

PHP Introduccion.

Introducción

- 1995 Rasmus Lerdorf desarrolla a través de Perl un conjunto de scripts para el control de acceso a sus páginas personales, al conjunto de scripts se les denominó PHP (Personal Home Pages).
- 1997 se desarrolla la versión 2.0 denominada PHP/ FI(Forms Interpreter).
- 1998 Zeev Suraski y Andi Gutmans renuevan el lenguaje creando PHP 3.0 y a partir de aquí se le denomina Hypertext Preprocessor.
- 1999 Zeev y Andi crean un nuevo núcleo para el lenguaje (Zend Engine), y se libera la versión 4.0.
- 2004 Se crea el motor Zend 2.0 y se libera PHP 5.0, se introduce un modelo de orientación a objetos muy similar al de Java.

CGI Common Gateway Interface

CGI es una norma para establecer comunicación entre un servidor web y un programa, de tal modo que este último pueda interactuar con internet. También se usa la palabra CGI para referirse al programa mismo, aunque lo correcto debería ser script.

CGI

Es un programa que se ejecuta en tiempo real en un Servidor Web en respuesta a una solicitud de un Navegador. Cuando esto sucede el Servidor Web ejecuta un proceso hijo que recibirá los datos que envía el usuario (en caso de que los haya), pone a disposición del mismo algunos datos en forma de variables de ambiente y captura la salida del programa para enviarlo como respuesta al Navegador.

Propósitos de CGI

Generar páginas de forma dinámica.

Procesamiento de formularios.

Interacción con Bases de datos.

Comercio electrónico.

Lectura y escritura de archivos.

Motores de búsqueda.

Foros de discusión

LENGUAJES DE PROGRAMACIÓN INTERPRETADOS

Los lenguajes interpretados son aquellos que van siendo codificados por la computadora al tiempo en que se están ejecutando. Es decir, un "intérprete" es un traductor de lenguajes de programación de alto nivel, el cual realiza la operación de compilación paso a paso, los intérpretes ejecutan un programa línea por línea. Para cada sentencia que compone el texto inicial, se realiza una traducción, se ejecuta la sentencia y se vuelve a iniciar el proceso con la sentencia siguiente.

El programa siempre permanece en su forma original (programa fuente) y el intérprete proporciona la traducción al momento de ejecutar cada una de las instrucciones. Es decir, el programa será ejecutado sin necesidad de ser codificado antes, y de encontrarse un error la ejecución se detendrá en el comando o acción errónea.

No se genera un archivo binario.

Ejemplos: ASP, PHP, PERL.

