

TEMA 5

Cookies & Sesiones

`$_COOKIE` y `$_SESSION` en PHP

COOKIES

¿Qué son las Cookies?

- Son datos que se guardan en el navegador del usuario. Los distintos navegadores no las comparten, ya que cada uno tiene las suyas.
- Están formadas por sólo texto.
- Son archivos menores de 4 Kb.
- Las Cookies se asocian al dominio donde las instalemos, no pudiendo ver cookies de otros dominios.

APLICACION
=
FUNCIONALIDAD
+
DATOS

Funcionalidad

La funcionalidad es lo que hemos estado viendo hasta ahora en lo que llevamos de curso.

Hemos aprendido la funcionalidad de la aplicación usando:

- Estructuras de Control.
- Funciones.
- APIs de PHP.

Datos

Los datos los estamos almacenando por medio de los distintos tipos de variables que hemos visto.

Pero...

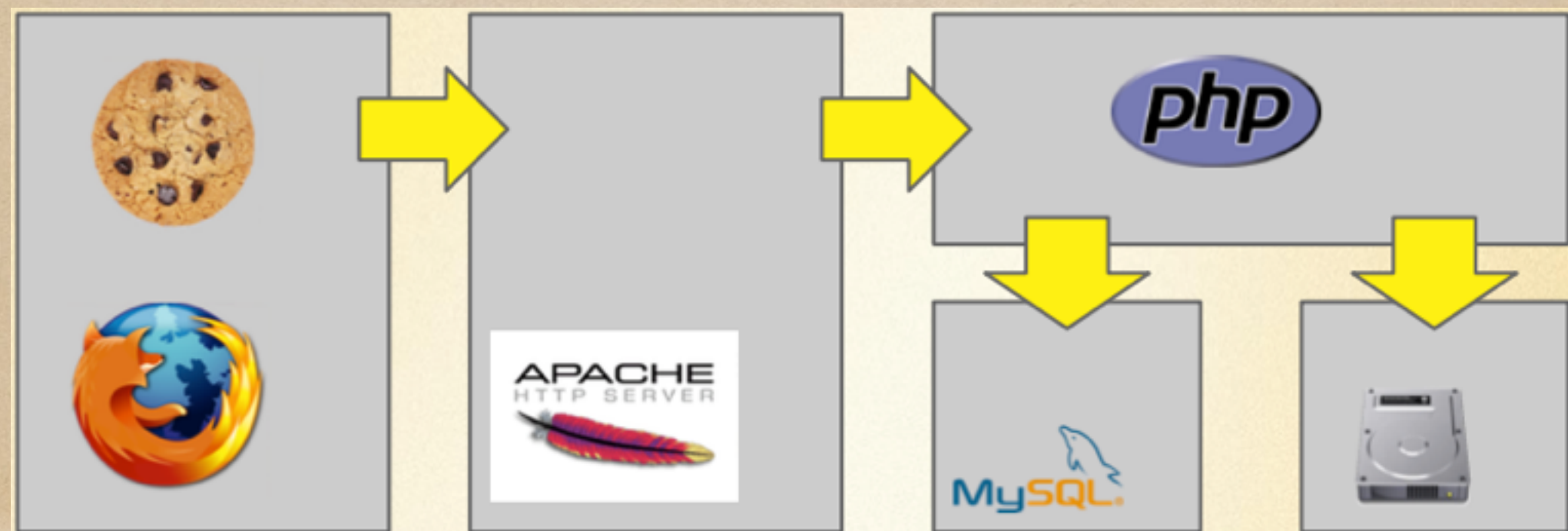
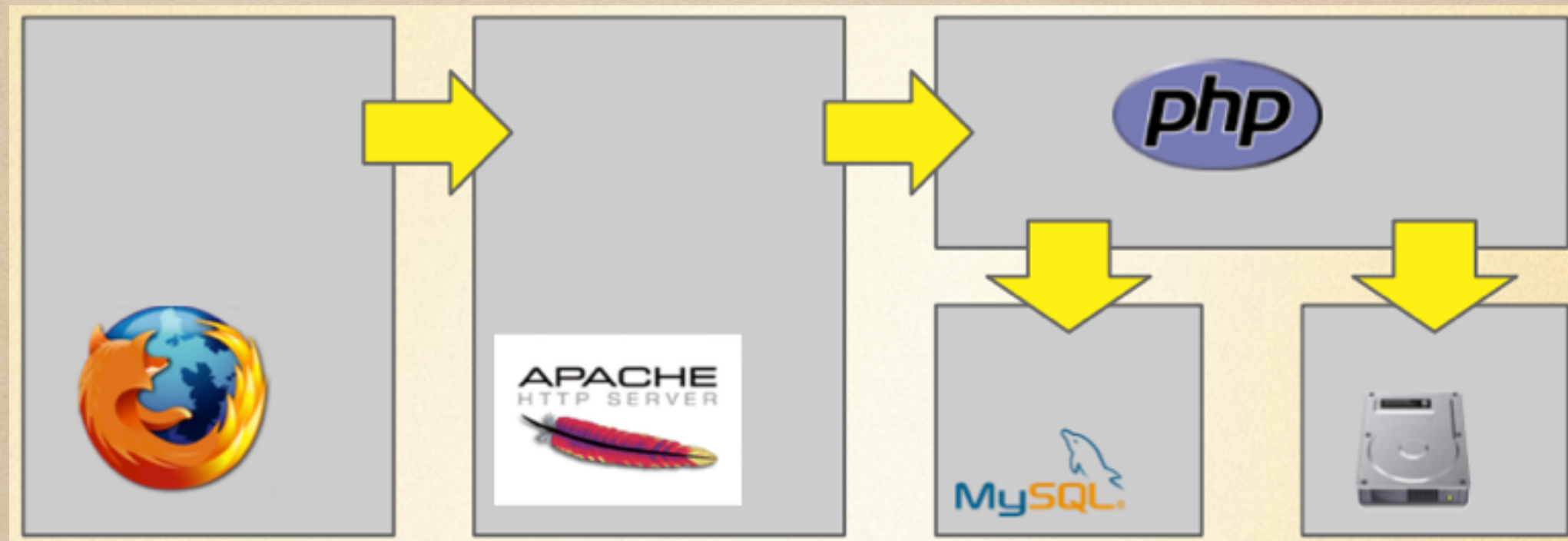
¿Qué ocurre cuando un script se acaba?

