

# PHP: Cookies

# Cookies: qué son

- Una cookie es una pequeña cantidad de información (no más de 4KB) definida por el servidor y enviada al navegador del cliente en donde la guardará hasta alcanzar su fecha de caducidad.
- Cuando el navegador solicita una página a ese sitio web, enviará automáticamente todas las cookies relacionadas con ese servidor.
- Están vinculadas a una combinación de ordenador-navegador, lo que proporciona 'indirectamente' una vinculación con el usuario.

# Cookies: qué son

- Son otra forma de preservar el estado de una aplicación entre peticiones del navegador. Recuerda que antes vimos cadenas de consulta y campos ocultos en formularios).
- Permiten guardar información del usuario durante más tiempo.
- Actualmente las cookies también se pueden definir en el navegador mediante un lenguaje como, p.e., javascript, (Ver `document.cookie`).

# Cookies: leyendas

## Ideas equivocadas

Desde su introducción en Internet han circulado ideas equivocadas acerca de las cookies:

- Las cookies son similares a gusanos y virus, pueden borrar datos de los discos duros de los usuarios.
- Las cookies son un tipo de spyware, porque pueden leer información personal almacenada en el ordenador de los usuarios.
- Las cookies generan popups.
- Las cookies se utilizan para generar spam.
- Las cookies solo se utilizan con fines publicitarios.

# Cookies: transporte

Las cookies se envían como cabeceras HTTP (headers).  
Ejemplo de cookie generada en el servidor:

Set-Cookie: fontSize=3; expires=Tuesday, 6-Jan-2016 17:50 GMT; path=/;  
domain=.ejemplo.com; HttpSecure; httpOnly

# Cookies: componentes

Son los siguientes:

	Campo	Descripción
1	name	Requerido. Nombre de la cookie. Es como el nombre de campo de un formulario o una clave en una tabla asociativa.
2	value	Opcional. Valor de la cookie. Similar al valor de un campo de un formulario o al valor en una tabla asociativa.
3	expires	Opcional. Tiempo de expiración de la cookie, y que superado, se elimina del navegador no enviándose al servidor en peticiones futuras. Si se establece a 0, u omite, la cookie dura mientras se ejecuta el navegador, y se elimina automáticamente cuando lo cerramos.
4	path	Opcional. Ruta de acceso para la que el navegador devolverá la cookie. Si se especifica, el navegador, solamente enviará la cookie a las URL que contengan esa ruta. Por ejemplo, si se especifica la ruta <i>/admin/</i> , solo scripts en la carpeta <i>admin</i> (y subcarpetas) recibirán la cookie. Si no se especifica un valor, se asume el directorio actual del script. Generalmente es una buena idea especificar una ruta. Especificando un valor de <i>'/'</i> estará disponible en todas las URL de su sitio web.

# Cookies: componentes

## Continuación:

	Campo	Descripción
5	domain	<p>Opcional. Por defecto, un navegador envía una cookie de vuelta al mismo servidor que la envió.</p> <p>Añadiendo al dominio un punto ('.') como prefijo, se enviará la cookie a todas las URL de ese dominio, incluidos sus subdominios.</p>
6	secure	Opcional, si está presente, indica que la cookie se enviará solo si el navegador estableció una conexión segura (https) con el servidor.
7	HttpOnly	Opcional, si está presente, le dice al navegador que los datos de la cookie no serán accesibles vía javascript. Esto reduce la vulnerabilidad de la aplicación en ataques XSS (cross-site-scripting).

# Cookies: componentes

- Aunque solo se requiere el argumento “name”, es recomendable proporcionar al menos, *name*, *value*, *expires* y *path* para evitar cualquier ambigüedad.
- *Expires* debe estar en formato de marca de tiempo UNIX, es decir, como el número de segundos transcurridos entre la medianoche del 1 de enero de 1970 (en el huso horario UTC) y la fecha/hora a representar. PHP proporciona funciones para calcular este valor, p.e., **time()**.



# Cookies: creación

Para crear una cookie usaremos la función PHP `setcookie()`. Ejemplo:

```
setcookie("fontSize", 3, time() + 60 * 60 * 24 * 365, "/", ".ejemplo.com",  
false, true);
```

La función `setcookie()` debe ser llamada antes de enviar cualquier resultado al navegador (al igual que cualquier cabecera).

# Cookies: acceso

Las cookies recibidas se guardan en el servidor en la variable superglobal `$_COOKIE`.

Esta superglobal contiene la lista de todas las cookies enviadas por el navegador en la petición actual, identificadas por el 'name' de la cookie. Ejemplo:

```
echo $_COOKIE['paginasVistas'];
```

NOTAS:

- 1) accesibles también en `$_REQUEST`, aunque depende de la configuración de la directiva `'request_order'` en `php.ini`. Ver también directiva `'variables_order'`.
- 2) Una cookie recién creada no estará disponible para los scripts vía `$_COOKIE` hasta la siguiente petición del navegador.
- 3) De modo similar, si actualizamos una cookie, la tabla `$_COOKIE` mantiene el valor antiguo hasta que el navegador solicite nuevamente la página.

