

LAMP

Radamantis Torres

7 de Agosto del 2006

1. Breve Introducción, Apache, Php, Mysql

Vamos a dar una pequeña introducción a cada uno de estos para empezar a entrar en calor y homogeneizar conocimientos

1.1. Apache

Apache es uno de los servidores web que mas se usan en el mundo, nace del servidor web de la NCSA (National Center for Supercomputing Applications). Existe para una gran cantidad de sistemas operativos, Linux, Solaris, BSD y hasta windows, claro que generalmente en los servidores windows se usa IIS.

1.2. PHP

Php es un lenguaje de programación que originalmente fue creado por Rasmus Lerdorf en 1995 y eran solo un conjunto de scripts en perl, poco a poco fué evolucionando hasta lo que conocemos ahora como php. El nombre original era Personal Homepage, pero desde la version 3 se cambió por PHP Hypertext Preprocesor, recuerdan GNU Not Unix? Php 4 fue liberado en el año 2000.

1.3. Mysql

Es un motor de Bases de Datos, actualmente es uno de los mas usados, debido a grandes características, como:

- Gran cantidad de tipos de datos
- Seguridad al momento de agregar usuarios
- multiplataforma
- gran apego al standard de lenguaje de sql

2. Cómo se que tengo Apache funcionando con php y conectándome a mysql?

Hay una forma muy sencilla de saber si apache ya esta funcionando con php, hay que crear un archivo foo.php con solo tres líneas:

```
<?php
phpinfo();
?>
```

La línea

```
<?php
```

significa que estamos abriendo un tag, donde todo lo que esté dentro, Apache lo enviará al intérprete de php para que haga lo suyo. La función `phpinfo()`; nos manda todas las características con las que tenemos corriendo php, con esto nos daremos cuenta si lo tenemos funcionando y con que librerías. Finalmente

```
?>
```

nos indica que estamos cerrando el tag de php y Apache empezará a interpretar lo siguiente como HTML.

Para saber si tenemos Mysql solo hay que mandarlo llamar con el siguiente comando: `mysql -u usuario -p` una vez con el prompt de mysql podemos empezar a crear bases, alterar tablas, agregar usuarios, etc.

3. Breve introducción a HTML

Quizá muchos de ustedes ya estén familiarizados con HTML y algunos de ustedes no, es algo muy sencillo, HTML es un lenguaje basado en *tags* en realidad son muy pocas las tags que podemos ocupar vamos a ver las básicas para utilizar en conjunto con nuestro php Todas las tags deben de cerrarse, mas adelante veremos como.

3.1. tags que siempre deben estar presentes

- **HTML** Este tag es el que le avisa al servidor web que empieza el documento en HTML, y es el primer tag que nos encontraremos en cualquier documento html *HTML* lo cerramos con */HTML*
- **HEAD** es el encabezado de nuestra página de igual manera se cierra */HEAD*
- **TITLE** dentro de las tags de **TITLE** pondremos el título de la página web también se cierra */TITLE*
- **BODY** encerrado por la etiqueta de **BODY** se encuentra todo el contenido de la página web que estamos construyendo.

3.2. Tags de uso frecuente

La siguiente lista de tags, no necesariamente las debemos de usar en nuestros documentos, pero son de las que mas se usan y siempre nos ayudarán para organizar la información

- table
- tr
- td
- br
- p
- div
- a
- img
- bueno, no es un tag
- sobre la marcha iré recordando otras

Vamos a escribir un pequeño archivo html

4. Breve introducción a bases de datos

Este no es un curso que vamos a enfocar al estudio de bases de datos, simplemente vamos a ver un panorama para que tengan una idea. Antes de crear una base de datos, debemos tener bien en claro el problema que vamos a *abstraer*. Es bueno pensar el diseño de una base de datos como si estuviéramos diseñando clases entre mejor podamos abstraer esto, es mejor. Por ejemplo. como pensarían una base de datos para una librería?, ahora que tal para un directorio?. Profundizemos un poco en todo esto.

5. Creando una base de datos y un usuario en mysql

Es muy facil crear una base de datos en mysql simplemente se hace lo siguiente:

```
create database elnombre;
```

pero eso no es todo, ya que tenemos la base creada, debemos de darle a algun usuario privilegios de acceso

```
grant all privileges on elnombre.* to unusuario@localhost idetified by 'unpassword '
```

Porque hay que crear usuarios con niveles de restriccion a nuestra base de datos?

6. Empezando con php

Php es un lenguaje de programación no tipado, esto quiere decir que no es necesario declarar los tipos de variables que estamos usando. Para asignar un valor a una variable solo hay que hacer lo siguiente:

```
$mivaribale="esta es una variable";
```

para imprimir el contenido de una variable usamos *echo*:

```
echo $mivariable;
```

6.1. Estructuras de control

Como en cualquier lenguaje de programación contamos con diferentes estructuras de control, como las que ya conocemos.