¿Y si vuelvo a entrar en la misma página de antes, o de ayer, no recuerda mi usuario y a veces mi contraseña?

Nos falta la **persistencia** de los datos.

Las variables superglobales nos permiten comunicar datos entre scripts pero no tienen persistencia.

Almacenar datos con PHP



Capas de persistencia de los Datos

\$var, \$i, \$miarray
\$_POST / \$_GET

La ejecución de una página
Enviar datos entre páginas

\$_SESSION

Guardar datos en la sesión

MySQL

Persiste lo que quieras

FileSystem

Persiste lo que quieras

\$_COOKIES

Persiste lo que quieras *

Asociar datos persistentes a usuario



Requiere que tu usuario se registre en la web para asociarle esos datos



Se guardan en el navegador. Por lo que se asocian directamente.

¿Qué pasa si el usuario usa dos navegadores?
¿Leyes de cookies? ¿Analytics?....

¿Quiénes acceden a las cookies?

{ JS }



Usos de las cookies

- Información del perfil/comportamiento.
- "Recordar la clave".
- Preferencias del usuario.
- Saber si el usuario es recurrente.
- Un contador de visitas.

Crear cookies en PHP

```
bool setcookie(  
    string $name  
    string $value  
    int $expire = 0  
    string $path  
    string $domain  
    bool $secure = false  
    bool $httponly = false)
```


Cómo recuperar una cookie

```
Superglobal $_COOKIE
```

```
Superglobal $_REQUEST
```

```
$_COOKIE[ "nombre_cookie" ]
```


