

PHP: variables, tipos, operadores, expresiones, estructuras de control

1)

a_

Variables	Tipo
\$a	boolean
\$b	string
\$c	string
\$d	integer
\$f	integer (ya que la función doble retorna un entero)
\$g	integer

b_ Operadores

Unitarios: ++

Binarios: *, =, ==, +=

Ternarios: ?

c_ Funciones

Definidas por el usuario: *doble()* en la cual los parámetros se pasan por valor(en este caso se le pasa la variable \$d incrementada)

Funciones que dispone PHP: *gettype()*, *is_int()*, *is_string()*

d_ **Estructuras de control:** *if()*, *break*

e_ **La salida por pantalla es:**

booleanstringstringinteger1xyzxyz174444

2)

a- Son equivalentes. En el primer caso mostrará los números del 1 al 10 ya que inicializa la variable \$i en 1 y utiliza un while (cuya condición es que \$i no sea mayor a 10) y un print acompañado de un postincremento(\$i++). En el segundo caso utiliza el mismo while e inicializa \$i en 1 pero dentro del while primero usa un print y luego un preincremento(++\$i) dando el mismo resultado. En el tercer caso usa un do while (cuya condición es que \$i sea menor a 10) e inicializa la variable \$i en 0 , luego dentro del do while usa un print con un preincremento haciendo que no se muestre el 0 y arranque con 1.

b- Son equivalentes. El primer caso en la primera expresión del for, inicializa la variable \$i en 1 , la segunda expresión evalúa si \$i es menor o igual a 10 y la tercera realiza el incremento de \$i (++\$i) al final de cada bucle, luego dentro del for hay un print \$i que hará que se muestren los números del 1 al 10. El segundo caso se diferencia en que el print \$i se encuentra en la tercera expresión del for y dentro de este no habrá ninguna sentencia

dando el mismo resultado que el primer caso. En el tercer caso las expresiones del for estan iguales que en el primero exceptuando la segunda expresión del for que está vacía, el bucle se rompe gracias a un if cuya condición es que \$i no sea mayor a 10 (dentro de este hay un break) y luego continua con print %i consiguiendo el mismo resultado. El cuarto caso inicializa la variable \$i en 1 fuera de las expresiones del for ya que tiene las 3 expresiones de este están vacías , para romper el bucle utiliza el mismo if que en el caso 3 y el print se encuentra dentro del for además de un post incremento llegando al mismo resultado.

c- Son equivalentes. Ambas sirven para determinar si una variable \$i tiene como valor 0,1 o 2. La diferencia esta en que el primer caso utiliza las estructuras if y elseif para comprobar que \$i valga 0,1 o 2 y el segundo caso utiliza la estructura switch.

3)

a- Crea una tabla de 5 filas y 2 columnas.

El for externo crea las filas y el for interno crea las columnas dentro de cada fila. La instrucción "echo" se utiliza para imprimir el código HTML generado en la página.

b- Crea un formulario el cual solicita la edad del usuario, luego la compara a través de un if, si la edad es mayor o igual a 21 años se imprime el texto "Mayor de edad" a través de la instrucción echo, y si es menor a 21 se imprime "Menor de edad".

4)

El primer echo muestra: **EI**

Ya que se ejecuta antes de invocar al archivo datos.php

Mientras que el segundo con el archivo ya invocado muestra: **EI clavel blanco**

5) Lo primero que se puede ver es que tanto en visitas.php como en contador.php hay un error en común. Este error está en la etiqueta de apertura la cual está incompleta, en ambos aparece <? cuando debería ser <?php (la etiqueta abreviada " <? " se puede usar solo si se habilita con la directiva short_open_tag del fichero de configuración "php.ini"). Estos archivos lo que harán es funcionar como un contador de visitas hechas a una página web. El archivo contador.dat establece el valor inicial del contador el cual luego será modificado en el archivo contador.php. Este último archivo abre contador.dat, incrementa el valor del contador en 1 y guarda este cambio, luego muestra el contador y por último cierra el archivo contador.dat.

PHP: arrays, funciones

1) En este caso los códigos son equivalentes ya que las matrices son idénticas ,ambas son matrices asociativas con un elemento "indexado" (el elemento: 4) al cual se accede a través de la posición que ocupa en el arreglo, que en este caso es 0. La única diferencia es que están declaradas de formas diferentes, uno está hecho con el constructor de lenguaje : array() y el otro declara "manualmente" cada clave del arreglo asignandole un valor , por ejemplo; `$a['color'] = 'rojo'` .

2)

- a) Imprime: bar1 (1 es igual a verdadero)
- b) Imprime: 5942
- c) Imprime: No imprime nada.

3)

- a) Imprime: Has entrado en esta página a las (hora actual de tu pc), con (Minutos actuales de tu pc) minutos y (Segundos actuales de tu pc) segundos, del (Fecha actual en tu pc)
- b) Imprime: $5+6=11$