

DESARROLLO WEB SERVIDOR (DWS)



CURSO 2017-2018

10 / 12 / 2017

---

PRÁCTICA TEMA 2  
INTRODUCCIÓN  
A PHP

Por:  
José Luis Ferrete Olarte

**Licencia Creative Commons BY-NC-SA**

Los contenidos de este documento se publican bajo licencia CC BY-NC-SA: Creative Commons Reconocimiento – NoComercial – CompartirIgual.

**Cedemos los siguientes derechos sobre la obra:**

Derecho de reproducción, distribución y comunicación pública sobre la obra.

Derecho a incorporarla en una o más obras conjuntas o bases de datos y para su reproducción en tanto que incorporada a dichas obras conjuntas o bases de datos.

Derecho para efectuar cualquier transformación sobre la obra y crear y reproducir obras derivadas.

Derecho de distribución y comunicación pública de copias o grabaciones de la obra, como incorporada a obras conjuntas o bases de datos.

Derecho de distribución y comunicación pública de copias o grabaciones de la obra, por medio de una obra derivada.

**Siempre que lo hagas con estas condiciones:**

Reconocer la autoría, especificando la firma y el autor que lo publica (*"D. José Luis Ferrete Olarte"*). Si es en formato digital, debes añadir un enlace al contenido original.

Compartir bajo la misma licencia. Si reproduces o remezclas esta obra, sólo puedes distribuir la obra generada bajo una licencia como ésta (con las mismas condiciones). Hemos realizado este trabajo con fines educativos y queremos que, si nos utilizas como base para tus creaciones, también contribuyas a la comunidad difundiéndolas con licencias libres. Así, aprenderemos todos juntos.

No se permite un uso comercial de la obra original, ni de las posibles obras derivadas.

Ante cualquier duda sobre las condiciones de cesión de derechos, consultar: <https://creativecommons.org/licenses/by-nc-sa/2.5/es/legalcode.es>

## INDICE

I.	Introducción .....	3
II.	Ejercicios.....	4
A.	Crea una página PHP con cuatro variables numéricas y calcule el valor medio de las 4. Imprime el resultado por pantalla .....	4
B.	Crea una página PHP que asigne un número aleatorio a una variable (función rand) y devuelve los siguientes resultados: .....	5
C.	Crea una página PHP que calcule e imprima el área y el perímetro de un rectángulo. Para ello, defina dos variables que reciban un número aleatorio entre 0 y 12 (función rand) y haga los cálculos necesarios.....	6
D.	Indique cuáles de los siguientes nombres de variables no son válidos en php y por qué. ....	7
E.	Cree un fichero PHP que permita conocer toda la información de una variable (usa la función var_dump()), de forma que pueda obtenerse por pantalla una salida similar a la siguiente: .....	8
F.	Indique el resultado, tanto tipo como valor, de las siguientes conversiones: .....	9
G.	Indique el resultado de los siguientes casting: .....	10

## I. Introducción

El objetivo fundamental de esta práctica es la toma de contacto con el lenguaje de programación JavaScript mediante la resolución de una serie de ejercicios. Trataremos de afrontarlos comprendiendo la metodología propuesta para la resolución de los mismos y tratando de interiorizar los nuevos conceptos, así como el lenguaje propio a utilizar.

PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa CGI, como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies. Aunque PHP puede hacer mucho más.

Existen principalmente tres campos principales donde se usan scripts de PHP.

Scripts del lado del servidor. Este es el campo más tradicional y el foco principal. Son necesarias tres cosas para que esto funcione. El analizador de PHP (módulo CGI o servidor), un servidor web y un navegador web. Es necesario ejecutar el servidor con una instalación de PHP conectada. Se puede acceder al resultado del programa de PHP con un navegador, viendo la página de PHP a través del servidor.

Scripts desde la línea de comandos. Se puede crear un script de PHP y ejecutarlo sin necesidad de un servidor o navegador. Solamente es necesario el analizador de PHP para utilizarlo de esta manera. Estos scripts también pueden usarse para tareas simples de procesamiento de texto. Véase la sección Uso de PHP en la línea de comandos para más información.