Cómo Actualizar/Borrar una cookie

Misma función **setcookie()**

Actualizamos sobrescribiendo

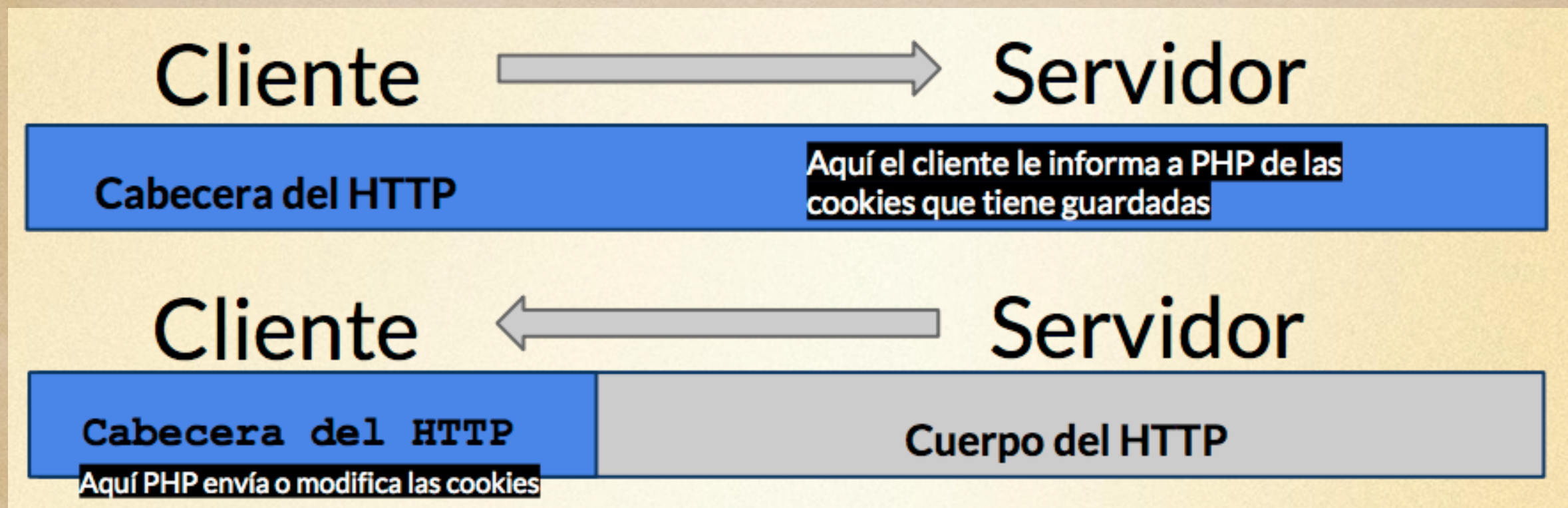
Borramos asignando **false** al valor

Errores comunes en el uso de cookies

- ♦ Están visibles en la siguiente carga de la página.
- ♦ Borramos con la misma configuración de la cookie, asignando la cadena vacía o false.
- ♦ Si queremos asignar false a la cookie, usa valor 0, ya que true se asigna con valor 1.
- ♦ Si no asignamos caducidad, la cookie no perdurará
- ♦ Se envían antes de escribir en la página.

Donde viajan las cookies

Cabeceras HTTP



Ejemplo de cabeceras de HTTP

GET http://www.desarrolloweb.com/kk33434	
Status: HTTP/1.1 404 Not Found	
Request Headers	
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding	gzip,deflate,sdch
Accept-Language	es-ES,es;q=0.8
Cookie	__gads=ID=c5db87d16e5c1856:T=1336507905:S=ALNI_Mb7VrfJ7N9Nz0AHUUWDD9K3jfGsYw; __ar_v4=
User-Agent	Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/29.0.1547.76 S
Response Headers	
Cache-Control	no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Connection	Keep-Alive
Content-Encoding	gzip
Content-Length	3735
Content-Type	text/html; charset=ISO-8859-15
Date	Thu, 26 Sep 2013 13:57:35 GMT
Expires	Thu, 19 Nov 1981 08:52:00 GMT
Keep-Alive	timeout=3, max=1000
Pragma	no-cache
Server	Apache
Vary	Accept-Encoding

