

Cos'è il PHP e come funziona

Cos'è Php?

Php è un modulo aggiuntivo per web servers che permette di creare delle pagine web dinamiche.

In pratica, una pagina in Php è composta sia da tag html, sia da parti in codice di programmazione Php. Ogni porzione di codice, limitato entro i tag "<?php" e ">", viene eseguito prima di essere inviato all'utente che ne fa richiesta.

Il funzionamento è praticamente lo stesso delle pagine Asp.

Un esempio di pagina Php è il seguente:

```
<html>
<body>

<!-- Parte di comandi Html -->

<? #parte di codice Php ?>

<!-- Altri comandi Html -->

</body>
</html>
```

Come funziona?

Per funzionare, Php necessita di un suo motore di scripting (script engine) che esegue le parti in codice prima che il web server invii la pagina all'utente. Attualmente esiste la versione 3.0 del motore Php (funzionante praticamente su tutti i web server) disponibile presso il sito <http://www.php.net/>.

Quando un utente richiede una pagina Php, il motore esegue il codice contenuto all'interno di quella pagina. Durante l'esecuzione, il codice produce delle informazioni in formato html. Infine il file (in completo formato html) viene inviato all'utente.

Se provate a visualizzare il sorgente di una pagina in Php potete notare che non vi compare nessuna riga di codice Php. Il vantaggio è proprio questo: nessun utente esterno, tranne il webmaster, può accedere al codice e modificarlo. Per l'utente esterno, la pagina in Php è esattamente uguale a una qualsiasi pagina in Html.

Un consiglio per comprendere al meglio il presente tutorial ed avere poi uno strumento efficace per programmare con esattezza i propri script è quello di scaricarsi il Manuale Php 3.0 dal sito <http://www.php.net/>.

Alcune particolarità

- I tag che indicano la presenza di codice Php sono "<?php" e ">", ma è possibile configurare Php in modo che si accorga anche dei tags "<?" e ">" di più semplice scrittura.
- I commenti all'interno del codice Php devono essere compresi tra i tags "/*" e "*/". Un altro tipo di commento si ottiene mettendo il tag "#" all'inizio di una riga. La riga intera non verrà eseguita.
- I comandi Php per restituire del codice Html sono: PRINT("<!-- html code --->"); ECHO("<!-- html code --->");
- Il comando per la concatenazione di stringhe è il punto (.). Ad esempio, il comando echo("Ciao"." ". "Alberto") produce la stringa "Ciao Alberto"
- Ogni comando puro in Php deve essere concluso con il punto e virgola (;)
- Per essere eseguiti, i file devono avere estensione ".php3" (a meno che il web server non sia configurato diversamente). Ogni file del nostro sito che contiene istruzioni Php deve avere quella estensione.

Primi esempi

Il primo esercizio potrebbe consistere nella realizzazione di un file prova.php3 contenente il seguente codice:

```
<HTML>
<BODY>
```

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

```
</BODY>  
</HTML>
```

Questo file produce una schermata in cui vengono visualizzati i parametri di configurazione di Php nel server in cui viene eseguito il codice. Può essere utile per verificare la presenza di Php in un server o per conoscere il valore delle variabili di sistema (per programmatori esperti).

Un altro esempio potrebbe essere il seguente.

```
<HTML>  
<BODY>
```

```
<? echo("Ultima modifica: ".date("d/m/Y",filemtime($PATH_TRANSLATED))); ?>
```

```
</BODY>  
</HTML>
```

Esso visualizza la data di ultima modifica del vostro file. Può essere utile nella costruzione di siti web in quanto spesso ci si dimentica, dopo una serie di aggiornamenti, di cambiare la data. In questo modo, la pagina visualizzerà correttamente la data, senza doversene preoccupare troppo.

Alcuni esempi iniziali

Questa sezione è dedicata ad alcuni esempi iniziali per affrontare il mondo Php. Sono abbastanza semplici, ma possono essere combinati tra loro e con un po' di fantasia si possono ottenere dei risultati efficaci per le nostre pagine.

Accodare un file

Può essere utile a tutti webmaster inserire alla fine di ogni file del proprio sito una riga di informazioni sull'autore (nome, cognome, email).

