



PHP


Hypertext Preprocessor

Matteo Caneva & Enrico Pezzano

- Linguaggio di programmazione interpretato
- Software libero
- Principalmente utilizzato per sviluppare applicazioni web lato server
 - Script a riga di comando
 - Applicazioni stand-alone con interfaccia grafica

Introduzione

- Creato nel 1994 da Rasmus Lerdorf


Rasmus Lerdorf

08/08/95

★

Traduci messaggio in italiano

Announcing the Personal Home Page Tools (PHP Tools) version 1.0.

These tools are a set of small tight cgi binaries written in C. They perform a number of functions including:

- . Logging accesses to your pages in your own private log files
- . Real-time viewing of log information
- . Providing a nice interface to this log information
- . Displaying last access information right on your pages
- . Full daily and total access counters
- . Banning access to users based on their domain
- . Password protecting pages based on users' domains
- . Tracking accesses ** based on users' e-mail addresses **
- . Tracking referring URL's - HTTP_REFERER support
- . Performing server-side includes without needing server support for it
- . Ability to not log accesses from certain domains (ie. your own)
- . Easily create and display forms
- . Ability to use form information in following documents

Here is what you don't need to use these tools:

- . You do not need root access - install in your ~/public_html dir
- . You do not need server-side includes enabled in your server
- . You do not need access to Perl or Tcl or any other script interpreter
- . You do not need access to the httpd log files

The only requirement for these tools to work is that you have the ability to execute your own cgi programs. Ask your system administrator if you are not sure what this means.

The tools also allow you to implement a guestbook or any other form that needs to write information and display it to users later in about 2 minutes.

The tools are in the public domain distributed under the GNU Public License. Yes, that means they are free!

For a complete demonstration of these tools, point your browser at: <http://www.io.org/~rasmus>

—
Rasmus Lerdorf
ras...@io.org
<http://www.io.org/~rasmus>

Annuncio degli strumenti personali della pagina iniziale (strumenti PHP) versione 1.0.

Questi strumenti sono un insieme di piccoli binari stretti scritti in C. Eseguono una serie di funzioni tra cui:

- . Registrazione degli accessi alle tue pagine nei tuoi file di log privati
- . Visualizzazione in tempo reale delle informazioni del registro
- . Fornire un'interfaccia piacevole a queste informazioni di registro
- . Visualizzazione delle ultime informazioni di accesso direttamente sulle tue pagine
- . Contatori di accesso totali giornalieri e totali
- . Vietare l'accesso agli utenti in base al loro dominio
- . Password che protegge le pagine in base ai domini degli utenti
- . Tracciamento degli accessi ** in base agli indirizzi e-mail degli utenti **
- . Tracciamento dell'URL di riferimento - Supporto HTTP_REFERER
- . Esecuzione di include sul lato server senza che sia necessario il supporto del server
- . Possibilità di non registrare gli accessi da determinati domini (ad esempio il tuo)
- . Creare e visualizzare facilmente moduli
- . Possibilità di utilizzare le informazioni del modulo nei seguenti documenti

Ecco cosa non è necessario utilizzare questi strumenti:

- . Non hai bisogno di un accesso root - installa nella tua directory ~/public_html
- . Non sono necessari gli accessi lato server abilitati nel server
- . Non hai bisogno di accedere a Perl o Tcl o qualsiasi altro interprete di script
- . Non è necessario accedere ai file di registro httpd

L'unico requisito per questi strumenti è che tu abbia la capacità di eseguire i propri programmi cgi. Chiedi al tuo sistema amministratore se non sei sicuro di cosa significhi.

Gli strumenti ti consentono anche di implementare un libro degli ospiti o qualsiasi altro modulo che ha bisogno di scrivere informazioni e mostrarlo agli utenti più tardi in circa 2 minuti.

Gli strumenti sono di dominio pubblico distribuiti sotto GNU Licenza pubblica. Sì, significa che sono gratis!

