

INTRODUCCIÓ AL PHP



Sara Rosés i Garriga sara@dunlock.com
Dunlock-Enginyeria Informàtica
Volcànica 2007

1. INTRODUCCIÓ AL PHP

Què és el Php?

El PHP (HyperText Preprocessor) és un llenguatge de scripting que permet la generació dinàmica de continguts en un servidor web.

•Característiques principals:

- Llenguatge potent
- Alt rendiment
- Facilitat d'aprenentatge
- Escàs consum de recursos

El codi PHP va col·locat entre les pàgines HTML. Per delimitar la secció de codi corresponent a aquest llenguatge de script s'utilitzen un parell d'etiquetes de inici i tancament que són:

<?php i ?>

1. INTRODUCCIÓ AL PHP

Arquitectura Client-Servidor



El client no pot veure el codi PHP, per això PHP és transparent per l'usuari, només veurà el resultat en HTML que genera l'intèrpret de PHP; d'aquesta forma qualsevol navegador pot interpretar el resultat obtingut en HTML.

1. INTRODUCCIÓ AL PHP

Avantatges del Php

- Disposa de múltiples eines que et permeten **accedir a bases de dades de forma senzilla**.
- És **multiplataforma**, funciona tant per Unix (amb Apache) com per Windows (amb Microsoft Internet Information Server).
- **Aprenentatge molt senzill**.
- És **més ràpid i gratuït**.
- També ofereix la **possibilitat d'usar programació de procediments o programació orientada a objectes**.

1. INTRODUCCIÓ AL PHP

Història

1994-->El sistema va ser desenvolupat per **Rasmus Lerdorf** com un CGI escrit en C que permetia la interpretació d'un nombre limitat de comandes.

1995-->El seu creador va dissenyar un sistema per processar formularis al qual va anomenar **FI (Form Interpreter)** i el conjunt d'aquestes dues eines, seria la primera versió compacta del llenguatge: **PHP/FI**.

1997-->Es va tornar a programar l'analitzador sintàctic, es van incloure noves funcionalitats com el suport a nous protocols d'Internet i el suport a la gran majoria de les bases de dades comercials. (**PHP3**)

2000-->Es desenvolupa el **PHP4** on cal destacar: més rapidesa, major independència del servidor web i un API més elaborat i amb més funcions.

2004-->Aparició del **PHP5**. Es dispara la seva utilització ja que és el complement ideal perquè el tàndem Linux-Apache sigui compatible amb la programació + servidor de pàgines web.

2. PROGRAMACIÓ EN PHP

Sintaxis bàsica

Per poder usar PHP i HTML conjuntament, el PHP utilitza una sèrie d'etiquetes que delimiten el codi, de manera que tot el que estigui situat entre aquestes etiquetes serà executat.

Existeix la possibilitat d'utilitzar quatre parells d'etiquetes però d'aquestes, només dos estan disponibles sempre:

```
<?php           <script language="php">  
  // Codi PHP      // Codi PHP  
?>              </script>
```

Les altres dues formes d'identificar el codi PHP no venen per defecte i per utilitzar-les s'han de modificar l'arxiu de configuració de php (php.ini).

```
<?>           <%>  
  // Codi PHP      // Codi PHP      Etiquetes utilitzades per ASP  
?>           %>
```

2. PROGRAMACIÓ EN PHP

Tipus de dades

Normalment el programador no indica el tipus d'una variable, en el seu lloc, ho decideix PHP en temps d'execució depenent del context en el que s'utilitzi aquesta variable.