A questo proposito ci viene in aiuto il comando INCLUDE di Php.

Inserendo il seguente codice alla fine delle vostre pagine verrà caricato automaticamente un codice html contenuto nel file /path/codice.txt:

```
<? include("/path/codice.txt"); ?>
```

il contenuto del file /path/codice.txt potrebbe essere:

```
<hr size="1">
```

```
Pagina realizzata da <a href="mailto:stain@dei.unipd.it">Alberto Mucignat</a>
```

Vedremo in seguito le specifiche del comando INCLUDE e del suo "simile" REQUIRE.

Contatore in ogni pagina

Per inserire un contatore nelle nostre pagine che venga modificato ad ogni visualizzazione, possiamo inserire il seguente codice:

```
<?  
if (!file_exists("/path/counter.txt")):  
$file=fopen("/path/counter.txt","w");  
$num=0;  
else:  
$file=fopen("/path/counter.txt","r+");  
$num=fgets($file,20);  
endif;  
$num++;  
print("Visitors: ".$num);
```

```
fputs($file,$num);  
fclose($file);  
?>
```

Il contatore non tiene conto delle visite ricevute dallo stesso utente nel corso della stessa sessione. Per fare ciò bisogna imparare a configurare i COOKIES o aspettare la nuova versione Php 4 per avere tutte queste funzionalità a disposizione con le variabili SESSION.

Esistono anche delle librerie che hanno implementato le funzionalità collegate alle variabili di sessione.

Caricare un'immagine

Può accadere di dover visualizzare, ad ogni apertura di una pagina, un'immagine diversa. Un semplice codice per gestire dinamicamente questa esigenza può essere il seguente:

```
<?  
$num=rand(1,10);  
print("<img src=\"images/img\".$num.\".gif\">");  
?>
```

Ovviamente bisogna nominare le immagini come images/img1.gif, images/img2.gif, ecc.

Un'altra possibilità è visualizzare un'immagine diversa a seconda del giorno della settimana in cui il visitatore richiede la pagina. Le pagine devono essere chiamate images/Mon.gif, images/Tue.gif, ecc, e il codice potrebbe essere:

```
<?  
$img=date("D");  
print("<img src=\"images/\".$img.\".gif\">");  
?>
```

Potete sbizzarrirvi in questo genere di giochetti a seconda della vostra fantasia.

Spedire mail

Con Php è possibile spedire mail da una pagina web.

Inizialmente si deve creare una pagina che chiameremo "master", completa di una FORM per l'invio dei dati, e di una pagina "slave" che riceve i dati e invia la mail.

Per quanto riguarda la sintassi di una form, vi rimando a un qualsiasi tutorial html.

La pagina "master" deve avere al suo interno un codice di questo tipo:

```
<FORM ACTION="send.php3" METHOD="POST">  
Nome e Cognome:<BR>  
<INPUT TYPE="text" NAME="name" SIZE="20" MAXLENGTH="30"><BR>  
Email:<BR>  
<INPUT TYPE="text" NAME="email" SIZE="20" MAXLENGTH="30"><BR>  
Titolo:<BR>  
<INPUT TYPE="text" NAME="subject" SIZE="20" MAXLENGTH="30"><BR>  
Testo della mail:<BR>  
<INPUT TYPE="text" NAME="text" ROWS="60" COLS="10" MAXLENGTH="200"><BR>  
<INPUT TYPE="submit" VALUE="INVIA">  
</FORM>
```

Nel file send.php3 (il nostro file "slave") sarà contenuto il seguente codice Php:

```
<?  
if (isset($email)):  
# l'indirizzo email a cui inviare la mail  
$target="info@sito.it";  
mail($target,$subject,"Nome: ".$nome."\\nTitolo: ".$subject."\\n\\n".$text);  
endif;  
?>
```

Come gestire un sito

Alcune importanti applicazioni di Php permettono ai webmaster di gestire i loro siti in maniera efficiente e senza troppo lavoro. Un problema nei siti composti da molte pagine è il fatto di dover fare piccoli cambiamenti grafici con corrispondente

aggiornamento di tutti i files e una mole di lavoro notevole da parte del webmaster.

