

Tema: [:: Tutorial ::] Manual PHP Por Riven wizard y NeoheX

Hola a Todos, Aqui les enseñaré lo básico de Php. Segun algunos, el mejor lenguaje de creacion de webs, y para mi tambien, jeje:D...

Bueno, empecemos:

I. Introduccion:

PHP es un lenguaje creado por una gran comunidad de personas. El sistema fue desarrollado originalmente en el año 1994 por Rasmus Lerdorf como un CGI escrito en C que permitía la interpretación de un número limitado de comandos. El sistema fue denominado Personal Home Page Tools y adquirió relativo éxito gracias a que otras personas pidieron a Rasmus que les permitiese utilizar sus programas en sus propias páginas. Dada la aceptación del primer PHP y de manera adicional, su creador diseñó un sistema para procesar formularios al que le atribuyó el nombre de FI (Form Interpreter) y el conjunto de estas dos herramientas, sería la primera versión compacta del lenguaje: PHP/FI.

Php es un lenguaje que se ejecuta en el servidor, osea primero escribes la direccion que tenga php, luego el navegador va al server, ve que la pagina es PHP, la transforma a HTML, y devuelve un resultado HTML a el navegador.

II. 1- Lo Básico

Para poder agregar algo de PHP en nuestra pagina, debemos abrir el codigo php kon un tag y un interrogante (<?), y para cerrarlo primero el interrogante y despues el tag (?>).No falta decir que hay 4 formas de iniciar un php:

```
<?php echo ('Primera forma'); ?>
```

```
<? echo ('Segunda Forma'); ?>
```

```
<script language="php"> echo ('Tercera Forma'); </script>
```

```
<% echo ('Forma Tipo ASP') %>
```

Aqui les mostraré un ejemplo de como escribir algo en una pagina html:

```
<html>
<head>
<title>Mi Primera Pagina Php</title>
...
</head>
<body>
```

Aki escribiremos nuestra pagina html

```
<!-- Aqui parte la sentencia PHP -->
```

```
<?php
  echo "Hola a Todos, Estoy corrigiendo las faltas de Riven,
  y este es mi primerCodigo PHP" ;
?>
```

```
</body>
</html>
```

Al terminar una sentencia PHP, SIEMPRE, **debemos escribir un punto coma ;** excepto cuando el codigo tiene una sola linea, por ejemplo:

```
<?php echo "Hola" ;?>
```

Tema: [:: Tutorial ::] Manual PHP Por Riven wizard y NeoheX

```
<?php echo "Hola" ?>
```

Estos 2 estarian bien, pero si escribimos:

```
<?php
$a = Hola

$b = Mundo

echo "$a + $b"

?>
```

Estaría mal, ya que no pusimos los punto coma (;)

1.1- Comentarios para PHP

Los comentarios en php son parecidos (mas bien los mismos) que se ocupan en C, C++ y en las Shells de Unix. Veremos unos Ejemplos:

```
<?php
    // Asi se pone un comentario de una linea
?>
```

```
<?php
    Para poner comentarios de mas de una linea:

/*
Aki estoi poniendo
Comentarios de mas de una linea en PHP y que no se verán por pantalla!!

*/
?>
```

```
<?php
    echo "Mas y Mas Comentarios" #Comentario tipo Shell de Unix
?>
```

1.2- Includes y Requires

Muchos de ustedes deben de haber visto alguna vez el "include", o el require, y pues bueno, aqui les enseñare a usarlos:

Include() :

Este sirve para Incluir un archivo, tal como su nombre lo dice.
Entre los parentesis iria el archivo a incluir.

Por ejemplo si queremos que nuestra página tenga un contador de visitas, pero el contador de visitas es un archivo aparte, escribiríamos dentro del archivo esto:

```
<html>
<head>
<title>Estamos aprendiendo el uso de Include</title>
</head>
<body>
```

```
<?php
include ('contador_visitas.php');
?>
Aqui el contenido de la pagina
</body>
</html>
```

Require() :

