

Utilización de los objetos predefinidos en JavaScript

Cookies

Módulo: Desarrollo Web en entornos cliente

CFGs Desarrollo de Aplicaciones Web

¿Qué es una cookie?

- Un fichero de texto muy pequeño almacenado en el cliente por una página Web.
- El nombre de una cookie se genera automáticamente por el navegador e identifica la URL que generó dicha cookie.

¿Para qué se usan?

- Habilitar el código JavaScript para almacenar datos de forma persistente en el disco duro del cliente, y para recuperar los datos que se almacenan previamente esta manera.
- Almacenamiento de datos de manera que se puede acceder por diferentes páginas de un sitio web (por ejemplo, para marcar si un usuario está conectado o no).
- Adaptación de una página web para el usuario / equipo específico.

¿En qué consiste una cookie?

- Nombre, Valor, Fecha de caducidad / Max-age, Path, Dominio, Seguridad.

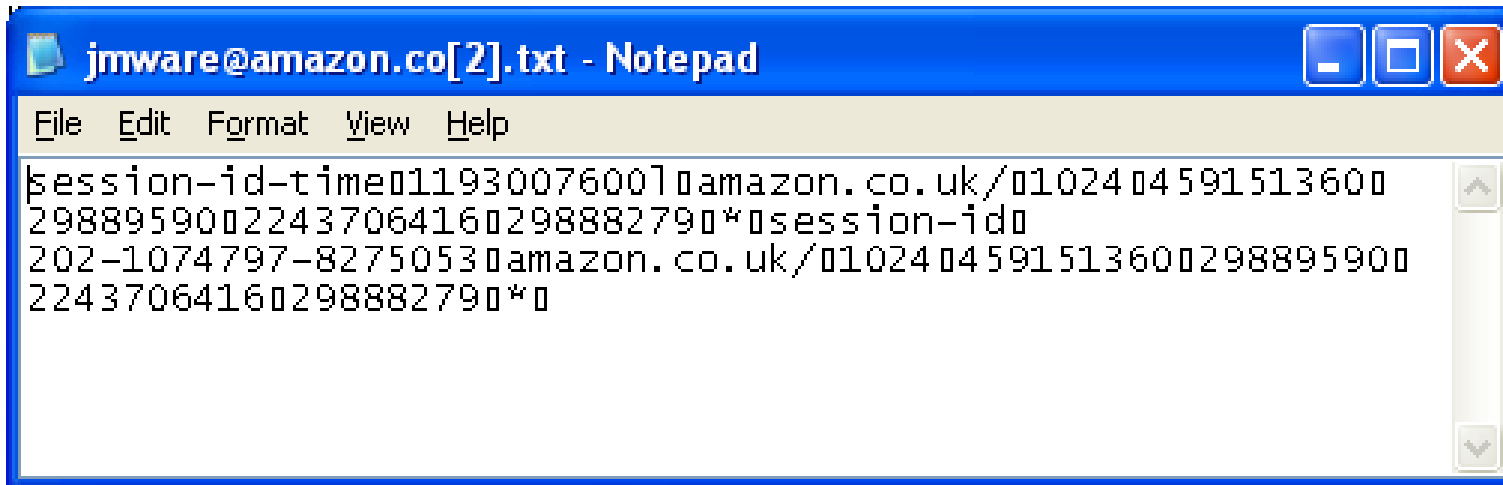
Cualquiera de los siguientes atributos opcionales puede ser escrito después del par clave-valor, afectando a la cookie a establecer o actualizar, precedidos de un punto y coma.

- `;path=path` (p. ej.: `'/'`, `'/midir'`). Si no se especifica, por defecto corresponde a la ruta del documento actual.
- `;domain=domain` (p. ej. `'example.com'`, `'.example.com'` (incluye todos los subdominios), `'subdomain.example.com'`) Si no se especifica, corresponde por defecto a la porción de la dirección web de la ubicación actual del archivo.
- `;max-age=duración-máxima-en-segundos` Por ejemplo: `60*60*24*365` para un año.
- `;expires=fecha-en-formato-GMTString` Si no se especifica expirará al terminar la sesión actual.
- `;secure` La cookie sólo será transmitida en un protocolo seguro (HTTPS, SSL).

El valor de la cookie puede ser evaluado mediante `encodeURIComponent()` para asegurarse de que dicha cadena no incluya comas, punto y coma, ni espacios en blanco (lo cual no está permitido en el valor de una cookie).

Puedes eliminar una cookie simplemente estableciendo su fecha de expiración a cero.

¿Qué aspecto tiene una cookie?

