



Introducción a la programación en PHP



Somos y formamos expertos en T.I.

¿Qué es PHP ?

Hypertext **P**re-processoes (**PHP**) es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en un servidor.

¿Qué permite?

Procesar información, generar contenidos dinámicos, consultas a bases de datos.

•Requisitos

- Servidor web Apache (www.apache.org)
- con el módulo PHP (www.php.net)
- y la base de datos MySQL (www.mysql.com) si se desea crear páginas dinámicas.

•Otras utilidades

- Herramientas para la gestión de MySQL, como PHPMyAdmin (www.phpmyadmin.net)
- Editores de PHP, como DevPHP (www.sourceforge.net)
- Manuales de PHP y MySQL

•Lenguajes de *script*

- PHP es un lenguaje de *script* del lado del servidor. Otros lenguajes similares son ASP, JSP o ColdFusion.
- Los scripts PHP están incrustados en los documentos HTML y el servidor los interpreta y ejecuta antes de servir las páginas al cliente.
- El cliente no ve el código PHP sino los resultados que produce

Funcionamiento de PHP

- PHP es un lenguaje de programación Web diseñado para que un intérprete que lo analice y ejecute.
- El servidor web, cuando detecta una extensión asociada a PHP, envía esta página al intérprete y éste se la devuelve la respuesta según la solicitud.
- Paralelamente se puede trabajar con una base de datos si es necesario.

Componentes de un Programa

Código del cliente

Los **lenguajes de lado de cliente** se situarán en líneas de código que ha de interpretar el ordenador del usuario. Para el lado del cliente suelen ser principalmente JavaScript y VBScript, lenguajes que el propio navegador tiene la capacidad de interpretar.

Código del servidor

En los **lenguajes de lado de servidor** ejecutan las instrucciones en el servidor haciéndole llegar al navegador paginas de código HTML puro. Por otro lado la carga de trabajo se realiza en el servidor pudiendo llegar a sobrecargarlo.

Los lenguajes más comunes de programación de lado de servidor son ASP, PHP y Perl para CGI.

Esquema del funcionamiento de PHP



1.1 Ventajas de PHP

Ventajas	Desventajas
Capacidad de conexión con la mayoría motores de base de datos.	No posee adecuado manejo del Unicode.
Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).	Por su diseño dinámico no puede ser compilado y es muy difícil de optimizar.
Biblioteca nativa de funciones sumamente amplia e incluida.	Por sus características promueve la creación de código desordenado y complejo de mantener.
Tiene manejo de excepciones.	
Es libre y se maneja con POO	

1.1.1 Configuración de Apache (Wamp Server)

WampServer es un entorno de desarrollo web para Windows en el cual se podrán crear aplicaciones web con Apache, PHP y base de datos en MySQL (*motor de base de datos*).



1: Descargar

- Conectar a www.apache.org
- MYSQL.
- PHP.
- Netbeans o Eclicpse o Dreamweaver.

WampServer ofrece a los desarrolladores herramientas necesarias para realizar aplicaciones web de manera local, con un sistema operativo (Windows), un manejador de base de datos (MySQL), un software de programación script web PHP.

Para su instalación y configuración, visualizar “MANUAL WAMPSERVER”



Estructura de un programa y de lo que se procura no hacer

SINTAXIS BÁSICA DE PHP

Ejemplo:

```
<HTML>
```

```
  <HEAD>
```

```
    <TITLE>Mi primer programa en PHP</TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

```
    <?PHP print("<P>Hola mundo</P>");?>
```

```
  </BODY>
```

```
</HTML>
```



TecGurus

Somos y formamos expertos en T.I.

1.1.2 Introducción a HTML

HTML es un **lenguaje de etiquetas**. Indica al navegador como tiene que mostrar el contenido. El HTML separa el "contenido" (palabras, imágenes, audio, video, etc.) de la "presentación" (la definición del tipo de contenido y las instrucciones de cómo esos contenidos tienen que mostrarse). El HTML emplea un conjunto de elementos predefinidos que permiten identificar los distintos tipos de elementos.

HTML



Estos elementos contienen una o más etiquetas que contienen o expresan el contenido. Estas etiquetas suelen ir encapsuladas entre los símbolos $\langle \rangle$, y las etiquetas de cierre (que indican el final de un determinado contenido) están precedidas por una barra /.



1.2 Estructura de un programa

Un archivo PHP normalmente suele contener HTML y trozos de código PHP. PHP es un lenguaje “embebido” en HTML, es decir, incrustado en código HTML.

Etiquetas PHP

- `<?PHP ... ?>` Apertura y cierre de un bloque de PHP
- Las instrucciones se separan con un ;
- Comentarios: `/* ... */` y `//`
- Para imprimir echo y print

