

EJERCICIO/EJEMPLO CARRITO DE LA COMPRA

Debéis desarrollar una pequeña “tienda” virtual donde el usuario podrá acumular una serie de productos en un carro de la compra. (en este ejemplo vamos a usar **cookies** para el mantenimiento del contenido del carrito, ya que se trata de aprender a manejarlas).



Básicamente, se dispondrá de una página en donde el usuario puede seleccionar una serie de artículos para incluir en su carrito de la compra, y de otra página en donde se muestra el contenido de la compra efectuada. Nuestro sistema, en este caso, va a estar formado por cinco *scripts*: *tienda.php*, *vercarrito.php*, *compra.php*, *funciones.php*

El primer script *tienda.php* realiza dos funciones básicas: una, mostrar los productos disponibles en la tienda para comprar. Y dos, mostrar el número de artículos que llevamos seleccionados por el momento, o bien la frase “El carrito está vacío”. Desde este *script* podremos realizar dos acciones:

- Añadir un artículo al carrito: pulsando sobre el enlace **comprar** que acompaña a cada artículo se producirá la ejecución del *script* *compra.php*, que debe recibir por url la referencia del artículo seleccionado, e introducirlo en el carrito. Una vez realizada esta operación, provoca de nuevo la ejecución del *script* *tienda.php*

- Ver carrito: provoca la ejecución del script *vercarro.php*. Este script muestra el contenido del carrito y, o bien te permite seguir comprando redirigiendonos de nuevo a *tienda.php*, o bien se puede finalizar la compra, lo que nos lleva al script *realizarcompra.php*

Para simplificar un poco el código y hacerlo más legible, vamos a desarrollar una librería de funciones denominada *funciones.php* que realizan operaciones auxiliares.

Funciones a incluir en *funciones.php*:

- `mostrar_carrito()`: esta función produce una tabla HTML realizando un recorrido por la cookie carrito (`$_COOKIE['carrito']`). En cada fila se debe incluir la referencia del artículo, y la cantidad adquirida por el momento. Finalmente se muestra el total de productos incluidos en la “cesta de la compra”. Si el carrito está vacío deberemos indicarlo.

- `escaparate()`: esta función genera una tabla HTML que en cada fila muestra la referencia, la descripción y el precio de cada producto (inventaros esta información por ahora), junto con un enlace (“comprar”) el cual nos permitirá incorporar otra unidad de dicho producto al “carrito de la

compra". (enviando dicha referencia a `compra.php`, como ya hemos comentado).

Una vez hecho esto, el script `tienda.php` es relativamente sencillo: tras incorporar la librería `funciones.php`, para mostrar los artículos disponibles sólo hay que utilizar la función `escaparate()`. También debe mostrar la cantidad de artículos comprados por el momento, o indicar si el carrito está vacío. Por último, se genera un formulario con un botón que permite acceder al script `vercarrito.php`.

Mi Tienda

El carrito está vacío

Referencia	Descripción	Precio	
ref1	Descripción Artículo 1	5 €	Comprar
ref2	Descripción Artículo 2	3 €	Comprar
ref3	Descripción Artículo 3	2 €	Comprar

Ver Carrito

O bien algo así:

Mi Tienda

Llevas 3 artículos seleccionados.

Referencia	Descripción	Precio	
ref1	Descripción Artículo 1	5 €	Comprar
ref2	Descripción Artículo 2	3 €	Comprar
ref3	Descripción Artículo 3	2 €	Comprar

Ver Carrito

El script `compra.php`, al que se llega al hacer click en "Comprar" recibe la referencia del artículo por query string, y actualiza el valor de la cookie, redirigiendote `tienda.php` una vez ha realizado su faena. Este será un script "transparente" para el usuario, ya que lo que el usuario percibe es que cuando pincha en comprar, se queda en la misma página pero se actualiza la cantidad de artículos comprados.

Por último, el script `vercarrito.php` muestra los productos adquiridos y proporciona un enlace de vuelta a la tienda para seguir comprando artículos, o bien un enlace a realizar compra, que nos lleva al script `realizarcompra.php`

Contenido del carrito

Referencia	Unidades
ref1	1
ref2	1
ref3	1
Número total de unidades: 3	

[Seguir comprando](#) [Realizar compra.](#)

El script `realizarcompra.php`, lo utilizaremos en este caso simplemente para borrar el carrito (borrar la cookie carrito), darle las gracias al usuario por su compra, y añadiremos un enlace a la pagina principal, `tienda.php` (la cual, si todo está correcto, nos dirá ahora que el carrito está vacío, ya que acabamos de finalizar la compra).

Pistas:

- Para crearnos la cookie, en este caso nos crearemos una **array asociativo**, donde el índice será la referencia del artículo, y el contenido las unidades compradas. En el ejemplo siguiente, el **nombre** que le damos a la cookie es `carrito['ref1']` (suponiendo que en este caso `$ref='ref1'`), por ejemplo, como **valor** le asignamos las unidades (lo que se guarda en la cookie) a través, en este caso, de la variable `$unidades`, y el **tiempo de vida** de la misma, en este caso una hora.

por ejemplo: `setcookie("carrito[$ref]", $unidades,time()+3600);`

- De esta manera se guarda todo en `$_COOKIE['carrito']`, y la manera de acceder será, por ejemplo `$_COOKIE['carrito']['ref1']`, o más bien `$_COOKIE['carrito'][$ref]`. Trabajando así, nos queda perfecto para comprobar si existe o no la cookie carrito (y entonces sabremos si hay algún artículo seleccionado, o no hay ninguno y el carrito está vacío). Y entonces, si existe la cookie carrito, (es decir, si existe

`$_COOKIE['carrito']`), la podremos recorrer con `foreach` y acceder a cada uno de sus índices, que será la referencia del artículo), y a su contenido, que será la cantidad de unidades de ese artículo seleccionadas por el momento.

- Hay otras formas de hacerlo, como usar un array a nuestro gusto, e ir trabajando sobre él, y “serializarlo” y guardarlo de esta manera una sola cookie. Os lo pido como array asociativo, para aprovechar y practicar de nuevo con ellos, ya que nos acompañarán todo el curso. Dejo el “serializar” como actividad optativa extra, para quién desee investigar sobre ello, y practicarlo. También os vendrá muy bien, y seguro que os gusta. Aunque según dicen no es muy seguro utilizar `unserialize()` con las cookies.

A trabajar, supongo saldrán dudas, buscad, investigad, preguntad...