



<http://www.php.net>

- 1 [Storia](#)
- 2 [Caratteristiche](#)
- 3 [Installazione](#)
- 4 [Inserire codice PHP in HTML](#)
- 5 [Linguaggio](#)
- 6 [Esempi di codice PHP](#)
- 7 [PHP e MySQL](#)

[PHP4 - ZEND Technology](#)

[Bibliografia - Sitografia](#)



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## Storia

**PHP e' l'acronimo di Hypertext Preprocessor.**

**La storia di PHP comincia nel 1994 con il PHP/FI (Personal Home Page / Form Interpreter) scritto da Rasmus Lerdorf.**

**A seguito del successo riscosso nel popolo della rete, Rasmus scrisse una seconda versione che offriva un maggior numero di funzioni ed il supporto per interfacciare il database miniSQL.**

**Nel 1997 la costituzione di un team di lavoro porta alla riscrittura completa del codice con la versione php.**

**Con la versione 4 il team di lavoro ha potenziato l'implementazione dell'impalcatura object-oriented e con il prodotto ZEND (PHP4) offre un notevole incremento delle prestazioni.**



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## Caratteristiche

- Open Source software;
- Linguaggio di scripting server- side ;
- linguaggio HTML - embedded :  
una pagina in PHP e' composta sia da tag HTML, sia da parti in codice PHP delimitate dai tag `<? ..... ?>` (anche `<?php ..... ?>`)
- fornisce un modo per inserire istruzioni nelle pagine HTML per creare contenuto dinamico;
- sintassi simile al linguaggio C, al Java ed al Perl;
- utilizzabile su piattaforme Unix e Windows 9X/Me/NT/2000;
- meccanismo di esecuzione degli script simile al linguaggio ASP in ambiente Windows;
- fornisce un'ampia varietà di funzioni (array, grafiche, matematiche, stringhe, HTTP, IMAP, XML ecc.);
- supporta un gran numero di funzioni Open Database Connectivity (ODBC) per la connessione ai più importati DBMS.

---

**PHP può essere configurato per essere eseguito come:**

- modulo del Web server
- script CGI separato





# Caratteristiche

La struttura del linguaggio comprende:

- **Tipi (array, floating point, integer, object, string);**
- **variabili, costanti, espressioni;**
- **operatori (arithmetic, assignment, bitwise, comparison, error control, incrementing/decrementing, logical, string operators);**
- **strutture di controllo (if, elsif, do, for, while,...);**
- **funzioni;**
- **classi ed oggetti.**

Caratteristiche aggiuntive:

- **gestione degli errori;**
- **gestione dei cookies**  
(cookies : meccanismo per memorizzare dati nel browser remoto e tenere traccia degli utenti);
- **gestione delle forms;**
- **interazione con il file system del server;**
- **creazione e gestione immagini;**
- **utilizzo di file remoti;**
- **interazione con i piu' importanti motori database relazionali:**
  - **dBase, Informix, InterBase, mSQL, MySQL, Oracle, PostgreSQL, etc.**



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





# Caratteristiche

## Linguaggio HTML - embedded

### Esempio:

```
<html>
<body>

<center>
<?phpinfo(); ?>
</body>
</html>
```

L'output dell'esempio precedente e' una pagina con una tabella con informazioni sul Web server e sulle variabili HTTP.

- l'utente esterno vede una pagina PHP (ad esempio con estensione .php ) come una qualsiasi pagina HTML;
- visualizzando il sorgente di una pagina PHP non compare nessuna traccia del codice;
- solo il webmaster puo' accedere al codice e modificarlo.

### Sequenza di analisi codice PHP

- le istruzioni PHP sono lette ed interpretate dal Web server, le fasi di analisi lessicale/sintattica e la fase di esecuzione del codice (PHP linguaggio interpretato), sono eseguiti contemporaneamente.
- Se durante la prima fase non vengono rilevati errori, il Web server passa all'esecuzione del codice PHP esegue la parte dinamica ed invia, al browser dell'utente, l'elaborato completamente in formato HTML.
- Nel caso di errore di sintassi, l'interpretazione viene interrotta inviando al client un messaggio di errore.





## Caratteristiche

### Successo del linguaggio

In questi ultimi tempi, PHP si sta affermando come linguaggio principe di scripting per la generazione di pagine dinamiche.

