bajo un protocolo seguro como https).

En lugar de expires se puede usar max-age. En este caso, en lugar de especificar una fecha concreta se especifica el número de segundos desde la creación de la cookie hasta su caducidad. Por ejemplo: "theme=blue; max-age=" + 60\*60\*24\*30 + "; path=/; domain=aprenderaprogramar.com; secure" serviría para indicar que el nombre de la cookie es theme, su valor blue, su fecha de expiración 30 días (expresado en segundos resulta 30 días \* 24 horas/día \* 60 minutos/hora \* 60 segundos/minuto").

Cuando se van creando cookies, éstas se van añadiendo a document.cookie (es decir, document.cookie no funciona como una propiedad que se vaya sobrescribiendo, sino que cada definición de document.cookie añade una cookie a la colección de cookies).

Una cookie se puede redefinir: para ello simplemente hemos de usar document.cookie indicando el mismo nombre de cookie que uno que existiera previamente. Los datos de esa cookie serán sobrescritos quedando reemplazados los anteriormente existentes por los nuevos.

Una cookie se puede eliminar: para ello hemos de usar document.cookie indicando el nombre de cookie que queremos eliminar y establecer una fecha de expiración ya pasada (por ejemplo del día de ayer, o simplemente indicar expires=Thu, 01 Jan 1970 00:00:00 UTC;). El navegador al comprobar que la fecha de caducidad de la cookie ha pasado, la eliminará.

## RECUPERAR EL CONTENIDO DE COOKIES

Para recuperar el contenido de cookies hemos de trabajar con `document.cookie` como si fuera una cadena de texto donde las cookies se organizan de la siguiente manera:

```
nombreCookieA=valorCookieA; nombreCookieB=valorCookieB; ... ; nombreCookieN = valorCookieN;
```

Para recuperar el valor de la cookie, hemos de buscar dentro de esa cadena de texto. Para ello usaremos las herramientas que nos proporciona JavaScript, y hay varias maneras de hacerlo. A continuación vamos a ver un ejemplo.

Escribe este código y guárdalo con el nombre de archivo `inicioCookie.html`:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">
<script type="text/javascript">
var miVariableGlobal = 8;
document.cookie = 'cookieNombreUsuario = Alberto; expires=Thu, 12 Aug 2049 20:47:11 UTC; path=/';
document.cookie = 'cookieEdadUsuario='+miVariableGlobal+'; expires=Thu, 12 Aug 2049 20:47:11 UTC; path=/';
alert ('El contenido de document.cookie es: '+document.cookie);
</script>
</head>
<body><div id="cabecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos JavaScript</h3></div>
<div style="color:blue;" id="pulsador" ><a href="item1Cookie.html"> Pulse aquí para visitar la sección "Animales salvajes"</a></div>
</body>
</html>
```

Con este código hemos definido dos cookies. Al ejecutar el código el resultado esperado es que se muestren por pantalla: << El contenido de `document.cookie` es: `cookieNombreUsuario=Alberto; cookieEdadUsuario=8` >>

Escribe ahora este otro código y guárdalo con el nombre de archivo `item1Cookie.html`:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">
<script type="text/javascript">
//Función para encontrar el valor de una cookie
function readCookie(nombre) {
    var nombrelgual = nombre + "=";
    var numeroCookies = document.cookie.split(';');
    for(var i=0;i < numeroCookies.length;i++) { //Recorremos todas las cookies
        var valorCookie = numeroCookies[i]; //Analizamos la cookie actual
        while (valorCookie.charAt(0)==' ') {valorCookie = valorCookie.substring(1,valorCookie.length); }
    //Eliminamos espacios
        if (valorCookie.indexOf(nombrelgual) == 0) {return
valorCookie.substring(nombrelgual.length,valorCookie.length);} //Devolvemos el valor
    }
    return null; //Si numeroCookies es cero se devuelve null
}
}
```

```
window.onload = function () {alert ('El valor de la cookieNombreUsuario es: '+readCookie('cookieNombreUsuario'));
alert ('El valor de la cookieEdadUsuario es: '+readCookie('cookieEdadUsuario'));
};

</script></head>
<body><div id="numeroCookiesbecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos
JavaScript</h3></div>
<div style="color:blue;" id="pulsador" ><a href="inicioCookie.html"> Pulse aquí para volver</a></div>
</body></html>
```

Prueba ahora a entrar en la página inicioCookie.html y pulsar sobre el enlace que nos lleva a item1Cookie.html. Al entrar en item1Cookie.html el resultado esperado es que se muestre por pantalla: << El valor de la cookieNombreUsuario es: Alberto, El valor de la cookieEdadUsuario es: 8 >>

Vamos a destacar algunas cuestiones del código que hemos usado:

- Las cookies se crean en una url (inicioCookie.html) y se almacenan en el navegador.
- Las cookies se recuperan en otra url (item1Cookie.html) usando la función readCookie. De esta manera tenemos una solución al problema de transmitir información entre distintas urls que habíamos planteado al principio de esta explicación. La función readCookie utiliza distintas construcciones y funciones nativas de JavaScript que deberías ser capaz de entender completamente si has venido siguiendo el curso desde el principio. Si no entiendes algo de este código, repasa las entregas anteriores del curso.
- Las cookies que se reciben son sólo las asociadas al dominio. ¿Y cuál es el dominio? En este caso el asociado a la ruta de nuestro computador donde estemos trabajando. Si estuviéramos trabajando sobre una web on-line, el dominio serían todas las páginas pertenecientes al sitio web.
- Las cookies que se han creado tienen fecha de caducidad 2049. Es decir, no se borrarán del navegador hasta 2049, excepto si realizamos el borrado de cookies del navegador (al borrado de este tipo de datos se le denomina en sentido amplio "Limpieza de la caché del navegador").

## EJERCICIO

Un programador ha desarrollado este código. Revísalo y responde a las siguientes preguntas:

```
function createCookie(name,value,days) {
    if (days) {
        var date = new Date();
        date.setTime(date.getTime()+(days*24*60*60*1000));
        var expires = "; expires="+date.toGMTString();
    }
    else var expires = "";
    document.cookie = name+"="+value+expires+"; path=/";
}

function eraseCookie(name) { createCookie(name,"",-1); }
```

- a) ¿Para qué sirve la función createCookie? Explica paso a paso qué es lo que hace esta función.
- b) ¿Para qué sirve la función eraseCookie? Explica paso a paso qué es lo que hace esta función.
- c) Escribe el código de una función de nombre mostrarTodasLasCookies() que muestre el nombre y valor de todas las cookies existentes. Por ejemplo, si hay dos cookies deberá mostrarse algo como esto:

Hay 2 cookies en el documento

Cookie 1 con nombre: cookieNombreUsuario y valor: Alberto

Cookie 2 con nombre cookieEdadUsuario y valor 8

- d) Usando las funciones createCookie, eraseCookie y mostrarTodasLasCookies() crea un script que pida 3 nombres y valores de cookies al usuario, y cree las cookies correspondientes. A continuación deberá mostrar las cookies existentes y sus valores. Finalmente, deberá borrar la última cookie existente y volver a mostrar todas las cookies y sus valores.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros [aprenderaprogramar.com](http://aprenderaprogramar.com).

**Próxima entrega:** CU01189E

**Acceso al curso completo** en [aprenderaprogramar.com](http://aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:  
[http://aprenderaprogramar.com/index.php?option=com\\_content&view=category&id=78&Itemid=206](http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=78&Itemid=206)