

1.) En el siguiente código identificar: las variables y su tipo, los operadores, las funciones y sus parámetros, las estructuras de control, cuál es la salida por pantalla

```
<?php
function doble($i) {
    return $i*2;
}
$a = TRUE;
$b = "xyz";
$c = 'xyz';
$d = 12;
echo gettype($a);
echo gettype($b);
echo gettype($c);
echo gettype($d);
if (is_int($d)) {
    $d += 4;
}
if (is_string($a)) {
    echo "Cadena: $a";
}
$d = $a ? ++$d : $d*3;
$f = doble($d++);
$g = $f += 10;
echo $a, $b, $c, $d, $f, $g;
?>
```

- a) \$a es booleana (bool)
\$b es cadena (string)
\$c es cadena (string)
\$d es número entero (int)
- b)
- * operador de multiplicación
 - + operador de suma
 - ? operador ternario
 - ++ operador de incremento
 - = operador de asignación.
- c)
- doble(\$i): función que toma un parámetro \$i y retorna \$i * 2.
gettype(\$variable): función nativa de PHP que devuelve el tipo de la variable.
is_int(\$variable): función nativa de PHP que verifica si una variable es de tipo entero.
is_string(\$variable): función nativa de PHP que verifica si una variable es de tipo cadena.

d)

if (is_int(\$d)) (verifica si \$d es entero.)

if (is_string(\$a)) (verifica si \$a es de tipo cadena.)

e)

boolean

string

string

integer

1

xyz

xyz

18

44

44

2)

Ejercicio 2:

Indicar si los siguientes códigos son equivalentes.

a)

```
<?php
$i = 1;
while ($i <= 10) {
    print $i++;
}
?>
```

```
<?php
$i = 1;
while ($i <= 10):
    print $i;
    $i++;
endwhile;
?>
```

```
<?php
$i = 0;
do {
    print ++$i;
} while ($i < 10);
?>
```

Los tres bloques de código son equivalentes en cuanto a la salida, ya que todos devuelven los números del 1 al 10. La única diferencia es la sintaxis y el valor inicial de \$i en el tercer bloque, pero como incrementa \$i antes de mostrar, el resultado final es el mismo.

b)

```
<?php
for ($i = 1; $i <= 10; $i++) {
    print $i;
}
?>
```

```
<?php
for ($i = 1; $i <= 10; print $i, $i++);
?>
```

```
<?php
for ($i = 1; ; $i++) {
    if ($i > 10) {
        break;
    }
    print $i;
}
?>
```

```
<?php
$i = 1;
for (;;) {
    if ($i > 10) {
        break;
    }
    print $i;
    $i++;
}
?>
```

Todos los bloques son equivalentes en términos de salida, ya que imprimen los números del 1 al 10. Aunque cada bloque tiene una estructura diferente, el resultado final es el mismo.

c)

```
<?php
...
...
if ($i == 0) {
    print "i equals 0";
} elseif ($i == 1) {
    print "i equals 1";
} elseif ($i == 2) {
    print "i equals 2";
}
?>
```

```
<?php
...
...
switch ($i) {
    case 0:
        print "i equals 0";
        break;
    case 1:
        print "i equals 1";
        break;
    case 2:
        print "i equals 2";
        break;
}
?>
```

Los dos códigos son funcionalmente equivalentes. Ya que las estructuras de control (if/elseif concatenados y un switch, son equivalentes).

Para códigos y aplicaciones grandes el switch posee más rendimiento porque requiere de menos verificación de condiciones.

3) Explicar para qué se utiliza el siguiente código:

a)

```
<html>
<head><title>Documento 1</title></head>
<body>
<?php
    echo "<table width = 90% border = '1' >";
    $row = 5;
    $col = 2;
    for ($r = 1; $r <= $row; $r++) {
        echo "<tr>";
        for ($c = 1; $c <= $col; $c++) {
            echo "<td>&nbsp;</td>\n";
        }
        echo "</tr>\n";
    }
    echo "</table>\n";
?>
</body></html>
```

Este código genera una tabla HTML.

Después recorre el número de filas especificado por \$row, inicia una nueva fila y por cada fila añade y crea las celdas correspondientes.

b)

```
<html>
<head><title>Documento 2</title></head>
<body>
<?php
if (!isset($_POST['submit'])) {
?>
    <form action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post">
    Edad: <input name="age" size="2">
    <input type="submit" name="submit" value="Ir">
    </form>
<?php
    }
else {
    Sage = $_POST['age'];
    if ($Sage >= 21) {
        echo 'Mayor de edad';
    }
    else {
        echo 'Menor de edad';
    }
}
?>
</body></html>
```

Este código PHP genera una página HTML con un formulario para que el usuario ingrese su edad. Luego, el programa determina si la persona es mayor o menor de edad

Como envía el formulario a `$_SERVER['PHP_SELF']`, los datos y la verificación se mostrarán (una vez enviado el form) recargando la misma página.

4)

Ejercicio 4:

Si el archivo `datos.php` contiene el código que sigue:

```
<?php
$color = 'blanco';
$flor = 'clavel';
?>
```

Indicar las salidas que produce el siguiente código. Justificar.

```
<?php
echo "El $flor $color \n";
include 'datos.php';
echo " El $flor $color";
?>
```

La salida será

“El

El clavel blanco”

El echo solo mostrará las variables definidas en `datos.php` una vez hecho el include `'datos.php'`.

5) -