

PROGRAMACIÓN EN PHP

Programando con PHP

INTRODUCCIÓN

¿QUÉ SE PUEDE HACER CON PHP?

- ✖ Principalmente se utiliza para programar scripts.
- ✖ También se puede utilizar para escribir programas desde línea de comandos.
- ✖ Donde ha conseguido más aceptación es en Web:
 - + Procesamiento de formularios.
 - + Manipulación de cookies.
 - + Generación de páginas con contenido dinámico.

Programando con PHP

INSTALACIÓN

INSTALACIÓN DE UN SERVIDOR PHP

- ✖ Nosotros vamos a utilizar el servidor Xampp Server.
 - + X: funciona sobre cualquier sistema Operativo.
 - + A: es un servidor Apache
 - + M: utiliza MySQL como SGBD.
 - + P: utiliza PHP como lenguaje de programación.
 - + P: permite también Perl como lenguaje de programación



INSTALACIÓN DE UN SERVIDOR PHP

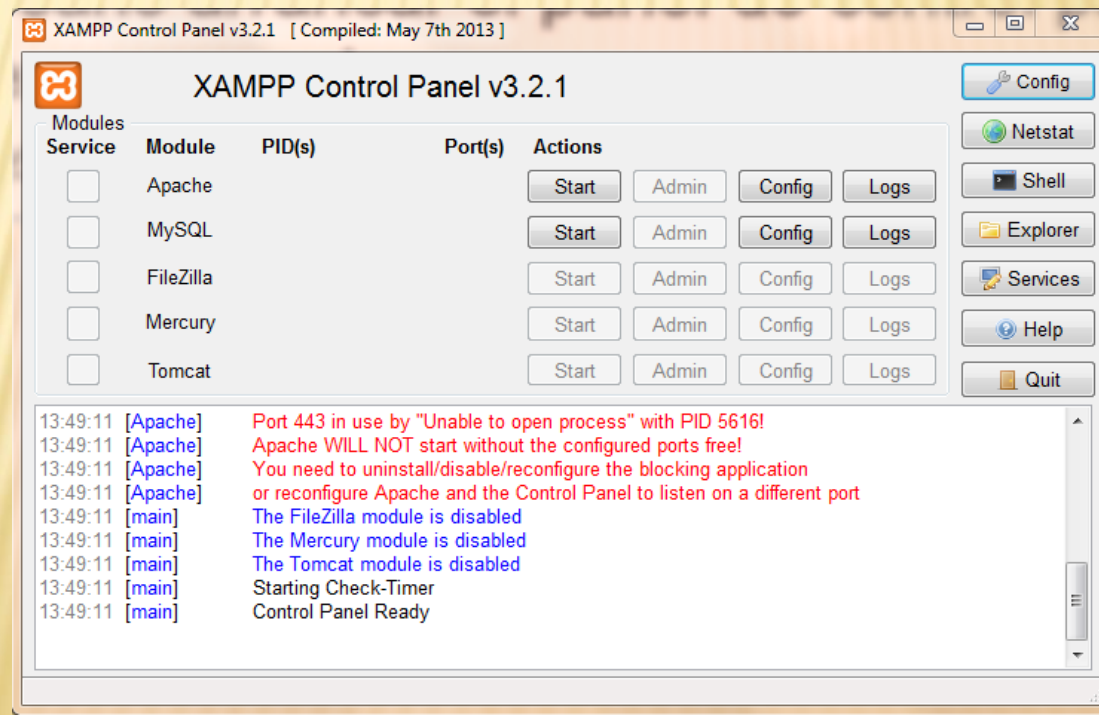
- ✖ Además Xampp Server es gratuito.
- ✖ Al ser código abierto, se puede descargar de cualquier página dónde lo tengan.

Recomendada:

www.apachefriends.org/es/xampp.html

ARRANCANDO EL SERVIDOR

- ✗ Una vez instalado Xampp Server es necesario arrancar el panel de control para poder arrancar los programas que necesitamos.



LOS DOCUMENTOS PHP

- ✖ Para que el servidor entienda que el documento solicitado contiene código PHP, la extensión de dicho documento será .php
- ✖ El documento deberá estar localizado en el directorio del servidor, normalmente:

`C:/xampp/htdocs`

Programando con PHP

FUNDAMENTOS DEL LENGUAJE PHP

FUNDAMENTOS DE PHP

- ✗ La sintaxis de PHP es muy parecida a otros lenguajes de programación como C o JAVA.
- ✗ Tiene un vocabulario relativamente pequeño y fácil de comprender.
- ✗ Ofrece las características básicas de un lenguaje de programación.
 - + Bucles
 - + Sentencias de control
 - + Etc.