- **IF** se usa para tomar decisiones:

```
if(condicion){
  realiza accion;
}else{
  si no se cumple realiza esta accion;
}
```

- **Switch** Dependiendo del valor de una sola variable podemos tomar varias decisiones:

```
switch(variable)
{
  case 'valor1':
    accion a realizar si la variable es valor1
  break;
  case 'valor2':
    accion a realizar si la variable es valor2
  break;
  default:
    si no se cumple ninguna de las anteriores es importante tener un
    comportamiento por default que se declara aqui.
  break;
}
```

- **for** Es un ciclo y se usa para hacer repeticiones

```
for ($i=1;i<=10;i++)
{
echo "el numero actual es:".$i;
}
```

- **while** Tambien es un ciclo y sirve para hacer repeticiones

```
while(condicion)
{
accion a realizar;
}
```

7. Creando la conexión a mysql

Php tiene funciones que nos ayudan a realizar la conexión a mysql, así como también tiene funciones para ejecutar sentencias en mysql, vamos a ver algunas de estas: Lo primero que debemos de hacer es establecer la conexión con la base de datos, para lo cual usamos lo siguiente:

```
$db=mysql_connect("localhost", "usuario", "password");
if(!$db)
{
echo "Error: no me puedo conectar al servidor de bases de datos";
exit();
}
if(!mysql_select_db("nombre de la base", $db))
{
echo "Error: seleccionando la base de datos";
exit();
}
```

Una vez establecida la conexión podemos empezar a realizar consultas y lo hacemos con las siguientes funciones:

```
$query="select * from tabla where condicion";
$result=mysql_query($query);
$row=mysql_fetch_object($result);
$num_rows=mysql_num_rows($result);
```

Ya que conocemos estas funciones vamos a crear unos cuantos formularios para ingresar información a la base de datos y consultarla.

8. Podemos crear funciones

Siempre es bueno crear funciones de todo lo que podamos, nos ayuda a tener un código mas limpio y ordenado, para crear una función en php tenemos que hacer lo siguiente:

```
declaramos la funcion
function nombre(parametros)
{
    Codigo de la funcion
}
```

```
para utilizar la funcion:
nombre(parametros);
```

9. utilizando Sesiones

Las sesiones son unas de las herramientas mas útiles con las que contamos en php, cuando generamos una sesión php lanza una cookie al cliente donde comienza a guardar, en primera instancia el id de la sesión y demas valores que nosotros le vayamos pidiendo que guarde, las sesiones tienen caducidad, después de determinado tiempo las sesiones caducan, así también nosotros podemos forzar a que caduque una sesión, vamos a ver como se usan las sesiones. Siempre que estemos usando sesiones, nuestro archivo de php debe contener la siguiente línea al principio del documento, es muy importante que esté al inicio si no el intérprete de php nos mandará un error

```
session start();
```

Con esto estamos inicializando la sesión y podemos empezar a utilizar variables de sesión. Para poder asignar valores a las variables de sesión lo hacemos del siguiente modo:

```
$_SESSION['nombre de la variable']="valor que queremos que tenga";
```

Otras funciones utiles de sesiones

- **session_destroy()** Elimina todos los datos de la sesión pero no borra la cookie
- **session_unset()** Limpia todas las variables de la sesión

10. Unos cuantos consejos de seguridad

Cuando estan realizando programación para web siempre es importante tener algunas medidas de seguridad. Contraseñas difíciles, esto es subjetivo puesto que para muchos una contraseña difícil puede ser algo como *supercalifragilisticoespialidoso* sin embargo no lo

es, cuando hablamos de contraseñas difíciles siempre intentamos que contenga números, mayúsculas y minúsculas, por ejemplo una contraseña difícil sería: *R42msL_a*. Por otro lado, no es bueno guardar contraseñas en la base de datos, y si definitivamente tenemos que hacerlo, les sugiero hacerlo con md5, veamos un ejemplo de esto.

```
md5(minombre);
```

Cuando estamos pasando variables entre varios scripts muchas veces se tiene la costumbre de hacerlo por la url:

```
<a href="mipagina.com?var1=valor1&var1=valor2">liga</a>
```

Cuando se trata de información sensible, no es recomendable pasar variables por la url, solo en casos extremos necesarios e información que no sea sensible es la que recomendamos pasar por url.

11. Donde puedo encontrar ayuda en internet?

Finalmente en internet existen muchos lugares donde encontrar información a continuación les pongo una lista de lugares donde encontrarán buena información, fácil y accesible:

- Página oficial de php, <http://www.php.net>
- Página oficial de mysql, <http://www.mysql.org>
- Página oficial de Apache, <http://www.apache.org>
- Página de ayuda de Ubuntu Linux, <https://help.ubuntu.com>
- Página con manuales y tutoriales, <http://www.desarrolloweb.com>
- Google, <http://www.google.com>