



# **Universidad Tecnológica Nacional**

## **Regional Rosario**

Cátedra: Entornos Gráficos  
Ingeniería en Sistemas de Información

### **Práctica N°4 - PHP**

Año: 2021

Docentes: Daniela Diaz - Julián Butti

Comisión: 4E03

#### **INTEGRANTES**

<b>Legajo</b>	<b>Apellido y nombre</b>	<b>Email</b>
41798	Schar, Juan	juanschar@gmail.com
42002	Scarafía, Mario Carande	mdc.mariio@gmail.com
42293	Caciorgna, Gerardo	gdcaciorgna@hotmail.com
46483	Vilchez, Joaquin	joaquinvilchez95@gmail.com
42133	Zoch, Leandro	leandrozoch18@gmail.com

Índice:

<b>Ejercicio 1</b>	<b>2</b>
Variables	3
Operadores	3
Funciones	4
Estructuras de control	4
Salida por pantalla	5
<b>Ejercicio 2</b>	<b>5</b>
<b>Ejercicio 3</b>	<b>6</b>
Ejercicio 3.a	6
Ejercicio 3.b	7
Ejercicio 4	8
Ejercicio 5	8
<b>Segunda Parte: Manejo de Arreglos</b>	<b>9</b>

# Práctica N°4 - PHP

## 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

```
<?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

Variable	Tipo
\$a	bool.
\$b	String Reemplaza variables que puede haber dentro del string
\$c	String literal. No reemplaza variables
\$d	Integer
\$f	Integer
\$g	Integer

---

## Operadores

Operador	Descripción
----------	-------------

<b>*</b>	Operador de multiplicación
<b>+=</b>	Equivale a sumar la misma variable más un argumento dado.
<b>,</b>	Concatena cadenas para imprimir por pantalla.
<b>++\$arg</b>	Incrementa el valor en 1 y luego realiza la asignación
<b>\$arg++</b>	Primero realiza la asignación y luego incrementa el valor en 1.

## Funciones

<b>Función</b>	<b>Argumentos</b>
<b>echo('texto')</b>	String a escribir por pantalla
<b>doble (\$arg)</b>	Valor numérico sobre el cual se devolverá el doble de su resultado
<b>is_int(\$arg)</b>	Devuelve 0 o 1 si el argumento es entero.
<b>is_string(\$arg)</b>	Devuelve 0 o 1 si el argumento es una cadena.

## Estructuras de control

1. **if{} - Sentencia if tradicional**
2. **\$a1? a2: a3    Abreviación de sentencia if.**

## Salida por pantalla

booleanstringstringinteger1xyzxyz184444

## Ejercicio 2

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);
?>
```

**Son equivalentes.** En los 3 casos se imprimen los números del 1 al 10 por pantalla.

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++;
}
?>
```

**Son equivalentes.** En los 4 casos se imprimen los números del 1 al 10 por pantalla.

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;
}
?>
```

**Son equivalentes.** La única diferencia es en las estructuras de control utilizadas.

## Ejercicio 3

### Ejercicio 3.a

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>
```

En un documento html se crea una tabla que ocupa el 90% del espacio y tiene un borde de 1 px. Posee 5 filas y dos columnas, definidas en dos variables. Por último, utilizando dos estructuras for anidadas, se crean 5 filas en la tabla y a cada celda se le inserta el carácter /nbsp;

La salida por pantalla sería la siguiente:

/nbsp/	/nbsp/
/nbsp/	/nbsp/
/nbsp/	/nbsp/
/nbsp/	/nbsp/
/nbsp/	/nbsp/

### Ejercicio 3.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>
```

En primer lugar verifica si en el documento actual se hizo el submit del formulario (haciendo click sobre el botón submit). Entonces, la primera vez que se ejecuta la página, como todavía no se hizo el submit, esa condición está en falso, por lo que al negarlo el if ingresa por el camino verdadero,



Luego, el usuario ingresará su edad y hará click en el botón “Ir”, se enviará el submit, y se le devolverá por pantalla si es mayor o 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 final del echo será “El clavel blanco”, una vez cargado el include de `datos.php`.

## Ejercicio 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>
```

Respuesta:

El código mencionado cuenta la cantidad de visitas a partir del número guardado en el disco dentro del archivo contador.dat. Luego, por cada usuario que solicita la página, se lee este archivo en modo lectura y se obtiene el número de visitas actual. Luego lo abre en modo escritura, para incrementar el contador en 1 y luego volver a guardarlo en disco.

Por último, se muestra por pantalla la cantidad de visitas hechas hasta el momento utilizando la función echo.

Es importante destacar, que en caso de que el navegador no tenga permisos de escritura, este no podrá ejecutar la función fwrite para guardar el nuevo número de visitas. En cambio, sí podrá ejecutar la función fopen, que no necesita permisos especiales.

## Ejercicio 1 - Parte 2

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;
?>
```

Si, ambos códigos son equivalentes.

## Ejercicio 2 - Parte 2

Indicar las salidas

a)

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

La salida es: "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"];
?>
```

La salida es "5942"

c)

```
<?php
$matriz = array(5 => 1, 12 => 2);
```

```
$matriz[] = 56;  
$matriz["x"] = 42; unset($matriz[5]); unset($matriz);  
?>
```

Este código no genera salidas, solo crea el arreglo y luego lo borra con la función unset().

## Ejercicio 3

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]";  
?>
```

Salida: Has entrado en esta pagina a las 20 horas, con 30 minutos y 36 segundos, del 12/5/2021

b.

```
<?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;
}

```

```

$nombres = array(1 => "gdcaciorgna", 2 => 'gdcaciorgna@hotmail.com', 3
=> 'ge', 4 => 'gerardoduiliocaciorgna@hotmail.com');
for($i = 1; $i <= sizeof($nombres); $i++){
    echo "<br>Prueba $i:";
    comprobar_nombre_usuario($nombres[$i]);
}

```

Salida:

Prueba 1:gdcaciorgna es válido

Prueba 2:gdcaciorgna@hotmail.com no es válido

Prueba 3:ge no es válido

Prueba 4:gerardoduiliocaciorgna@hotmail.com no es válido