Bibliografía:

www.w3schools.com

Cookies en PHP | Como crear y gestionar cookies en PHP (srcodigofuente.es)

PASO DE INFORMACIÓN ENTRE PÁGINAS

Para pasar información entre dos páginas se pueden usar varios mecanismos.

Una forma es **utilizar campos ocultos**:

<input type="hidden" value="valor que pasamos" name="nombre del dato">

Este dato se recoge en la página destino con el método indicado en el formulario ($_POST o $_GET)

Otra manera es **a través de un enlace**:

<a href="destino.php?nombre=Pepe&apellido=Lopez" >

Se pueden pasar parejas nombre=valor separados por &. Se recogen en destino con $_GET. Hay que tener cuidado de no sobrepasar el tamaño máximo permitido.

Cuando lo anterior no es posible o aconsejable se utilizan **cookies o sesiones**. Las cookies guardan la información en el cliente, mientras que las sesiones la guardan en el servidor.

COOKIES

Las cookies son información almacenada por un sitio web en el disco duro del usuario. Esta información es almacenada en un archivo tipo texto que se guarda cuando el navegador accede al sitio web.

En nuestros ordenadores se guardan muchos datos que necesitan conocer las páginas web cada vez que entramos en ellas, 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.
- Los textos nunca podrán ocupar más de 4K, de modo que nadie podría llenarnos el ordenador a base de cookies. El número máximo de cookies de un dominio está establecido en 20.

Un ejemplo típico de las cookies podría ser un contador de las veces que accede un usuario a una págin**a**. 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 misma

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.

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 rellenados total o parcialmente, redirección a determinadas páginas...).

Gestión de las cookies con PHP

Las cookies se envían en las cabeceras de las transacciones HTTP y, consecuentemente, deberemos generarlas en PHP antes de enviar ningún otro contenido al cliente. Su creación será previa a la apertura del documento HTML.

Para crear una cookie, llamamos a la función `setcookie` . En las posteriores peticiones que recibamos de ese cliente vendrá incrustada la cookie, a la que podremos acceder mediante la superglobal $_COOKIE.

IMPORTANTE: La llamada a setcookie se tiene que hacer antes de la etiqueta html.

La sintaxis de `setcookie` es la siguiente:

bool setcookie (string $nombre [, string $valor [, int $expira = 0 [, string $ruta [, string $dominio [, bool $segura = false [, bool $httponly = false]]]]])

Donde:

- nombre es el identificador de la cookie. Si terminamos este nombre en corchetes [] y escribimos un índice dentro de ellos (atención, sin comillas) podremos recibir los datos de las cookies con el mismo nombre en una variable de tipo array. No es que se pueda almacenar un array dentro de una cookie, sino que al crear varias cookies con el mismo nombre y distinto índice dentro de los corchetes, PHP recibe todos sus datos en un array con ese nombre.
- valor es el dato que queremos asignar al identificador mediante la cookie. Si le asignamos el valor FALSE, o “”, se borrara la cookie; consecuentemente, no

podremos almacenar directamente valores booleanos en las cookies, sino que tendremos que adoptar algún convenio alternativo (por ejemplo, 1 significa TRUE y 0 significa FALSE). Debemos tener en cuenta que técnicamente las cookies no se pueden modificar; sólo se pueden reescribir con un nuevo valor.

- expira es la fecha expresada en formato época UNIX en la que caduca la cookie y es borrada del cliente. Si no se indica una fecha concreta la cookie se borrará cuando se cierre el navegador. Para expresar la fecha de expiración generalmente se recurre a la función `time()` sumándole un cierto número de segundos (3600 para una hora, 86400 para un día, …).
- ruta es el path del servidor desde el que se podrá ver la cookie. Si le asignamos el valor / podrá verse desde todo el servidor; pero si le asignamos un valor /nombre/, sólo podrán verla los script PHP almacenados en esta carpeta y todas sus subcarpetas.
- dominio es el subdominio dentro del servidor al que el cliente enviará las cookies. Por ejemplo, si le asignamos el valor '.rosa.com' el cliente enviará la cookie para cualquier petición del dominio, pero si le asignamos el valor '.otro.rosa.com' sólo las enviará en las peticiones a este subdominio.
- segura puede recibir el valor TRUE si sólo queremos enviar la cookie en caso de que exista una conexión segura de tipo https, o FALSE si queremos enviarla en cualquier caso
- httponly debe tener el valor FALSE si no queremos que nuestras cookies sean accesibles mediante lenguajes de script del lado del cliente como JavaScript. Un tipo de ataque muy frecuente consiste en lograr ejecutar en el ordenador del cliente un script (que previamente hemos logrado incrustar en una página en la que el cliente confía) que accede a sus cookies y las aprovecha para realizar peticiones malintencionadas en su nombre; estos ataques se denominan XSS (*Cross Site Scripting*).

