

Desarrollo web en entorno servidor - DWES

DWES03 - Introducción PHP.

Profesor: Pepe Lluyot
pepe.lluyot@iescristobaldemonroy.es

Introducción a PHP - Relación de ejercicios 1

Para poder realizar los siguientes ejercicios, vamos a elaborar una web con html5 con una estructura básica que nos servirá de base para incluir el código PHP.

Contenido:

Introducción a PHP - Relación de ejercicios 1	1
Comentarios	2
Variables	2
Variables no inicializadas	3
Asignación por copia y por referencia	3
variables especiales \$_SERVER[]	3
Ámbito de las variables	3
echo vs printf	4
Constantes	4
Operadores, expresiones	4
Arrays	5
cadenas de texto	5
Paso de variables por URL	6

Ejercicios (bloque I)

Código PHP embebido en HTML

1. Crea un documento HTML simple que incluya una sección de título (<h1>). Usa PHP para insertar el título dinámicamente. Por ejemplo, dentro de un **echo**, escribe `echo "<h1>Mi primera página PHP</h1>";`

Visualízalo a través del navegador, tanto desde el servidor, como abriendo el archivo directamente en local.

Comentarios

2. Incluye diferentes **comentarios** `//`, `/* */`, `#` en un archivo php y visualiza el código fuente generado en el navegador.
3. Crea un archivo PHP que contenga varias líneas de código (similar a la del ejercicio anterior), añade diferentes tipos de comentarios (`//`, `/* */`, `#`) que expliquen cada paso. Luego, abre el código fuente en el navegador para observar cómo se visualiza.
4. Modifica el ejercicio anterior para que añadas un bloque de comentarios al principio del archivo, explicando el propósito del script, y usa comentarios para "desactivar" temporalmente alguna línea de código.

Variables

5. Asigna a una **variable** tu color favorito y muestra un encabezado (`<h2>`) con el mensaje: "Mi color favorito es el nombre_del_color". Usa la variable dentro de un `echo`.
6. Utilizar el **operador** `.` para concatenar cadenas de caracteres con variables.
7. Crea dos variables, una de tipo entero y otra de tipo cadena de texto, que almacenen tu edad y tu ciudad. Inserta estas variables dentro de un párrafo HTML usando `echo`, como: `echo "<p>Vivo en $ciudad y tengo $edad años.</p>";` Probar el texto con comillas simples y dobles.
8. Declara dos variables numéricas, suma sus valores y muestra el resultado en una oración descriptiva dentro de un párrafo HTML. Por ejemplo: `echo "La suma de $numero1 y $numero2 es $suma."`
9. Modifica del ejercicio anterior el valor de la variable `numero1` por un valor decimal. Muestra la suma. Actualiza la variable `suma` con el valor: `$suma = (int) $numero1 + $numero2;` y muestra el resultado
10. Declara variables de varios tipos (entero, cadena, flotante, booleano) y utiliza **`var_dump()`** para mostrar el tipo de cada variable y su valor. Explica lo que muestra la función en cada caso.
Modificar para cada variable contenido con un tipo de dato diferente y mostrar de nuevo su tipo con la función **`gettype()`**

-
11. Crea tres variables (\$x, \$y, \$z) y asignarles el mismo valor usando la sintaxis `$x = $y = $z = "PHP es interesante";`. Muestra el contenido de las tres variables en la misma línea usando concatenación.

Variables no inicializadas

12. Declara una variable sin asignarle ningún valor y utiliza la función **`isset()`** para comprobar si está definida y es distinta de null. Muestra en pantalla el mensaje "La variable está definida y no es null" o "La variable no está definida o es null" según el caso.
13. Declara una variable y asígnale un valor vacío (como "", 0 o null). Utiliza la función **`empty()`** para comprobar si la variable está vacía. Muestra en pantalla el mensaje "La variable está vacía" o "La variable no está vacía" según el caso.

Asignación por copia y por referencia

14. Crea una variable llamada \$a y asígnale un valor numérico. Luego, crea otra variable \$b que sea una copia de \$a (**asignación por copia**). Modifica \$b y muestra ambos valores para verificar si \$a se ve afectada.
15. Declara una variable \$x con un valor de cadena. Luego, crea una variable \$y usando **asignación por referencia** (operador **`&`**). Modifica el valor de \$y y muestra el valor de ambas variables para comprobar si \$x se ve afectada.

variables especiales **`$_SERVER`**

16. Usa la variable especial **`$_SERVER`** para mostrar la ruta del archivo PHP actual y el nombre del servidor. Por ejemplo: `echo $_SERVER['PHP_SELF'];` y `echo $_SERVER['SERVER_NAME'];`
17. Utiliza otras variables especiales de **`$_SERVER`** para mostrar información sobre la dirección IP del cliente y el agente de usuario: `$_SERVER['REMOTE_ADDR']` y `$_SERVER['HTTP_USER_AGENT']`.