# Cookies: eliminación

Utilizaremos la misma función que para crearla “setcookie()” con el mismo “name”, cualquier valor (incluso una cadena vacía), y con un “expires” anterior al momento actual. Ejemplo:

```
setcookie("fontSize", 3, time() - 3600, "/", ".ejemplo.com", false, true);
```

## NOTAS:

- 1) Debe pasarse exactamente los mismos valores que cuando se creó, excepto “expires” y “value”, para garantizar que se elimina correctamente.
- 2) Eliminar una cookie no la elimina de la tabla \$\_COOKIE mientras el script se está ejecutando. Sin embargo, la siguiente vez que el navegador visite la página, no se enviará la cookie al servidor y no se creará, por tanto, el elemento de tabla \$\_COOKIE correspondiente.

# Cookies: edición

Es posible editar una cookie previa instalación de la correspondiente extensión; disponible al menos en los principales navegadores.

Googlea un poco y la encontrarás.

# Cookies: enlaces

## **Para repasar:**

<http://desarrolloweb.dlsi.ua.es/cookies-que-son-y-para-que-sirven>

# PHP: Sesiones

# Sesiones: qué son

- Las sesiones nos permiten almacenar datos en el servidor que pueden utilizarse entre peticiones.
- PHP asocia a estos datos una cadena generada por el motor PHP (Identificador de sesión) conocido como SID y que es único, aleatorio y casi imposible de adivinar.
- Cuando iniciamos sesión, php enviará al navegador una cookie con este SID para que lo almacene.
- Cuando el navegador solicita una url de este sitio, envía esta cookie llamada PHPSESSID (predeterminado) al servidor, permitiendo recuperar los datos guardados en la sesión.
- Esta información se guarda (normalmente) en un fichero temporal ubicado en el directorio indicado por la directiva ***session.save\_path***.

# Sesiones: creación

Para crear una sesión usaremos la función PHP `session_start()`. Ejemplo:

```
session_start();
```

Al igual que `setcookie()`, debe ser llamada antes de enviar nada al navegador ( o puede generar un error).

Si es una sesión nueva, esta función genera un único **SID** para la sesión y la envía al navegador como una cookie denominada **PHPSESSID** (nombre predeterminado).

Pero si el navegador ya había enviado un cookie **PHPSESSID** al servidor, porque existe ya la sesión, `session_start()` utiliza esa sesión creada previamente.



# Sesiones: acceso

Las sesiones se guardan en el servidor en la variable superglobal `$_SESSION[ ]`.

Los datos son almacenados como pares clave/valor, de modo similar a los arrays.

```
$_SESSION['username'] = 'admin';
```

Admiten cualquier tipo de dato, incluso tablas y objetos. En caso de almacenar un objeto hay que incluir la definición de clase antes de recuperar los objetos de la clase.

# Sesiones: almacenamiento

Predeterminadamente, PHP almacena los datos de cada sesión en un archivo temporal en el servidor.

La ubicación de estos archivos temporales se especifica en la directiva ***session.save\_path*** del archivo de configuración PHP.

Podemos consultar ese valor con:

```
echo ini_get("session.save_path");
```

# Sesiones: eliminación--1

Las sesiones se eliminan automáticamente cuando el usuario sale del navegador, porque el campo *expire* de la cookie PHPSESSID está establecido a 0.

Para destruir una sesión inmediatamente llamaremos a la función

`session_destroy();`

Esto eliminará exclusivamente los datos de sesión del disco. Los datos continuarán en la tabla `$_SESSION` hasta que la ejecución actual del script termine.

# Sesiones: eliminación--2

Por tanto, para asegurarnos de que los datos sean eliminados, también deberemos inicializar la tabla \$\_SESSION.

```
$_SESSION = array();
```

```
session_destroy();
```

# Sesiones: eliminación--3

Aún así, queda un rastro de la sesión en el navegador del usuario, a través de la cookie PHPSESSID.

**Conclusión:** para eliminar una sesión tanto en el servidor como en el navegador, además, debemos destruir la cookie de sesión.

```
if (isset ($_COOKIE[session_name()])) {  
    setcookie( session_name(), "", time()-3600, "/");  
    unset($_COOKIE[session_name()]);  
}  
  
session_unset(); // o bien $_SESSION = array();  
session_destroy();
```

NOTA: session\_name() devuelve el nombre de la cookie de sesión (el predeterminado es PHPSESSID).

# Sesiones: directivas php.ini

## Modifican el comportamiento de una sesión:

Directiva	Descripción
session.cookie_lifetime	Si utilizas la URL para propagar el SID, éste se perderá cuando cierres tu navegador. Sin embargo, si utilizas cookies, el SID se mantendrá mientras no se destruya la cookie. En su valor por defecto (0), las cookies se destruyen cuando se cierra el navegador. Si quieres que se mantenga el SID durante más tiempo, debes indicar en esta directiva ese tiempo en segundos
session.cookie_path	
session.cookie_domain	
session.cookie_httponly	
session.name	Determina el nombre de la cookie que se utilizará para guardar el SID. Valor predeterminado PHPSESSID
session.auto_start	Predeterminado a <i>false</i> . En este caso deberás usar la función <code>session_start</code> para gestionar el inicio de las sesiones. Si usas sesiones en el sitio web, puede ser buena idea cambiar su valor

# Sesiones: directivas php.ini

## Otras directivas de php.ini:

Directiva	Descripción

# Sesiones: enlaces

## **Pendiente de incluir Sesiones sin cookies.**

Directivas modificables dinámicamente.

Completar selección y descripción de directivas de sesión.