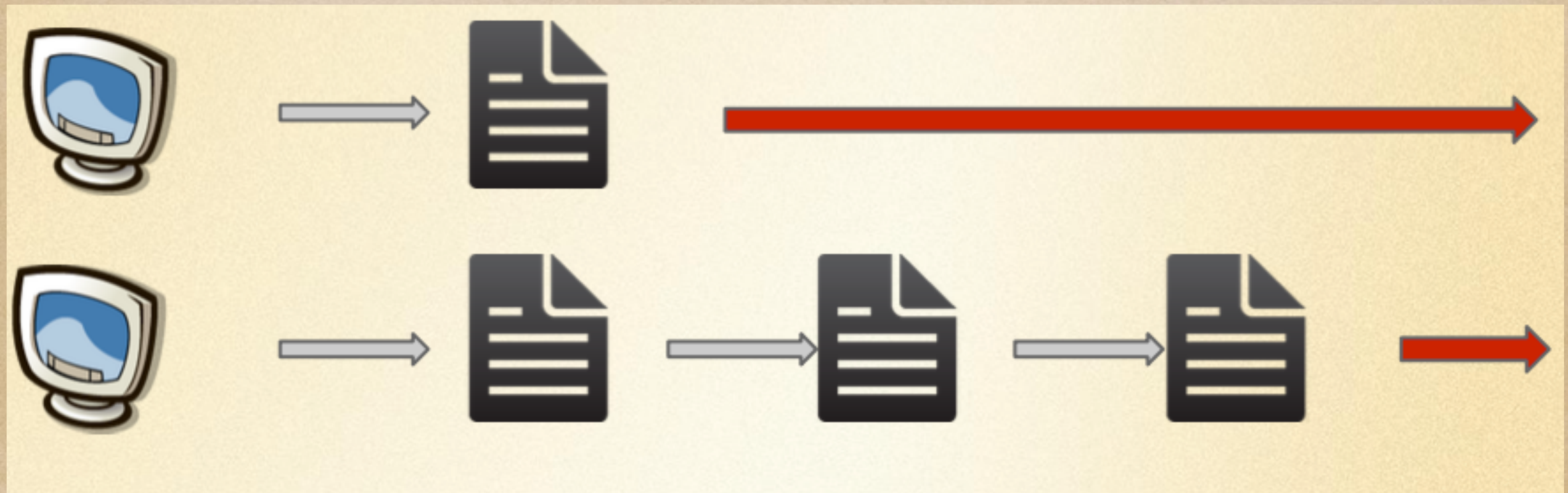
Más información sobre Cookies

<http://www.php.net/manual/es/features.cookies.php>

<http://www.desarrolloweb.com/articulos/cookies-en-php.html>

SESIONES

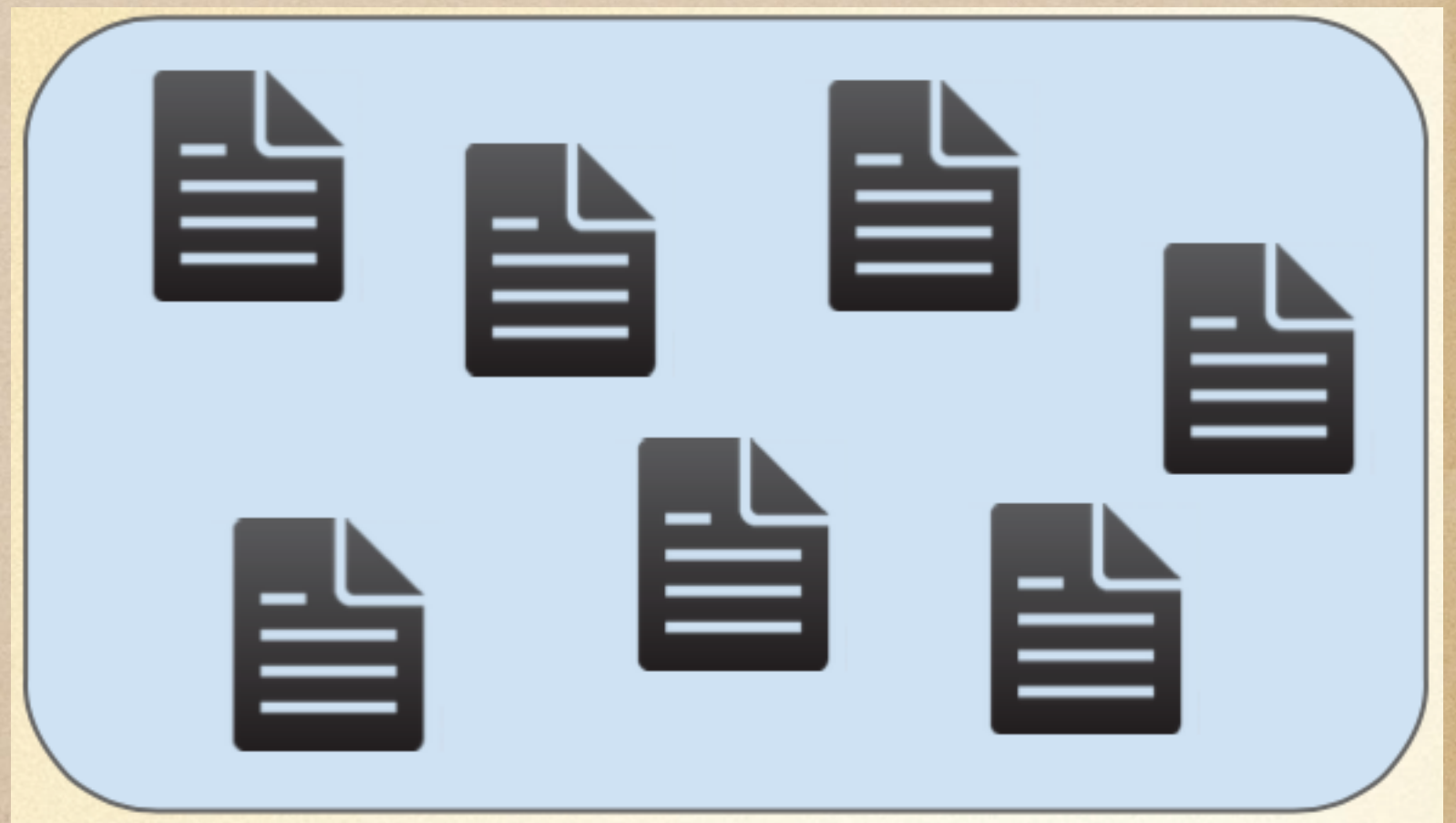
¿Qué es una sesión?



En las aplicaciones web es esencial poder almacenar datos que persistan a lo largo de todas las páginas que visita un usuario dentro de un sitio.

Aplicaciones Web

Necesitamos algún mecanismo para que todas las páginas formen un mismo sistema, lo que llamamos una APLICACIÓN.



Persistencia de datos

Hasta ahora hemos visto:

- `$_GET`. Los datos se mandan de una página a otra de la aplicación a través de la URL. Cualquiera ve los datos.
- `$_POST`. Los datos se envían a través de un formulario de una página a otra. Se necesita por tanto un formulario.
- `$_COOKIE`. Los datos los podemos guardar en un archivo de texto, pero pueden estar disponibles cada vez que accedamos a esa página.

Sesiones

Lo que ahora pretendemos es que ciertos datos, los cuales no queremos que sean visibles (descartamos `$_GET`), no usar formularios (descartamos `$_POST`) y que sólo se mantengan mientras estoy dentro de la página web (descartamos `$_COOKIE`) persistan.