Per una dimostrazione completa di questi strumenti, punta il tuo browser a: <http://www.io.org/~rasmus>

—
Rasmus Lerdorf
ras...@io.org
<http://www.io.org/~rasmus>

Storia

- Il pacchetto venne riscritto da Lerdorf stesso poco tempo dopo
 - Nuova versione: PHP 2.0
 - Integrare il codice PHP con il codice HTML
- Nel 1998 oltre il 10% dei siti web presenti su internet utilizzava PHP 3.0
- Versione più recente: PHP 7.2.X

Storia

- Sintassi molto simile a C/C++
- Si integra anche con altri linguaggi di programmazione
- Dispone di un archivio (Pear)
 - Librerie
 - Estensioni

Caratteristiche

- Sono precedute dal simbolo \$
- Il nome della variabile non può essere un numero
- Una variabile può contenere qualsiasi carattere
- Le variabili sono casesensitive

Variabili

- Non è necessario specificare la tipologia di variabile che si vuole inizializzare

```
$hello = "Ciao mondo!";
```

- Per mostrare il contenuto della variabile, è possibile utilizzare l'istruzione *echo*

```
echo $hello;
```

Variabili

- Esistono 2 modi per inizializzare le stringhe in PHP:
 - Racchiusa da apici doppi, vengono interpretati anche i caratteri speciali.
 - Racchiusa da apici singoli, il testo all'interno viene salvato così com'è

Stringhe

- Apici doppi

```
echo "Hello vale: $hello";
```

- Apici singoli

```
$one = 'Laura';  
$two = 'è';  
$three = 'bella';  
  
echo $one . ' ' . $two . ' ' . $three;  
echo "$one $two $three";
```

Stringhe

- PHP gestisce gli array come mappe associative
- Sono estremamente versatili
- I singoli elementi possono essere rimossi con *unset()*;
- Possono essere “attraversati” tramite *foreach*

Array

- Esempio di “attraversamento” tramite *foreach*

```
<?php
$residents = array(
    'Torino' => 908551,
    'Milano' => 1315791,
    'Roma' => 2753000
);

$sum = 0;

foreach($residents as $key=>$value) {
    echo "La città $key ha $value abitanti. ";
    $sum += $value;
}

echo "E in tutto ci sono $sum abitanti!";
?>
```

Array

- Le funzioni possono avere un mix di parametri opzionali
- Il concetto di overloading è inutile
- Qualunque variabile di qualsiasi tipo creata dentro ad una funzione, è considerata locale
- Per specificare variabili globali all'interno di una funzione si utilizza *global \$variabile*;

Funzioni

- Esempio

```
<?php
function esperimento() {
    global $a;
    $a = "tre";
    $b = "quattro";
    echo "Durante: $a $b";
}

$a = "uno";
$b = "due";

echo "Prima: $a $b";
esperimento();
echo "Dopo: $a $b";
?>
```

Funzioni

- È possibile includere una pagina PHP all'interno di un'altra

```
<?php  
include 'header.php';  
echo "Ciao mondo!";  
include 'footer.php';  
?>
```

- include / included_once / require / require_once

Inclusioni

- Include → segnale di avviso
- Require → segnale di errore fatale di compilazione
- Included_once/require_once → quando il comando è ripetuto più volte

Inclusioni (utilizzi)

- Funzionalità di passaggio (supportate da HTTP)
- Tipo GET
- Tipo Post
- Cookie

Gestione dei Parametri

```
<?php
/*
 * Contollo se è stato ricevuto il parametro "pippo" tramite metodo
GET.
 * Se è rilevato lo visualizzo; se non è rilevato avverto l'utente.
 */

if( isset( $_GET['pippo'] ) ) {
    $a = $_GET['pippo']; // Salvo nella variabile $a il parametro
    "pippo"

    $a = htmlentities( $a ); // Questo "disinnesca" eventuali
    contenuti HTML contenuti in $a

    echo $a; // Mostro a video
} else {
    echo "Nessun parametro di nome pippo trovato. Devi inviarlo!";
}
?>
```

Gestione dei Parametri

```
<?php
/*
 * In questo caso la variabile $n viene stampata a video,
 successivamente
 * viene incrementata di 1, il ciclo termina quando $n è uguale a 9
 (il ciclo conta da 0 fino a 9 quindi 10 volte).
 */

for($n=0; $n<=9; $n++){
    echo $n;
}

// Output -> 0123456789
?>
```