Ámbito de las variables

18. Crea una **variable global** llamada `$saludo` con el valor "Hola desde fuera". Luego, crea una función que intente modificar esa variable. Explica cómo usar la palabra clave global dentro de la función para permitir que la función modifique la variable. Hacer uso de **`global`** y del array **`$GLOBALS`**

-
19. Crea una función llamada contador() que utilice una **variable estática** para llevar la cuenta de cuántas veces ha sido llamada la función.
- Dentro de la función, declara una variable estática llamada \$cuenta e inicializarla con el valor 0.
 - Incrementa el valor de \$cuenta cada vez que se llame la función y luego muestra un mensaje que diga: "La función ha sido llamada X veces", donde X es el valor actual de la variable estática.
 - Llama a la función tres veces seguidas y observa el resultado.

echo vs printf

20. Define una variable con tu nombre y muestra un mensaje en pantalla utilizando **echo**, **print** y **printf**. Compara los métodos en cuanto a la sintaxis y el formato. Por ejemplo, puedes usar: `printf("Mi nombre es %s", $nombre);`. Realiza distintas pruebas usando comillas dobles y simples.
21. Crea una variable con un número entero y otra con un número decimal. Utiliza **printf** para mostrar ambos números en diferentes formatos. Por ejemplo, muestra el número entero como un entero y el número decimal con dos cifras decimales. Realiza pruebas usando comillas dobles y simples.

Constantes

22. Define una constante para el IVA (21%) y úsala para calcular el precio final de un producto cuyo precio base se almacene en una variable. Muestra el precio final aplicando la constante del IVA.
23. Crea una constante para almacenar el nombre de una ciudad y luego intenta modificarla. Explica por qué no es posible cambiar el valor de una constante en PHP.

Operadores, expresiones

24. Operadores aritméticos. Crea un programa que declare dos números y realice las siguientes operaciones entre ellos: suma, resta, multiplicación, división, módulo y exponenciación. Muestra cada operación en una nueva línea.
25. Operadores de asignación. Declara una variable con un valor inicial. Luego, realiza las siguientes operaciones utilizando operadores de asignación: a) Incrementa el valor de la variable con el operador `+=`. b) Multiplica el valor de la variable con el operador `*=`. c) Calcula el módulo del valor de la variable con el operador `%=`. Muestra el resultado de cada operación en pantalla.
26. Declara dos variables con valores diferentes y realiza las siguientes comparaciones:

a) Compara si las dos variables son iguales (==). b) Compara si las dos variables son estrictamente iguales en valor y tipo (===). c) Compara si las dos variables son diferentes (!=). d) Compara si las dos variables son estrictamente diferentes en valor o tipo (!==).

27. Utiliza operadores de comparación (>, <, >=, <=) para crear condiciones que comparen dos números e imprime un mensaje diferente según cuál sea mayor o si son iguales.

Arrays

28. Crea un **array** que contenga los nombres de tres ciudades: "Sevilla", "Málaga", y "Cádiz". Usa echo para mostrar en pantalla el nombre de la primera ciudad.
29. Crea un array con tres colores: "rojo", "verde" y "azul". Luego, cambia el valor del segundo elemento del array a "amarillo" y muestra el contenido actualizado del array usando **print_r()**
30. Crea un array que almacena los nombres de tres frutas. Luego, añade una cuarta fruta al array y muestra todos los elementos usando un echo. Usar la función **implode**.
31. Crea un **array asociativo** que almacene el nombre de una persona, su edad y su ciudad. Usa echo para mostrar la ciudad de la persona.
32. Crea un array con tres países: "España", "Francia", "Italia". Usa un bucle **foreach** para recorrer y mostrar cada uno de los países en pantalla.
33. Crea un array con cinco números. Usa la función **count()** para mostrar cuántos elementos tiene el array.
34. Crea un array con los siguientes valores: "gato", "perro", "loro". Elimina el elemento "perro" del array usando **unset()** y muestra el array resultante usando **print_r()**.
35. Crea dos arrays, uno con tres frutas y otro con tres colores. Usa **array_merge()** para combinarlos y mostrar el resultado usando **print_r()** y **var_dump()**

cadena de texto

36. Define una cadena de texto y usa la función **strlen()** para mostrar la longitud de la cadena. Después, usa **strtoupper()** para convertir en mayúsculas y **strtolower()** para convertir a minúsculas.
37. Crea una variable llamada \$frase que contenga la siguiente cadena de texto: "PHP es un lenguaje divertido y poderoso." Usa la función **str_replace()** para reemplazar la palabra "divertido" por "fácil" en la cadena almacenada en \$frase.

Almacena el resultado en una nueva variable `$nuevaFrase` y muestra ambas frases (la original y la modificada) en pantalla. Reemplaza múltiples palabras a la vez: usa `str_replace()` para cambiar "PHP" por "JavaScript" y "poderoso" por "versátil" en la misma frase. **`str_replace([a,b],[a',b'],$cadena)`**.

Paso de variables por URL

38. Crear una página con un enlace que pase por parámetros dos variables "Nombre" y "Apellidos" a otra página que las mostrará en un mensaje de texto