A questo vengono in aiuto i programmi tipo **Dreamweaver** che permettono la creazione dei cosiddetti templates. I templates sono dei file "tipo" che rappresentano la struttura portante (e comune) di tutti i file di un sito. I templates permettono di gestire il design di un sito semplicemente definendo tutti gli attributi comuni alle pagine di un sito.

In Php è utile usare sistemi di questo tipo, facilitati dal linguaggio di programmazione.

Due istruzioni importanti per questi sistemi sono la INCLUDE e la REQUIRE.

L'istruzione INCLUDE (già vista in precedenza), quando inserita all'interno di pagine web, permette di includere un determinato file.

L'istruzione REQUIRE differisce solamente per il fatto che il file incluso può contenere a sua volta delle istruzioni in Php che verranno eseguite prima di restituire la pagina all'utente.

Questo ci permette, ad esempio, di gestire dei banner pubblicitari. Il metodo è semplice. Inizialmente si deve fare in modo che tutte le pagine del nostro sito abbiano al loro interno la seguente riga:

```
<? if (file_exists("/path/banner.inc")): require("/path/banner.inc"); endif; ?>
```

nel file banner.inc può essere contenuto un codice Php che restituisce un'immagine con un link esterno e nel contempo effettua altre operazioni utili (nel caso di banner pubblicitari: selezione random, conteggio delle impression, ecc).

Ma torniamo ai templates e cerchiamo di strutturare tutti i file del sito in modo che siano utilizzabili da Php:

```
<? include("/path/header.inc"); ?>
```

```
<!-- Html Code --->
```

```
<? include("/path/footer.inc"); ?>
```

In questo modo basta creare i file header.inc e footer.inc e il gioco è fatto. Il file header potrebbe contenere le seguenti righe:

```
<HTML>
<HEAD>
<TITLE>My Site</TITLE>
<META NAME="author" CONTENT="Alberto Mucignat">
<META NAME="description" CONTENT="Questo è il mio sito gestito in Php">
</HEAD>

<BODY BGCOLOR="RED">
<CENTER><B><H2>My Site</H2></B></CENTER>
<HR SIZE="1">
```

E il file footer.inc potrebbe contenere le righe:

```
<CENTER>
<HR SIZE="1">
Sito realizzato da <A HREF="mailto:stain@dei.unipd.it">Alberto Mucignat</A>
</CENTER>
</BODY>
</HTML>
```

In questo modo ogni aggiornamento alla grafica del nostro sito è facile e immediato. Basta infatti andare a modificare i files header.inc e footer.inc per fare sì che tutte le pagine del sito vengano aggiornate automaticamente.

Un'altra utilità nella gestione dei siti è la realizzazione delle cosiddette "restricted areas", ovvero aree riservate a cui serve una password per accedere.

Per fare ciò basta creare un file index.php nella directory del sito protetta e inserire la form:

```
<FORM ACTION="main.php3" METHOD="post">
Inserire password:<BR>
<INPUT TYPE="password" NAME="pwd" SIZE="20" MAXLENGTH="30"><BR>
<INPUT TYPE="submit" VALUE="ENTRA">
</FORM>
```

Il file main.php3 è il file che fa la verifica della password e contiene il codice:

```
<? if ((isset($pwd)) AND ($pwd=="pass128")): ?>
```

```
<!-- Restricted area - Html code --->
```

`<? else: ?>`

`<!-- Error message -->`

`<? endif; ?>`

Per maggior sicurezza, la password può essere memorizzata in un file e il codice diventa:

`<? if ((isset($pwd)) AND ($pwd==include("passwd.txt"))): ?>`

Nel caso di sistemi unix/linux è possibile criptare la password con il comando CRIPT e in tal caso, se il file passwd.txt contiene la password precedentemente criptata, la sintassi diventa:

`<? if ((isset($pwd)) AND (crypt($pwd)==include("passwd.txt"))): ?>`

Questi sono solo alcuni degli accorgimenti che si possono usare nella costruzione e gestione dei siti web. Il consiglio è di non fermarsi alle cose viste, ma di stimolare la fantasia ricercando nuove strategie e soluzioni.

