

PHP

PHP: Hypertext Preprocessor

- **PHP** - acronimo ricorsivo di "PHP: Hypertext Preprocessor" (preprocessore di ipertesti)
- è un linguaggio di *scripting interpretato*, con licenza open source originariamente concepito per la realizzazione di *pagine web dinamiche*
- attualmente è utilizzato principalmente per sviluppare *applicazioni web lato server*

Wikipedia

- linguaggio di scripting *server side*
 - gli script vengono eseguiti lato server
- il server *genera dinamicamente codice html* che viene inviato al client (il client non riceve script)
- il *motore php* è *open source* e si affianca al server http (es Apache)
- disponibile su diverse piattaforme
- *html embedded* (il codice è inserito all'interno di pagine html)
- è un'alternativa a Perl, Python, JSP, ASP e ASP.NET

- sintassi simile ai linguaggio C e Java
- fornisce connettività a vari database (MySQL, PostgreSQL ...)
- pagine facili da creare e non necessitano di compilazione
- è stato introdotto dal danese *Rasmus Lerdorf (1995)*
- il linguaggio di scripting è meno strutturato di un linguaggio di programmazione classico



Subject: Announce: Personal Home Page Tools (PHP Tools)

Date: 1995/06/08

newsgroups: comp.infosystems.www.authoring.cgi

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

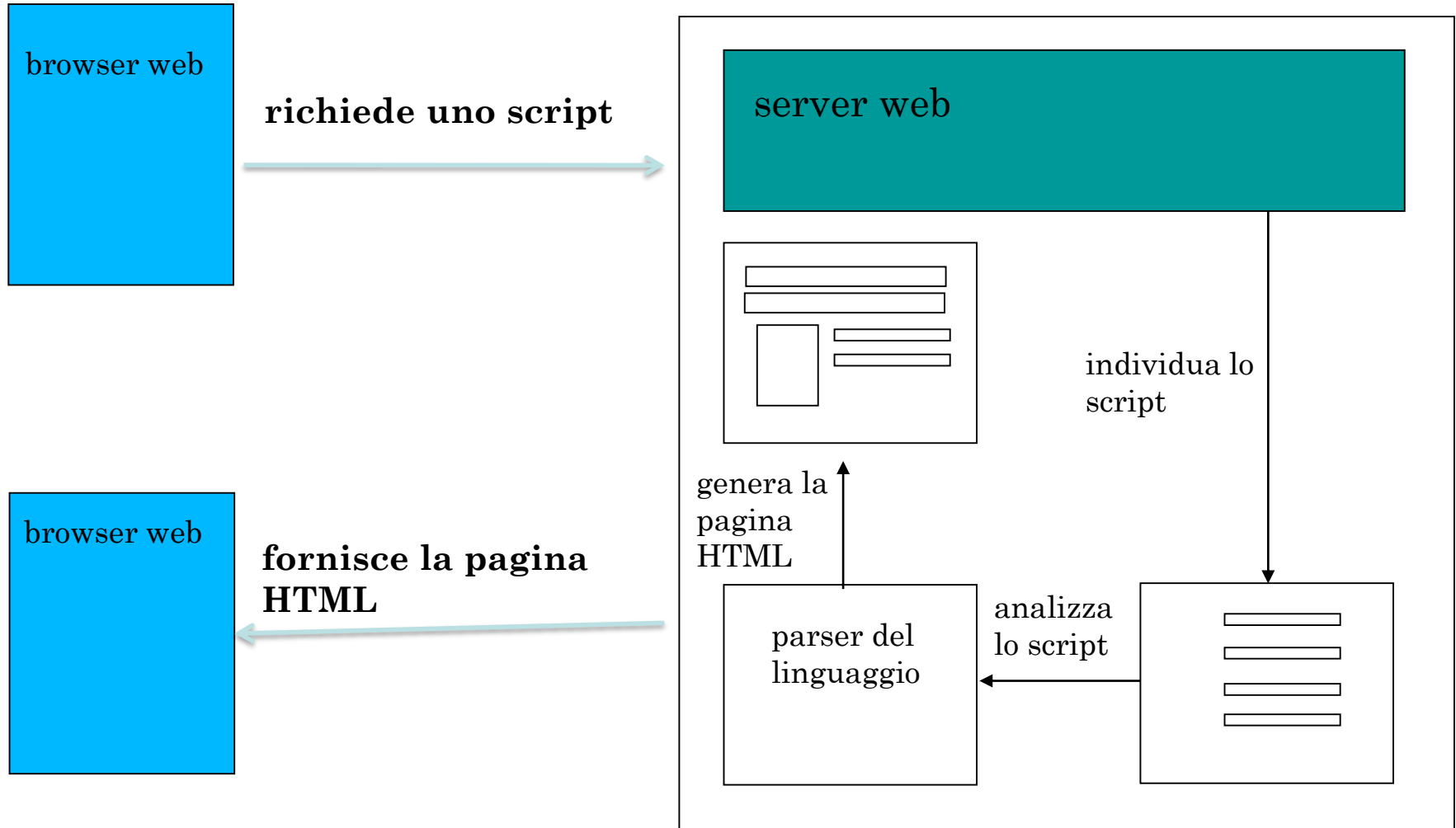
The only requirement for these tools to work is that you have the ability to execute your own cgi programs. Ask your system administrator if you are not sure what this means.