Definir una cookie ya existente implica el borrado de la antigua. Del mismo modo, el crear una primera cookie conlleva la generación automática del archivo texto.

Si queremos eliminar una cookie, la sobrescribimos con fecha anterior a la actual.

Para crear una cookie que sólo tenga existencia mientras no cerremos la ventana del navegador, pasaremos como fecha de expiración de la cookie, el valor cero.
Una vez que la instancia del navegador se cierra, dicha cookie desaparecerá.
Este tipo de cookie puede ser muy útil para validar un usuario en un conjunto de páginas, si previamente ingresó correctamente su nombre de usuario y clave. Es decir, una vez validado el usuario, se verifica en páginas sucesivas si existe la cookie. Una vez que el usuario cierra el navegador, no hay posibilidad de solicitar las páginas recorridas sin previa validación nuevamente de clave y usuario.

Hay que tener cuidado de no definir variables en nuestro script con el mismo nombre que las cookies puesto que PHP da privilegio a el contenido de la variable local con respecto a la cookie y no dará un mensaje de error.

Las cookies son una herramienta para personalizar nuestra 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.

Ejemplos:

https://www.w3schools.com/php/php_cookies.asp

SESIONES

Hasta ahora las variables, como tales, sólo han sido visibles en los scripts donde las "declarábamos". Al cargar otra página, sus valores se pierden a menos que los pasemos (vía URL, con un formulario). Estos métodos, no son los más adecuados en todas las ocasiones: cuando queremos utilizar el valor de una variable en varios scripts de distintas páginas. Podríamos resolver este problema con las cookies, pero no todos los navegadores las aceptan y podrían estar desactivadas.

Necesitamos, por tanto, poder declarar ciertas variables que puedan ser reutilizadas tantas veces como queramos dentro de una misma sesión. Este tipo de situaciones son solventadas a partir de las variables de sesión. Una sesión es considerada como el intervalo de tiempo empleado por un usuario en recorrer nuestras páginas hasta que abandona nuestro sitio o deja de actuar sobre él durante un tiempo prolongado o bien, sencillamente, cierra el navegador.

PHP nos permite almacenar variables llamadas de sesión que, una vez definidas, podrán ser utilizadas durante este lapso de tiempo por cualquiera de los scripts de nuestro sitio. Estas variables serán específicas del usuario de modo que varias variables sesión del mismo tipo con distintos valores pueden estar coexistiendo para cada una de las sesiones que están teniendo lugar simultáneamente. Estas sesiones tienen además su propio identificador de sesión que será único y específico.

Para cada usuario PHP internamente genera un identificador de sesión único (SID), que sirve para saber las variables de sesión que pertenecen a cada usuario. Para conservar

el identificador de sesión durante toda la visita de un usuario a una página PHP almacena la variable de sesión en una cookie, o bien la propaga a través de la URL. Esto se puede configurar desde el archivo php.ini. Si tenemos activada la directiva session.use_trans_sid y desactivada la directiva session_use_only_cookies, PHP se encargará de añadir automáticamente el id a todos los URLs relativos (en la forma ?PHPSESSID=id_sesion o como un campo oculto en los formularios con método POST) Si no tenemos activada la directiva session.use_trans_sid (está desactivada por defecto) y están desactivadas las cookies, o si queremos pasar el SID en un header('Location: …), tendremos que incluir el SID manualmente recurriendo a la constante SID, que contiene el id de la sesión actual ya en el formato PHPSESSID=id_sesion.

Gestión de las sesiones en PHP

Cuando queremos utilizar variables de sesión en una página tenemos que iniciar la sesión con la siguiente función:

session_start ()
Inicia una sesión para el usuario o continúa la sesión que pudiera tener abierta en otras páginas. Al hacer session_start() PHP internamente recibe el identificador de sesión almacenado en la cookie o el que se envíe a través de la URL. Si no existe tal identificador se sesión, simplemente lo crea.

Si en el php.ini se ha definido la variable session.auto_start = 1 se inicializa automáticamente la sesión sin que se tenga que hacer el session_start()

Una vez inicializada la sesión, podemos utilizar variables de sesión, es decir, almacenar datos para ese usuario, que se conserven durante toda su visita o recuperar datos almacenados en páginas que haya podido visitar.

La sesión se tiene que inicializar antes de escribir cualquier texto en la página. Esto es importante y de no hacerlo así corremos el riesgo de recibir un error, porque al iniciar la sesión se deben leer las cookies del usuario, algo que no se puede hacer si ya se han enviado las cabeceras del HTTP. Las variables de sesión pueden crearse y utilizarse en cualquier parte del script

Una vez iniciada la sesión podemos utilizar variables de sesión a través de $_SESSION,

Otras funciones útiles para la gestión de sesiones son:

Función	Descripción
Session_id()	Nos devuelve el identificador de la sesión

Session_destroy()	Da por abandonada la sesión eliminando variables e identificador.
Session_unregister('variable')	Abandona una variable sesión