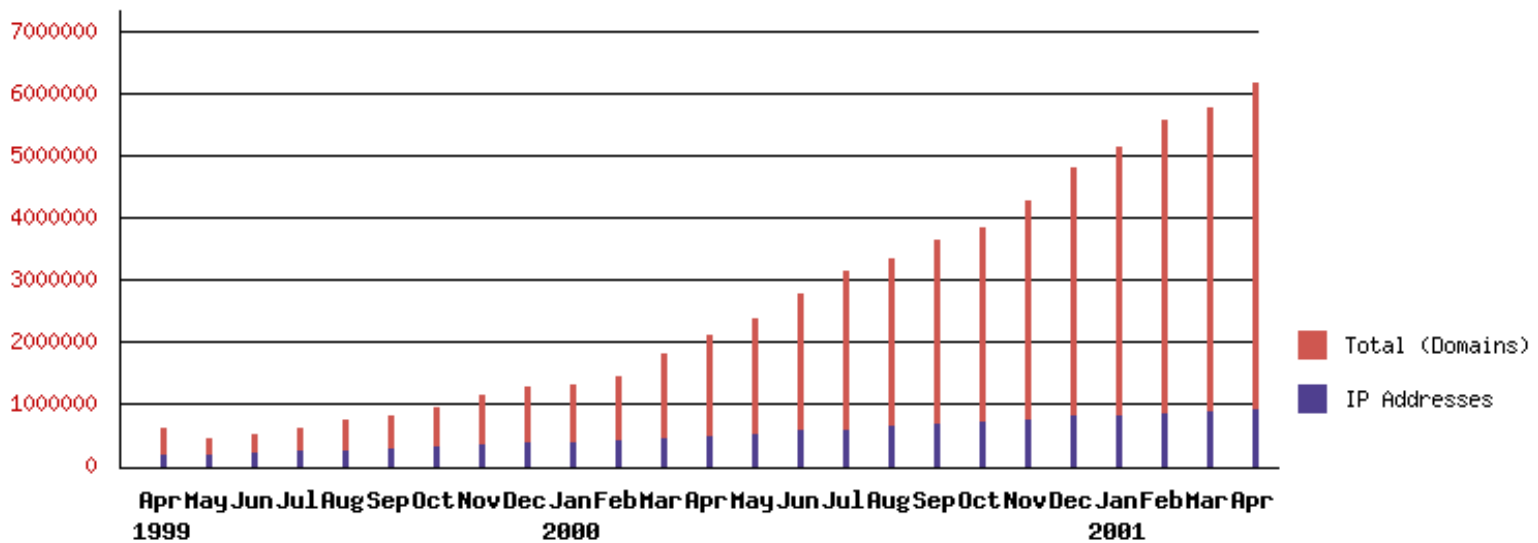
Inesorabilmente sta superando l'uso dei CGI.

- PHP linguaggio embedded, non necessita di ulteriori file esterni per essere eseguito.
- Ogni script CGI deve essere caricato sul server in determinate directory con determinati permessi.
- Con PHP, una volta istruito il Web server, ogni pagina con all'interno il codice PHP, potrà essere eseguito in ogni directory (a partire dalla `document_root`).

## Usage Stats for April 2001

PHP: 6156321 Domains, 914146 IP Addresses

Source: [Netcraft](#)



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
 Daniela.Beatrici@polarnet.cnr.it





## Installazione - Configurazione (1)

L'utilizzo dell'interprete PHP all'interno del HTTP server, vale a dire Apache, fa sì che il file richiesto dal client venga elaborato direttamente all'interno del server senza effettuare chiamate di sistema, riducendo, in questo modo, l'utilizzo di risorse e quindi ottimizzando i tempi di risposta.

Il codice sorgente utilizzato è : `php-3.0.12.tar.gz`

Tutte le informazioni sulla configurazione e compilazione di PHP si trovano nel file `INSTALL`.

Steps per l'installazione:

- `gzip -d php-3.0.12.tar.gzip` (unzip del file compresso)
- `tar -C /usr/local -xvf php-3.0.12.tar` (installazione del codice sorgente nella directory `php-3.0.12`)
- `cd php-3.0.12` (posizionamento sulla directory di installazione)

Esecuzione dello script di configurazione:

- `./configure --with-apache=/usr/local/apache_1.3.9` (indica la posizione di Apache)  
`--with-mysql=/usr/local/mysql` (abilitazione utilizzo librerie per interfacciare il database MySQL)  
`--with-gd=path gd library` (utilizzo della libreria grafica GD)
- `make` (compilazione)
- `make install` (installazione)
- `cp php.ini-dist /usr/local/lib/php.ini` (copia il file che contiene le direttive per l'interprete)

Alla fine dell'installazione vengono copiati i files necessari per l'inclusione del modulo PHP all'interno del HTTP server Apache.