Sirve para decirle al archivo que Requiere de algo, por ejemplo cuando un usuario mete sus datos, y los guardamos en un archivo aparte, debemos decirle al archivo que requiere de ese archivo para escribir sus datos, entonces ponemos:

tunombre.php

```
<?php

$nombre = $_GET ['nombre'];
$apellido = $_GET ['apellido'];
$pais = $_GET ['pais'];
$telefono = $_GET ['telefono'];

?>
```

procesa.php

```
<html>
<head>
<title>Requires</title>
</head>
<body>
```

```
<?php

require (tunombre.php);
echo "Tu te llamas $nombre $apellido , Vives en $pais, y tu telefono es $telefono." ;

?>
</body>
</html>
```

Otra cosa importante sobre los includes y requires, es, que cuando no queramos que salga un error cuando la página no se encuentra, debemos poner una @ (Arroba) antes del include o el require,

Ejemplo:

```
<?php
@include ('pagina.php') ;
?>
```

III- 1. Variables

Aqui entraremos a lo esencial de todos los lenguajes, **las Variables.**

En una variable uno puede guardar algo, despues sumarlo, restarlo, ver si es igual a otra variable o otra cosa, etc...

Las variables deben tener un signo de plata o dolar al principio, SIEMPRE

Tema: [:: Tutorial ::] Manual PHP Por Riven wizard y Neohex

\$variable

Aqui veremos un ejemplo:

```
<?php
$a = Hola; // Le asignamos el valor Hola a la variable A
$b = Mundo; // Le asignamos el valor Mundo a la variable B
$c = $a + $b; // Le asignamos el valor de A y B a la variable C
echo " $c " ; // Esto imprimiria "Hola Mundo"
?>
```

Otra cosa muy importante, es que **las variables son sensibles a Mayúsculas y Minúsculas**, osea que si escribimos lo siguiente:

```
<?php
$uno = Hola;
$Uno = Mundo;
echo "$uno $Uno";
?>
```

Nos daría como resultado:

Hola mundo

Lo que quiere decir, que no es lo mismo escribir **uno** que **Uno** (php lo interpretaría como variables diferentes).

III- 1.1 Cadenas

El unico operador de cadenas que existe, es el punto.

Ejemplo:

```
<?php
$a = Hola ;
$b = $a . "Mundo" ; //Esto tiene Hola Mundo
?>
```

Aqui hay que aclarar una cosa, **es distinto hacer una cadena con comilla simple, y una con comilla doble**, por ejemplo, si escribimos esto:

```
<?php
$a = Mundo ;

echo 'Hola $a'; //Escribiriamos Hola $a
echo "Hola $a"; //Esto si escribira Hola Mundo
```

III- 1.2 Caracteres Protegidos

Al igual que en otros lenguajes, la barra invertida, (slash), o como lo llamen (\), se usa para ocupar los caracteres protegidos:

| | |
|------------------|-------------------------|
| <code>\n</code> | Esto crearia otra linea |
| <code>\r</code> | Retorno |
| <code>\t</code> | Tabulacion |
| <code>\\</code> | Barra Invertida |
| <code>\\$</code> | Signo de Plata o dolar |
| <code>\"</code> | Comillas dobles |

IIII- 1.1- Operadores de Comparacion

| | |
|----------------------------|----------------------------|
| <code>\$a < \$b</code> | \$a es menor que \$b |
| <code>\$a > \$b</code> | \$a es mayor que \$b |
| <code>\$a <= \$b</code> | \$a es menor o igual a \$b |
| <code>\$a >= \$b</code> | \$a es mayor o igual a \$b |
| <code>\$a == \$b</code> | \$a es igual a \$b |
| <code>\$a != \$b</code> | \$a es distinto a \$b |

1.2 Operadores Logicos

| | |
|---------------------------------|---|
| <code>\$a AND \$b</code> | Verdadero si los 2 son verdaderos |
| <code>\$a && \$b</code> | Verdadero si los 2 son verdaderos |
| <code>\$a !! \$b</code> | Verdadero si alguno de los 2 es verdadero |
| <code>\$a OR \$b</code> | Verdadero si alguno de los 2 es verdadero |
| <code>\$a XOR \$b</code> | Verdadero si solo uno es verdadero |
| <code>!\$a</code> | Verdadero si \$a es falso |