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!

- una pagina PHP è un file con estensione .php e contiene:
 - testo
 - tag HTML
 - comandi script
- in PHP la parte di script può apparire in qualsiasi punto della pagina html e viene delimitato dai delimitatori di script:
 - `<? e ?>`
 - es. il comando: `<? $x=10 ?>`
assegna alla variabile x il valore 10

- quando il *web server* riceve la richiesta di una pagina con estensione .php sa che la pagina deve essere processata dal motore PHP
- il *risultato* dell'esecuzione dello script viene restituito al web server che lo invia al client
- gli elementi HTML sono passati direttamente al server web mentre le parti di script vengono interpretate dal motore di scripting PHP



- l'output di uno script viene "scritto" sulla finestra del client
- l'istruzione di stampa di PHP è echo (o print)
 - `<? echo "Ciao"; ?>`
- oppure
 - `<? echo("Ciao"); ?>`
- oppure
 - `<? print "Ciao"; ?>`

```
<html>
<head>
  <title>Hello World in PHP</title>
</head>
<body>
  <p>
    <?php
      /* esempio script PHP */
      echo "Oggi è il ";
      print(Date("l F d, Y"));
      echo "<br>";
    ?>
    esempio script PHP!
  </p>
</body>
</html>
```

- le variabili in PHP si denotano con una sequenza di caratteri preceduti dal simbolo **\$**
- ***case sensitive*** (maiuscole e minuscole sono diverse)
- devono iniziare con una lettera o il carattere sottolineatura () possono contenere caratteri numerici
 - **`$miavariabile`**
 - **`$_MiaVariabile`**
 - **`$var25`**

```
<?php
  $FatturaNum = 548;
  $Nome = "Paolino";
  $Cognome = "Paperino";
  $GiorniLavorati = 234;
  $CostoGiornaliero = 145;
?>
<html>
<head>
  <title>Fattura in PHP</title>
</head>
<body>
  <?php
    print("<h2>Fattura al signor $Nome $Cognome</h2>\n");
    print("<h3>Giorni lavorati: $GiorniLavorati</h3>\n");
    print("<h3>Costo Giornaliero: $CostoGiornaliero euro</h3>\n");
    print("<h3>Costo Totale: ");
    print($GiorniLavorati * $CostoGiornaliero);
    print(" euro</h3>\n");
  ?>
</body>
</html>
```

- l'ambito (*scope*) di una variabile in PHP è la *pagina* stessa
- ogni variabile esiste solo per lo *script* dove è definita, alla fine della computazione scompare
 - possono coesistere variabili con lo stesso nome se definite in pagine diverse
 - non è possibile usare il valore di una variabile in uno script diverso da dove è stata definita
- le uniche variabili globali permesse sono i superglobalarray
 - array globali predefiniti che sono visibili da qualsiasi pagina dell'applicazione
- vecchie versioni di PHP permettevano la definizione esplicita di variabili globali
 - nelle versioni più recenti le globali sono disattivate di default nel file php.ini con la direttiva `register_globals = Off`.