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## Installazione - Configurazione (2)

Finita l'installazione di PHP bisogna procedere con la nuova compilazione di Apache generando i nuovi Makefile contenenti i dati relativi all'utilizzo del modulo PHP.

Attivare, nel file di configurazione di Apache `httpd.conf`, uno speciale tipo MIME in modo che tutti i file con estensione `.php` vengano riconosciuti ed eseguiti come applicazioni:

**AddType application/x-httpd-php .php**

Steps per nuova compilazione di Apache:

- `cd /usr/local/Apache_1.3.9` (posizionamento sulla directory del sorgente di Apache)
- `./configure --disable-module=include` (esclusione modulo per il Server Side Include)  
`--activate-`  
`module=src/modules/php/libphp.a`  
(inclusione modulo PHP)
- Controllare che nel file `config.layout` sia inserito `prefix=/usr/local/httpd/bin`
- `make` (compilazione)
- `cd /usr/local/httpd/bin` (posizionamento sulla directory che contiene il daemon)
- `cp httpd httpd.old` (copia precedente versione del daemon)
- `cp /usr/local/apache_1.3.9/src/httpd .` (sostituisce il nuovo daemon - con integrati moduli)
- `./httpd -f /usr/local/httpd/conf/httpd.conf` (attiva nuovo daemon)



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## Installazione - Configurazione (3)

### File di configurazione **php.ini**

- situato su `/usr/local/lib/`
- nel caso di PHP generato come modulo di Apache, il file viene letto solamente quando il web server viene avviato o reinizializzato. Se si modifica la configurazione di PHP, occorre riavviare il web server Apache.
- La maggior parte delle direttive del file `php.ini` sono disponibili come direttive native del server Apache.
- In Apache, le direttive di PHP sono composte dal prefisso `php_` seguito dal nome che assume nel file di configurazione `php.ini`

Documentazione sulle direttive del file di configurazione:

<http://www.zend.com/manual/configuration.php#configuration.file>



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





# Inserire codice PHP nell' HTML

## Stili:

- tag `<? ..... ?>`
- tag `<php ..... ?>`  
(utilizzato nel caso di conflitto con il codice XML; e' possibile disattivare il primo stile utilizzando la direttiva `short_open_tag` del file `php.ini`)
- tag `<SCRIPT LANGUAGE="php">.....</SCRIPT>`

E' possibile inserire piu' istruzioni separandole con il punto e virgola.

### Esempio n. 1 :

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

### Esempio n. 2 :

```
.....
<? echo "Visualizza prima riga"; echo "Visualizza seconda riga" ?>
.....
```



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## Inserire codice PHP nell' HTML

### Inclusione di file

E' possibile includere file che possono contenere, a loro volta, ulteriori tag HTML e/o PHP.

### Comando INCLUDE

#### Esempio :

```
.....  
<? include "autor.html";?>  
.....
```

### Regole

- tutte le variabili del file contenitore sono automaticamente disponibili nel file incluso;
- ogni file incluso deve iniziare con i tag HTML, di seguito si puo' inserire codice PHP.



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## LINGUAGGIO

- Sintassi
- Variabili
- Tipi di dati
- Operatori
- Strutture di controllo
- Funzioni
  - Funzioni per MySQL

## SINTASSI

### Alcune regole

- nomi variabili case sensitive precedute dal carattere \$ (\$nome="Mario Ghezzi")
- nomi delle funzioni non sono case sensitive
- fine istruzioni delimitate dal ; (punto e virgola)

### Commenti

- /\* ..... \*/
- // ..... //
- # .....

