

MANEJO DE COOKIES

Programando PHP



¿Qué son las cookies?

Introducción

- Las **cookies** son un mecanismo por el cuál se almacenan información en el cliente para así rastrear a usuarios que vuelven.
- Esta información se guarda en un archivo de tipo texto en el disco duro del ordenador desde el que se accede al sitio Web.
- Al identificar al usuario que accede por segunda vez, el sitio puede adaptarse dinámicamente a sus preferencias: idioma, colores, formularios rellenos...

Introducción

- Por el **peligro** que conlleva que las páginas Web a las que accedemos puedan guardar cosas en nuestro ordenador, las cookies están altamente restringidas.
 - ▣ Sólo se puede guardar texto (ni programas, ni imágenes, etc)
 - ▣ No podrán ocupar más de 1K
 - ▣ Deben tener fecha de caducidad

Ejemplo típico

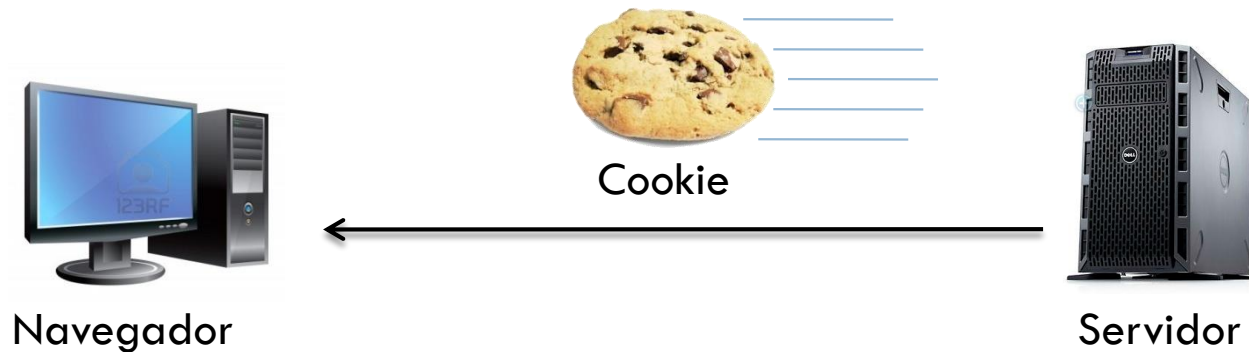
- Un ejemplo típico de las cookies podría ser un **contador de las veces que accede un usuario a una página.**
- Podríamos poner una cookie en el ordenador del cliente donde tendríamos una variable que lleva la cuenta de las veces que ha accedido a la página.
- Cada vez que se accede se incrementa en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



Funcionamiento de una Cookie

Funcionamiento de una Cookie

- 1 . La cookie se envía al navegador desde el servidor.



Si el navegador la acepta, se queda en él.

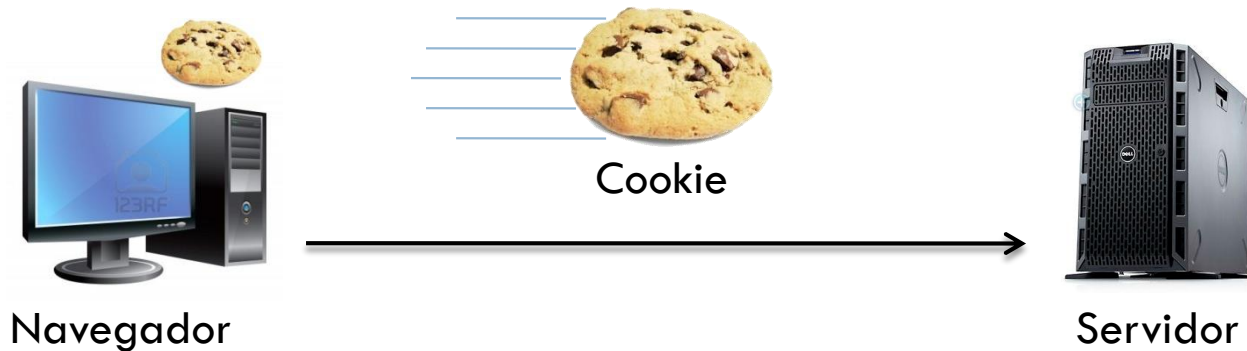
Funcionamiento de una Cookie

2. En próximos accesos, el sitio Web solicita la cookie al navegador



Funcionamiento de una Cookie

3. El navegador la envía, junto con la información de usuario.



El Servidor ya puede utilizarla como variable local

Funcionamiento de una Cookie

- Las cookies se envían junto con el resto de cabeceras de HTML
- No puede haber **nada antes** del envío de una cookie, ni siquiera:
 - `<html>`
 - `<head>`
 - Espacios en blanco
- Una vez enviada la cookie, se puede acceder a ella en la próxima carga de la página.



