

DAWeb práctica 11, día dos de mayo de 2018

Carrito de compra

Posiblemente uno de los usos más comunes de las *cookies* de sesión es la capacidad de rastrear los elementos que los clientes pretenden comprar mientras navegan a través de una tienda en línea. Al igual que los carritos de compras antiguos, el carrito de compras en línea debería permitir a los clientes colocar uno o más artículos en el carrito, ver su contenido en cualquier momento y eliminar cualquier artículo, todo con el beneficio de no tener que escuchar una rueda de carro chirriante, o el desplazamiento hacia un lado.

Se muestra cómo usar las cookies de sesión para implementar carros de compras en aplicaciones web dinámicas.

Creando un carro

Para crear un carrito de compra en línea, solo se necesita usar dos características de PHP: cookies de sesión y matrices. La idea es crear una cookie de sesión como una variable de matriz vacía. A medida que los compradores colocan nuevos elementos en el carrito, el código agrega un nuevo elemento a la matriz, estableciendo la cantidad del elemento seleccionado como valor en la matriz. Para ello, crea una cookie de sesión de matriz multidimensional. Eso suena como complicado, pero en realidad es muy fácil de crear:

```
$_SESSION['compra'] = array();
```

Esta única línea de código crea una cookie de sesión llamada compra y la define como una variable de matriz. Ese es el comienzo del carrito de compras.

Colocación de elementos en el carrito

Al crear la cookie de sesión del carrito de compras, está listo para comenzar a colocar artículos en él. Para colocar un artículo en el carrito, creará un nuevo elemento de la matriz vinculado con un valor. La clave del elemento de matriz será el nombre del producto colocado en el carro, y el valor del elemento de matriz será la cantidad del producto que se comprará. Eso se ve así:

```
$_SESSION['compra']['peras'] = 10;
```

Esta declaración crea un elemento de matriz en la cookie de sesión `$_SESSION['compra']` con el nombre `peras` y le asigna el valor 10. Se pueden crear tantos elementos de matriz como se desee agregar a la variable de matriz de cookies de sesión.

Recuperando elementos de un carrito

Ahora que se tiene una matriz de *cookies* de sesión multidimensional que contiene los productos que se han colocado en el carro, todo lo que se necesita hacer es extraer los valores almacenados en la matriz para ver qué hay allí. Sin embargo, eso puede ser un poco complicado.

Debido a que la matriz es una matriz asociativa, no se puede simplemente recorrer el elemento de la matriz utilizando una declaración simple `for` o `while` porque no se sabe qué nombres de clave están en la matriz. Aquí es donde **foreach** es útil. Permite iterar a través de todos los elementos de matriz sin tener que saber cuáles son:

```
foreach($_SESSION['compra'] as $key => $valor) {  
    echo "<p>$key - $valor</p>\n";  
}
```

La instrucción `foreach` itera a través de la matriz, extrayendo cada clave y valor en cada iteración. A continuación, se puede usar la clave individual y los pares de valores en el código para enumerar los artículos y sus cantidades.

Eliminar elementos del carro

Debido a que cada producto en el carro es un elemento de matriz separado de la *cookie* de sesión, es posible manejar cada producto individualmente, siempre que se sepa el nombre del producto que se utilizó para la clave de la matriz. Para eliminar un producto individual, simplemente se indica en la función `unset` (`()`):

```
unset($_SESSION['carro']['peras']);
```

Esta instrucción solo elimina la clave del conjunto de *peras* y su valor de la matriz, dejando cualquier otro elemento todavía en la matriz. Si se desea eliminar todo el carrito de compras, se usaría lo siguiente:

```
session_unset($_SESSION['carro']);
```

Esta declaración elimina toda la *cookie* de sesión del carrito. Para comenzar un carrito nuevo, el programa necesitaría crear una nueva *cookie* de sesión de carrito y convertirla en una variable de matriz.

Se ha de tener precaución al eliminar elementos individuales del carrito de compra o la *cookie* de sesión completa, porque no hay vuelta atrás. Cuando se elimina una *cookie* de sesión, se borra y no se puede recuperar.


```

    }
}
?>
<fieldset style="width:300px">
<legend>Su carrito de la compra</legend>
<?php
    if (!isset($_SESSION['carro'])) {
        $_SESSION['carro'] = array();
        echo "El carrito está vacío\n";
    } else {
        echo "<form
action=\"carroprueba.php\" method=\"post\">\n";
        echo "<table>\n";
        echo
"<tr><th>Producto</th><th>Cantidad</th><th/></tr>\n";
        foreach($_SESSION['carro'] as
$key => $valor) {
            echo "<tr><td>$key</td><td>$valor</td>\n";
            echo "<td><input type=\"submit\"
name=\"\$key\" value=\"Remove\"></td>
</tr>\n";
        }
        echo "</table>\n";
        echo "</form>\n";
    }
?>
</fieldset>
</body>
</html>

```

La primera parte del programa crea un formulario simple para seleccionar los productos para comprar. El código enumera dos productos: peras y plátanos, y proporciona un cuadro de texto para indicar la cantidad de cada uno que se desea colocar en el carrito de compra.

La siguiente sección usa código PHP para verificar si el formulario ya ha sido enviado. Si el visitante del sitio ha enviado el formulario, el código PHP verifica si alguno de los productos se ha seleccionado para la compra. Si se ha seleccionado uno, el código PHP almacena el nuevo número de cantidad en la cookie de sesión del carrito para ese producto:

```

if (isset($_POST['peras'])) {

```

```
if (is_numeric($_POST['peras'])) {  
    $_SESSION['carro']['peras'] = $_POST['peras'];
```

A continuación, el código muestra el estado del carrito de compras. Si no hay una cookie de sesión de carrito de compras, se crea una:

```
$_SESSION['carro'] = array();
```

Si existe una *cookie* de sesión del carrito de compra, el programa crea un formulario que contiene los elementos del carrito, junto con un botón Eliminar. La instrucción `foreach` se usa para iterar a través de cada uno de los artículos de la compra:

```
foreach($_SESSION['carro'] as $key => $value) {  
    echo "<tr><td>$key</td><td>$value</td>\n";  
    echo "<td><input type='\"submit\"' name='\"$key\"'  
value='\"Remove\"'></td></tr>\n";  
}
```

Debido a que hay dos formularios en la página web, se debe agregar más código para verificar si el comprador hizo clic en el botón Eliminar. Esto se agregó al código que busca los datos del otro formulario:

```
} elseif ($_POST['peras'] == "Remove") {  
    unset($_SESSION['carro']['peras']);  
}
```

Los siguientes pasos muestran cómo probar el programa `carroprueba.php`:

1. Abrir un editor e ingresar el código del listado.
2. Guardar el archivo como `carroprueb.php` en la carpeta raíz del servidor web. Es importante usar este nombre de archivo exacto porque los formularios lo emplean como el atributo de acción.
3. Verificar que el servidor web Apache se esté ejecutando, y luego se abre un navegador indicando el siguiente URL: `http://localhost:8080 /carroprueba.php`
4. Ingresar una cantidad para comprar para uno de los artículos, y luego hacer clic en el botón "Añadir al carrito" para agregar al carro.
5. Ingresar una cantidad para comprar para el otro artículo, y luego confirmar de forma análoga al paso anterior.
6. Hacer clic en el botón Eliminar para uno de los elementos.
7. Repetir el proceso para agregar o quitar productos en el carrito.

8. Cerrar el navegador y el servidor web Apache cuando se haya terminado. Cuando se abra por primera vez el archivo cartrodeprueba.php, el carro de compra debería mostrar que está vacío.