1.3 Operadores de Asignacion

| | |
|-------------------------|---|
| <code>\$a = \$b</code> | Asigna \$a a \$b |
| <code>\$a += \$b</code> | Le suma \$a a \$b |
| <code>\$a -= \$b</code> | Le resta \$a a \$b |
| <code>\$a *= \$b</code> | Multiplica \$a por \$b y el resultado va para \$a |
| <code>\$a /= \$b</code> | Divide \$a por \$b y el resultado va para \$a |
| <code>\$a .= \$b</code> | Añade \$a a \$b |

IV. 1- Sentencias de Control

Con esto podremos ver si algo es verdadero o falso, y depende del resultado hacer algo:

Se trata de las condiciones

If, Else

Con esto podemos ejecutar un código **si algo es verdadero, y si es falso ejecutar otro código**, por ejemplo:

```
<?php
$a = 3 ;

if ($a == 3) { // si $a es igual a 3
echo "La Variable A es igual a 3";
}
else {
echo "La variable A no es igual a 3";
}

?>
```

If, Elseif, Else

Es parecido a If, Else, solo que en este podemos hacer varios Ifs con Elseif:

```
<?php
$a = 8 ;

if ($a == 1) {
echo "A es igual a 1";
}
elseif ($a == 2) {
echo "A es igual a 2";
}
else {
echo "A no es igual ni a 1 ni a 2";
}

?>
```

Switch, Case, Default

Este evalúa y compara cada una de las expresiones de la sentencia **case** con la expresión que evaluamos, si llegamos al final de la lista y no encuentra una condición verdadera, ejecuta el código que hay en **default**, pero si encuentra una condición que sea verdadera, la ejecuta y ocupa **break** para que no siga ejecutándose el código:

```
<?php
switch ($mes) {
    case "Enero":
        echo "Estamos en el mes de Enero";
```


Tema: [:: Tutorial ::] Manual PHP Por Riven wizard y Neohex

```
break;
case "Febrero":
echo "Estamos en el mes de Febrero";
break;
case "Marzo":
echo "Estamos en el mes de Marzo";
break;
case "Abril":
echo "Estamos en el mes de Abril";
break;
case "Mayo":
echo "Estamos en el mes de Mayo, Es mi cumpleaños!! Cheesy";
break;
case "Junio":
echo "Estamos en el mes de Junio";
break;
case "Julio":
echo "Estamos en el mes de Julio";
break;
case "Agosto":
echo "Estamos en el mes de Agosto";
break;
case "Septiembre":
echo "Estamos en el mes de Septiembre";
break;
case "Octubre":
echo "Estamos en el mes de Octubre";
break;
case "Noviembre":
echo "Estamos en el mes de Noviembre";
break;
case "Diciembre":
echo "Estamos en el mes de Diciembre, El ultimo mes del año!";
break;
default:
echo "Pos que raro, no estamos en ningun mes xD";
}
```

?>

While

Esta sentencia ejecuta un codigo mientras una condición se este cumpliendo (while, en castellano es MIENTRAS... Cheesy).

Ejemplo:

```
<?php
$num = 1;

while ($num < 5) {
echo " $num ";
$num++
}
?>
```

Al igual que en Switch, aqui tambien podemos parar el codigo con **break:**

```
<?php
```

Tema: [:: Tutorial ::] Manual PHP Por Riven wizard y NeoheX

```
$num = 1;

        while ($num < 5) {
            echo " $num ";
            if ($num == 3) {
                echo "Aki paramos";
                break;
            }
            $num++
        }
?>
```

V- Variables y Formularios, Un excelente equipo xD

Si sabemos trabajar con variables y formularios, podriamos hacercasi cualquier cosa (bueno, estoy exagerando jejeje)... por ejemplo, podriamos hacer un recomendador de webs, un programilla que adivine "las posibilidades de amor" (yo hice uno xDDD <http://www.elextremo.cl/adi/adivina.htm>)

Algo que pregunte el nombre de una persona y lo escriba, etc...