- le stringhe sono sequenza di caratteri alfanumerici
- sono racchiuse tra i caratteri ' oppure "
- l'unione di stringhe si effettua con il carattere punto (.)
- `$nome = "Paolo";`
- `$cognome='Rossi';`
- `$nomeintero=$nome.$cognome;`

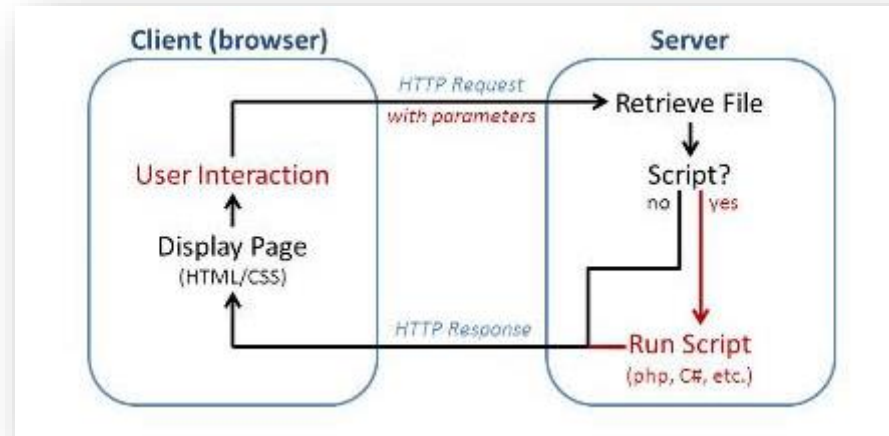
- **strlen(stringa)**
 - restituisce il numero di caratteri della stringa
- **trim/ltrim/rtrim(stringa)**
 - trim elimina spazi all'inizio e alla fine della stringa, ltrim all'inizio (sinistra) rtrim alla fine (destra)
- **substr(stringa, intero1 [,intero2])**
 - restituisce la sottostringa che inizia alla posizione intero1 eventualmente fino a intero1+intero2
- **str_replace(str1, str2, str3)**
 - restituisce una nuova stringa dove sostituisce tutte le occorrenze di str1 con str2 in str3
- **strtolower/strtoupper(stringa)**
 - converte tutti i caratteri in minuscolo/maiuscolo
- il confronto tra stringhe si effettua con gli usuali operatori di confronto **==, <, >**

```
<html>
<body>
<?php
    $txt1="Benvenuto";
    $txt2="al corso di php";
    echo ($txt1 . " " . $txt2."<br \>");
    echo ("La stringa Jack e' lunga ".strlen("Jack")."<br \>");
    echo ("La sottostringa (corso) nella stringa ($txt2) ");
    echo ("inizia alla posizione ".strpos($txt2,"corso"));
?>
</body>
</html>
```


- sono disponibili variabili globali definite al di fuori dello script (*superglobalarray*)
- variabili del server, sono definite dal server web e quindi variano a seconda del server usato
 - sono definite come l'array `$_SERVER`
- `$_SERVER["PHP_SELF"]`
 - nome dello script corrente
- `$_SERVER["SERVER_NAME"]`
 - nome del server
- `$_SERVER["HTTP_USER_AGENT"]`
 - browser che ha inoltrato la richiesta
- `phpinfo()`
 - fornisce informazioni sullo stato di PHP, tra cui tutte le variabili predefinite
 - è utile per vedere se MySQL è installato e viene visto correttamente da PHP

```
<html>
<body>
<?php
    $eta["Aldo"] = "32";
    $eta["Giovanni"] = "30";
    $eta["Giacomo"] = "34";
    foreach ($eta as $indice => $valore)
        echo $indice , " ha " , $valore , " anni<br/>";
?>
</body>
</html>
```

