**Universidad Tecnológica Nacional**

**FRRO**



Electiva: Entornos Gráficos

TRABAJO PRACTICO N°1

HTML

INTEGRANTE: ALANIZ WALTER -    32.916

DOCENTES: DANIELA DIAZ; JULIAN BUTTI

PRÁCTICA N°4 PHP:

Variables, tipos, operadores, expresiones, estructuras de control PHP: arrays, funciones.

## Ejercicio 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

• las variables y su tipo

$a = TRUE; - BOOLEAN.

$b = "xyz" - STRING

$c = 'xyz'- STRING

$d = 12- INTEGER

$f – INTEGER

$g - INTEGER

• los operadores :

+= 4 (UNARIO)

$i\*2 (BINARIO)

$d = $a ? ++$d : $d\*3; (TERNARIO)

• las funciones y sus parámetros

function doble($i)

{ return $i\*2; } –(PARAMETROS: $I).

• las estructuras de control:

if (is\_int($d)) { $d += 4; }

if (is\_string($a)) { echo "Cadena: $a"; }

• cuál es la salida por pantalla

booleanstringstringinteger1xyzxyz184444.

## Ejercicio 2: Indicar si los siguientes códigos son equivalentes

<?php

$i = 1;

while ($i <= 10) {

 print $i++;

}

?>

Salida: 12345678910

<?php

$i = 1;

while ($i <= 10):

 print $i;

 $i++;

endwhile;

?>

Salida: 12345678910

<?php

$i = 0;

do {

 print ++$i;

} while ($i<10);

?>

**SALIDA**: 12345678910

Los tres códigos son equivalentes entregan la misma salida.

b)

<?php

for ($i = 1; $i <= 10; $i++) {

 print $i;

}

?>

Salida: 12345678910

<?php

for ($i = 1; $i <= 10; print $i, $i++) ;

?>

Salida: 12345678910

<?php

for ($i = 1; ;$i++) {

 if ($i > 10) {

 break;

 }

 print $i;

}

?>

Salida: 12345678910

<?php

$i = 1;

for (;;) {

 if ($i > 10) {

 break;

 }

 print $i;

 $i++;

}

?>

Salida: 12345678910

Todos los códigos tienen la misma salida, por lo tanto son equivalentes.

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 equivalentes.

## Ejercicio 3: Explicar para qué se utiliza el siguiente código:

<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 muestra en el lado del cliente una tabla de 5 filas y 2 columnas. La construcción se hace del lado del servidor.

<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>

Este código muestra un formulario para enviar con el campo

EDAD:……………..

Cuando se envía el formulario y si la edad ingresada es mayor igual a 21, nos mostrara un cartelito ‘mayor de edad’

De lo contario se mostrara ‘menor de edad’

## 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 es:

“El ”

“El clavel blanco”.

En la primera salida no tenemos definido las variables $flor, $color,

Pero antes de la segunda salida tenemos un include que define en su archivo las varibles $flor, $color.

# PHP: arrays, funciones

Ejercicio 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;

?>

Ambos códigos tienen la misma finalidad, en el primero crea la array con sus elementos.

En el segundo el array está creada y lo que se hace línea a línea es ir llenando los campos del array.

## Ejercicio 2: En cada caso, indicar las salidas correspondientes:

<?php

$matriz = array("x" => "bar", 12 => true);

echo $matriz["x"];

echo $matriz[12];

?>

Salida: bar ture

<?php

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

echo $matriz["unamatriz"][6];

echo $matriz["unamatriz"][13];

echo $matriz["unamatriz"]["a"];

?>

Salida: 5 9 42

<?php

$matriz = array(5 => 1, 12 => 2);

$matriz[] = 56;

$matriz["x"] = 42; unset($matriz[5]); unset($matriz);

?>

Salida: 2 42

## Ejercicio 3: En cada caso, indicar las salidas correspondientes:

<?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]";

?>

Salida: "Has entrado en esta página a las 20 horas, con 54 minutos y 30 segundos, del 02/10/2022

<?php

function sumar($sumando1,$sumando2){

 $suma=$sumando1+$sumando2;

 echo $sumando1."+".$sumando2."=".$suma;

}

sumar(5,6);

?>

Salida: 5 + 6 = 11

## Ejercicio 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;

}