PHP suporta els següents tipus de dades:

- Array `$var = new array ("patates","tomàquets")`
- Nombres amb punt flotant `$var = 23.45`
- Enter `$var = 33`
- Objecte `$var = new NombreClase ()`
- Cadena `$var = "Hola Antoni"`

2. PROGRAMACIÓ EN PHP

Variables

- Són **contenidors d'informació** on podem guardar-hi nombres enters, nombres decimals, caràcters...
- El **contingut de les variables es pot llegir i es pot canviar** durant l'execució d'una pàgina PHP.
- En PHP totes les variables **comencen amb el símbol del dòlar “\$” i no cal definir-les abans d'utilitzar-les.**
- **No tenen tipus**, és a dir que una mateixa variable pot contenir un nombre i després contenir caràcters,
- Hem de tenir en compte que **el nom de les variables és sensible a minúscules i majúscules.**

2. PROGRAMACIÓ EN PHP

Variables

Exemple sobre variables

```
<html>
  <head><title>Exemple de PHP</title></head>
  <body>
    <?php
      $a = 1;           // guardem a $a un enter
      $b = 3.34;       // guardem a $b un float
      $c = "Hola Món"; // guardem a $c una cadena
      echo $a,"<br>",$b,"<br>",$c;
      // mostrem en pantalla el contingut de les variables
    ?>
  </body>
</html>
```

2. PROGRAMACIÓ EN PHP

Variables

En PHP les variables **s'assignen per valor** (els canvis que s'efectuïn a una d'aquestes variables no afectarà a l'altra).

Amb PHP4 podem **assignar per referència** (la nova variable simplement fa referència a la variable original). Es fa posant un & davant de la variable.

```
<?php
    $foo = 'Bob';           // Assigna el valor 'Bob' a $foo
    $bar = &$foo;          // Referència $foo via $bar.
    $bar = "El meu nom és $bar"; // Modifica $bar...
    echo $foo;             // $foo també es modifica.
    echo $bar;
?>
```

Només les variables amb nom poden assignar-se per referència!

2. PROGRAMACIÓ EN PHP

Variables Predefinides

<i>argc</i>	Conté la cadena de petició en una línia de comandes.
<i>argv</i>	Conté el nombre d'arguments de la línia de comandes, que són variables molt utilitzades en C.
<i>HTTP_GET_VARS</i>	És un array associatiu amb les variables passades mitjançant el mètode GET, bé a través d'un formulari o bé passat directament per la url.
<i>HTTP_POST_VARS</i>	És un array associatiu amb les variables passades mitjançant el mètode POST a través d'un formulari.
<i>HTTP_COOKIE_VARS</i>	És un array associatiu format per les variables passades mitjançant cookies.

2. PROGRAMACIÓ EN PHP

Variables de Sistema

Variable	Descripció
\$HTTP_USER_AGENT	Ens informa principalment sobre el sistema operatiu i tipus i versió del navegador utilitzat per l'internauta. La seva principal utilitat radica en que, a partir d'aquesta informació, podem redirigir els nostres usuaris cap a pàgines optimitzades per el seu navegador o realitzar qualsevol altre tipus d'acció en el context d'un navegador determinat.
\$HTTP_ACCEPT_LANGUAGE	Ens retorna les abreviacions de la llengua considerada com a principal per el navegador. Aquesta llengua o llengües principals poden ser escollides en el menú d'opcions del navegador. Aquesta variable resulta també extremadament útil per enviar a l'internauta a les pàgines escrites en la seva llengua, si és que existeixen.
\$HTTP_REFERER	Ens indica la URL des de la qual l'internauta ha tingut accés a la pàgina. Molt interessant per generar botons "Enrere" dinàmics o per crear els nostres propis sistemes d'estadístiques de visites.
\$PHP_SELF	Ens torna una cadena amb la URL de l'script que s'està executant. Molt interessant per crear botons per recarregar la pàgina.
\$HTTP_GET_VARS	Es tracta d'un array que emmagatzema els noms i continguts de les variables enviades a l'script per URL o per formularis GET.
\$HTTP_POST_VARS	Es tracta d'un array que emmagatzema els noms i continguts de les variables enviades a l'script mitjançant un formulari POST.
\$HTTP_COOKIE_VARS	Es tracta d'un array que emmagatzema els noms i continguts de les cookies.
\$PHP_AUTH_USER	Emmagatzema la variable usuari quan s'efectua l'entrada a pàgines d'accés restringit. Combinat amb \$PHP_AUTH_PW resulta ideal per controlar l'accés a les pàgines internes del lloc.
\$PHP_AUTH_PW	Emmagatzema la variable password quan s'efectua l'entrada a pàgines d'accés restringit. Combinat amb \$PHP_AUTH_USER resulta ideal per controlar l'accés a les pàgines internes del lloc.
\$REMOTE_ADDR	Mostra la direcció IP del visitant.
\$DOCUMENT_ROOT	Ens torna el path físic en el que es troba allotjada la pàgina en el servidor.
\$PHPSESSID	Guarda l'identificador de sessió de l'usuari.

