

INTRODUCCIÓN A LA PROGRAMACIÓN CON PHP

COOKIES

PRÁCTICA N° 7

Ejemplo:

Vamos a crear una página que puede configurarse con distintos estilos CSS.

El usuario es quien decide qué aspecto desea que tenga la página, por medio de un formulario.

Luego la página es capaz de recordar, entre los distintos accesos que realice el usuario, el aspecto que había elegido para mostrar la web.

Para hacer memoria...



¿A qué se llama cookies?

Las cookies son pequeñas informaciones de texto que se pueden almacenar en el navegador del visitante, siempre que este tenga configurado el browser para aceptarlas.

Las cookies son unas informaciones almacenadas por un sitio web en el disco duro del usuario.

¿Qué pueden almacenar las cookies?

Las cookies sólo pueden almacenar caracteres, pero aun así resultan de vital importancia para recordar estados o variables de una visita a otra del usuario.

Esta información es almacenada en un archivo tipo texto que se guarda cuando el navegador accede al sitio web.

Es interesante poder guardar información de los usuarios como nombre, edad, color preferido, etc., en el propio ordenador del mismo usuario.

Luego en nuestros ordenadores se guardan muchos datos que necesitan conocer las páginas web cada vez que entramos en la página, estas pequeñas informaciones son las cookies: **estados de variables que se conservan de una visita a otra en el ordenador del cliente.**

Como es un poco peligroso que las páginas web a las que accedemos se dediquen a introducir cosas en nuestro ordenador, las cookies están altamente restringidas.

Solamente podemos guardar en ellas textos, nunca programas, imágenes, etc.

Además, los textos **nunca podrán ocupar más de 1 K,** de modo que nadie podría inundarnos el ordenador a base de cookies.

Estas restricciones, unidas a la necesidad de **poner una fecha de caducidad** a las cookies para que estas se guarden, hacen que el aceptar cookies no signifique un verdadero problema para la integridad de nuestros sistemas



Ejemplos de uso de las cookies

Un ejemplo típico de las cookies podría ser un **contador de las veces que accede un usuario a una página**. Podríamos poner una cookie en el ordenador del cliente donde tendríamos una variable que lleva la cuenta de las veces que ha accedido a la página y cada vez que se accede se incrementa en uno.



Otro ejemplo típico es el que **guarda el perfil del usuario**. Si el usuario accede a contenidos determinados podemos enviarle una cookie que le marca como interesado en un tema. A medida que va accediendo a distintos sitios le vamos encasillando como joven, adulto, hombre, mujer, o lo que proceda. Conociendo el perfil de un usuario se le pueden ofrecer un tipo de productos o servicios orientados a sus gustos o necesidades.

¿Cómo se utilizan las cookies?

Para trabajar con cookies tenemos que utilizar un lenguaje de programación avanzado como Javascript o ASP, PHP, etc.

No podemos entonces trabajar con cookies si solamente nos dedicamos a utilizar el HTML.

¿Qué utilidad tienen las cookies?

La utilidad principal de las cookies es la de poder identificar al navegador una vez éste visita el sitio por segunda vez y así, en función del perfil del cliente dado en su primera visita, el sitio puede adaptarse dinámicamente a sus preferencias (lengua utilizada, colores de pantalla, formularios rellenos total o parcialmente, redirección a determinadas páginas...).

Polémica de las cookies

Existe con las cookies el problema que impiden la privacidad.

Lo que señalábamos acerca de guardar el perfil de un usuario puede llegar a ser un problema, dado que de alguna manera se está vigilando y apuntando cada uno de los movimientos de los usuarios, lo cual puede convertirse en un abuso de información que no tiene por qué pertenecer a nadie más que al usuario mismo.

Para crear **cookies con PHP**, modificar o generar una nueva cookie lo podemos hacer a partir de la función SetCookie:

```
setcookie("nombre_de_la_cookie",valor,expiracion);
```

Ejemplo: supongamos que queremos introducir en una variable cookie el nombre del visitante.

El nombre ha podido ser previamente recogido por un formulario:

```
setcookie("persona",$nombre,time()+86400*365);
```

Creamos una cookie php llamada *persona* que tiene como *valor el contenido de la variable \$nombre* y tendrá una duración de 1 año a partir de su creación (el tiempo time() actual en segundos sumado a un año en segundos).

Es importante que la creación de la cookie sea previa a la apertura del documento HTML.

Es decir, las llamadas a la función `setcookie()` deben ser colocadas antes de la etiqueta HTML.

Es interesante señalar que el hecho de que definir una cookie ya existente implica el borrado de la antigua.

Luego el crear una primera cookie conlleva la generación automática del archivo texto.

Para utilizar el valor de la cookie en nuestros scripts tan sólo tendremos que llamar la variable que define la cookie.

Hay que tener cuidado sin embargo de no definir variables en nuestro script con el mismo nombre que las cookies puesto que PHP privilegiará el contenido de la variable local con respecto a la cookie y no dará un mensaje de error (difícil de detectar)

Es posible recopilar en una variable tipo array el conjunto de cookies almacenadas en el disco duro del internauta mediante la variable de servidor `$HTTP_COOKIE_VARS`

Las cookies son una herramienta muy interesante para personalizar una página pero hay que ser cautos ya que, por una parte, no todos los navegadores las aceptan y por otra, se puede deliberadamente impedir al navegador la creación de cookies.

Por lo tanto resultan un complemento y no una fuente de variables infalible para el sitio.

Crear cookies en PHP

En **PHP las cookies se controlan por medio de una función**, que nos sirve para generarlas y guardarlas en el navegador del usuario. Dicha función, como ya se explicó en diapositivas anteriores es **setcookie()**, que recibe varios parámetros, entre ellos, el nombre de la cookie, el valor y la caducidad.