Lo que tenemos que hacer es hacer un formulario dentro de una web html

```
<form action="pagina_que_procese.php" method=get/post>
```

Primero preguntamos que nombre tiene, con un input de text (aqui ya entramos en html... xD)

NOTA: Si no sabes que es un input o un form, etc, en <http://foro.elhacker.net> (seccion webmaster, subforos), tienes tutoriales de html...ve hay!!:D)

y despues, en la pagina que procese, recuperamos los datos, con **\$_GET**, aqui les explicaré como, con un ejemplo:

nombre.htm

```
<html>
<head>
<title>Como te llamas?</title>
</head>
<body>
Como te llamas?
<form action="nombre.php" method="get">
<input type="text" size="30" name="tunombre">
<!-- name=tunombre es muy importante, ya que esa sera la variable que tomaremos
en la pagina procesadora -->
<input type="submit" value="Enviar">
</form>
</body>
</html>
```

nombre.php

```
<?php
```

```
$tunombre = $_GET ['tunombre'];
```


Tema: [:: Tutorial ::] Manual PHP Por Riven wizard y Neohex

```
/*  
Con este codigo recuperamos los datos de  
nombre.htm y los guardamos en $tunombre  
*/
```

```
echo "Tu te llamas $tunombre";
```

```
?>
```

Este era un ejemplo con un **input text**, ahora veamos uno con un **input checkbox**

que.htm

```
<html>  
<head>  
<title>que haras Hoydia?</title>  
</head>  
<body>  
Que haras Hoy Dia?  
<form action="que.php" method="get">  
Ire al cine<input type="checkbox" name="check1">  
Ire al Colegio<input type="checkbox" name="check2">  
Jugare Futbol<input type="checkbox" name="check3">  
Dormire todo el dia Cheesy<input type="checkbox" name="check4">  
<input type="submit" value="Enviar">  
</form>  
</body>  
</html>
```

que.php

```
<?php  
  
$check1 = $_GET ['check1'];  
$check2 = $_GET ['check2'];  
$check3 = $_GET ['check3'];  
$check4 = $_GET ['check4'];  
    if (isset ($check1))  
        echo "$chek1<br><br>";  
    if (isset ($check2))  
        echo "$chek2<br><br>";  
    if (isset ($check3))  
        echo "$chek3<br><br>";  
    if (isset ($check4))  
        echo "$chek4<br><br>";  
?  
>
```

Como pueden ver, para ver si esta marcado el **checkbox**, utilizo el "Isset" (en castellano, **esta puesto, colocado (malpensaos)**), por lo que seria **"si esta puesto el check 1, etc...."**

Después veremos mas ejemplos.

VI- Almacenamiento y Envio de Datos

Tema: [:: Tutorial ::] Manual PHP Por Riven wizard y NeoheX

Veremos el uso de **fopen**, **fread** y **fwrite**, y como enviar mails.

Fopen

Abre un archivo o Url

Descripcion:

```
int fopen (string filename, string mode [, int use_include_path])
```

Si filename empieza con http://, se abre una conexión HTTP 1.0 hacia el servidor dado y devuelve un apuntador al archivo al comienzo del texto de respuesta.

Siempre se debe incluir una barra final.

Si filename empieza con ftp:// se abre una conexión ftp hacia el servidor dado y se devuelve el apuntador del archivo requerido.

Si filename no empieza con lo dado anteriormente, el archivo se abre de los archivos que se encuentran en el servidor.

Mode, puede ser cualquiera de lo siguientes:

"r" - Abre el archivo solo para lectura

"r+" - Abre el archivo para lectura y escritura

"w" - Abre el archivo solo para escritura, si el archivo no existe, trata de crearlo.

"w+" - Abre el archivo para lectura y escritura, si el archivo no existe, trata de crearlo.

"a" Abre el archivo solo para escribir (añadir), si el archivo no existe, trata de crearlo.

"a+" Abre el archivo para escribir (añadir) y para lectura. Si el archivo no existe, trata de crearlo.

fwrite

Escribe en modo Binario

```
int fwrite (int fp, string string [, int length])
```

fwrite() Escribe el contenido de string al fichero apuntado por fp. Si se da el argumento **length**, la escritura acaba antes de que length bytes sean escritos o se alcance el final de string, lo que pase primero.

fread

Lee archivos en modo Binario

Descripción: **string fread (int fp, int length)**

fread()

Lee hasta length bytes del apuntador del fichero referenciado por fp.
La lectura acaba cuando length bytes se han leído o se alcanza EOF, lo que pase primero.