2. PROGRAMACIÓ EN PHP

Operadors aritmètics

Ens permeten realitzar operacions numèriques entre variables i nombres.

Signe	Operació
+	Suma
-	Resta
*	Multiplicació
/	Divisió
%	Retorna el residu de la divisió

2. PROGRAMACIÓ EN PHP

Operadors de comparació

S'utilitzen principalment en les sentències condicionals (if) per comparar dos variables i verificar si compleixen o no la propietat de l'operador.

Signe	Operació
==	Igualtat
!=	Desigualtat
<	Menor que
<=	Menor o igual que
>	Major que
>=	Major o igual que

2. PROGRAMACIÓ EN PHP

Operadors lògics

S'utilitzen en combinació amb els operadors de comparació quan l'expressió de la condició ho requereix.

Signe	Operació
And, &&	I
Or,	O
!	No

2. PROGRAMACIÓ EN PHP

Operadors d'increment

Serveixen per augmentar o disminuir en una unitat el valor d'una variable.

Podem distingir dos tipus: Operadors de pre-increment i de post-increment.

El 1er tipus primer incrementa el valor i després retorna \$a mentre que el 2on

tipus primer retorna el valor actual i després incrementa la variable en un.

Funcionament	Tipus	Acció que realitza
++\$a	Pre-increment	Incrementa \$a en un i després retorna \$a.
\$a++	Post-increment	Retorna \$a i després incrementa \$a en un.
--\$a	Pre-decrement	Decrementa \$a en un i després retorna \$a.
\$a--	Post-decrement	Retorna \$a i després decrementa \$a en un.

2. PROGRAMACIÓ EN PHP

Operadors combinats

Aquest tipus d'expressions no són més que abreviacions d'altres formes més clàssiques ja que escriure `$variable += 10` és el mateix que fer : `$variable = $variable+10`.

Funcionament	Acció que realitza
<code>\$variable += 10</code>	Suma 10 a <code>\$variable</code>
<code>\$variable -= 10</code>	Resta 10 a <code>\$variable</code>
<code>\$variable .= "afegeixo"</code>	Concatena les cadenes <code>\$variable</code> i "afegeixo"

3. ESTRUCTURES DE CONTROL

Tot arxiu de comandes de PHP es compona d'una sèrie de **sentències**.

Una sentència pot ser **una assignació, una crida a funció, un bucle, una instrucció condicional** i inclús una sentència que no faci res.

Les sentències normalment acaben amb punt i coma. També es poden col·locant-les entre claus.

Les sentències que tractarem són:

- If
- Switch
- While
- For

3. ESTRUCTURES DE CONTROL

Sentència condicional IF

Permet l'execució condicional de fragments de codi, és a dir, podem executar un codi o un altre, depenent del compliment o no d'una condició (simple o composta).

Exemple 1

```
if ($a > $b)  
print "a és major que b";
```

//Si el contingut de la variable \$a és major que el de \$b s'executa la instrucció print, la qual escriuria en pantalla: “a és major que b”

Exemple 2

```
if ($a > $b) {  
    print "a és major que b";  
    $b = $a;  
}
```

//Si volem que dins la instrucció if hi hagi varies sentències les podem agrupar utilitzant claus {...}

3. ESTRUCTURES DE CONTROL

Sentència condicional IF

Les sentències if es poden anidar indefinidament posant-les unes dins de les altres sentències.