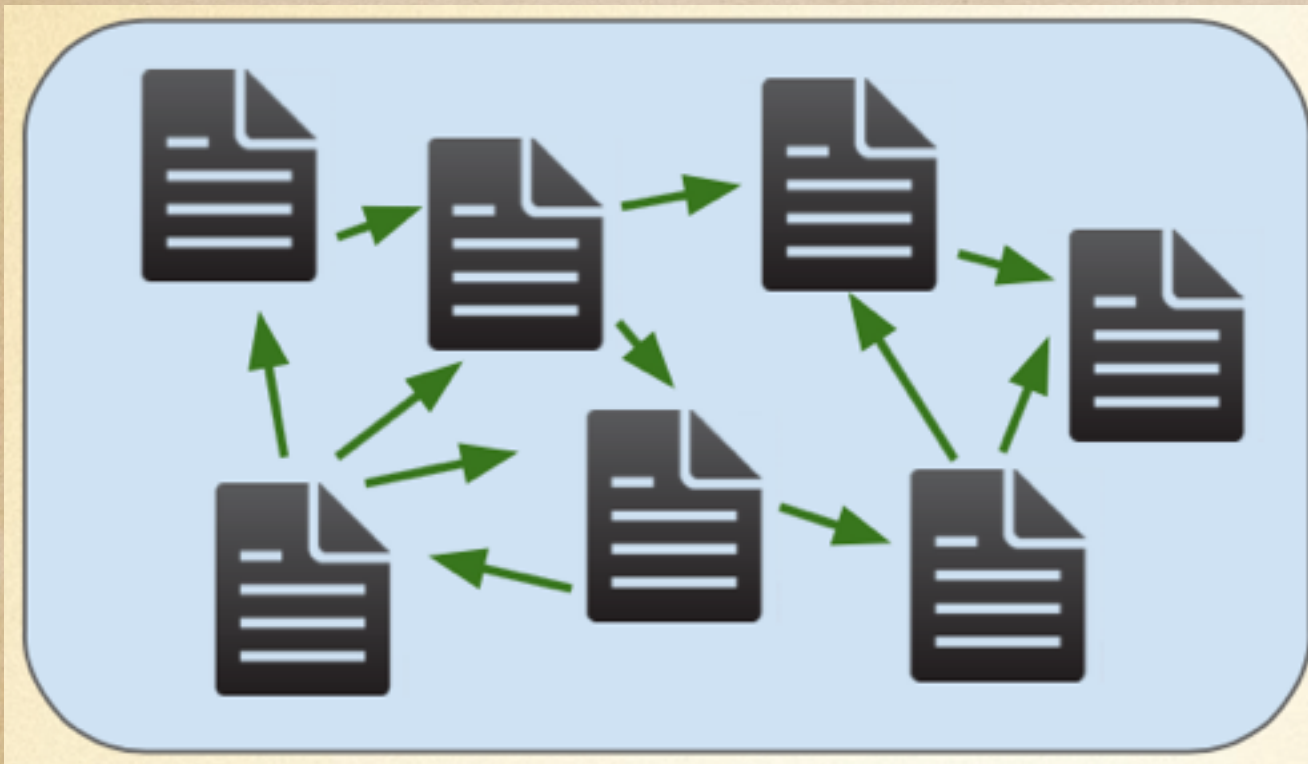
Queremos que esos datos estén disponibles a lo largo de todas las páginas que visite del mismo dominio. Por ejemplo, guardar los elementos de un carrito de la compra en Amazon o eBay.

\$_SESSION



No necesitamos mandar los datos explícitamente de una página a otra, simplemente guardarlos en la sesión.

\$_SESSION



La sesión está presente durante toda la visita y la persistencia es transparente para el desarrollador.

EJEMPLOS:

- Login de usuarios.
- Carrito de la compra.

Sesiones en PHP

PHP implementa las sesiones de una manera sencilla.

Activar variables de sesión:

```
session_start()
```

Almacenar datos en la sesión:

```
$_SESSION
```

Destruir una sesión:

```
session_destroy()
```


Sesiones en PHP

La variable `$_SESSION`, por ser superglobal, siempre está presente, pero para trabajar con sesiones debemos iniciarla.

Para iniciar en una página el trabajo con sesiones se usa `session_start()`; y debe ser escrito en el script antes que cualquier otra cosa, incluido el HTML.

PHP asigna un identificador de sesión a cada sesión abierta, y ese identificador es lo único que se guarda en el navegador del cliente. Todos los datos de la sesión se guardan en el servidor. Cuando el cliente cierre el navegador, la sesión acaba.

id de sesión

Para conocer el identificador de la sesión actual tenemos `session_id()`

- Su mantenimiento es transparente al desarrollador.
- Solamente se renueva si cerramos el navegador.
- El identificador de sesión es lo único que se guarda en el navegador (por medio de una cookie).

Serialización automática

En `$_SESSION` podemos guardar cualquier cosa. Una variable, un array, un objeto, etc.

PHP realiza la serialización automática de los datos a guardar.

Tiempo de una sesión

El tiempo que una sesión permanece activa se puede configurar en PHP a través del archivo `php.ini`

```
session.gc_maxlifetime = 1440
```

Hay otras configuraciones para la sesión en el `php.ini`