

¿ Porque PHP ?

PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios web. PHP es el (acrónimo recursivo de "PHP: Hypertext Preprocessor", inicialmente PHP Tools, o, Personal Home Page Tools) es un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web, y últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la librería GTK+.

Características de PHP

- La principal característica es que se basa en ser un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad.
- Leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una muy buena documentación en su página oficial (<http://www.php.net/docs.php>).
- Es Libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.

Los principales usos del PHP son los siguientes:

- Programación de páginas web dinámicas, habitualmente en combinación con el motor de base datos MySQL, aunque cuenta con soporte nativo para otros motores, incluyendo el estándar ODBC, lo que amplía en gran medida sus posibilidades de conexión.
- Programación en consola, al estilo de Perl, en Linux, Windows y Macintosh.
- Creación de aplicaciones gráficas independientes del navegador, por medio de la combinación de PHP y GTK (GIMP Tool Kit), que permite desarrollar aplicaciones de escritorio tanto para los sistemas operativos basados en Unix, como para Windows y Mac OS X.

Requerimientos de Software y Hardware

Los requerimientos de Hardware son mínimos, prácticamente en cualquier PC que pueda correr un sistema operativo (en particular linux) podremos tener un servidor web con PHP instalado.

En cuanto a los requerimientos de software podemos nombrar los siguientes como una alternativa mínima como para empezar con la programación en el lenguaje PHP.

- Servidor Web, recomendamos Apache (www.apache.org)
- PHP (www.php.net)
- Editor, puede ser un editor de texto simple o algo más complejo y especializado para el lenguaje.
- Base de Datos, es opcional. Recomendamos MySQL (www.mysql.com)

El Lenguaje de PHP

PHP se escribe dentro de la propia página web, junto con el código HTML y, como para cualquier otro tipo de lenguaje incluido en un código HTML, en PHP necesitamos especificar cuáles son las partes constitutivas del código escritas en este lenguaje. Esto se hace, como en otros casos, delimitando nuestro código por etiquetas. Podemos utilizar distintos modelos de etiquetas en función de nuestras preferencias y costumbres. Hay que tener sin embargo en cuenta que no necesariamente todas están configuradas inicialmente y que otras, como es el caso de `<% y %>` sólo están disponibles a partir de una determinada versión (3.0.4.).

Estos modos de abrir y cerrar las etiquetas son:

```
<?           y           ?>
<%           y           %>
<?php       y           ?>
<script language="php">   y   </script>
```

Este último modo está principalmente aconsejado a aquellos que tengan el valor de trabajar con Front Page ya que, usando cualquier otro tipo de etiqueta, corremos el riesgo de que la aplicación nos la borre sin más debido a que se trata de un código incomprensible para ella.

El modo de funcionamiento de una página PHP, a grandes rasgos, no difiere del clásico para una página dinámica del lado servidor: El servidor va a reconocer la extensión correspondiente a la página PHP (phtml, php, php4,...) y antes de enviarla al navegador va a encargarse de interpretar y ejecutar todo aquello que se encuentre entre

las etiquetas correspondientes al lenguaje PHP. El resto, lo enviara sin más ya que, asumirá que se trata de código HTML absolutamente comprensible por el navegador.

Otra característica general de los scripts en PHP es la forma de separar las distintas instrucciones. Para hacerlo, hay que acabar cada instrucción con un punto y coma ";". Para la ultima expresión, la que va antes del cierre de etiqueta, este formalismo no es necesario.

Incluimos también en este capítulo la sintaxis de comentarios. Un comentario, para aquellos que no lo sepan, es una frase o palabra que nosotros incluimos en el código para comprenderlo más fácilmente al volverlo a leer un tiempo después y que, por supuesto, el ordenador tiene que ignorar ya que no va dirigido a él sino a nosotros mismos. Los comentarios tienen una gran utilidad ya que es muy fácil olvidarse del funcionamiento de un script programado un tiempo atrás y resulta muy útil si queremos hacer rápidamente comprensible nuestro código a otra persona.

Pues bien, la forma de incluir estos comentarios es variable dependiendo si queremos escribir una línea o más. Veamos esto con un primer ejemplo de script:

```
<?php
 $mensaje = "Tengo hambre!!"; //Comentario de una línea
 echo $mensaje; #Este comentario también es de una línea
 /*En este caso
 mi comentario ocupa
 varias líneas, lo ves? */
?>
```

Si usamos doble barra (//) o el símbolo # podemos introducir comentarios de una línea. Mediante /* y */ creamos comentarios multilínea. Por supuesto, nada nos impide de usar estos últimos en una sola línea.

No se preocupen si no comprenden el texto entre las etiquetas, es sólo a modo de ejemplo. Adelantamos que las variables en PHP se definen anteponiendo un símbolo de dólar (\$) y que la instrucción "echo" sirve para sacar en pantalla lo que hay escrito a continuación.