Si volem que realitzi alguna acció quan no es compleixi una condició, podem utilitzar la paraula reservada else, que és una extensió de if.

Exemple 3

```
if ($a > $b) {  
    print "a és major que b";  
}else {  
    print "a NO és major que b";  
}
```

//Si \$a és major que \$b imprimirà per pantalla: "a és major que b"

En qualsevol altre cas (que sigui menor o igual que b), mostrarà: "a NO és major que b"

3. ESTRUCTURES DE CONTROL

Sentència condicional IF

També és possible realitzar **comparacions de cadenes** en les que es tenen en compte les majúscules i minúscules.

La comparació es realitza amb els codis ascii de forma que:

"albert"
"A"
"pepe"
"abc"

serà menor que
serà menor que
serà menor que
serà diferent de

"joan"
"a"
"pepe2"
"ABC"

3. ESTRUCTURES DE CONTROL

Sentència condicional SWITCH

En moltes ocasions, es vol comparar una variable (o expressió) amb molts valors diferents, i executar una part de codi diferent depenent de a quin valor és igual.

Amb la sentència switch, podem **executar unes o altres instruccions depenent del valor de la variable que estiguem utilitzant.**

3. ESTRUCTURES DE CONTROL

Sentència condicional SWITCH

Exemple utilitzant Switch

```
$posicio = "amunt";

switch($posicio) {
    case "amunt": // Bloc 1
        echo "La variable conté el valor amunt";
        break;
    case "avall": // Bloc 2
        echo "La variable conté el valor avall";
        break;
    default: // Bloc 3
        echo "La variable conté un altre valor";
        echo " diferent de amunt i avall";
}
```

Equivalent utilitzant varios if

```
$ posicio = "amunt";

if($ posicio == "amunt") { // Bloc 1
    echo "La variable conté el valor amunt";
}
else{
    if($ posicio == "avall") { // Bloc 2
        echo "La variable conté el valor avall";
    }
    if(($ posicio != "amunt")&&
($ posicio != "avall"))
    { // Bloc 3
        echo "La variable conté un altre valor";
        echo " diferent de amunt i avall";
    }
}
```

3. ESTRUCTURAS DE CONTROL

Control d'iteracions WHILE

La instrucció while és un senzill bucle que avalua una condició, mentre la condició d'una instrucció while és certa es reiterarà el bucle, la sintaxis és la mateixa que la utilitzada en C.

Exemple 4

```
$i=0;  
while ($i<10)  
{  
    echo "El valor de i és ", $i,"<br>";  
    $i++;  
}
```

3. ESTRUCTURES DE CONTROL

Control d'iteracions FOR

És la instrucció de bucles més completa. En una sola instrucció ens permet controlar tot el funcionament del bucle.

El primer paràmetre del for, serveix per inicialitzar la variable del bucle, el segon paràmetre indica la condició que s'ha de complir perquè el bucle segueixi executant-se i el tercer paràmetre és una instrucció que s'executa al final de cada iteració i serveix per modificar el valor de la variable d'iteració.

Exemple 5

```
for($i=0 ; $i<10 ; $i++)  
{  
    echo "El valor de i és ", $i,"<br>";  
}
```

4. FUNCIONS

Una Funció és un **bloc de codi** al que **li passem una sèrie de paràmetres** i ens **retorna un valor**.

Declaració: la instrucció **function** seguida del nom que li vulguem donar, i després **entre parèntesis la llista d'arguments** separats per comes, encara que també hi haurà funcions que no tinguin cap argument.

```
function nom_de_la_funcio (arg_1, arg_2, ..., arg_n)
{
    bloc de codi
}
```

En PHP no podem redefinir una funció prèviament declarada, i a més en PHP3, les funcions han de definir-se sempre abans de que s'invoquin, en PHP4 aquest requeriment ja no existeix.