Funciones de manejo de Cookies

Funciones de manejo de cookies

setcookie(nombre, valor, expiración, path)

- **Nombre:** Nombre de la cookie. Si creamos una cookie solamente con el nombre, en el cliente se eliminara la cookie que exista con ese nombre.
- **Valor:** Valor que almacenará la cookie en el cliente.

Funciones de manejo de cookies

- **Expiración:** indica la hora en que se eliminara la cookie. Siempre en formato timestamp.
 - ▣ Normalmente se usa `time() + N`.
 - ▣ Siendo N el número de segundos que queremos que esté viva la cookie
- **Path:** Ruta dentro del servidor donde estará la cookie disponible.
 - ▣ Cuando una cookie se crea sólo está disponible para los archivos que se encuentren en el mismo directorio en el que se creó
 - ▣ Si en la ruta ponemos '/', la cookie estará disponible para todo el sitio.

Incisos

- Es importante recordar:
 - ▣ El código de creación de cookies deberá ir antes de cualquier etiqueta HTML
 - ▣ Crear por primera vez una cookie implica la generación automática del archivo de texto.
 - ▣ El hecho de definir una cookie ya existente implica el borrado de la antigua.

Ejemplo

- Crear una cookie llamada “migalleta” y que tenga como contenido el valor 49:

```
<?php
    setcookie("migalleta", 49);
?>
```

- Si al crear una cookie, no indicamos caducidad, ésta caducará al finalizar la sesión.
- Simplemente con ejecutar este documento, se crea la cookie en el navegador:

Ejemplo

Migalleta

^

×

Nombre	Migalleta
Contenido	49
Dominio	localhost
Ruta	/
Enviar para	Cualquier tipo de conexión
Accesible para secuencia de comandos	Si
Creada	lunes, 18 de diciembre de 2017, 9:37:11
Caduca	Al finalizar la sesión de navegación

Ejemplo

- ❑ Crear una cookie que tenga como nombre “persona” como contenido el nombre del usuario y que tenga como caducidad un año desde su creación:

```
<?php  
  
$nombre = 'Julia';  
setcookie('persona', $nombre, time() + (365*24*60*60));  
  
?>
```

Ejemplo

← localhost ha almacenado datos de forma local ELIMINAR TODO

Migalleta	▼	×
persona	^	×
Nombre persona		
Contenido Julia		
Dominio localhost		
Ruta /		
Enviar para Cualquier tipo de conexión		
Accesible para secuencia de comandos Sí		
Creada lunes, 18 de diciembre de 2017, 9:38:25		
Caduca martes, 18 de diciembre de 2018, 9:38:25		



Recuperación de Cookies

Recuperación de Cookies

- Para utilizar el valor de una cookie en siguientes accesos, sólo tendremos que acceder a la variable que define la cookie.
- Hay que tener cuidado con no crear variables que tengan el mismo nombre que la cookie ya que siempre se utilizará primero la variable.
- Todas las cookies que existan en el ordenador del cliente se almacenan en el array asociativo `$_COOKIE`

Recuperación de Cookies

- Sólo tendremos que acceder a la posición que deseemos del array:

```
<?php
    echo "la cookie 'persona' tiene el valor: ";
    echo $_COOKIE['persona'];
?>
```

la cookie 'persona' tiene el valor Julia

Recuperación de Cookies

- Para saber si una cookie existe o no, podemos utilizar la función **isset**
- **isset** determina si una variable existe y tiene valor.
 - ▣ Devuelve true en el caso de que la variable exista
 - ▣ Devuelve false en caso de que no esté definida o su contenido sea NULL



Ejercicio

Ejercicio

- ❑ Crear una página Web que permita al usuario registrarse. Cuando el usuario se registra le pide su nombre y su edad.
- ❑ La página mostrará después un saludo al usuario. En caso de que ya esté registrado utilizará los datos de la última visita.


```
<?php
    if(isset($_COOKIE['nombre']))
    {
        echo "Hola $_COOKIE[nombre] tienes $_COOKIE[edad]";
    }
    else
    {
        if(!isset($_POST['enviar']))
        {
            echo "<form action='#' method='post'>
                Nombre:<input type='text' name='nombre'>
                <br>
                Edad: <input type='text' name='edad'>
                <br>
                <input type='submit' name='enviar'>
            </form>";
        }
        else
        {
            echo "Hola $_POST[nombre] tienes $_POST[edad]";
            setcookie('nombre', $_POST['nombre'], time()+3600, '/');
            setcookie('edad', $_POST['edad'], time()+3600, '/');
        }
    }
?>
```

MANEJO DE COOKIES

Programando PHP