FORMATO DEL CÓDIGO PHP

✖ Delimitadores:

- + El código PHP va incrustado en HTML por lo que necesita de etiquetas para ser encontrado:
- + Las etiquetas son:

```
<?php
```

```
    instrucciones  PHP
```

```
?>
```

```
1 <html>
2   <head>
3     <title> Prueba Apache </title>
4   </head>
5   <body>
6     <center>
7       <h2>
8         
9         <?php
10          echo "El módulo de PHP funciona correctametine...";
11        ?>
12       </h2>
13     </center>
14   </body>
15 </html>
```

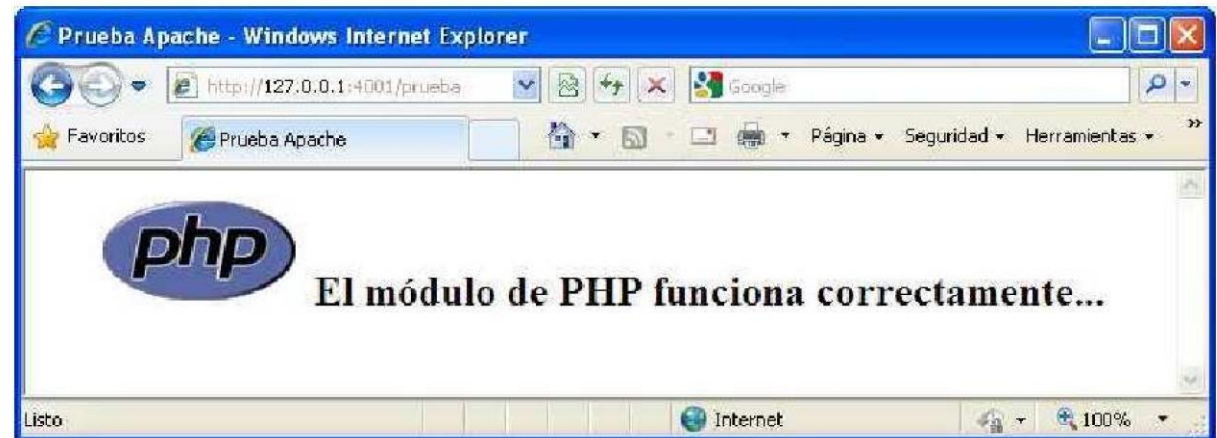


Fig. 3. Ejecución de prueba1.php

COMENTARIOS

✖ Existen en PHP dos tipos de comentarios

+ Comentario de una sola línea.

✖ //

+ Comentario de varias líneas.

✖ /* */

✖ **NOTA IMPORTANTE:** Todas las líneas de PHP deben terminar con punto y coma (;)

EJERCICIOS

- ✖ Crear una página en PHP que muestre el saludo: “hola mundo”. La página deberá tener
 - + Fondo rojo
 - + Texto tamaño 7 y color verde.

EJERCICIO

```
1 <HTML>
2   <HEAD>
3     <TITLE> Ejercicio 1 PHP </TITLE>
4   </HEAD>
5   <BODY bgcolor='red' text='green'>
6     <font size=7>
7       <?php
8         echo 'HOLA MI MUNDO'
9       ?>
10    </font>
11  </BODY>
12 </HTML>
```

Programando con PHP

SINTAXIS BÁSICA: VARIABLES

VARIABLES

- ✖ En PHP no es necesario declarar las variables antes de su utilización.
- ✖ Se crean en el instante que son utilizadas por primera vez.
- ✖ Para inicializarlas se utiliza el operador de asignación (=).
- ✖ A partir de ese momento podemos utilizarlas para cualquier cosa.