4. FUNCIONS

La instrucció RETURN

S'utilitza per acabar l'execució de la funció en qualsevol punt d'aquesta.

```
function major ($x, $y)
{
    if ($x > $y) {
        return $x." és major que".$y;}
    else {
        return $y." és major que".$x;}
}
```

```
function major ($x, $y) //Millor que l'altra
{
    $msg = "";
    if ($x > $y) {
        $msg = $x." és major que".$y;}
    else {
        $msg = $y." és major que".$x;}
    return $msg;
}
```

Pot retornar-se qualsevol tipus de valor,[}] incloent taules i objectes.

Només permet retornar un valor, si en volem retornar més d'un haurem d'utilitzar una taula (array).

```
}
```

4. FUNCIONS

Paràmetres de les funcions

Existeixen dues formes de passar els paràmetres a una funció, per **valor** o per **referència**.

- **Variable per valor**--> Passi el que passi en aquesta no modificarà per res el contingut de la variable.

```
//paràmetres per valor  
echo suma ($a, $b);
```

- **Variable per referència**--> Qualsevol canvi en la funció sobre la variable ho farà per sempre.

```
//paràmetres per referència  
suma (&$a, $b)
```

En PHP, per defecte, les variables es passen per valor. Per fer-ho per referència hem de posar al davant de la variable un **ampersand** (&).

4. FUNCIONS

Valors per defecte

- Han de ser una **expressió constant**, i no una variable o membre d'una classe.
- **Han d'estar a la dreta de qualsevol paràmetre sense valor per defecte**, si no ho fem així el PHP en retornarà un error.

```
<?php
function suma ($y, $x=1)
{
    $x = $x + 1;
    return $x+$y;
}
?>
```

4. FUNCIONS

Recursivitat

El PHP també permet la recursivitat: **una funció es pot cridar a sí mateixa.**

Exemple sobre una funció que comprova si un nombre és enter o no.

```
function esEnter($nombre) {  
    if ($ nombre > 1) {  
        return (esEnter($ nombre -1));  
    } elseif ($ nombre < 0) {  
        /* com que els nombres són simètrics el convertim a positiu */  
        return (esEnter((-1) * $ nombre -1));  
    } elseif (($ nombre > 0) AND ($ nombre < 1)) {  
        return ("NO");  
    } else {  
        return ("SÍ"); /* el zero és enter per definició */  
    }  
} //fi function
```

4. FUNCIONS

Variables estàtiques i variables globals

- **Variables estàtiques (static)**--> Es defineixen dins la funció.
- **Variables globals (global)**--> No es poden declarar dins d'una funció, agafem el valor que tinguin en aquell moment i que pot ser modificat en la funció.

```
function comptador ()  
{  
    static $count = 0;  
    $count = $count + 1;  
    return $count;  
}  
echo comptador()."<BR>"; // imprimirà 1  
echo comptador()."<BR>"; // imprimirà 2  
echo comptador()."<BR>"; // imprimirà 3
```

```
var $a = 1;  
function ver_a()  
{  
    global $a;  
    echo $a."<BR>"; // imprimirà el valor de $a  
    $a += 1; // sumem 1 a $a  
}  
echo ver_a(); // imprimirà 1  
echo ver_a(); // imprimirà 2  
$a = 7;  
echo ver_a(); // imprimirà 7  
echo ver_a(); // imprimirà 8
```

5. INCLUDES

Els includes són trossets de codi escrits en php que fan una crida a una altra pàgina diferent. Així si modifiquem aquesta pàgina, totes les altres pàgines que l'inclueixin es modificaran.

Tot seguit adjuntarem un exemple per veure com s'aplica:

```
<html>
  <head><title>Prova include</title></head>
  <body>
    <table width="100%" border="0" cellspacing="0">
      <tr>
        <td width="25%"><?php include("menu.php"); ?></td>
        <td width="75%">   </td>
      </tr>
    </table>
  </body>
</html>
```

6. ACCÉS A LA BASE DE DADDES

Com accedim a les Bases de Dades:

- A partir de les funcions que ens proposa el PHP per cada tipus de base de dades.
- L'establiment d'un diàleg a partir d'un idioma universal: SQL (Structured Query Language) el qual és comú a totes les bases de dades. Aquest llenguatge resulta molt potent i fàcil d'aprendre.