## VARIABILI

- Variabili di Apache
- Variabili d'ambiente
- Variabili PHP





## Variabili di APACHE

**Benvenuto !**

Il tuo indirizzo IP e' :	150.146.133.10
Il tuo dominio e' :	mandrake.polarnet.cnr.it.133.146.150.in-addr.arpa
Il tuo sistema e' :	Mozilla/3.0 (compatible; WebCapture 2.0; Windows)
Hai richiesto lo script :	/srv/www/htdocs/CORSI/PHP/php12.php
Al server :	www.polarnet.cnr.it
In ascolto sulla porta :	80
Web Server :	Apache/1.3.26 (UnitedLinux) mod_ssl/2.8.10 OpenSSL/0.9.6g mod_python/2.7.10 Python/2.2.1 PHP/4.2.2 mod_perl/1.27
La root di questo server e' :	/srv/www/htdocs
Il nostro amministratore di sistema e' :	beatrici@polarnet.cnr.it

**Powered by : Apache/1.3.26 (UnitedLinux) mod\_ssl/2.8.10 OpenSSL/0.9.6g mod\_python/2.7.10 Python/2.2.1 PHP/4.2.2 mod\_perl/1.27**

<html>

.....  
Benvenuto !

```
<TABLE BORDER=1 cellpadding=2 cellspacing=2>
<TR><TD>Il tuo indirizzo IP e' : <TD> <?php echo $_SERVER['REMOTE_ADDR']; ?>
<TR><TD>Il tuo dominio e' :<TD> <?php echo $_SERVER['REMOTE_HOST']; ?>
<TR><TD> Il tuo sistema e' :<TD> <?php echo $_SERVER['HTTP_USER_AGENT']; ?>
<TR><TD>Hai richiesto lo script :<TD> <?php echo $_SERVER['SCRIPT_FILENAME']; ?>
<TR><TD>Al server :<TD> <?php echo $_SERVER['HTTP_HOST']; ?>
<TR><TD>In ascolto sulla porta :<TD> <?php echo $_SERVER['SERVER_PORT']; ?>
<TR><TD>Web Server :<TD> <?php echo $_SERVER['SERVER_SOFTWARE']; ?>
<TR><TD>La root di questo server e' :<TD> <?php echo $_SERVER['DOCUMENT_ROOT']; ?>
<TR><TD>Il nostro amministratore di sistema e' :<TD> <?php echo $_SERVER['SERVER_ADMIN']; ?>
</TR></TABLE>
```

Powered by : <?php echo \$\_SERVER['SERVER\_SOFTWARE']; ?>



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
Daniela.Beatrici@polarnet.cnr.it





## Variabili d'ambiente

Con PHP è possibile specificare ed utilizzare le variabili d'ambiente (dipendono dalla shell utilizzata).

Si possono richiamare all'interno di uno script PHP.

Esempio :

```
<?echo "Il mio path è : $PATH";?>
```

Il mio path è :

## Variabili PHP

PHP crea automaticamente variabili globali per tutti i dati ricevuti in una richiesta HTTP.

- `argv` : un array contenente i parametri passati allo script;
- `argc` : un array contenente i parametri passati dalla linea di comando;
- `PHP_SELF` : il nome dello script attualmente in esecuzione;
- `HTTP_COOKIE_VARS` : un array associativo contenente le variabili passate allo script tramite i cookies HTTP;
- `HTTP_GET_VARS` : un array associativo contenente le variabili passate allo script tramite una richiesta GET;
- `HTTP_POST_VARS` : un array associativo contenente le variabili passate allo script tramite il metodo POST.

Le ultime tre variabili hanno senso solamente se è stato abilitato il tracking delle variabili o nel file di configurazione, con `track_vars=On`, oppure all'interno dello script con la direttiva `php_track_vars`.



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## Tipi di Dati

Il PHP fornisce tre tipi di dato primitivi:

- **Integer**

Gli Integers, o interi, possono assumere diversi valori numerici esprimibili in differenti notazioni:

- `$a = 18;` # decimale
- `$a = -18;` # decimale negativo
- `$a = 022;` # notazione ottale; equivalente a 18 decimale
- `$a = 0x12;` # notazione esadecimale, equivalente a 18 decimale

- **Floating Point number**

- `$numero=0.033`
- `$numero=17.0E- 3`

- **String**

Esempi :

- `<? $numero = 10;`  
`$stringa = "Il numero è $numero";?>`  
 visualizza : Il numero è 10
- `<? $stringa = "Il numero è \"$numero\"";?>`  
 visualizza : Il numero è "10"
- `<? $stringa = 'Il numero è $num';?>`  
 visualizza : Il numero è \$num
- Altri caratteri speciali sono:

```

\n -> newline
\r -> carriage return
\t -> tabulazione
\\ -> backslash
\$ -> simbolo del dollaro

```





## Tipi di Dati

PHP fornisce due tipi di dato composti : ARRAY e OBJECT

### • Array (1)

Il PHP supporta sia gli array scalari che gli array associativi. Un array e' un tipo di dato composto che puo' contenere piu' di un valore di dato, indicizzati sia numericamente sia con stringhe.