VARIABLES

- ✖ PHP es un lenguaje débilmente tipado:
 - + una variable podrá almacenar cualquier tipo de información.
- ✖ En PHP todas las variables van precedidas del símbolo \$, seguido de una letra o un (_)

Válidos	Inválidos	Motivo
\$Valor_actual	\$Valor actual	Contiene un espacio
\$NumeroDeDatos	\$#Datos	Contiene carácter no válido
\$N	\$2Saldcs	Empieza por número
\$n	\$Prueba,valor	Contiene carácter no válido

VARIABLES

- ✖ Hay que tener en cuenta que PHP distingue entre mayúsculas y minúsculas.
 - + No es la misma variable \$n que \$N

Programando con PHP

SINTAXIS BÁSICA: TIPOS DE DATOS

TIPOS DE DATOS

× Enteros

- + La variable de tipo 'integer' contiene números enteros que van desde los -2 billones hasta 2 billones.
- + Se pueden representar en decimal, octal o hexadecimal.

```

<HTML><HEAD>
    <TITLE>Variables enteras</TITLE>
</HEAD>
<BODY><CENTER>
    <H2>Formato de los números Enteros</H2>
    <TABLE BORDER='1' CELLPADDING='2' CELLSPACING="2">
        <TR ALIGN="right">
            <TD BGCOLOR="yellow">Decimal Posivito</TD>
            <TD><?php
                $num = 502; //número decimal
                echo $num; //mostramos el valor de num
            ?></TD>
        </TR>
        <TR ALIGN="right">
            <TD BGCOLOR="yellow">Decimal negativo</TD>
            <TD><?php
                $num = -502; //número decimal negati
                echo $num; //mostramos el valor de
            ?></TD>
        </TR>
        <TR ALIGN="right">
            <TD BGCOLOR="yellow">Decimal Octal (0512)</TD>
            <TD><?php
                $num = 0502; //número octal
                echo $num; //mostramos el valor de num
            ?></TD>
        </TR>
        <TR ALIGN="right">
            <TD BGCOLOR="yellow">Decimal Hexadecimal (0x12)</TD>
            <TD><?php
                $num = 0x12; //número hexadecimal
                echo $num; //mostramos el valor de num
            ?></TD></TR></TABLE></CENTER></BODY>
</HTML>

```

Formato de los números Enteros

Decimal Posívito	502
Decimal negativo	-502
Decimal Octal (0512)	322
Decimal Hexadecimal (0x12)	18

TIPOS DE DATOS

✖ Cadenas

- + Representan a un conjunto de 0 o más caracteres.
- + Se encierran entre (“) o (‘).

✖ Con “:

- + Podremos incluir en la cadena nombres de variables que serán evaluados a la hora de mostrar la cadena.
- + Podremos incluir en la cadena ‘.

✖ Con ‘:

- + No se evaluará nada de lo que aparezca en la cadena.

<H2>Trabajando Con Caderas de caracteres </H2>

<?php

```
$lenguaje = "PHP";  
$ver = "v5";  
echo "<B>Estamos 'trabajando' con $lenguaje $ver"  
echo "</B><BR><BR>";  
echo 'La variable $lenguaje contiene: ';  
echo "$lenguaje <BR>";  
echo 'La variable $ver contiene: ';  
echo $ver;
```

?>

Trabajando Con Caderas de caracteres

Estamos 'trabajando' con PHP v5

La variable \$lenguaje contiene: PHP

La variable \$ver contiene: v5

<h2> Comillas dobles y simples </h2>

<?php

```
$cadena=' un tesoro ';  
echo "la cadena '$cadena' contiene: $cadena"  
echo "<br><br>";  
echo "La cadena \"$cadena\" contiene: $cadena"  
echo "<br><br>";  
echo 'La cadena "$cadena" contiene: $cadena'  
echo '<br><br>';
```

?>

Comillas dobles y simples

la cadena ' un tesoro ' contiene: un tesoro

La cadena " un tesoro " contiene: un tesoro

La cadena "\$cadena" contiene: \$cadena

CARACTERES ESPECIALES

- ✖ Las cadenas pueden contener caracteres especiales como:

Código de escape	Significado
\b	Espacio hacia atrás (backspace)
\f	Cambio de página (form feed)
\n	Cambio de línea (line feed)
\r	Retorno de carro (carriage return)
\t	Tabulación
\\	Barra inversa (backslash)
\'	Comilla simple
\"	Comilla doble
\\$	Carácter \$

Programando con PHP

SINTAXIS BÁSICA: OPERADORES

OPERADORES ARITMÉTICOS

Operador	Ejemplo	Descripción
+	$\$a + \b	Suma dos operandos
-	$\$a - \b	Resta dos operandos
*	$\$a * \b	Multiplica dos operandos
/	$\$a / \b	Divide dos operandos
%	$\$a \% \b	Resto de la división entera
.	$\$a . \b	Concatenación

