Práctica 4 – PHP (parte 1)

PHP: variables, tipos, operadores, expresiones, estructuras de control

PHP: arrays, funciones

Universidad Tecnológica Nacional – Facultad Regional Rosario

Asignatura: Entornos Gráficos

Año 2022 – 1° Cuatrimestre

Alumno: Tadeo Schlieper

# Índice

Índice 2

PHP: variables, tipos, operadores, expresiones, estructuras de control 3

Ejercicio 1 3

Ejercicio 2 3

Ejercicio 3 4

Ejercicio 4 4

Ejercicio 5 5

PHP: arrays, funciones 5

Ejercicio 1 5

Ejercicio 2 5

Ejercicio 3 5

Ejercicio 4 6

# PHP: variables, tipos, operadores, expresiones, estructuras de control

## Ejercicio 1

*En el siguiente código identificar:*

* *las variables* (en verde declaraciones y usos) *y su tipo* (comentado en cian)
* *los operadores* (en azul)
* *las funciones* (en morado declaraciones, en magenta invocaciones) *y sus parámetros* (en rojo declaraciones, en naranja usos dentro de la función)
* *las estructuras de control* (en amarillo)
* *cuál es la salida por pantalla*

Código:

<?php

function doble($i) { // Parámetros por valor: $i

return $i\*2;

}

$a = TRUE; // Tipo de variable $a: boolean

$b = "xyz"; // Tipo de variable $b: string

$c = 'xyz'; // Tipo de variable $c: string

$d = 12; // Tipo de variable $d: integer

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;

?>

Salida:

boolean

string

string

integer

1

xyz

xyz

18

44

44

## Ejercicio 2

*Indicar si los siguientes códigos son equivalentes.*

1. Los 3 códigos producen la misma salida. Pero tienen una diferencia respecto al valor de la variable que utilizan. La variable $i en los primeros 2 códigos inicia en 1 y termina valiendo (fuera del bucle) 11, en cambio en el tercer código la variable inicia en 0 y termina valiendo (dentro y fuera del bucle) 10.
2. Los 4 códigos son equivalentes pues producen la misma salida (y la misma salida que los del apartado anterior) y la variable que utilizan tiene los mismos valores dentro y fuera del bucle.
3. Los 2 códigos son equivalentes, puesto que comparan la variable utilizada por igualdad contra los mismos valores y producen la misma salida.

## Ejercicio 3

*Explicar para qué se utiliza el siguiente código.*

1. Se utiliza para generar, dentro del elemento body del HTML, una tabla (etiqueta table) de ancho 90% con borde, con 5 filas (elementos tr) y cuyas filas tendrán 2 columnas (celdas de datos; elemento td) sin contenido en ellas más que un espacio en blanco.
2. Ejecuta código condicional, basado en si existe la variable “$\_POST['submit']”, lo cual indicaría que se recibió un parámetro por método POST llamado “submit”.

* Si no existe dicha variable, la acción del formulario del HTML final será enviar los datos al propio archivo (el cual es realmente PHP), de manera que se recibirían los parámetros “age” y “submit” (determinados por los elementos input del HTML) cuando el usuario lo envíe.
* Si sí existe dicha variable, el formulario no tendrá acción, pero en su lugar se tomará el valor recibido en el parámetro POST llamado “age”, y el código mostrará en el HTML “Mayor de edad” o “Menor de edad” si el valor es mayor incluido o menor a 21 respectivamente.

## Ejercicio 4

*Si el archivo datos.php contiene el código que sigue… Indicar las salidas que produce el siguiente código. Justificar.*

Salida producida en consola:

PHP Notice: Undefined variable: flor in php shell code on line 1

Notice: Undefined variable: flor in php shell code on line 1

PHP Notice: Undefined variable: color in php shell code on line 1

Notice: Undefined variable: color in php shell code on line 1

El

El clavel blanco

La razón es que la primera sentencia del código quiere mostrar dos variables que todavía no están definidas: $flor y $color. Por eso PHP informa esto (sólo se ve en consola, no en HTML) y luego muestra “El ”. La segunda sentencia busca el archivo datos.php y se ejecuta, por lo cual se declaran con valores las variables $flor y $color. Finalmente, la tercer sentencia usa ambas variables, mostrando “ El clavel blanco”.

## Ejercicio 5

*Analizar el siguiente ejemplo: Contador de visitas a una página web. contador.php. visitas.php. En la misma carpeta, crear el archivo de texto contador.dat, con el valor inicial del contador y con permisos de lectura y escritura.*

Se puede utilizar esto en páginas web HTML, y tener diferentes contadores para cada página o grupo de páginas, si tuviéramos un archivo contador.dat para cada grupo, lo cual requeriría que cada grupo esté en una carpeta única junto a su contador.dat, y/o se transformara el código de contador.php para que se ejecute dentro de una función cuyo parámetro sería el nombre del archivo de guardado.

# PHP: arrays, funciones

## Ejercicio 1

*Indicar si los siguientes códigos son equivalentes.*

Sí, ambos códigos son equivalentes. Ambos producen la declaración de la variable array $a. Dicho array tendrá una parte asociativa y otra indexada. En la parte asociativa tendrá las claves “color”, “sabor”, “forma” y “nombre”, con sus respectivos valores. Y en la parte indexada tendrá un elemento de índice “0”, cuyo valor será 4.

## Ejercicio 2

*En cada caso, indicar las salidas correspondientes.*

1. Salida (valor bool marcado en azul):

bar

1

1. Salida:

5

9

42

1. No produce salida. La primera sentencia declara la variable array $matriz indexada, con índices 5 y 12, y valores respectivos 1 y 2. La segunda sentencia agrega el siguiente elemento (índice 13) con valor 56. Luego agrega como parte asociativa la clave “x” y valor 42. Luego se remueve el elemento del array con índice 5. Por último se remueve la variable array completamente.

## Ejercicio 3

*En cada caso, indicar las salidas correspondientes.*

1. Salida:

Has entrado en esta pagina a las 17 horas, con 54 minutos y 30 segundos, del 27/4/2022

1. Salida:

5+6=11

## Ejercicio 4

*Analizar la siguiente función, y escribir un script para probar su funcionamiento.*

Script PHP para comprobar el funcionamiento de dicha función:

<?php

// Declaración de un array donde cada clave es un nombre de usuario a probar y su valor es el resultado esperado de la función.

$nombres\_prueba = array(

"usuArio\_1" => true,

"usuArio 1" => false,

"123-user" => true,

"123&user" => false,

"\_username" => true

);

// Comparación de cada valor esperado con el resultado real de la función.

foreach ($nombres\_prueba as $nombre => $es\_valido\_esperado) {

if (comprobar\_nombre\_usuario($nombre) === $es\_valido\_esperado) {

$resultados\_esperados[] = $nombre;

} else {

$resultados\_no\_esperados[] = $nombre;

}

}

// Muestra resultado final del script.

if (isset($resultados\_no\_esperados)) {

echo "La función no dio el resultado esperado para los siguientes nombres: \"" . implode("\", \"", $resultados\_no\_esperados) . "\".";

} elseif (isset($resultados\_esperados)) {

echo "La función funcionó correctamente con todos los nombres.";

} else {

echo "No se introdujo ningún nombre para comprobar la función.";

}

¿>