Bueno, sabiendo esto ya podríamos Crear un contador... Aquí les dejo uno

```
<?
$archivo = "contador.txt";
$abre = fopen($archivo, "r");
$total = fread($abre, filesize($archivo));
fclose($abre);
$abre = fopen($archivo, "w");
$total = $total + 1;
$grabar = fwrite($abre, $total);
fclose($abre);
echo "<font face='verdana' size='2'>Hemos tenido <b>".$total."</b> visitas
desde FECHA DE CREACION</font>";
?>
```

VI- Envio de mails

Esta es una función muy fácil, y el código "mail" hace casi todo por nosotros... aquí vá

mail (destinatario, tema, mensaje, remitente);

Así de simple, Aquí un ejemplo:

email.htm

```
<html>
<head>
<title>Manda un Mail</title>
</head>
<body>
<form action=email.php method=get>
Introduce tu dirección: <input type=text size=20 name=direccion><br>
Introduce el mensaje:<br>
<textarea rows=5 name=mensaje><br>
Introduce el Email de destino: <input type=text size=20 name=destino>
</form>
</body>
</html>
```

email.php

```
<?php

//recojemos los datos
$direccion = $_GET ['direccion'];
$mensaje = $_GET ['mensaje'];
$destino = $_GET ['destino'];

//Enviamos el mail
mail ($destino, "Hola!", $mensaje, "FROM: $direccion");
```

Tema: [:: Tutorial ::] Manual PHP Por Riven wizard y NeoheX

```
//Le decimos al user que su mail ha sido enviado con exito
echo "Tu mensaje ha sido enviado con exito ha: $destino !!";
?>
```

Bueno, hasta aqui llega el manual por hoy, de vez en cuando le iré agregando cosas de utilidad, espero que les haya gustado mi primer manual y que hayan quedado sabiendo php!! xD...

Salu2...

<?php

```
//recojemos los datos
$direccion = $_GET ['direccion'];
$mensaje = $_GET ['mensaje'];
$destino = $_GET ['destino'];

//Enviamos el mail
mail ($destino, Hola!, $mensaje, $direccion);

//Le decimos al user que su mail ha sido enviado con exito
echo "Tu mensaje ha sido enviado con exito ha: $destino !!";
?>
```

Tengo una duda, yo en un mini tuto de PHP, vi que la sintaxis es esta:

Citar

```
<? mail(destinatario, tema, texto del mensaje, información adicional de
cabecera);
?>
```

Y, creo que me confundí un poco con lo que has escrito Embarrassed , como es eso de "\$destino, Hola!, \$mensaje, \$direccion"

yo le entiendo que es algo así como:

```
destino = midireccion@server.com
mensaje = mi mensaje Grin
direccion = midireccion@server.com
cuando, debería de ser
```

```
destino = midireccion@server.com
asunto = asunto del mensaje
mensaje = pos mi mensaje Grin
```

y, lo de mas Grin, solo para completar, le pondriamos algo así...

```
<?
mail (" $destino", " $asunto", " $cuerpo, e-Mail enviado de la web en
contruccion", "FROM: Comentarios <informes@nemrodweb.tk>\n"); ?>
```

Creo que es así Embarrassed no estoy seguro.

No se si os fijasteis, pero antes de esa pagina que enviaria los datos, **hay una en la que tienes que entrar los datos**, y hay esta eso de \$destinario, \$mensaje, etc... entonces en el otro recogemos los datos con el **Get**, y los pasamos a variables, y en vez de poner un destinatario fijo, ponemos el que el user quiera..

Tema: [:: Tutorial ::] Manual PHP Por Riven wizard y NeoheX

»

Bueno pues este mini manual va dirigido a personas que kieran empezar a utilizar php pero no sepan como...así que aki les dejo algunas cositas básicas...

=====

1- Introducción

1.1- Ejemplo de pagina en php

1.2- Comentarios

2- Variables

3- Operadores

3.1- Operadores aritmeticos

3.2- Otros Operadores

4- Condicionantes

4.1- If, Else, Else if

4.2- For

4.3- While

5- Introduccion a las Cookies

6- Introduccion a Mysql

6.1- Conectar con el servidor

6.2- Principales Funciones SQL

=====

1- Introducción

- PHP es un lenguaje que, a diferencia de otros como javascript, se ejecuta en el servidor y no en el navegador...