Recordamos que todo el texto insertado en forma de comentario es completamente ignorado por el servidor. Resulta importante acostumbrarse a dejar comentarios, es algo que se agradece con el tiempo.

Variables

Las variables son definidas anteponiendo el símbolo dólar (\$) al nombre de la variable que estamos definiendo.

Dependiendo de la información que contenga, una variable puede ser considerada de uno u otro tipo:

Variables numéricas

Enteros \$entero = 2002; Números sin decimales
Real \$real = 3.14159; Números con o sin decimal

Variables alfanuméricas

Almacenan textos compuestos de números y/o cifras
\$cadena = "Hola amigo";

Arrays

Almacenan series de informaciones numéricas y/o alfanuméricas

```
$sentido[1] = "ver";  
$sentido[2] = "tocar";  
$sentido[3] = "oir";  
$sentido[4] = "gusto";  
$sentido[5] = "oler";
```

Objetos

Se trata de conjuntos de variables y funciones asociadas. Presentan una complejidad mayor que las variables vistas hasta ahora pero su utilidad es más que interesante.

A diferencia de otros lenguajes, PHP posee una gran flexibilidad a la hora de operar con variables. En efecto, cuando definimos una variable asignándole un valor, el ordenador le atribuye un tipo. Si por ejemplo definimos una variable entre comillas, la variable será considerada de tipo cadena:

```
$variable = "5"; //esto es una cadena
```

Sin embargo si pedimos en nuestro script realizar una operación matemática con esta variable, no obtendremos un mensaje de error sino que la variable cadena será asimilada a numérica:

```
<?php  
  $cadena = "5"; //esto es una cadena  
  $entero = 3; //esto es un entero  
  echo $cadena + $entero  
?>
```

Este script dará como resultado "8". La variable cadena ha sido asimilada en entero (aunque su tipo sigue siendo cadena) para poder realizar la operación matemática. Del mismo modo, podemos operar entre variables tipo entero y real. No debemos preocuparnos de nada, PHP se encarga durante la ejecución de interpretar el tipo de variable necesario para el buen funcionamiento del programa.

Sin embargo, en contraste, hay que tener cuidado en no cambiar mayúsculas por minúsculas ya que, en este sentido, PHP es sensible. Conviene por lo tanto trabajar ya sea siempre en mayúsculas o siempre en minúsculas para evitar este tipo de malentendidos a veces muy difíciles de localizar.

Variables asignadas por referencia

En PHP también podemos asignar variables por referencia. En ese caso no se les asigna un valor, sino otra variable, de tal modo que las dos variables comparten espacio en memoria para el mismo dato.

La notación para asignar por referencia es colocar un "&" antes del nombre de la variable.

```
<?php
    $foo = 'Bob'; // Asigna el valor 'Bob' a $foo
    $bar = &$foo; // Referencia $foo vía $bar.
    $bar = "Mi nombre es $bar"; // Modifica $bar...
    echo $foo; // $foo también se modifica.
    echo $bar;
?>
```

Esto dará como resultado la visualización dos veces del string "Mi nombre es Bob". Algo como:

Mi nombre es BobMi nombre es Bob

Conexión a MySQL

Este sencillo ejemplo muestra cómo conectarse, ejecutar una consulta, imprimir las filas resultantes y desconectarse de una base de datos MySQL.

```
<?php
// Conexión, selección de base de datos
$enlace = mysql_connect('host', 'usuario', 'clave')
    or die('No pudo conectarse : ' . mysql_error());

echo 'Conexión exitosa';

mysql_select_db('base_datos')
    or die('No pudo seleccionarse la BD.');
```

```
// Realizar una consulta SQL
$consulta = 'SELECT * FROM mi_tabla';
$resultado = mysql_query($consulta)
    or die('La consulta falló: ' . mysql_error());

// Impresión de resultados
while ($registro = mysql_fetch_array($resultado, MYSQL_ASSOC)) {
    foreach ($registro as $valor_col) {
        echo $valor_col." - ";
    }
    echo "<br/>";
}

// Liberar conjunto de resultados
mysql_free_result($resultado);

// Cerrar la conexión
mysql_close($enlace);
?>
```

El material aquí presentado es una extracción, copia fiel de textos publicados en las siguientes paginas:

- Desarrollo Web (<http://www.desarrolloweb.com/>)
- Wikipedia (<http://es.wikipedia.org/wiki/PHP>).
- Mi Primer Script en PHP
[http://www.foros.frc.utn.edu.ar/topic.asp?\\$\\$sid=&id=577](http://www.foros.frc.utn.edu.ar/topic.asp?$$sid=&id=577)
- ¿Qué es PHP? ¿Para qué se lo usa? y ¿Cuáles son sus ventajas?
[http://www.foros.frc.utn.edu.ar/topic.asp?\\$\\$sid=&id=576](http://www.foros.frc.utn.edu.ar/topic.asp?$$sid=&id=576)