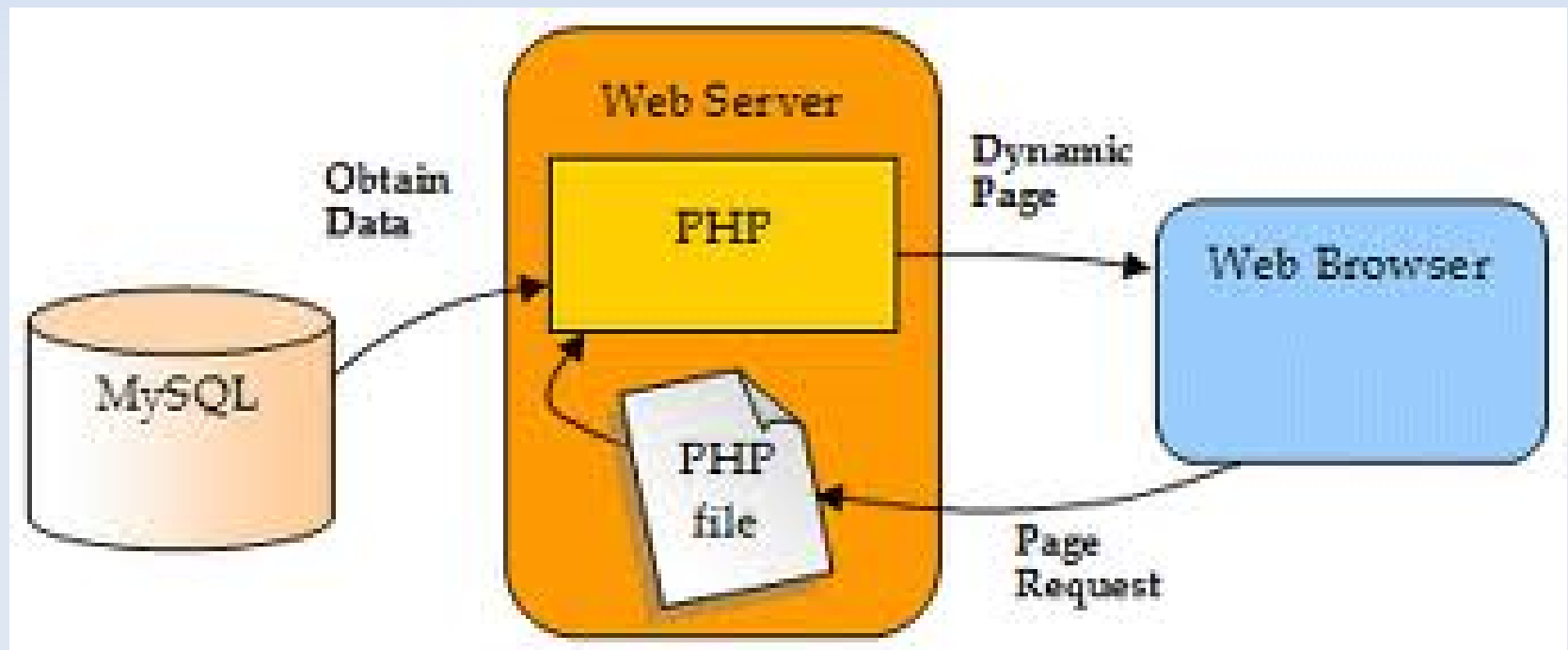
LENGUAJES DE PROGRAMACIÓN COMPILADOS

Los lenguajes compilados son aquellos que necesitan ser codificados, antes de ser ejecutados y obtener resultados, de encontrarse un error a la hora de codificar los comandos del programa, éste nunca podrá ser ejecutado.

Un compilador es un programa que traduce un lenguaje de alto nivel al lenguaje máquina de una computadora. Según va ejecutando la traducción verifica, comprueba y coteja los errores hechos por el programador; un compilador traduce un programa una sola vez, generalmente. Un programa compilado indica que ha sido traducido y está listo para ser ejecutado. La ejecución de los programas compilados es más rápida que la de los interpretados, ya que el intérprete debe traducir mientras está en la fase de ejecución.

Ejemplos: Java, C, C++ .

ARQUITECTURA CLIENTE/ SERVIDOR



FORMATO DEL CÓDIGO EN PHP

Delimitadores de código php:

`< ?php ?>`

`<? ?> (1)`

`<% % > (2)`

`< script language= "php">`

`< / script>`

NOTA: Para utilizar (1) directiva `short _open_tag= on`

Para utilizar (2) directiva `aspt_tags= on`

delimitadores.php

```
< html>
< head>
< title> Delimitadores de php< / title>
< / head>
< body>
< h1 align= "center"> DELIMITADORES DE ETIQUETAS EN PHP< / h1> < hr>
<?
echo "Utilizando la etiqueta corta";
?>
< br>
< script language= "php">
echo "Utilizando la etiqueta de html script con php";
< / script>
< br>
< ?php
echo "< h1> utilizando etiquetas de html dentro de php, con la etiqueta";
echo "tradicional de php< / h1> < br> ";
?>
<%
echo "Utilizando las etiquetas de asp y de jsp";
%>
< / body>
< / html>
```

VARIABLES

Una variable es el nombre que se le da a una posición de la memoria en la cual se almacena información.

PHP es un lenguaje débilmente tipado, es decir, las variables no tienen asociada la naturaleza del tipo de información que almacenan.

En PHP una misma variable podrá almacenar distintos tipos de información.

DEFINICIÓN DE VARIABLES

Sintaxis:

`$nombre_de_variable`

El nombre de las variables puede tener letras, números y `_`.

NOTA: PHP es case sensitive, por lo que diferencia entre minúsculas y mayúsculas.

TIPOS DE DATOS

Soporta 3 tipos de datos simples:

- Integer

- Float(sustituye a double desde v 4.2.0).

- String

Y 2 tipos de datos compuestos:

- Array

- Object

Además hace uso de un tipo boolean que no está en la sintaxis del lenguaje.

Integer (Enteros)

Números que varían entre – 2 billones y 2 billones, se pueden representar en formato decimal, octal o hexadecimal.

Ejemplos:

- 33,2139 Decimal

071,0336 Octal

0x 768, 0x AA Hexadecimal

Float (Flotantes)

Números con decimales.

Ejemplos:

3405.673, - 1.98

8.32e+ 11, 5.6 e- 3 Notación científica

String (Cadenas)

Cadenas de caracteres, una cadena está formada por cero o más caracteres encerrados entre “” ó ‘’.

Caracteres especiales para las cadenas:

`\n` Cambio de linea

`\t` Tabulación

`\\` Barra inversa

`\'` Comilla simple

`\"` Comilla doble

`\$` caracter \$

variables.php

```
< html>
< head>
< title> VARIABLES< / title>
< / head>
< body>
< h1
align= "center"> VARIABLES< / h1> < br/>
<?
/*
Definimos las variables:
Demostración del lenguaje débilmente tipado
*/
$entero= - 230;
echo 'Valor de la variable $entero : < br/> ';
echo "$entero< br /> ";
$entero= 4.74e3;
echo "$entero < br /> ";
$entero= "hola";
echo $entero;
?>
```

```
< br />
<?
echo "Utilizando caracteres especiales para las cadenas< br> ";
$entero= 0xFF;
echo "Valor de la variable $entero :
< br /> \ t";
echo "$entero< br/> \ t";
$entero= 4.74e- 3;
echo "$entero< br /> \ n";
$entero= "hola";
echo '$entero\ t';
?>
< /body>
< /html>
```