- Todos los archivos deben tener una extension *.php

- El contenido de una pagina en php se ejecuta entre las siguientes etiquetas:

<? o <?php --> esta marca el inicio de la pagina.

?> --> esta marca el fin de la pagina.

- En una página en php tambien se pueden introducir etiquetas en html.

- Cada linea de un codigo en php debe acabar con ;

1.1- Ejemplo de pagina en php

```
<?
echo "Ejemplo de pagina en php...blabla...";
?>
```

Resultado:

Ejemplo de pagina en php...blabla...

1.2- Comentarios

Tema: [:: Tutorial ::] Manual PHP Por Riven wizard y NeoheX

- Los comentarios son, como su propio nombre indica, comentarios sobre el código o sobre lo que se quiera poner...mejor verlo con un ejemplo...

- Comentarios de una línea: se escriben después de `//`
- Comentarios de más de una línea: se escriben entre `/* y */`

Ejemplo:

```
<?
// este es un comentario de una línea

/* este comentario ocupa más
de una línea y los hacen que
el texto que haya en ellos
no se muestre en el navegador */

echo "Ejemplo de página en php...blabla...";
?>
```

Resultado:

Ejemplo de página en php...blabla...
como se ve...no se muestra el contenido del comentario...

2- Variables

- Las variables son como un almacén de datos...para declarar una variable se utiliza el signo \$ seguido del nombre de la variable y su valor...

Ejemplo:

```
<?
// declaramos la variable "nombre" y le damos el valor "Pedro"
$nombre = "Pedro";

// mostramos en pantalla la variable "nombre"
echo "hola $nombre que tal estas?";
?>
```

Recuerda poner siempre `;` al final...

Resultado:

hola Pedro que tal estas??

También podemos mostrar una variable de estas formas:

```
<?
$nombre = "Pedro";
echo "hola ".$nombre."<br>hola otra vez";
?>
```

Resultado:

hola Pedro
hola otra vez

3- Operadores

- Los operadores son comandos para operar con los datos...

3.1- Operadores aritmeticos

- Los principales operadores aritmeticos son la suma, la resta, la multiplicacion, la division y las potencia...

Operacion

\$a

\$b

Sintaxis

Resultado

Suma

5

3

\$a + \$b

8

Resta

5

3

\$a - \$b

2

Multiplicacion

5

3

\$a * \$b

15

Division

5

3

\$a / \$b

1,6666

Potencia

5

3

```
pow($a,$b)
```

125

Ejemplo:

```
<?
$a = 2;
$b = 3;
$c = $a + $b;
echo "la suma de $a y $b es $c";
?>
```

Resultado:

la suma de 2 y 3 es 5

3.2- Otros Operadores

- Los principales son:

!\$a

\$a no se cumple

\$a == \$b

\$a tiene el mismo valor que \$b

\$a === \$b

\$a tiene el mismo valor que \$b y son del mismo tipo

\$a != \$b

\$a es lo opuesto de \$b

\$a < \$b;

\$a es menor de \$b

\$a > \$b

\$a es mayor que \$b

\$a <= \$b

\$a es menor o igual a \$b

\$a >= \$b

\$a es mayor o igual a \$b

\$a++

incrementa en 1 a \$a

\$a--

decrementa en 1 a \$a

4- Condicionantes

- Los condicionantes son una forma de ejecutar un código siempre y cuando se cumplan unas condiciones...

4.1- If, Else, Else if

La sintaxis de IF es:

```
<?
if (condicion){
lo que se ejecuta si es cierta la condicion
}
?>
```

Ejemplo:

Código:

```
<?
$a = 2;
$b = 3;
if ($a == $b){
echo "Son iguales";
}
?>
```

Explicacion:

Si el valor de \$a es igual al valor de \$b imprime en pantalla "Son iguales"

La sintaxis de ELSE es:

```
<?
if (condicion){
lo que si ejecuta si es cierta la condicion
}else{
lo que se ejecuta si no es cierta
}
?>
```

Ejemplo:

```
<?
$a = 2;
$b = 3;
if ($a == $b){
echo "Son iguales";
}else{
echo "No son iguales";
}
?>
```

Explicacion:

Si el valor de \$a es igual al valor de \$b imprime en pantalla "Son iguales", y si no lo es imprime en pantalla "No son iguales"

La sintaxis de ELSE IF es:

```
<?
if (condicion1){
lo que si ejecuta si es cierta la condicion1
}elseif (condicion2){
lo que se ejecuta si es cierta la condicion2
}else{
lo que se ejecuta si no es cierta ninguna condicion
}
?>
```

Ejemplo:

Código:

```
<?
$a = 2;
$b = 3;
if ($a == $b){
echo "Son iguales";
}elseif ($a == 7){
echo "$a tiene el valor 7";
}else{
echo "$a no valor ni lo mismo que $b ni vale 7";
}
?>
```

Explicacion:

Si el valor de \$a es igual al valor de \$b imprime en pantalla "Son iguales", y si no se cumple la primera condicion y si se cumple la segunda imprime en pantalla "\$a tiene el valor 7", y si no se cumple ningun condicion imprime en pantalla "\$a no valor ni lo mismo que \$b ni vale 7"

4.2- For

Ejemplo de FOR:

```
<?
for ($a = 1 ; $a <= 10 ; $a++) {
echo $a;
}
?>
```

Explicacion:

\$a tiene el valor 1, \$a es menor o igual que 10, a se incrementa en 1...

Resultado:

12345678910

4.3- While

```
<?
$a = 3;
while ($a <= 7){
echo $a++;
}
?>
```

El mismo pero de otra forma:

```
<?
$a = 3;
while ($a <= 7)
echo $a++;
endwhile;
?>
```

Resultado:

34567

5- Introduccion a las Cookies

- Una cookie es un almacen de informacion que se guarda en el ordenador del usuario y en cualquier momento la pagina puede pedir la informacion que contiene la cookie al ordenador del usuario...

Colocar una cookie:

```
<?
setcookie("nombre","Pedro",time()+3600);
?>
```

Explicacion:

Este codigo crea una cookie llamada nombre con el valor Pedro y con una vida de 1 hora (3600 segundos).

Recoger la informacion de una cookie:

```
<?
echo "Hola ".$_COOKIE["nombre"]."<br>Esta es la mejor forma";
?>
```

Resultado:

Hola Pedro
Esta es la mejor forma

Eliminar una cookie:

```
<?
setcookie("nombre");
?>
```

Resultado:

Este codigo elimina la cookie llamada nombre

6- Introduccion a Mysql

- Mysql es la base de datos que utiliza php...esta base de datos funciona de la siguiente manera:

Tenemos filas y columnas...las columnas son el tipo de datos y las filas son los datos insertados...un ejemplo:

Tabla usuarios
id

Tema: [:: Tutorial ::] Manual PHP Por Riven wizard y NeoheX

nombre

email

1

neoheX

neoheX16@hotmail.com

2

pepe

pepe123@hotmail.com

3

mari

mari123@hotmail.com

4

manolo

manolito@hotmail.com

- aki id, nombre y email son las columnas y neoheX, pepe123@hotmail.com, ect son las filas...

para almacenar y mostrar datos de la base de datos en nuestra pagina .php
tenemos que hacerlo **mediante las consultas**

Ejemplo(sintaxis):

seleccionar email de la tabla usuarios donde nombre sea neoheX

Resultado:

neoheX16@hotmail.com

- para hacer las consultas primero tenemos que conectar con el servidor y luego con la base de datos...una vez echo esto la consulta la realizamos mediante el lenguaje SQL...

6.1 Conectar con el servidor

- Para conectar on el servidor crearemos una pagina .php con el contenido:

->Conectamos con el servidor:

```
<?
$conec = @mysql_connect("servidor", "usuario", "contraseña");
?>
```

->Conectamos con la base de datos:

```
<?
$conec = @mysql_connect("servidor", "usuario", "contraseña");
@mysql_select_db("nombre_base_datos", $conec);
?>
```


Una vez que ya estamos conectados podemos realizar una consulta SQL.

6.2- Principales Funciones SQL

Las principales funciones son las siguientes:

CREATE

con esta funcion podemos crear base de datos, tablas, etc...

