Facultad Regional Rosario
Universidad Tecnológica Nacional

Entornos Gráficos

Trabajo Práctico N°4:

"PHP"

Grupo N°7

Alumno	**Legajo**
Cuello Alejo	45572
Jimenez Dana	29694
Piccoli Enzo	42850

Ejercicios:

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
<?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;
?>
```

Variables y sus tipos:

$a: boolean
$b: string
$c: string
$d: integer
$f: integer
$g: integer

Operadores:

Unarios:

++

Binarios:

*

=

+=

Ternarios:

? :

Funciones y sus parámetros:

doble($i): esta función cuando es llamada, recibe la variable $d con valor 17.

gettype($a): esta función recibe las distintas variables definidas en el código.
is_int($d): recibe la variable $d con valor 12.
is_string($a): recibe la variable con valor TRUE.

Estructuras de control:

if

Salida por pantalla:

booleanstringstringinteger1xyzxyz184444

2) Indicar si los siguientes códigos son equivalentes

a)

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

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

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

Sí, son equivalentes ya que muestran la misma salida en pantalla.

b)

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

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

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

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

Sí, son equivalentes. La primera sería la forma más clara de escribirla ya que separa las expresiones y la sentencia.

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;
}
?>
```

Sí, son equivalentes. En este caso en particular, la utilización de la estructura de control switch, muestra de una forma más clara la lógica a ejecutar.

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> </td>\n";
         }        echo "</tr>\n";
   }
   echo "</table>\n";
?>
</body></html>
```

Dicho código se utiliza para crear una tabla en HTML, estableciendo mediante las variables $row y $column, la cantidad de filas y columnas que tendrá respectivamente. Se utilizan dos estructuras for anidadas para definir primero las filas y luego las columnas dentro de las mismas.

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 {
   $age = $_POST['age'];
   if ($age >= 21) {
      echo 'Mayor de edad';
      }
   else {
      echo 'Menor de edad';
   }
}
?>
</body></html>
```

Dicho código se utiliza para mostrar un formulario en caso que no haya sido enviado. Una vez enviado un valor de edad mediante este formulario, se mostrará si corresponde a una mayoría o minoría de edad.

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 es la siguiente:

Warning**: Undefined variable $flor in **C:\xampp\htdocs\4.php** on line **7

Warning**: Undefined variable $color in **C:\xampp\htdocs\4.php** on line **7

El El clavel blanco

La salida muestra estas advertencias ya que se intenta acceder a las variables $flor y $color antes de ser inicializadas. Dichas variables se inicializan en el momento que se incluye el archivo datos.php.

5) Analizar el siguiente ejemplo: Contador de visitas a una página web

contador.php

```
<?
// Archivo para acumular el numero de visitas
$archivo = "contador.dat";
// Abrir el archivo para lectura
$abrir = fopen($archivo, "r");
// Leer el contenido del archivo
$cont = fread($abrir, filesize($archivo));
// Cerrar el archivo
fclose($abrir);
// Abrir nuevamente el archivo para escritura
$abrir = fopen($archivo, "w");
// Agregar 1 visita
$cont = $cont + 1;
// Guardar la modificación
$guardar = fwrite($abrir, $cont);
// Cerrar el archivo
fclose($abrir);
// Mostrar el total de visitas
echo "<font face='arial' size='3'>Cantidad de visitas:".$cont."</font>";
?>
```

visitas.php

```
<!-- Página que va a contener al contador de visitas  -->
<html>
<head></head>
<body>
<? include("contador.php")?>
</body>
</html>
```

En la misma carpeta, crear el archivo de texto contador.dat, con el valor inicial del contador y con permisos de lectura y escritura

PHP: arrays, funciones

1) Indicar si los siguientes códigos son equivalentes.

```
<?php
$a = array( 'color'  => 'rojo',
            'sabor'  => 'dulce',
            'forma'  => 'redonda',
            'nombre' => 'manzana',
            4
          );
?>
```

```
<?php
$a['color']  = 'rojo';
$a['sabor']  = 'dulce';
$a['forma']  = 'redonda';
$a['nombre'] = 'manzana';
$a[]         = 4;
?>
```

Sí, son equivalentes. Cabe destacar que ambas son matrices asociativas, teniendo un error en la última asignación ya que no se proporciona un conjunto de clave-valor.

2) En cada caso, indicar las salidas correspondientes:

a)

```
<?php
$matriz = array("x" => "bar", 12 => true);
echo $matriz["x"];
echo $matriz[12];
?>
```

bar1

b)

```
<?php
$matriz = array("unamatriz" => array(6 => 5, 13 => 9, "a" => 42));

echo $matriz["unamatriz"][6];
echo $matriz["unamatriz"][13];
echo $matriz["unamatriz"]["a"];
?>
```

5942

c)

```
<?php
$matriz = array(5 => 1, 12 => 2);
$matriz[] = 56;
$matriz["x"] = 42;  unset($matriz[5]);  unset($matriz);
?>
```

No tiene ninguna salida. La función unset eliminar el valor asociado con la key 5 dentro de la matriz, y luego elimina la matriz entera.

3) En cada caso, indicar las salidas correspondientes:

a)

```
<?php
$fun = getdate();
echo "Has entrado en esta pagina a las $fun[hours] horas, con $fun[minutes] minutos y $fun[seconds]
segundos, del $fun[mday]/$fun[mon]/$fun[year]";
?>
```

Has entrado en esta pagina a las 2 horas, con 47 minutos y 41 segundos, del 10/5/2022

b)

```
<?php
function sumar($sumando1,$sumando2){
  $suma=$sumando1+$sumando2;
  echo $sumando1."+".$sumando2."=".$suma;
}
sumar(5,6);
?>
```

5+6=11

4) Analizar la siguiente función, y escribir un script para probar su funcionamiento

```
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;
}
```

El script se encuentra en el archivo arrays.4.php de esta carpeta.