OPERADORES DE ASIGNACIÓN

Operador	Ejemplo	Descripción
=	$\$a = \b	$\$a$ toma el valor de $\$b$
+=	$\$a += \b	Equivale a $\$a = \$a + \$b$
-=	$\$a -= \b	Equivale a $\$a = \$a - \$b$
/=	$\$a /= \b	Equivale a $\$a = \$a / \$b$
*=	$\$a *= \b	Equivale a $\$a = \$a * \$b$
%=	$\$a \% = \b	Equivale a $\$a = \$a \% \$b$
.=	$\$a .= \b	Equivale a $\$a = \$a . \$b$

OPERADORES DE INCREMENTO Y DECREMENTO

Operador	Ejemplo	descripción
++	++\$a	Preincremento: incrementa \$a en uno y después devuelve \$a
	\$a++	Postincremento: devuelve \$a y después incrementa en uno \$a
--	--\$a	Predecremento: decrementa en uno \$a y después devuelve \$a
	\$a--	Postdecremento: devuelve \$a y después decrementa en uno \$a

OPERADORES DE COMPARACIÓN

Operador	Ejemplo	Devuelve TRUE cuando
==	\$a == \$b	Los operandos son iguales
!=	\$a != \$b	Los operandos son distintos.
===	\$a === \$b	Los operandos son idénticos: iguales y del mismo tipo.
!==	\$a !== \$b	Los operandos no son iguales o del mismo tipo.
<	\$a < \$b	El operando de la izquierda es menor que el de la derecha
>	\$a > \$b	El operando de la izquierda es mayor que el de la derecha
<=	\$a <= \$b	El operando de la izquierda es menor o igual que el de la derecha
>=	\$a >= \$b	El operando de la izquierda es mayor o igual que el de la derecha

OPERADORES LÓGICOS

Operador	Ejemplo	Devuelve TRUE cuando
&&	\$a && \$b	\$a y \$b son ambos true
And	\$a and \$b	
	\$a \$b	\$a o \$b son true
Or	\$a or \$b	
!	!\$a	\$a es false, niega el valor lógico de la variable
xor	\$a xor \$b	\$a es true o \$b es true, pero no lo son los dos a la vez

EJERCICIOS

- ✖ Crear una página Web en PHP que tome dos números y muestre la suma de ellos.
 - + Mostrar la suma dentro de una tabla del tipo:

Número 1	Número 2	Suma
25	13	38

- ✖ Crear una Web en PHP que muestre:

Operación	Resultado	Operación	Resultado
25 + 13	38	25 * 13	325
25 - 13	12	25 / 13	1.9230

Programando con PHP

LOS ARRAYS

ARRAYS POSICIONALES

- ✗ Conjunto de datos almacenados bajo el mismo nombre.
- ✗ Puesto que PHP es un lenguaje débilmente tipado, los arrays pueden estar compuestos por elementos de diferente tipo.

León	Seat		2.500	V6
0	1	2	3	4

ARRAYS POSICIONALES

- ✖ La declaración de un array posicional se hace como si cada posición fuera una variable independiente.
- ✖ El acceso a las distintas posiciones de un array posicional se hace de indicando entre [] la posición a la que se quiere acceder.
- ✖ Para asignar valor a la última posición del array no es necesario indicar dicha posición, dejaremos los corchetes vacíos.

ARRAYS POSICIONALES: ASIGNACIÓN

```
<?php
```

```
$matriz1[0]="rojo";  
$matriz1[1]="azul";  
$matriz1[2]="violeta";  
$matriz1[3]="";  
$matriz1[]="rosa";  
$matriz1[]="amarillo";
```

```
?>
```

ARRAYS POSICIONALES: ACCESO

Trabajando con matrices o *arrays*

	Posi 1	Posi 2	Posi 3	Posi 4	Posi 5	Posi 6
Matriz 1	rojo	azul	violeta		rosa	amarillo

```
<TD BGCOLOR="yellow">Matriz 1</TD>
<TD> <?php echo $matriz1[0] ?> </TD>
<TD> <?php echo $matriz1[1] ?> </TD>
<TD> <?php echo $matriz1[2] ?> </TD>
<TD> <?php echo $matriz1[3] ?> </TD>
<TD> <?php echo $matriz1[4] ?> </TD>
<TD> <?php echo $matriz1[5] ?> </TD>
```

```
</TR>
```

```
</TABLE></CENTER></BODY>
```

