

PRIMER REPASO: VARIABLES, OPERADORES Y EXPRESIONES.

Temas a repasar:

- Incrustar PHP en HTML
- Agregar contenido dinámico
- Accederemos a variables de formulario
- Crear variables declaradas por el usuario
- Los tipos de variables,
- Asignar valores a las variables
- Declarar y utilizar constantes
- Ámbito de las variables
- Operadores y su orden de precedencia
- Evaluar expresiones
- Utilizar funciones de variables
- Tomar decisiones if, else, switch
- Iteración mediante bucles while, do y for
- Validaremos los datos introducidos en el formulario

PRESENTAMOS UN EJEMPLO DE UN FORMULARIO DE PEDIDOS

Una de las aplicaciones más comunes de cualquier lenguaje de servidor es el procesamiento de formularios HTML.

Vamos a hacer un formulario para una empresa llamada Bob's AutoParts

Se trata de una compañía ficticia para repuestos de coches

Es un formulario de pedidos para los repuestos que pide la empresa

1. Partimos de un formulario como el siguiente

elemento	cantidad
llantas	<input type="text"/>
aceite	<input type="text"/>
bujias	<input type="text"/>
Como conocí a Bob's	Soy cliente habitual <input type="checkbox"/>
<input type="button" value="Enviar pedido"/>	

150 ▼ 15 (144) ▼

2. Se quiere saber qué pide el cliente, obtener el total del pedido incluidos impuestos y creamos un php como este
3. La acción llama a un doc PHP, que se carga cuando el usuario pulse el botón de envío.
4. Se envía como un paquete a parte con post para que no sean visibles los datos durante el proceso del envío (con get se adjuntarían la ruta al final de URL).

5. Se incluye en el formulario los nombres de cada elemento “llantas”, “aceite” y “bujías”, que utilizaremos luego en el docum php
6. Función DATE (Profundizaremos en esta función, muy utilizada, más adelante: cada argumento del parámetro representa una parte de l fecha y la hora:
 1. **H** es la hora en formato 24 horas
 2. **i** son los minutos precedidos de un cero cuando resulte necesario (ej 07),
 3. **j** es el día del mes sin utilizar un cero inicial
 4. **S** representa el sufijo ordinal (en este caso th del inglés) y
 5. **F** es el nombre completo del mes
7. Accedemos a los name del formulario a través de las variables php estilos de variable para acceder a los datos de un formulario: (se pueden denominar largo intermedio y corto
 Por ejemplo se accedería a los datos del campo tireqty de las siguientes posibles maneras (profundizaremos en ellas más adelante).
 1. \$tireqty //estilo corto
 2. \$_POST['\$tireqty']//estilo intermedio
 3. \$_HTTP_POST_VAR['tireqty']//estilo largo
8. TIPOS DE VARIABLES
 El tipo de variable hace referencia al tio de datos que se almacenan en ella. PHP cuenta con un completo conjunto de tipos de datos. En los distintos tipos de datos se pueden almacenar distintitos datos.
 PHP admite los siguientes tipos de datos:
 1. Entero Utilizado para números enteros
 2. Flotante o doble: para números reales
 3. Cadena: para cadenas de caracteres
 4. Booleano: utilizado para valores verdaderos o falsos
 5. Matriz : utilizado para almacenar conjuntos de datos del mismo tipo (lo veremos más adelante)
 6. Objeto: para almacenar instancias de clases

Además existen dos tipos especiales:

- NULL. Las variables a las que no se les ha asignado un valor no están definidas o se les ha asignado el valor NULL, serán de tipo NULL
- y de recurso. aquellas que representan recursos externos como conexiones de bases d datos

Lo curioso en PHP es que si añadiéramos una línea como esta

\$totalqty = “hola”; pasaría de ser entero a ser string, de tipo cadena

Es también usual que hagamos una conversión de tipo de variable, así

\$totalqty = 0;

\$totalamount = (float)\$totalqty

La segunda línea toma el valor \$totalqty lo convierte en flotante y lo almacena en \$totalamount

9. CONSTANTES

- a. Los nombres de las constantes utilizan siempre mayúsculas . NO es obligatorio pero contribuye a que la lectura y el mantenimiento del código resulten más sencillos
- b. No necesita anteponer el símbolo del dólar, cuando hacemos referencia a ella, como la variable .
por ejemplo para utilizar una de las constantes anteriores podemos escribir echo TIREPRICE
- c. Otra diferencia es que las constantes solo pueden almacenar datos de tipo Booleano, entero, flotante o de cadena
- d. PHP establece un gran número de constantes propias (ejecuta el comando phpinfo() para verlas . Es una función que devuelve una lista de variables y constantes predefinidas de PHP además de otra información útil

10. ÁMBITO DE VARIABLES:

hace referencia a los lugares dentro de las secuencias de comandos en los que resulta visible una variable dada

Existen 6 reglas que se aplican a los ámbitos de PHP:

- a. Las variables superglobales incorporadas resultan siempre visibles dentro de una secuencia de comandos
- b. Las constantes una vez declaradas, siempre resultan visibles de forma global y se pueden utilizar dentro y fuera de una función
- c. Las variables globales declaradas en una secuencia de comandos resultan visibles a lo largo de la secuencia de comandos pero no dentro de las funciones
- d. Las variables utilizadas dentro de funciones que se declaran como globales hacen referencia a la variable global del mismo nombre.
- e. Las variables creadas dentro de funciones tienen restringido su ámbito a la función y dejan de existir cuando ésta desaparece.
- f. LAS MATRICES \$_GET Y \$_POST así como otras variables especiales llevan asignadas sus propias reglas de ámbito. Estas se conocen como superglobales y se pueden ver en todas partes tanto dentro como fuera de las funciones.

LISTA COMPLETA DE VARIABLES GLOBALES

- f.i. \$GLOBALS: Es una matriz con todas las variables globales. AL igual que la palabra clave global, nos permite acceder a las variables globales dentro de una función, ej como
\$GLOBALS['myvariable']
- f.ii. \$_SERVER: Es una matriz con las variables de entorno servidor.
- f.iii. \$_GET: Es una matriz con las variables pasadas a la secuencia de comandos a través del método GET
- f.iv. \$_POST Es una matriz con las variables pasadas a la secuencia de comandos a través del método POST
- f.v. \$_COOKIE Es una matriz con las variables de cookies

- f.vi. `$_FILES` Es una matriz donde se almacenan las variables relacionadas con las cargas de archivos
- f.vii. `$_ENV` Es una matriz de variables de entorno
- f.viii. `$_REQUEST`. Es una matriz con todas las variables de entrada de usuario incluyendo los contenidos de entrada de `$_GET`, `$_POST` y `$_COOKIE`
- f.ix. `$_SESSION`: Es una matriz con las variables de sesión