Esempi :

- `$a = ("abc", "def", "ghi");`
- `$a[0] = "abc";`  
`$a[1] = "def";`  
`$a[2] = "ghi";`
- utilizzano le parentesi quadre vuote, i dati vengono accodati all'array:  
`$a[] = "lmn";`  
`$a[] = "opq";`  
In questo caso, l'array si allunga di 2 elementi:  
`$a[0] = "abc";`  
`$a[1] = "def";`  
`$a[2] = "ghi";`  
`$a[3] = "lmn";`  
`$a[4] = "opq";`
- E' possibile trattare le stringhe come un tipo di dato composto (stringa vista come un array di caratteri):  
`$stringa[2];` (seleziona il terzo carattere della stringa)





## Tipi di Dati

- Array (2)

**Esempi :**

- Gli array associativi sono indicizzati utilizzando stringhe:

```
$mesi["Gennaio"]=1;  
$mesi["Febbraio"]=2;  
$mesi["Marzo"]=3;  
$mesi["Aprile"]=4;
```

**E' possibile utilizzare una combinazione di indici numerici e stringa in un singolo array.**

- Visita di un array:

```
<? while (list($indice, $valore)=each ($mesi))  
    {  
        echo "contenuto array:[$indice]=$valore <br> \n";  
    } ?>
```

**Produce:**

```
contenuto array: [Gennaio]=1  
contenuto array: [Febbraio]=2  
contenuto array: [Marzo]=3  
contenuto array: [Aprile]=4
```

**PHP fornisce diverse funzioni per la manipolazione degli array.**





## Tipi di Dati

### • CLASSES and OBJECTS

Una classe e' una collezione di variabili e funzioni che utilizzano le variabili stesse.

Esempio:

```
<?
class visualizza {

    function start_visualizza () {
        echo ' Questo il messaggio visualizzato ';
    }
}
$oggetto=new visualizza;
$oggetto->start_visualizza();
?>
```

Si definisce la classe `visualizza`, che contiene la funzione `start_visualizza`. L'istruzione `new` inizializza l'oggetto `$oggetto` e richiama la funzione `start_visualizza` utilizzando l'operatore `->` su `$oggetto`.



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## Casting dei tipi

In PHP non e' necessario dichiarare esplicitamente il tipo delle variabili. E' possibile, comunque, specificarlo esplicitamente utilizzando un casting stile linguaggio C.

```
<? $variabile = (int) "12345abcde"; ?>
```

(con il cast esplicito si crea una variabile intera con valore 12345)

**Operatori di cast:**

- (int), (integer)
- (real), (double), (float)
- (string)
- (array)
- (object)



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## Operatori

### • Operatori aritmetici

- $\$a + \$b$  -> **addizione**
- $\$a - \$b$  -> **sottrazione**
- $\$a * \$b$  -> **moltiplicazione**
- $\$a / \$b$  -> **divisione**
- $\$a \% \$b$  -> **resto della divisione** (il resto della divisione viene riportato da solo, senza il risultato della divisione stessa)

### • Operatori di assegnazione : simbolo uguale (=) attribuisce ad una variabile un valore

- $\$a = 2;$
- **Errore frequente e' confondere l'assegnazione con il simbolo di eguaglianza**  
`if ($a == 2) {do_something; }`

### • Operatori booleani

- $\$a \& \$b$  (operatore "And" )
- $\$a \&\& \$b$  ( And , ma con una precedenza più alta)
- $\$a \|\| \$b$  (Or, ma con una precedenza più alta)
- $\$a \wedge \$b$  ( Xor - \$a oppure \$b ma non entrambi)
- $!\$a$  ( Not - vero se \$a non è vera)
- $\$a == \$b$  (uguaglianza, vero se \$a ha lo stesso valore di \$b)
- $\$a != \$b$  (opposto di quanto sopra)

### • Operatori matematici

- $\$a < \$b; a <= \$b; \$a > \$b; \$a >= \$b;$

### • Operatori incremento- decremento

- $++\$a$  (incrementa di uno \$a e la restituisce)
- $\$a++$  (restituisce \$a e la incrementa di uno)
- $--\$a$
- $\$a--$





## Strutture di controllo (1)

Possono essere utilizzate in due modalita' sintattiche:

1. parentesi graffe per racchiudere i blocchi di dichiarazione;
2. dichiarazioni di chiusura esplicite.