MySql

És gratuïta, eficient i la més estesa en combinació amb PHP.

Tot seguit explicaré com connectar-se a una BDD MySql des de Php.

Cal dir que suposarem que tenim la BDD ja creada i ubicada en un hipotètic servidor.

6. ACCÉS A LA BASE DE DADOS

Veiem primer com realitzar la **connexió al servidor MySQL**:

```
$link=mysql_connect("nom de host","usuari","password");
```

Una vegada fet això, haurem de **seleccionar la base de dades** que utilitzarem:

```
mysql_select_db("nom de la base de dades",$link);
```

Ara estudiarem com realitzar les operacions més bàsiques sobre la nostra BDD: inserció, cerca, modificació i eliminació.

Base de Dades inicial que utilitzarem: taula persones

Nom	Cognoms	Sexe	Edat
Juan	Puertas	Home	22
Francisco Javier	Orellana	Home	22

6. ACCÉS A LA BASE DE DADOS

Inserció d'un nou registre

Per afegir un nou registre ho farem amb la següent instrucció en PHP.

```
mysql_query("INSERT INTO persones(Nom,Cognoms,Sexe,Edat)
VALUES ('Marta','García',Dona','24')", $link);
```

Nom	Cognoms	Sexe	Edat
Juan	Puertas	Home	22
Francisco Javier	Orellana	Home	22
Marta	García	Dona	24

6. ACCÉS A LA BASE DE DADOS

Obtenció de dades de la BDD

Com realitzar una consulta sobre la BDD:

```
$result=mysql_query("SELECT * FROM persones WHERE  
Edat='22'", $link);
```

MySQL permet obtenir els resultats de cada registre guardats en la variable `$result` de tres formes diferents:

- 1- `mysql_fetch_row($result)`:** Retorna el resultat en un array indexat (0,1,2,...)
- 2- `mysql_fetch_array($result)`:** Retorna el resultat en un array associatiu
- 3- `mysql_fetch_object($result)`:** Retorna el resultat en un registre.

La única diferència entre les tres instruccions anteriors és la forma en la que les dades són retornades.

6. ACCÉS A LA BASE DE DADDES

Obtenció de dades de la BDD

Ara mostrarem les tres formes diferents de com recuperar els valors de la variable:

mysql_fetch_array(\$result)	mysql_fetch_row(\$result)	mysql_fetch_object(\$result)
<pre>while(\$row=mysql_fetch_array(\$result)) { echo " \$row[Nom] "; echo " \$row[Cognoms] "; echo " \$row[Sexe] "; };</pre>	<pre>while(\$row=mysql_fetch_row(\$result)) { echo " \$row[0] "; echo " \$row[1] "; echo " \$row[2] "; };</pre>	<pre>while(\$row=mysql_fetch_array(\$result)) { echo " \$row->Nombre" ; echo " \$row->Apellidos "; echo " \$row->Sexo "; };</pre>

6. ACCÉS A LA BASE DE DADES

Modificació de registres de la BDD

Podem actualitzar les dades d'un registre ja existent, de la forma següent:

```
mysql_query ("UPDATE persones SET Edat=23 WHERE  
Nom='Juan'", $link);
```

Nom	Cognoms	Sexe	Edat
Juan	Puertas	Home	23
Francisco Javier	Orellana	Home	22
Marta	García	Dona	24

6. ACCÉS A LA BASE DE DADOS

Eliminació de registres de la BDD

La sintaxis per eliminar registres de la BDD és:

```
mysql_query ("DELETE FROM persones WHERE  
Sexe='Home'", $link);
```

Nom	Cognoms	Sexe	Edat
Marta	García	Dona	24

INTRODUCCIÓ AL PHP



Sara Rosés i Garriga sara@dunlock.com
Dunlock-Enginyeria Informàtica
Volcànica 2007