Php e database

L'impiego più importante di Php si ottiene nel campo delle interrogazioni di database presenti sul server. Php consente al programmatore e di interfacciarsi con numerosi database tra i più importanti nel mondo della programmazione:

- PostgreSQL
- MySql
- Oracle
- Adabas
- filePro
- ODBC

più alcuni altri meno noti.

In particolare, la connessione ODBC permette di interfacciarsi con database tipo Access.

L'utilizzo dei comandi per la connessione ai database prescinde dalla conoscenza dei database stessi. Il linguaggio SQL è richiesto per formulare le query, come pure la conoscenza del significato di indici, chiavi, relazioni, ecc.

In buona sostanza, un webmaster che intende organizzare un sistema che preveda la consultazione di un database via web, nella maggioranza dei casi deve conoscere approfonditamente la struttura stessa del database, nonché il significato dei dati contenuti.

Per quanto riguarda SQL, il linguaggio standard di interrogazione sui database, esistono molti tutorial e manuali in rete. Mi limiterò a dire che per interrogazione o query si intende una serie di comandi che un qualsiasi database esegue restituendo una serie di tuple (in generale righe o parti di righe di una o più tabelle del database).

In questo tutorial consideriamo operazioni su un database di tipo MySQL e diamo per scontata la conoscenza del database stesso. Al lettore attento non risulterà difficile trasformare le operazioni spiegate per ricondurle al suo database preferito.

In pratica le operazioni Php sui database potrebbero suddividersi in tre livelli:

- apertura/chiusura connessione
- invio interrogazioni SQL
- lettura risultati di una query

Le operazioni di apertura/chiusura di una connessione a un database servono ad aprire/chiedere un collegamento con il database prescelto, al fine di poter fare delle operazioni di lettura/scrittura e trasmettere i dati in formato html.

Per aprire/chiedere una connessione in MySql sono disponibili i comandi:

`mysql_connect(host,login,password)`
`mysql_close()`

Guide

Per inviare interrogazioni si possono usare i comandi:

```
mysql_db_query(database,query,$db)
```

Per leggere i risultati si usano dei comandi che vedremo in seguito.

Una prima ricerca

Per compiere la nostra prima ricerca nel database poniamo di conoscere l'esistenza, la configurazione e il significato del contenuto di una tabella chiamata Annunci.

La tabella Annunci contiene i campi:

- ID chiave primaria
- Nome nome dell'autore dell'annuncio
- Email email
- Testo testo dell'annuncio

Una ricerca all'interno della tabella potrebbe essere fatta mediante il codice seguente:

```
<?  
$db=mysql_connect("localhost","","") or die("Errore durante la connessione a  
MySQL");  
$result=mysql_db_query("database_annunci","SELECT * FROM Annunci");  
mysql_close;  
?>
```

Questi comandi permettono di memorizzare all'interno della variabile \$result tutti gli annunci presenti nella tabella Annunci.

Per scriverli occorre prima fare l'operazione:

```
$row=mysql_fetch_row($result);
```

e, successivamente, la prima riga del risultato sarà disponibile nella variabile array \$row. Per scrivere il risultato sarà necessaria una riga tipo:

```
print($row[1]."/".$row[2]."/".$row[3]."/".$row[4]."/");
```

Per conoscere il numero di righe presenti nel risultato di una query basta usare il comando:

```
$num_rows=mysql_num_rows($result);
```

Un'altra operazione molto utile è quella che restituisce il numero di campi presenti nel set di risultati della query appena compiuta:

```
$num=mysql_num_fields($result);
```

A questo punto siamo pronti per formulare query più complesse. Ad esempio, creando una form che restituisca una parola su cui fare la ricerca nella variabile \$ricerca, si può scrivere:

```
$sql="SELECT * FROM Annunci WHERE Testo LIKE '%".$stringa.%'"  
$result=mysql_db_query("db_annunci",);
```

\$result sarà composto da tutti gli annunci che contengono la parola memorizzata in \$stringa.

Un sito di annunci

Alla fine dell'ultima lezione accennavamo alla costruzione di un database di annunci e alla sua consultazione tramite pagine web.