```
<html>
  <head>
    <title>Prueba de PHP</title>
  </head>
  <body>
    <?php echo '<p>Hola Mundo</p>'; ?>
  </body>
</html>
```

1.2.1 Librerías php

El uso de librerías, nos permiten agrupar varias funciones y variables en un mismo fichero, de manera que luego podemos incluir esta librería en distintas páginas y disponer de esas funciones fácilmente.

Algunos ejemplos de librerías son:

- pChart
- Upload
- PHP Excel Reader
- Gaufrette
- Omniplay
- FPDF
- PHPMailer
- Krumo
- Carbon
- Underscore



1.2.2 Aplicación instalada en servidor

Para su instalación de la aplicación visualizar “MANUAL WAMPSEVER”

Estructura del lenguaje

• Sintaxis básica

```
<html>  
  <head>  
    <title> Página de prueba</title>  
  </head>  
  
  <?php  
    sentencias de PHP;  
  ?>  
</html>
```

Declarar e inicializar una variable

Variable

Es una estructura de datos a la cual se le asigna un nombre y puede almacenar un valor, este valor puede cambiar durante la ejecución del programa.

- Las variables en PHP empiezan con el signo de pesos (\$)
- Los nombres de las variables pueden contener letras, números o el guion bajo.
- Los nombres de las variables no pueden empezar con un número pero sí por el guion bajo (_).

```
$texto = "variable de texto"; // string  
$num = 5; // integer  
$doble = 5.0021; // double  
$valor = true; // booleana
```

Sintaxis del lenguaje

Tipo de datos

- Booleanos
- Enteros
- String
- array
- Objetos
- Null

Permite la definición de constantes.

Soporta los operadores

- Aritméticos
- Asignación
- De cadena
- Incremento y decremento
- Comparación
- Comparación de los datos en forma

binaria

- Lógicos

Estructuras de control

Estructuras de ciclos

- For
- Foreach
- While
- Do while
- Break y continue

Estructuras de control

- If-else
- If elseif
- Switch

if-else

- Mismo comportamiento que en C.
- Las sentencias compuestas se encierran entre llaves.
- elseif puede ir todo junto.

```
if (expresión1)
    sentencia 1
else if (expresión2)
    sentencia 2
...
else if (expresión n)
    sentencia n
else
    sentencia n+1
```

while

```
while (expresión)  
    sentencia
```

for

```
for (expresión1; expresión2; expresión3)  
    sentencia
```

switch

```
switch (expresión)
{
    case valor 1:
        sentencia 1
        break;
    case valor 2:
        sentencia 2
        break;
    ...
    case valor n:
        sentencia n
        break;
    default
        sentencia n+1}
```

Funciones

- Ejemplo:

```
function suma ($x, $y)
{
    $s = $x + $y;
    return s;
}
$a=1;
$b=2;
$c=suma ($a, $b);
print $c;
```

- Por defecto los parámetros se pasan por valor
- Paso por referencia:

```
function incrementa (&$a)  
{  
    $a = $a + 1;  
}  
$a=1;  
incrementa ($a);  
print $a; // Muestra un 2
```

- Argumentos por defecto

```
function muestranombre ($titulo = "Sr.")  
{  
    print "Estimado $titulo:\n";  
}
```

```
muestranombre ();  
muestranombre ("Prof.");
```

- Salida:

Estimado Sr.:

Estimado Prof.:

- Los argumentos con valores por defecto deben ser siempre los últimos:

```
function muestranombre ($nombre, $titulo= "Sr.")  
{  
print "Estimado $titulo $nombre:\n";  
}  
muestranombre ("Fernández");  
muestranombre ("Fernández", "Prof.");
```

- Salida:

Estimado Sr. Fernández:

Estimado Prof. Fernández:

- Sintaxis:

array ([clave =>] valor, ...)

- La clave es una cadena o un entero no negativo. El valor puede ser de cualquier tipo válido en PHP, incluyendo otro array

- Ejemplos:

\$color = array ('rojo'=>101, 'verde'=>51, 'azul'=>255);

\$medidas = array (10, 25, 15);

- Acceso:

\$color['rojo'] // No olvidar las comillas

\$medidas[0]

- El primer elemento es el 0

La estructura de control **foreach** permite iterar sobre arrays

Sintaxis:

foreach (expresión_array as \$valor) sentencia

foreach (expresión_array as \$clave => \$valor) sentencia

Salida:

Valor: 101, Valor: 51, Valor: 255

Clave: rojo; Valor: 101, Clave: verde; Valor: 51, Clave: azul; Valor: 255

Ejemplos:

foreach (\$color as \$valor)

**print “Valor: \$valor
\n”;**

foreach (\$color as \$clave => \$valor)

**print “Clave: \$clave; Valor: \$valor
\n”;**