El único parámetro obligatorio es el nombre de la cookie, los demás son opcionales

Lista de parámetros de setcookie():

Nombre: string con el nombre que queremos darle a la cookie a guardar.

Valor: cadena de caracteres que es el valor que va a tener la cookie.

Caducidad: es un timestamp con el valor de la fecha en la que **caducará la cookie**. Lo normal es utilizar la función `time()`, que genera el timestamp actual y sumarle el número de segundos que queramos que dure la cookie.

Ruta: camino o ruta donde la cookie se podrá utilizar dentro del dominio. Por defecto, la cookie se podrá utilizar en el directorio donde se ha creado y sus subdirectorios. Si indicamos "/" la cookie tendrá validez dentro de todo el dominio.

Dominio: subdominio donde se podrá acceder a la cookie. Las cookies sólo se pueden generar y utilizar para el dominio de la página donde está colocado el script, pero podemos hacerlo visible para todos los subdominios del dominio de la web .

Seguro: booleano que, si es true, indica que la cookie sólo puede ser transmitida por `https` (`http` seguro).

Sólo http: booleano que sirve para indicar que la cookie sólo puede ser accedida por medio de las cabeceras del `http`, lo que la haría inalcanzable para lenguajes de script en el cliente como Javascript. Este parámetro fue añadido en PHP 5.2.0

La function setcookie() de PHP genera y envía la cookie al navegador y devuelve un booleano, si es true indica que se pudo incluir en el navegador del usuario y si es false indica que no ha podido colocarla en el sistema.

Pero este valor no indica que luego el visitante la haya aceptado o no, puesto que el navegador puede haberlo configurado para no aceptar cookies y esto no lo puede detectar setcookie() directamente.

Ejemplos de diferentes llamadas a setcookie():

```
setcookie("migalleta", "mivalor");  
setcookie("cookie2", "mivalor2", time() + 3600);
```

Para enviar una cookie al navegador se debe hacer antes de haber enviado las cabeceras del http al cliente, es decir, antes de haber escrito cualquier texto en la página. Si no, PHP podrá lanzar un error de headers already sent (cabeceras ya enviadas).

Recuperar cookies con PHP

Para recibir las cookies que el navegador del usuario pueda tener creadas en el sistema se utiliza el array asociativo \$_COOKIE. En este array están todas las cookies que tiene disponible la página PHP en el dominio y el directorio donde está colocado.

Por medio del nombre de la cookie accedemos a su valor:

```
$_COOKIE["migalleta"];  
$_COOKIE["cookie2"];
```

■ Ejercicio 1

- *Vamos a crear una página que puede configurarse con distintos estilos CSS*
- *El usuario es quien decide qué aspecto desea que tenga la página, por medio de un formulario*
- *Luego la página es capaz de recordar, entre los distintos accesos que realice el usuario, el aspecto que había elegido para mostrar la web*

Ejemplo de uso de cookies en PHP para almacenar la hoja de estilos css que queremos utilizar

Aquí puedes seleccionar el estilo (color de fondo) que deseas la página:

Verde ▾ Actualizar el estilo

Verde
Rosa
Negro



■ El formulario HTML para seleccionar el estilo

```
<form action="taller-cookies-php.php" method="post">  
  Aquí puedes seleccionar el estilo (color de fondo) que deseas la  
  página:  
  <br>  
  <select name="estilo">  
    <option value="verde">Verde  
    <option value="rosa">Rosa  
    <option value="negro">Negro  
  </select>  
  <input type="submit" value="Actualizar el estilo">  
</form>
```

- Recibir el formulario y crear la cookie: si recibimos un valor del formulario tendríamos que crear una cookie para recordar qué estilo desea el usuario para visualizar la página

```
//Veo si recibo datos del formulario
if(isset($_POST["estilo"])){
    //es que estoy recibiendo un estilo nuevo, lo tengo que meter en
    las cookies
    $estilo = $_POST["estilo"];
    //meto el estilo en una cookie
    setcookie("estilo", $estilo, time() + (60 * 60 * 24 * 90));
}
```


- La cookie la guardamos con el nombre "estilo" y con el valor captado del formulario. Hemos configurado la cookie para que permanezca en el ordenador del usuario 90 días
- Ahora, si la página no ha recibido por post el estilo con el que se debe mostrar, tendremos que acceder al array `$_COOKIE` para ver si está creada la correspondiente galletita y saber el estilo que el visitante había configurado en anteriores accesos

```
}else{  
    //si no he recibido el estilo que desea el usuario en la página,  
    miro si hay una cookie creada  
    if (isset($_COOKIE["estilo"])){  
        //es que tengo la cookie  
        $estilo = $_COOKIE["estilo"];  
    }  
}
```

- En cualquier caso, el estilo lo hemos guardado en una variable global al script PHP \$estilo, que la vamos a utilizar para crear el enlace con la hoja de estilos CSS deseada

```
if (isset($estilo)){  
    echo '<link rel="stylesheet" type="text/css" href="' . $estilo .  
    '.css">';  
}
```

- Si hay un estilo definido en \$estilo, hacemos la inclusión del mismo con la correspondiente etiqueta HTML

■ Finalmente, es necesario definir los distintos estilos

verde.css

```
body{
    background-color: Green;
    color: #ffffff;
    font: 10pt Verdana, Geneva, Arial, Helvetica, sans-serif;
}
```

rosa.css

```
body{
    background-color: #ffdddd;
    color: #666666;
    font: 10pt "Times New Roman", Times, serif;
}
```

negro.css

```
body{
    background-color: #000000;
    color: #ffffff;
    font: 12pt Garamond, arial;
}
```

¿dudas?