Vediamo ora di affrontare meglio il problema e di costruire un sito che permetta sia l'inserimento che la ricerca.

La tabella Annunci che memorizza gli annunci è stata già descritta e contiene i campi:

- ID chiave primaria
 - Nome nome dell'autore dell'annuncio
 - Email email
 - Testo testo dell'annuncio
- Passiamo alla pratica ricordandoci che lo strumento necessario per inviare dati da un file html/php3 ad un altro è la form, come descritto nelle lezioni precedenti.

Per fare funzionare il tutto creiamo dei file nel seguente modo:

- show.php3 mostra gli annunci presenti
- insert.php3 inserisce un annuncio
- search.php3 cerca una o più parole nel database

Inserimento di un annuncio

Supponiamo di ricevere dei dati da una form nelle variabili \$nome,\$email,\$testo. Per inserire un nuovo annuncio basta fare le seguenti operazioni:

```
<?
# dati da modificare a seconda del database
$host="localhost";
$user="";
$pass="";

$db=mysql_connect($host,$user,$pass) or die ("Errore nella connessione al
database");
$sql="INSERT INTO Annunci (Nome,Email,Testo)
VALUES('".$nome."','".$email."','".$testo."')";
mysql_db_query("db_annunci",$sql);
mysql_close();
?>
```

In questo caso il database inserisce automaticamente una chiave unica che contraddistingue l'annuncio nel campo ID della tabella Annunci.

Ovviamente si possono anche fare delle operazioni di controllo sulla validità dell'indirizzo email o per verificare che alcuni campi non siano nulli.

Mostrare gli annunci

Per mostrare gli annunci presenti nel database basta inserire il seguente codice:

```
<?
# dati da modificare a seconda del database
$host="localhost";
$user="";
$pass="";

$db=mysql_connect($host,$user,$pass) or die ("Errore nella connessione al
database");
$sql="SELECT * FROM Annunci";
$result=mysql_db_query("db_annunci",$sql);
```

Guide

```
while ($row=mysql_fetch_array($result) {
print("<B>Nome</B>: <A HREF=\"\".$row[\"Email\"].\"\\\">\".$row[\"Nome\"].\"</A><BR>");
print("<B>Testo dell'annuncio</B>:<BR>");
print(htmlentities($row[\"Testo\"])."
");
print("<HR SIZE=\\\"1\\\">");
}
mysql_close();
?>
```

La funzione htmlentities() permette di "tradurre" le lettere accentate e gli altri caratteri particolari nella loro codifica html, in modo da poter venire visualizzate correttamente dai vari browser.

Ricerca di un annuncio

Per la ricerca di un annuncio particolare, pensiamo di ricevere una stringa \$str da una form, in modo che \$str contenga una serie di parole da ricercare separate da spazi.

Il seguente codice permette di effettuare la ricerca e visualizzare i risultati:

```
<?
# dati da modificare a seconda del database
$host="localhost";
$user="";
$pass="";

$db=mysql_connect($host,$user,$pass) or die ("Errore nella connessione al
database");
$sql="SELECT * FROM Annunci ";
$where="";
$words=explode(" ", $str);
$j=0;
while (isset($words[$j])) {
$where=$where."Testo LIKE '%".$words[$j]."%";
if (isset($words[$j+1])):
$where=$where." OR ";
endif;
$j++;
}
$result=mysql_db_query("db_annunci",$sql.$where);
while ($row=mysql_fetch_array($result) {
print("<B>Nome</B>: <A HREF=\"\".$row[\"Email\"].\"\\\">\".$row[\"Nome\"].\"</A><BR>");
print("<B>Testo dell'annuncio</B>:<BR>");
print(htmlentities($row[\"Testo\"])."<BR>");
print("<HR SIZE=\\\"1\\\">");
}
mysql_close();
?>
```

Ovviamente le varianti sono molteplici e ognuno può sbizzarrirsi come meglio crede.

Volendo dare una traccia da seguire per creare un sistema del genere crediamo di aver detto abbastanza. Vorremmo solo far presente che il passaggio del codice da un database all'altro richiede un'attenta traduzione da parte del programmatore perché è possibile che le operazioni richiedano un numero diverso di operatori a seconda del database.