A screenshot of a Windows Notepad window. The title bar reads "jmware@amazon.co[2].txt - Notepad". The menu bar includes "File", "Edit", "Format", "View", and "Help". The text area contains two lines of a cookie file, separated by a blank line. The first line is "session-id-time0119300760070amazon.co.uk/010240459151360029889590022437064160298882790*0session-id0" and the second line is "202-1074797-82750530amazon.co.uk/010240459151360029889590022437064160298882790*0".

```
jmware@amazon.co[2].txt - Notepad
File Edit Format View Help
session-id-time0119300760070amazon.co.uk/0102404591513600
29889590022437064160298882790*0session-id0
202-1074797-82750530amazon.co.uk/0102404591513600298895900
22437064160298882790*0
```

Limitaciones

- Relacionarse a un equipo en particular y no a un usuario
- Pueden deshabilitarse
- 300 cookies por navegador
- 20 cookies por servidor
- 4 KB de datos por cookie

¿Cómo crear/borrar/acceder a las cookies?

- En JavaScript usamos el método **document.cookie**
- Hay otras formas – por ejemplo, directamente en el servidor como parte de la respuesta HTTP

Cookies de Sesión

- Si no se encuentra la fecha de vencimiento, entonces la cookie se elimina cuando el explorador se cierre. Esto se llama cookie de sesión.
- Puede ser utilizado como un medio para almacenar datos utilizado en más de una página html (por ejemplo, recordando que ya ha iniciado la sesión).
- Ejemplo sencillo de cookies de sesión - **1a abierta en un navegador**, a continuación, **1b abierto en otro navegador**

```
<html>
<head>
<title>Session Cookie 1a </title>

</head>

<body>

<script>
document.cookie="logged_in=yes"
</script>


</body>
</html>
```

```
<html>
<head>
<title>Session Cookie 1b </title>
</head>
<body>

<script>
index = document.cookie.indexOf("logged_in=yes");
if (index == -1)
    document.write("Not logged in")
else {
    document.write("Your bank account details are:<br>")
    document.write("Current account: OVERDRAWN<br>")
    document.write("Deposit account: OVERDRAWN")
}
</script>


</body>
</html>
```

```
<body>
<script language="javascript">
1) var YourUserID = getCookie("UserID");
2) if (YourUserID == null) {
    YourUserID = prompt ("Por favor introduce tu UserID", "");
    setCookie("UserID",YourUserID,30);
}
else
3) alert ("Hola usuario " + YourUserID + "\nBienvenido");
</script>
</body>
```



- 1) Comprueba si el archivo de cookies contiene una cookie con el nombre de "UserID". Esto se hace llamando a la función **getCookie**.
 - Si no hay un archivo de cookies que corresponde entonces se devuelve un valor NULL.
 - Si existe el archivo de cookies, pero la cookie UserID no existe, entonces se devuelve un valor NULL.
 - Si la cookie de UserID no existe, entonces el valor asociado con el UserID se asigna a la variable YourUserID.


```
<body>
<script language="javascript">
1) var YourUserID = getCookie("UserID");
2) if (YourUserID == null) {
    YourUserID = prompt ("Por favor introduce tu UserID:", "");
    setCookie("UserID",YourUserID,30);
}
else
3) alert ("Hola usuario " + YourUserID + "\nBienvenido");
</script>
</body>
```



2) Si la cookie no existe, getCookie devuelve un valor de null, en cuyo caso:

- Se pide al usuario el número identificador, y se asigna a la variable YourUserID.
- la cookie “UserID” se crea y se le asigna el valor de name=YourUserID y caduca a los 30 días a partir de ahora. Esto se hace mediante la función setCookie.


```
<body>
<script language="javascript">
1) var YourUserID = getCookie("UserID");
2) if (YourUserID == null)
    {
        YourUserID = prompt ("Por favor introduce tu UserID ", "");
        setCookie("UserID",YourUserID,30);
    }
    else
3)      alert ("Hola usuario " + YourUserID + "\nBienvenido");
</script>
</body>
```



3) Si el usuario ha visitado la página antes, entonces la cookie existe y se le muestra al usuario un mensaje de bienvenida personalizado

```
function getCookie(name) {  
    1) var index = document.cookie.indexOf(nombre+"=");  
    2) if (index == -1) return null;  
    3) index = document.cookie.indexOf("=", index) + 1;  
       var endstr = document.cookie.indexOf(";", index);  
       if (endstr == -1)  
           endstr = document.cookie.length;  
       return decodeURIComponent(document.cookie.substring(index, endstr));  
}
```