### • if

<pre>IF (espr) {     dichiarazione } ELSEIF (espr) {     dichiarazione } ELSE {     dichiarazione }</pre>	<pre>IF (espr):     dichiarazione ELSEIF (espr):     dichiarazione ELSE:     dichiarazione ENDIF;</pre>
-----------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------

### • switch

Utilizzato in sostituzione di una lunga sequenza di dichiarazioni if. Per ogni espressione case viene confrontata con l'espressione di switch.

(se uguale viene eseguito il codice che segue il case, se nessuna espressione dei case viene soddisfatta il controllo passa al case default.)

<pre>SWITCH (espr) {     case valore:         dichiarazione     BREAK;     DEFAULT:         dichiarazione     BREAK; }</pre>	<pre>switch (espr):     case valore:         dichiarazione     BREAK;     DEFAULT:         dichiarazione     BREAK; ENDSWITCH;</pre>
------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------





## Strutture di controllo (2)

- while**

Costrutto ciclico che esegue ripetutamente del codice finche' una particolare espressione e' true.

**while** (espressione) statement

```
while (espr) {
dichiarazione
}
```

```
while (espr) :
dichiarazione
endwhile;
```

Esempi:

```
$b = 1;
while ($b <= 10) {
print $i++;
}
```

```
$b = 1;
while ($b <= 10)
print $i++;
endwhile;
```

- for**

Costrutto piu' complesso del ciclo while; sono richieste tre espressioni.

```
for (esp_inizio; esp_condizionale;
esp_iterazione) {
dichiarazione
}
```

```
for (esp_inizio; esp_condizionale;
esp_iterazione) :
dichiarazione
endfor;
```

Esempi:

```
for ($a = 0 ; $a <=10 ;
$a++) {
print $a;
}
```

```
for ($a = 0 ; $a <=10 ; $a++)
:
print $a;
endfor;
```





# FUNZIONI

Le funzioni sono una delle parti più importanti di un linguaggio di programmazione.

Una funzione è una sequenza di dichiarazioni di codice, a cui viene assegnato un nome, che può accettare parametri e/o restituire valori.

Il PHP fornisce un gran numero di funzioni interne.

(Lista di tutte le funzioni disponibili: <http://www.php.net/manual/it/funcref.php>)

Il PHP consente all'utente la definizione di funzioni. Nell'assegnare il nome alla funzione, occorre essere sicuri che non entri in conflitto con quelli delle funzioni interne di PHP. La dichiarazione di una funzione avviene tramite la parola chiave `function`; la sintassi generale è la seguente:

```
function <nome-funzione> () {
    <corpo-della-funzione>
}
```

Esempio di definizione di una funzione per la generazione di un numero random compreso tra 10 e 20:

```
<?
    function random () {
        $a = rand(10,20);
        return $a;
    }
?>
```

Questa funzione che non necessita di nessuna variabile esterna, genera una variabile. Il comando `return` indica alla funzione di restituire il valore, in questo caso `$c`.





## FUNZIONI

### • Visibilita' delle variabili

Le variabili globali sono quelle definite all'esterno della funzione (nella parte principale dello script); viceversa, le variabili locali sono quelle definite all'interno della funzione.

- Se una variabile e' definita nel corpo principale di uno script PHP (non all'interno di una funzione o di una classe), e' visibile globalmente.
- L'unico modo per rendere le variabili di uno script disponibili per un altro script, e' di passarle con i cookies, con il metodo GET oppure con il metodo PUT.
- Per accedere ad una variabile globale dall'interno di una funzione, occorre utilizzare la parola **global**.

Esempio:

```
<?// $numero e' una variabile globale nello script principale
    $numero = 3;

    function prova() {

    global $numero;    // Con l'istruzione global si rende visibile la variabile
                       // all'interno della funzione

    echo $numero;
    }
?>
```

- L'array **\$GLOBALS** (array associativo), e' un meccanismo alternativo per accedere alle variabili con visibilita' globale.