- un *modulo* HTML è un insieme di oggetti che contengono valori
- il modulo ha un attributo *action* che specifica l'azione da compiere al momento della conferma del modulo
- se nell'attributo action è specificato il nome di un file php questo riceve i valori contenuti nel modulo
- il passaggio di dati può avvenire utilizzando due metodi:
 - GET
 - POST



- il metodo **GET** consiste nell'accodare i dati all'indirizzo della pagina richiesta
- il nome della pagina è seguito da un punto interrogativo e dalle coppie nome/valore dei dati che si vogliono inviare
 - nome e valore sono separati dal simbolo '='
 - le diverse coppie nome/valore sono separate dal simbolo '&'.
- ``
- la stringa che si trova dopo il punto interrogativo, contenente nomi e valori dei parametri, viene detta *query string*

- con il metodo è POST i dati vengono inviati in maniera da non essere direttamente visibili per l'utente, attraverso la richiesta HTTP che il browser invia al server
- il metodo POST è utilizzabile *solo con i moduli*

```
<html>
<body>
<form action="recupera-post.php" method="post">
Nome: <input type="text" name="campoNome" id="campoNome" />
Eta': <input type="text" name="campoEta" id="campoEta" />
<input type="submit" />
</form>
</body>
</html>
```

html

```
<html>
<body>
Benvenuto <?php echo $_POST["campoNome"]; ?>!<br />
La tua eta' e' <?php echo $_POST["campoEta"]; ?>
</body>
</html>
```

php

- si possono passare i parametri da url (metodo get)
- `http://nome_host/recupera_get.php?nome=a1&eta=18`
- se il metodo del form è get, allora si deve usare la variabile **`$_GET`**
- La variabile **`$_REQUEST`** include gli elementi di **`$_GET`**, **`$_POST`**, **`$_COOKIE`**

- all'apertura di un *file*, bisogna specificare le seguenti informazioni:
 - modalità
 - in sola lettura, in sola scrittura, o in lettura e scrittura
 - dove posizionare il puntatore
 - all'inizio o alla fine del file
 - cosa fare se il file non esiste

- **fopen(filename, mode)**
- mode
 - **'r'** solo lettura, puntatore all'inizio del file
 - **'r+'** lettura e scrittura, puntatore all'inizio del file
 - **'w'** solo scrittura, puntatore all'inizio del file
 - se il file non esiste, tenta di crearlo
 - **'w+'** lettura e scrittura, puntatore all'inizio del file
 - se il file non esiste, tenta di crearlo
 - **'a'** solo scrittura, puntatore alla fine del file
 - se il file non esiste, tenta di crearlo
 - **'a+'** lettura e scrittura, puntatore alla fine del file
 - se il file non esiste, tenta di crearlo

```

<html>
<body>
<?php
    $mio_file=fopen("diritti.txt","r")
        or exit("Errore nell'apertura del file");
echo "<h1>File correttamente aperto</h1>";
echo "<hr />";
//Lettura riga per riga
while(!feof($mio_file))
{
    $riga = fgets($mio_file);
    echo $riga. "<br />";
}
echo "<hr />";
fclose($mio_file);
echo "<h1>File chiuso</h1>";
?>
</body>
</html>

```

```
<html>
<body>
<?php
    $mio_file=fopen("nuovo.txt","w")
        or exit("Errore nell'apertura del file");
echo "<h1>File nuovo.txt creato</h1>";
echo "<hr />";
fwrite($mio_file, "ITIS");
fwrite($mio_file, " Leonardo ");
fwrite($mio_file, "da Vinci");
echo "<hr />";
fclose($mio_file);
echo "<h1>File chiuso</h1>";
?>
</body>
</html>
```

```
<html>
<body>
<?php
    $mio_file=fopen("nuovo.txt","a+")
        or exit("Errore nell'apertura del file");
echo "<h1>File nuovo.txt aperto</h1>";
echo "<hr />";
fwrite($mio_file, " - Parma");
echo "<hr />";
fclose($mio_file);
echo "<h1>File chiuso</h1>";
?>
</body>
</html>
```