Ejemplo crear base de datos:

```
CREATE DATABASE neohex_db
```

Resultado:

Esto crearia una base de datos con el nombre de neohex_db

Ejemplo crear tablas:

```
CREATE TABLE usuarios (id int not null auto_increment primary key,nombre  
varchar(50) not null,email varchar(50) not null)
```

Resultado:

Esto crearia una tabla llamada usuarios con las columnas id,nombre y email

La tabla seria de la siguiente manera:

usuarios

id

nombre

email

1

neohex

neohex16@hotmail.com

2

pepe

pepe123@hotmail.com

3

mari

mari123@hotmail.com

4

manolo

manolito@hotmail.com

INSERT

Tema: [:: Tutorial ::] Manual PHP Por Riven wizard y Neohex

Con esta funcion podemos insertar datos dentro de una tabla:

Ejemplo:

```
INSERT INTO usuarios (nombre,email) values ('Bartolo','bartolito@hotmail.com')
```

Resultado:

Esto insertaria una nueva fila en la que el nombre tendria el valor Bartolo y su email correspondiente seria bartolito@hotmail.com

SELECT

Con esta funcion podemos mostrar el contenido de una tabla en la base de datos

Ejemplo:

```
select email from usuarios
```

Resultado:

```
neohex16@hotmail.com
pepel23@hotmail.com
mari123@hotmail.com
manolito@hotmail.com
esto nos devuelve los email de la tabla usuarios...
```

Otro:

```
select email from usuarios where nombre='neohex'
```

Resultado:

```
neohex16@hotmail.com
```

este nos devuelve el email, de la tabla usuarios, donde el nombre vale neohex

Otro mas:

```
select email,nombre from usuarios where id='3'
```

Resultado:

```
mari
```

```
mari123@hotmail.com
```

este nos devuelve el email y el nombre donde la id vale 3

DELETE

Con esta funcion podemos borrar filas de una tabla:

Código:

Tema: [:: Tutorial ::] Manual PHP Por Riven wizard y NeoheX

DELETE FROM usuarios WHERE nombre='neoheX'

Esto borraría la fila, en la tabla usuarios, correspondiente a donde nombre tenga el valor neoheX

Bueno pues solo keda mostrar con realizar una consulta SQL desde una pagina en php, para esto primero conectamos al servidor, a la base de datos y despues realizamos la consulta...

Ejemplo borrar.php:

```
<?
$sql = "DELETE FROM usuarios WHERE id='2'";
$conec = @mysql_connect("servidor", "usuario", "contraseña");
@mysql_select_db("nombre_base_datos", $conec);
mysql_query("$sql", $conec);
?>
```

Resultado:

Al entrar en esta pagina se borraría la fila donde la id fuera igual a 2

=====

Bueno pues esta ultima parte no la he extendido mucho pero es solo para dar una idea de como va el tema...cuando tenga tiempo la haré un poco mas largita...
pequeñas ideas de:

Citar
htmlpoint.com, electros.tk

Manual by neoheX -> <http://www.neoheX.net>

=====

neoheX16@hotmail.com

Re:[:: Tutorial ::] Manual PHP (Actualizado el 19/02/04)
« Respuesta #14 en: 12 Marzo 2004, 19:08 »

He encontrado una buena cantidad de errores de sintaxis en el primer tutorial que me parece deben de ser corregidos, pues no permiten la ejecución de los ejemplos y por tanto que un newbie aprenda.

Por ejemplo:

```
Citar
<?php

$a = 8 ;

if ($a == 1) {
echo "A es igual a 1";
}
elseif ($a == 2); {
echo "A es igual a 2";
}
else {
echo "A no es igual ni a 1 ni a 2";
}
?>
```

Tema: [:: Tutorial ::] Manual PHP Por Riven wizard y NeoheX

Citar

```
elseif ($a == 2); {
```

Después de la comparación se debe de eliminar el ";"
y esto sucede en varios ejemplos...

En los primeros ejemplos las variables están mal definidas

Citar

```
<?php
```

```
$a = Hola
```

```
$b = Mundo
```

```
echo "$a + $b"
```

```
?>
```

*No hay forma de que Hola sea una constante

*Falla a la primera regla de terminar cada línea con una ";"

Sería bueno que se corrigiera para no frustrar a la gente que intente aprender con este tutorial...

Estoy con el segundo y no me ha dado problemas con los ejemplos al momento.

Morris