Esempio:

```
function test() {
echo $GLOBALS ["var1"];
echo '<p>';
echo $GLOBALS ["var2"];
}

$var1="prima variabile";
$var2='seconda variabile';
test();
```

- Ogni funzione possiede una propria visibilita'. La variabile creata all'interno di una funzione ha una visibilita' locale, vale a dire che e' disponibile solamente all'interno della funzione.
  - Qualsiasi variazione ad una variabile interna ad una funzione, non modifica il valore di un'eventuale variabile globale con lo stesso nome.





# FUNZIONI

## • Passaggio di argomenti

### ○ Per valore:

Esempio:

```
function tripla ($scampo) {
    $scampo=$scampo*3;
    return $scampo;
}
```

```
$var=10;
$risultato=tripla($var);
echo $risultato;
```

### ○ Per riferimento:

Il carattere **&** che precede la variabile nella chiamata alla funzione, provoca il passaggio per riferimento dell'argomento. Il parametro nella funzione fa riferimento direttamente al valore della variabile; qualsiasi modifica nell'ambito della funzione e' visibile anche all'esterno della stessa. Esempio:

```
<?
function visualizza(&$stringa) {
    $stringa .= 'e questa è la seconda stringa.';
}
$str = 'Questa è la prima stringa, ';
visualizza($str);
echo $str;
?>
```

Questa è la prima stringa, e questa è la seconda stringa.





## FUNZIONI (1)

Documentazione : <http://www.php.net/manual/en/html/>

### Lista di alcune classi di funzioni

- **Apache- specific Functions**
- **Array Functions**
- **Calendar, Date and Time functions**
- **DBM Functions**
- **Directory functions**
- **filePro functions**
- **Filesystem functions**
- **FTP functions**
- **HTTP functions**
- **Image functions**
- **IMAP, POP3 and NNTP functions**



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## FUNZIONI (2)

..... classi di funzioni

- **Mail functions**
- **Mathematical Functions**
- **Mcrypt Encryption Functions**
- **Microsoft SQL Server functions**
- **mSQL functions**
- **MySQL functions**
- **Network Functions**
- **Unified ODBC functions**
- **Oracle 8 functions**
- **PDF functions**
- **PostgreSQL functions**
- **Satellite CORBA client extension**
- **SNMP functions**
- **String functions**
- **Sybase functions**
- **XML parser functions**



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## FUNZIONI per DATABASE

### Funzioni MySQL ([Documentazione on-line](#))

`mysql_affected_rows` – Get number of affected rows in previous MySQL operation  
`mysql_change_user` – Change logged in user of the active connection  
`mysql_close` – Close MySQL connection  
`mysql_connect` – Open a connection to a MySQL Server  
`mysql_create_db` – Create a MySQL database  
`mysql_data_seek` – Move internal result pointer  
`mysql_db_name` – Get result data  
`mysql_db_query` – Send a MySQL query  
`mysql_drop_db` – Drop (delete) a MySQL database  
`mysql_errno` – Returns the numerical value of the error message from previous MySQL operation  
`mysql_error` – Returns the text of the error message from previous MySQL operation  
`mysql_fetch_array` – Fetch a result row as an associative array, a numeric array, or both.  
`mysql_fetch_assoc` – Fetch a result row as an associative array  
`mysql_fetch_field` – Get column information from a result and return as an object  
`mysql_fetch_lengths` – Get the length of each output in a result  
`mysql_fetch_object` – Fetch a result row as an object  
`mysql_fetch_row` – Get a result row as an enumerated array  
`mysql_field_flags` – Get the flags associated with the specified field in a result  
`mysql_field_name` – Get the name of the specified field in a result  
`mysql_field_len` – Returns the length of the specified field  
`mysql_field_seek` – Set result pointer to a specified field offset  
`mysql_field_table` – Get name of the table the specified field is in  
`mysql_field_type` – Get the type of the specified field in a result  
`mysql_free_result` – Free result memory  
`mysql_insert_id` – Get the id generated from the previous INSERT operation  
`mysql_list_dbs` – List databases available on a MySQL server  
`mysql_list_fields` – List MySQL result fields  
`mysql_list_tables` – List tables in a MySQL database  
`mysql_num_fields` – Get number of fields in result  
`mysql_num_rows` – Get number of rows in result  
`mysql_pconnect` – Open a persistent connection to a MySQL Server  
`mysql_query` – Send a MySQL query  
`mysql_result` – Get result data  
`mysql_select_db` – Select a MySQL database  
`mysql_tablename` – Get table name of field





## Esempi di codice PHP

- Contatore di accessi in ogni pagina
- Visualizzazione di immagini
- Spedire una E-Mail
- Utilizzo delle date
  - Script per calcolo giorni..