- i **cookie HTTP** sono utilizzati dalle applicazioni web lato server per archiviare e recuperare informazioni a lungo termine **sul lato client**
- *il primo uso di cookie in HTTP risale al 1994, usati per controllare se i lettori del sito di Netscape lo avessero già visitato in precedenza*
- *i cookie, e in particolare i cookie di terza parte, sono comunemente usati per memorizzare le ricerche di navigazione degli utenti; questi dati sensibili, possono essere una **potenziale minaccia alla privacy** degli utenti; proprio questo ha indotto le autorità europee e degli Stati Uniti a regolamentarne l'uso mediante una legge nel 2011*
 - *la legislazione europea impone a tutti i siti degli stati membri di informare gli utenti che il sito usa certi tipi di cookie*



```
<?php
    setcookie("Scuola", "ITIS", time()+3600);
    setcookie("Citta", "Parma", time()+3600);
?>
<html>
<head>
    <title>esempio cookie</title>
</head>
<body>
    <p>setcookie deve essere scritto prima del tag html</p>
</body>
</html>
```

```
<html>
<head>
  <title>Visualizzazione di tutti i cookie</title>
</head>
<body>
  <?php
  /* Visualizzazione di tutti i cookie */
  foreach ($_COOKIE as $nome => $valore)
  {
    print("il cookie $nome ha valore ($valore) <br>");
  }
  ?>
</body>
</html>
```

- il protocollo http è *stateless*: non ci sono variabili globali alle pagine
- una *variabile di sessione* è utilizzata per memorizzare informazioni valide *per tutte le pagine* di un singolo utente (*sessione*) in un'applicazione (*sito web*)
- le variabili session sono *temporanee* e vengono eliminate quando l'utente lascia il sito web
- nell'array associativo `$_SESSION` è possibile salvare le variabili di sessione
- la sessione va inizializzata con `session_start()` ;
- l'inizializzazione della sessione va inserita *prima* del tag `<html>`


```
<?php
session_start();
?>
<html>
<head>
    <title>Esempio variabile session</title>
</head>
<body>
<?php
    if (isset($_SESSION["contatore"]))
        $_SESSION["contatore"]=$_SESSION["contatore"]+1;
    else
        $_SESSION["contatore"]=1;
    $cont = $_SESSION["contatore"];
    echo "Contatore=$cont" ;
?>
</body>
</html>
```

- eliminazione di una variabile di sessione
 - `<?php unset($_SESSION["contatore"]); ?>`
- chiusura forzata della sessione
 - `<?php session_destroy(); ?>`



php - mysql

```
<?php
    $host = "nome_host";
    $user = "nome_utente";
    $password = "password_utente";

    $connect = new mysqli($host, $user, $password);
    if ($connect->connect_errno) {
        die("Impossibile connettersi all' host"
            . $connect->connect_error());
    }
?>
```

```
...
if ($connect->select_db("motociclismo")) {
    echo("Database esistente");
    exit;
}

$comandoCreazioneDB = "CREATE DATABASE motociclismo "
    . "DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci";

if($connect->query($comandoCreazioneDB) === false) {
    echo("errore nella creazione del database");
    exit;
}
```

```
$sql = "SELECT * FROM moto INNER JOIN pilota ON codmoto = moto"
      . "where marca = '" . $_GET["marca"] . "'";
$result = $connect->query($sql);

if ($result->num_rows > 0) {
    echo "<table><tr><th>Pilota</th><th>Nome</th><th>Marca</th></tr>";
    while($row = $result->fetch_assoc()) {
        echo "<tr><td>". $row["Pilota"] . "</td><td>". $row["nome"]
            . "</td><td>". $row["marca"] . "</td></tr>";
    }
    echo "</table>";
}
else {
    echo "0 risultati";
}
```



https://www.w3schools.com/php/php_mysql_intro.asp

DOCUMENTAZIONE