- 1) Se busca el fichero cookie asociado – empleando el método `indexOf`– para un nombre de cookie correspondiente almacenado en la variable **nombre** (en este ejemplo contiene el string “UserID”) – observa que a `document.cookie` se le trata como un string.
- Si se encuentra la cadena (es decir, la cookie existe), entonces `indexOf` devuelve la posición del primer carácter de la cadena.
 - Si el archivo de cookies no existe o si no se encuentra la cadena (es decir, la cookie no existe) devuelve el valor -1.

```
function getCookie(nombre) {  
    1) var index = document.cookie.indexOf(nombre+"=");  
    2) if (index == -1) return null;  
    3) index = document.cookie.indexOf("=", index) + 1;  
        var endstr = document.cookie.indexOf(";", index);  
        if (endstr == -1)  
            endstr = document.cookie.length;  
        return decodeURIComponent (document.cookie.substring(index, endstr));  
}
```

2) Si la cookie no se encontró la función devuelve un **null**.

```
function getCookie(name) {  
    1) var index = document.cookie.indexOf(name+"=");  
    2) if (index == -1) return null;  
    3) index = document.cookie.indexOf("=", index) + 1;  
       var endstr = document.cookie.indexOf(";", index);  
       if (endstr == -1)  
           endstr = document.cookie.length;  
       return decodeURIComponent (document.cookie.substring(index, endstr));  
}
```

3) Recupera el valor de la cookie en particular y devuelve el valor al código de llamada.

```
function setCookie(name, value, days)
{
  1) var expdate = new Date();
  2) expdate.setTime(expdate.getTime() + days*24*60*60*1000);
  3) document.cookie = name+"="+encodeURIComponent (value)+"";
    expires=" + expdate.toUTCString();
}
```

La función *setCookie* recibe 3 parámetros - el nombre de la cookie, el valor de la cookie y la fecha de caducidad (en términos de número de días a partir de hoy).

```
function setCookie(name, value, days)
{
  1) var expdate = new Date();
  2) expdate.setTime(expdate.getTime() + days*24*60*60*1000);
  3) document.cookie = name+"="+encodeURIComponent (value)+"";
    expires=" + expdate.toUTCString();
}
```

1) Se crea una nueva instancia del objeto Date.

```
function setCookie(name, value, days)
{
  1) var expdate = new Date();
  2) expdate.setTime(expdate.getTime() + days*24*60*60*1000);
  3) document.cookie = name+"="+encodeURIComponent (value)+"";
    expires=" + expdate.toUTCString();
}
```

2) Se calcula la fecha de expiración de la Cookie (en milisegundos desde la fecha actual).

```
function setCookie(name, value, days)
{
  1) var expdate = new Date();
  2) expdate.setTime(expdate.getTime() + days*24*60*60*1000);
  3) document.cookie = name+"="+encodeURIComponent (value)+"";
    expires=" + expdate.toUTCString();
}
```

3) La cookie es asignada – esto es, se forma una cadena con el nombre de la cookie, el valor de la cookie y la fecha de caducidad de la cookie.

Ejemplo de Cookie

```
<html>
<head>
<title>Cookie Example</title>
<script language="javascript">
```

```
function setCookie(nombre,value,dias) {
    var expdate = new Date();
    expdate.setTime(expdate.getTime() + dias*24*60*60*1000);
    document.cookie = nombre+"="+encodeURIComponent (value)+"; expires=" + expdate.toUTCString();
}
```

```
function getCookie(nombre) {
    var index = document.cookie.indexOf(nombre+"=");
    if (index == -1)
        return null;
    index = document.cookie.indexOf("=", index) + 1;
    var endstr = document.cookie.indexOf(";", index);
    if (endstr == -1)
        endstr = document.cookie.length;
    return decodeURIComponent(document.cookie.substring(index, endstr));
}
```

```
</script>
</head>
```

```
<body>
<script language="javascript">
    var tuNIF= getCookie("Beneharo");
    if (tuNIF == null) {
        tuNIF = prompt ("Por favor, introduce tu NIF", "");
        setCookie("Beneharo",tuNIF,30);
    }
    else
        alert ("Hola usuario" + tuNIF + "\nBienvenido");
</script>
</body>
<html>
```

- Crear un script que cuente el número de visitas a la página con una cookie que haga de contador
- Añadir al script anterior un enlace que nos muestre el número de visitas que llevamos
- Añadir al script una nueva cookie que almacene el nombre del visitante y mostrarlo. Si el nombre ya existe no se debe de solicitar.
- Añadir un enlace que elimine las cookies creadas.