Ciclo *for*

```
<?php
/*
 * In questo caso vediamo come stampare a video dei risultati
 estratti
 * da un database MySQL, eseguendo una query SQL.
 * Questo script è incompleto, in quanto prima è necessario
 effettuare
 * il collegamento al database MySQL tramite mysqli.
 */

$result = $mysqli->query("SELECT id FROM persone");

while($row = $result->fetch_assoc()) {
    echo $row['id'] . "<br />";
}

/*
 * In questo modo estraiamo dei dati da un database MySQL
 * sotto forma di array che verrà poi stampato tramite un ciclo
 while
 */
?>
```

Ciclo *while*

```
<?php
/*
 * Questo è un commento. Altri modi per commentare sono i simboli // e #
 * Questo tipo di commenti non ha bisogno degli asterischi (*) all'inizio
 * di ogni riga, ma lo si fa per convenzione. I simboli // e #
 * commentano solo il testo che si trova dopo di essi e fino alla fine della riga;
 * non c'è bisogno di finire il commento con qualche carattere speciale.
 */

/*
 * Prima definiamo una funzione chiamata "plural".
 * Restituisce una "s" se l'argomento passato non è 1.
 */
function plural($number) {
    // L'operatore ternario (!= ? :) è una struttura condizionale
    // simile a if-else: (condizione ? true : false)
    return ($number != 1) ? 's' : '';
}
```

90 *bottle of beer*

```
// Definiamo una variabile chiamata $a_capo che contiene un a capo HTML,  
// un carattere di ritorno e uno di nuova riga:  
$a_capo = "<br />\r\n";  
  
for($i=99; $i>0; $i--) {  
    echo "$i bottle" . plural($i) . " of beer on the wall,$a_capo";  
    // Non abbiamo bisogno di ripetere il comando echo per ogni riga:  
    echo "$i bottle" . plural($i) . " of beer . $a_capo  
    Take one down, pass it around,$a_capo" .  
    (((($i - 1) != 0) ? $i - 1 : 'no more') .  
    ' bottle' . plural($i - 1) . " of beer on the wall.$a_capo$a_capo";  
    /*  
    * PHP permette di creare stringhe su più righe, finché  
    * trova un punto e virgola (;) che termina l'istruzione.  
    * Un punto (.) concatena insieme più stringhe.  
    * Le variabili, che iniziano con il carattere "$", sono interpretate  
    * anche dentro i doppi apici ("), ma non dentro gli apici singoli (').  
    * Le funzioni, come plural(), non sono interpretate fra gli apici.  
    */  
}  
echo "Go to the store, $a_capo buy some more, $a_capo 99 bottles of beer on the wall!";  
?>
```

90 *bottle of beer*

```
<?php
// Connessione a MySQL con l'estensione MySQLi
$mysqli = new mysqli('localhost', 'nome_utente_database', 'password_database', 'nome_database');

/*
 * Questo è il modo ufficiale di controllare se una connessione MySQL è fallita,
 * MA $connect_error non è supportato fino a PHP 5.2.9 e 5.3.0.
 */
if ($mysqli->connect_error) {
    die("Connessione fallita, motivi: " . $mysqli->connect_error);
}

// Proviamo una query ipotizzando che stiamo elaborando dati provenienti da una simil-rubrica:
$SQLQuery = $mysqli->query("SELECT indirizzo, telefono FROM utenti WHERE nome = 'Francesca' ");

// Da questa query otterremo l'indirizzo e il telefono della persona di nome Francesca,
// ipotizzando sia presente nella tabella utenti del database in questione.

// Ora "trasformiamo" i dati ricevuti dalla query in un array associativo utilizzabile successivamente
$row = $SQLQuery->fetch_assoc();

// Mostriamo a video per esempio il numero di telefono dall'array
echo "Il numero di telefono di Francesca è: " . $row['telefono'];

// Chiusura della connessione al database
$mysqli->close();
?>
```

Esempio di *Database*