Escribir aplicaciones de escritorio. Probablemente PHP no sea el lenguaje más apropiado para crear aplicaciones de escritorio con una interfaz gráfica de usuario, pero si se conoce bien PHP, y se quisiera utilizar algunas características avanzadas de PHP en aplicaciones del lado del cliente, se puede utilizar PHP-GTK para escribir dichos programas.

PHP puede emplearse en todos los sistemas operativos principales, incluyendo Linux, muchas variantes de Unix (incluyendo HP-UX, Solaris y OpenBSD), Microsoft Windows, Mac OS X, RISC OS y probablemente otros más. PHP admite la mayoría de servidores web de hoy en día, incluyendo Apache, IIS, y muchos otros. Esto incluye cualquier servidor web que pueda utilizar el binario de PHP FastCGI, como lighttpd y nginx.

De modo que, con PHP, se tiene la libertad de elegir el sistema operativo y el servidor web. Además, se tiene la posibilidad de utilizar programación por procedimientos o programación orientada a objetos (POO), o una mezcla de ambas. Con PHP no se está limitado a generar HTML. Entre las capacidades de PHP se incluyen la creación de imágenes, ficheros PDF e incluso películas Flash (usando libswf y Ming) generadas sobre la marcha. También se puede generar fácilmente cualquier tipo de texto, como XHTML y cualquier otro tipo de fichero XML. PHP puede autogenerar estos ficheros y guardarlos en el sistema de ficheros en vez de imprimirlos en pantalla, creando una caché en el lado del servidor para contenido dinámico.

## II. Ejercicios

- A. Crea una página PHP con cuatro variables numéricas y calcule el valor medio de las 4. Imprime el resultado por pantalla

```
<?php
$num1=3;
$num2=5;
$num3=7;
$num4=11;
$media=($num1+$num2+$num3+$num4)/4;

echo "El valor medio de las 4 variables es $media";
?>
```

B. Crea una página PHP que asigne un número aleatorio a una variable (función rand) y devuelve los siguientes resultados:

1. Si el número es primo, muestra el mensaje: "El número es primo."

2. Si no lo es, muestra: "El número no es primo."

```
<?php
{
$numA=rand(1,30);

if(primo($numA))
echo "El numero ".$numA." es primo";
else
    echo "El numero ".$numA." NO es primo";
}
function primo($num)
{
    $cont=0;
    # Funcion que recorre todos los numero desde el 2 hasta el valor recibido
    for($i=2;$i<=$num;$i++)
    {
        if($num%$i==0)
        {
            # Si se puede dividir por algun numero mas de una vez, no es primo
            if(++$cont>1)
                return false;
        }
    }
    return true;
}
?>
```

C. Crea una página PHP que calcule e imprima el área y el perímetro de un rectángulo. Para ello, defina dos variables que reciban un número aleatorio entre 0 y 12 (función rand) y haga los cálculos necesarios.

```
<?php

$base=rand(0,12);
$altura=rand(0,12);

while($base==0)
{
    $base=rand(0,12);
}

while($altura==0)
{
    $altura=rand(0,12);
}

echo "El area del rectangulo es ".$base*$altura."<BR/>";

echo "El perimetro del rectangulo es ".$(($base*2)+($altura*2));

?>
```

D. Indique cuáles de los siguientes nombres de variables no son válidos en php y por qué.

<p>Posible nombre de variable -> Es incorrecto, porque usa espacios en blanco y no empieza por \$</p>

<p>\$4variable -> Es incorrecto, ya que empieza por un numero</p>

<p>\$\_4variable -> Es posible usarlo, aunque no me parece muy recomendable</p>

<p>\$variable4 -> Es correcto usarlo</p>

<p>\$otra -> Es correcto usarlo</p>

<p>\$1\_otra -> Es incorrecto, ya que empieza por un numero</p>

<p>\$variable\_de\_nombre\_muy\_largo -> Es correcto, aunque personalmente, usaría la regla del camello</p>

<p>\$ABC -> Es correcto su uso</p>

<p>\$ A B C -> Es incorrecto, ya que usa espacios en blanco</p>

<p>\$A\_y\_B\_x\_C -> Es correcto su uso</p>



E. Cree un fichero PHP que permita conocer toda la información de una variable (usa la función `var_dump()`), de forma que pueda obtenerse por pantalla una salida similar a la siguiente:

*Información de la variable "nombreVariable":*

*Strin (4) "Juan"*

*Contenido de la variable: Juan*

*Después de asignarle un valor nulo: NULL*

```
<?php
```

```
$palabra = 'Juan';  
echo "Informacion de la variable 'palabra'";  
var_dump($palabra);  
echo "Contenido de la variable: ".$palabra."<BR/>";  
$palabra = null;  
$result = varDumpToString($palabra);  
echo "Despues de asignarle un valor nulo: ".$result;
```

```
function varDumpToString($var) {
```

```
    ob_start();
```

```
    /*
```

Esta función activará el almacenamiento en búfer de la salida. Mientras dicho almacenamiento esté activo, no se enviará ninguna salida desde el script (aparte de cabeceras); en su lugar la salida se almacenará en un búfer interno.

El contenido de este búfer interno se puede copiar a una variable de tipo string usando `ob_get_contents()`. Para producir la salida de lo almacenado en el búfer interno se ha de usar `ob_end_flush()`. De forma alternativa, `ob_end_clean()` desechará de manera silenciosa el contenido del búfer.

```
    */
```

```
    var_dump($var);
```

```
    $result = ob_get_clean();
```

```
    /*
```

Obtiene el contenido del búfer actual y elimina el búfer de salida actual. `ob_get_clean()` básicamente ejecuta `ob_get_contents()` y `ob_end_clean()`. El búfer de salida debe estar iniciado por `ob_start()` con el indicador `PHP_OUTPUT_HANDLER_CLEANABLE` Si no, `ob_get_clean()` no funcionará.

```
    */
```

```
    return $result;
```

```
}
```

```
?>
```

F. Indique el resultado, tanto tipo como valor, de las siguientes conversiones:

*\$var = 34 + "1 manzana"*

*\$var = "200" + 0.5*

*\$var = "14" + 15*

*\$var = 20 + "manzanas"*

*\$var = 20. "manzanas"*

*<?php*

```
$var = 34 + "1 manzana";  
echo 'Expresion a evaluar : 34 + "1 manzana";  
var_dump($var);
```

```
$var = "200" + 0.5;  
echo ('Expresion a evaluar : "200" + 0.5');  
var_dump($var);
```

```
$var = "14" + 15;  
echo ('Expresion a evaluar : "14" + 15');  
var_dump($var);
```

```
$var = 20 + "manzanas";  
echo ('Expresion a evaluar : 20 + "manzanas"');  
var_dump($var);
```

```
$var = 20 . "manzanas";  
echo ('Expresion a evaluar : 20 . "manzanas"');  
var_dump($var);  
?>
```

Expresión a evaluar	Tipo devuelto	Valor devuelto
34 + "1 manzana"	Integer (Int)	35
"200" + 0.5	Float	200.5
"14" + 15	Integer (Int)	29
20 + "manzanas"	Integer (Int)	20
20 . "manzanas"	String (longitud 10)	'20manzanas'

G. Indique el resultado de los siguientes castings:

```
$var1=false  
(string) $var1
```

```
$var2="hola"  
(bool) $var2  
(int) $var2
```

```
$var3="3 metros"  
(float) $var3  
(bool) $var3
```

```
<?php
```

```
$var1=false;  
echo("variable1");  
var_dump($var1);  
//(string) $var1;  
echo("primer cast variable1");  
var_dump((string)($var1));
```

```
$var2="hola";  
echo("variable2");  
var_dump($var2);  
//(bool) $var2;  
echo("primer cast variable2 ");  
var_dump((bool)($var2));  
//(int) $var2;  
echo("segundo cast variable2");  
var_dump((int)($var2));
```

```
$var3="3 metros";  
echo("variable2");  
var_dump($var3);  
//(float) $var3;  
echo("primer cast variable3");  
var_dump((float)($var3));  
//(bool) $var3;  
echo("segundo cast variable3");  
var_dump((bool)($var3));
```

```
?>
```

Variable			Cast 1		Cast 2	
<i>Expresión</i>	<i>Tipo</i>	<i>Valor</i>	<i>Tipo</i>	<i>Valor</i>	<i>Tipo</i>	<i>Valor</i>
\$var1=false	boolean	false	string	"		
\$var2="hola"	string	'hola'	boolean	true	int	0
\$var3="3 metros"	string	'3 metros'	float	3	boolean	true