EJERCICIOS

- ✖ Crear el siguiente array posicional y mostrarlo dentro de una tabla igual a la siguiente:

0	1	2	3	4	5	6
Jorge	Pérez	35	1,77	80	Moreno	Soltero

ARRAYS ASOCIATIVOS

- ✖ Los índices de los arrays asociativos son cadenas.
- ✖ Cada posición está definida por el par (valor, clave)

León	Seat		2.500	V6
modelo	marca	fecha	Cc	Motor

ARRAYS ASOCIATIVOS

- ✖ La declaración de un array asociativo como si cada posición fuera independiente.
- ✖ Se indica en cada caso cómo se llamará la posición.
- ✖ El acceso a las distintas posiciones de un array asociativo se hace de indicando entre [] el nombre de la posición a la que se quiere acceder.

ARRAYS ASOCIATIVOS: ASIGNACIÓN

```
<?php
```

```
$matriz2['modelo']="León";  
$matriz2['marca']="Seat";  
$matriz2['fecha']=null;  
$matriz2['cc']="2.500";  
$matriz2['motor']="V6";
```

```
?>
```

ARRAYS ASOCIATIVOS: ACCESO

Trabajando con matrices o *arrays*

	Mtodelo	Marca	Fecha	CC	Motor
Matriz 2	León	Seat		2.500	V6

cha </TD>

```

<TABLE border="1">
  <TR>
    <th><?php echo $matriz2[0]>
    <th>Mtodelo
    <th>Marca
    <th>Fecha
    <th>CC
    <th>Motor
  </tr>
  <tr>
    <td>Matriz 2
    <td>León
    <td>Seat
    <td>
    <td>2.500
    <td>V6
  </tr>
  <tr align="center">
    <td bgcolor="yellow">Matriz 2</td>
    <td><?php echo $matriz2['modelo'] ?></td>
    <td><?php echo $matriz2['marca'] ?></td>
    <td><?php echo $matriz2['fecha'] ?></td>
    <td><?php echo $matriz2['cc'] ?></td>
    <td><?php echo $matriz2['motor'] ?></td>
  </tr>
</TABLE></CENTER></BODY>

```

EJERCICIOS

- ✗ Crear el siguiente array asociativo y mostrarlo dentro de una tabla igual a la siguiente:

Nombre	Apellido	Edad	Altura	Peso	Pelo	Estado
Jorge	Pérez	35	1,77	80	Moreno	Soltero

Programando con PHP

FUNCIONES ASOCIADAS A VARIABLES

FUNCIONES ASOCIADAS A VARIABLES

- ✗ PHP ofrece una serie de funciones muy útiles a la hora de trabajar con variables.
 - + Gettype(variable) : devuelve el tipo de la variable.
 - + Settype (variable, tipo): asigna a la variable el tipo.
 - + isset(variable): true si la variable ha sido inicializada con un valor.
 - + Unset(variable): destruye la variable.
 - + Empty(variable): true si la variable tiene valor.

FUNCIONES ASOCIADAS A VARIABLES

✖ Más funciones:

- + `is_int(variable)`: true si la variable es int.
- + `is_integer`, `is_long`, `is_float`, `is_numeric`, `is_real`, `is_double`.
- + `is_bool`, `is_array`, `is_string`, `is_objet`.

Programando con PHP

LAS CONSTANTES

CONSTANTES

- ✖ Crear una constante en PHP es tan fácil como usar una de las siguientes funciones:
 - + Define(constante, valor): crea la constante con el valor.
 - + Defined(constante): true si la constante está definida.
- ✖ Para hacer referencia a una constante no hay que poner \$ delante de su nombre.

CONSTANTES PREDEFINIDAS

Constante	Significado
PHP_VERSION	Cadena que representa la versión del intérprete de PHP en uso.
PHP_OS	Cadena con el nombre del sistema operativo en el que se está ejecutando el intérprete de PHP
TRUE	Verdadero
FALSE	Falso
E_ERROR	Información sobre errores distintos a los de interpretación del cual no es posible recuperarse.
E_PARSE	Informa que el intérprete encontró una sintaxis inválida en el archivo de comandos. Finaliza la ejecución.
E_NOTICE	Informa que se produjo algo incorrecto que puede provenir o no de un error. A ejecución continúa.
E_WARNING	Denota un error que no impide que continúe la ejecución.
E_ALL	Conjunto con todos los errores que se han producido.

PROGRAMACIÓN EN PHP