

Ejercicio 1

Variables:

\$a = booleana

\$b = string

\$c = string

\$d = integer

\$f = integer

\$g = integer

Operadores:

++ Incremento

+= Incremento a una variable

\$i * 2 Operador binario

= \$a ? ++\$d : \$d*3 Operador ternario

Funciones

function doble(\$i) \$i ← Parametro

Estructuras de Control

```
if f (is_string($a)) {  
echo "Cadena: $a"; }
```

```
if (is_int($d)) { $d += 4; }
```

Salida por pantalla

booleanstringstringinteger1xyzxyz184444

Ejercicio 2

a-

Aunque los tres códigos se realizan de manera y con estructuras de control diferentes, el resultado termina siendo el mismo. Los primeros dos ambos con while, la única diferencia entre ambos es que el print y la acumulacion se hace en la misma linea en el código 1. El último código lo realiza con un do-while, pero las sentencias internas son las mismas.

b-

Al igual que el ejemplo anterior los cuatro códigos resultan en la misma salida, una secuencia de números del 1 al 10. El primero utiliza un for con solo una secuencia interna, la de imprimir la variable. El segundo tambien usa un for, pero en este caso la instrucción de print esta dentro de la condición del for, sin sentencias internas. El tercero utiliza además de un for un if interno el cual corta si la variable llega a 10. Y el ultimo tambien utiliza un for y un if interno pero en este caso la acumulacion de la variable se hace fuera de su condición.

c-

En este caso ambos códigos deberían generar el mismo resultado suponiendo que la variable tenga el mismo valor. Lo que serían salidas de `i equals 0` `1` o `2` dependiendo del valor de la variable. En el primer caso lo resuelve con un `if elseif`, y el segundo con un `switch case` y `breaks` internos.

Ejercicio 3

a- El código se utiliza para crear una tabla de un ancho del 90% del permitido con un borde de grosor 1 la cual posee 5 filas y 2 columnas. Las filas y columnas son generadas por dos `for` los cuales a medida que se van haciendo las iteraciones van aumentando la cantidad de filas y columnas respectivamente.

b- Este código crea un input para ingresar una edad y dependiendo de si esta es mayor o menor a 22 va a devolver si la persona es mayor o menor de edad. Esto es realizado con un input Edad de nombre `age` y tamaño 2 y el tipo es `submit`. Esto se encuentra en la parte del `if`, en la parte del `else`, se realiza el `if` de la edad, si esta es `>= 21` devuelve mayor de edad de lo contrario devolverá menor de edad.

Ejercicio 4

En la salida del 2do código incluyendo al primero se puede ver, la primera linea imprime un `El` y dos errores de variables indefinidas, lo cual sucede porque todavía no se incluyo el archivo `datos.php`.

Ya en la segunda salida se ve la frase `el clavel blanco`, ya que la definición y contenido de las variables se incluyeron con exito para este segundo `echo`.

Ejercicio 5

Este programa cuenta con tres archivos, 2 de ellos `php` y uno `txt`. El `txt` guarda el valor inicial del contador, el `php` contador se encarga de abrir el valor del `txt` e ir sumando 1 cada vez que es abierto el archivo `visitas`, el cual crea el archivo `html` y llama al `contador.php` para que se muestre el valor actualizado del mismo de la siguiente manera: Cantidad de visitas:3 por ejemplo.

PHP: arrays, funciones

Ejercicio 1

Los dos códigos descritos producen la misma funcionalidad. Ambos crean un arreglo `$a`, el cual lo llenan con los datos `color`, `sabor`, `forma`, `nombre` y un número. La diferencia es que el segundo código hace asignación individual a cada posición del arreglo, `pej: $a['color'] = 'rojo';`

El primer código lo realiza en una sola línea llenando cada posición dentro de un solo paréntesis, en este caso cada asignación en vez de ser con un `=` es con un `=>`. `Pej: ('color' => 'rojo',)`

Ejercicio 2

a- bar 1

b- 5942

c- vacío

Ejercicio 3

a- Has entrado en esta pagina a las 19 horas, con 35 minutos y 42 segundos, del 27/10/2021

En este caso muestra el valor de la fecha actual obtenido con `getdate()` y con `$fun[]`, este valor de la hora se encuentra sin embargo en el horario de región Europea, por lo que hay que restarle 5hs a la misma.

b- $5+6=11$

Este crea la función sumar, la cual recibe 2 parámetros 5 y 6, los suma y los muestra

Ejercicio 4

Esta función comprueba si se le ingresa un nombre valido el cual debe ser de mas de 3 caracteres y menor a 20 haciendo dos if: el primero se encarga de controlar el tamaño del script y el segundo esta dentro de un for que controla cada caracter del valor del parametro de entrada y chequea por cada caracter revisa que se encuentre dentro de los caracteres permitidos dentro de `$permitidos`.

El programa luego de llamar a la función devuelve el nombre del parametro y si es valido o no. Pej: franco es válido