Práctica N° 4 PHP

1-

**Variables y su tipo**

1. **$a**: Booleano (TRUE)
2. **$b**: String ("xyz")
3. **$c**: String ('xyz')
4. **$d**: Entero (12)
5. **$f**: Entero (resultado de la función **doble($d++)**)
6. **$g**: Entero (resultado de **$f += 10**)

**Operadores**

1. **Asignación**: **=**
2. **Aritméticos**: **+**, **\***, **++**, **+=**
3. **Concatenación**: **.**
4. **Comparación**: **==**
5. **Ternario**: **? :**
6. **Incremento**: **++**
7. **Post-incremento**: **$d++**

**Funciones y sus parámetros**

1. **doble($i)**: Función definida por el usuario que toma un parámetro **$i** y devuelve el valor de **$i** multiplicado por 2.
2. **gettype($var)**: Función incorporada de PHP que devuelve el tipo de la variable **$var**.
3. **is\_int($var)**: Función incorporada de PHP que comprueba si **$var** es de tipo entero.
4. **is\_string($var)**: Función incorporada de PHP que comprueba si **$var** es de tipo string.

2) a) los códigos **1)**, **2)** y **3)** son equivalentes en términos de su salida por pantalla y funcionamiento, a pesar de las diferencias en la estructura del bucle y el punto de inicialización/incremento de la variable **$i**.

b) Todos los código 1), 2), 3), y 4) son equivalentes en términos de su salida por pantalla y funcionalidad, aunque emplean diferentes estructuras y mecanismos para lograr el mismo resultado. En cada uno de ellos, el valor de **$i** se incrementa de 1 a 10 y se imprime en cada iteración.

c) los dos códigos son equivalentes. Ambos imprimirán exactamente el mismo mensaje para los valores de **$i** que están siendo comprobados (0, 1, y 2).

3-

a)

Este código PHP genera dinámicamente una tabla HTML con las siguientes características:

* La tabla tiene un ancho del 90% del contenedor padre.
* La tabla tiene un borde de 1 píxel.
* Contiene 5 filas.
* Cada fila contiene 2 columnas.
* Cada celda de la tabla (**<td>**) está vacía, pero contiene un espacio en blanco (**&nbsp;**) para asegurar la visibilidad.

b)

El propósito de este código es crear una página web que solicite al usuario su edad a través de un formulario. Una vez que el formulario se envía, el script verifica si la edad ingresada es mayor o igual a 21, e imprime un mensaje correspondiente indicando si el usuario es mayor o menor de edad.

4-

* La primera parte “**El”** seguida de dos espacios en blanco se debe a la impresión de las variables no definidas.
* La segunda parte “**El clavel blanco”** se imprime después de incluir el archivo **datos.php**, donde las variables **$flor** y **$color** han sido definidas y contienen los valores **'clavel'** y **'blanco'**.

Por lo tanto, la salida completa del código es:

**El**

**El clavel blanco**

5-

**contador.php**

<?php

$archivo = "contador.dat";

$abrir = fopen($archivo, "r");

$cont = fread($abrir, filesize($archivo));

fclose($abrir);

$abrir = fopen($archivo, "w");

$cont = $cont + 1;

$guardar = fwrite($abrir, $cont);

fclose($abrir);

echo "<font face='arial' size='3'>Cantidad de visitas: ".$cont."</font>";

?>

**visitas.php**

<html>

<head></head>

<body>

<?php include("contador.php"); ?>

</body>

</html>

PHP: arrays, funciones

1- Sí, los dos fragmentos de código son equivalentes. Ambos generan el mismo array con los mismos elementos y claves.

2-

* A) bar, 1
* B) 5, 9, 42
* C) no tiene salida visible

3-

* A) Has entrado en esta página a las [horas] horas, con [minutos] minutos y [segundos] segundos, del [día]/[mes]/[año]
* B) 5+6=11

4- La función **comprobar\_nombre\_usuario** comprueba si un nombre de usuario dado es válido o no. Aquí está el análisis de la función:

* Comprueba si el tamaño del nombre de usuario está dentro del rango permitido (entre 3 y 20 caracteres).
* Comprueba si el nombre de usuario contiene solo caracteres permitidos. Los caracteres permitidos son letras minúsculas y mayúsculas del alfabeto inglés, dígitos del 0 al 9, y los caracteres especiales **-** y **\_**.
* Si el nombre de usuario es válido, la función devuelve **true**. Si no es válido, la función devuelve **false**.

Script de prueba:

<?php

function comprobar\_nombre\_usuario($nombre\_usuario){

// Compruebo que el tamaño del string sea válido.

if (strlen($nombre\_usuario) < 3 || strlen($nombre\_usuario) > 20){

echo $nombre\_usuario . " no es válido<br>";

return false;

}

// Compruebo que los caracteres sean los permitidos.

$permitidos = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-\_";

for ($i = 0; $i < strlen($nombre\_usuario); $i++){

if (strpos($permitidos, substr($nombre\_usuario, $i, 1)) === false){

echo $nombre\_usuario . " no es válido<br>";

return false;

}

}

echo $nombre\_usuario . " es válido<br>";

return true;

}

// Pruebas de la función comprobar\_nombre\_usuario

comprobar\_nombre\_usuario("usuario1"); // Válido

comprobar\_nombre\_usuario("user-name"); // Válido

comprobar\_nombre\_usuario("user\_name123"); // Válido

comprobar\_nombre\_usuario("us"); // No válido (demasiado corto)

comprobar\_nombre\_usuario("usuario\_completo\_con\_muchos\_caracteres"); // No válido (demasiado largo)

comprobar\_nombre\_usuario("usuario@123"); // No válido (carácter no permitido)

?>