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## Esempi di codice PHP

### • Contatore di accessi in ogni pagina

```

<?           //file ascii contenente il numero delle visite
if(!file_exists("counter.txt"))           // Se il file NON esiste
{
    $File=fopen("counter.txt","w")           // apre il file in sola scrittura
        or die ("errore scrittura file");
    $Tmp=sprintf("%d",1);                     // trasforma l'intero in una stringa
    fwrite($File,$Tmp);                       // scrive nel file
    fclose($File);                           // chiude il file
    $Conteggio=1;                             // inizializza variabile
}
else           // file gia' esistente
{
    $File=fopen("counter.txt","r")
        or die ("errore lettura file");
    $Conteggio=fread($File,10);             //legge dal file una stringa per 10 caratteri
    fclose($File);
}           // il numero degli accessi viene salvato nella variabile "Conteggio"
           //aggiorna il conteggio sul file

$File=fopen("counter.txt","w")
    or die ("errore scrittura file");
$Tmp=sprintf("%d",$Conteggio+1);
fwrite($File,$Tmp);
fclose($File);
print("visitatore numero : " . $Tmp);?>

```





## Esempi di codice PHP

- **Visualizzazione randomica di immagini**

I file devono essere nominati nel seguente modo :/immagini/1.gif  
/immagini/2.gif ...

```
$num=rand(1,20);  
$file="./immagini/".$num . ".gif" ;  
$imgfile= "<img src=\" " . $file . "\" . ">";  
echo$imgfile;
```

- **Visualizzazione di un'immagine diversa a seconda del giorno della settimana**

I file devono essere nominati nel seguente modo :/immagini/Sun.gif  
/immagini/Mon.gif ...

```
$img=date("D");  
$file="./immagini/".$img . ".gif" ;  
$imgfile= "<img src=\" " . $file . "\" . ">";  
echo$imgfile;
```



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## Esempi di codice PHP

### • Invio di un E- Mail

```
<FORM ACTION="send.php" METHOD="POST">
  Nome e Cognome:<INPUT TYPE="text" NAME="nome" SIZE="20" MAXLENGTH="30"><BR>
  Email:<INPUT TYPE="text" NAME="email" SIZE="40" MAXLENGTH="40"><BR>
  Titolo:<INPUT TYPE="text" NAME="soggetto" SIZE="20" MAXLENGTH="30"><BR>
  Testo della mail:<TEXTAREA TYPE="textarea" NAME="testo" ROWS="2" COLS="50"
MAXLENGTH="100"></TEXTAREA>
  <BR>
<INPUT TYPE="submit" VALUE="INVIA">
</FORM>
```

Visualizza FORM :

Nome e Cognome:

Email:

Titolo:

Testo della mail:

Codice dello script `send.php` :

```
    if (isset($email)):
        mail("$email","$soggetto","Nome: ".$nome."\nTitolo:
".$soggetto."\n\n".$testo);
    endif;
```



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## Esempi di codice PHP

### • Utilizzo delle date (1)

- **Visualizza data corrente :**

```
<?echo "data : ". date("d/m/Y");echo'<br>'; echo "data : ". date("j/n/Y");?>
```

**Produce:**

data : 01/10/2004

data : 1/10/2004

- **Visualizza giorno della settimana :**

```
<?$giorni = array("Domenica", "Lunedì", "Martedì","Mercoledì","Giovedì",
"Venerdì", "Sabato"); echo " Oggi e' : " . $giorni[date("w")];?>
```

**Produce :**

Oggi e' : Venerdì

- **Utilizzo funzione `mktime()` :**

**Fornisce il timestamp di una determinata data.**

**Timestamp e' un numero intero che corrisponde al numero di secondi trascorsi dalla cosiddetta Unix epoch vale a dire 0:00:00 1 gennaio 1970.**

**`mktime ( hour, minute, second, month, day, year,0 )`**

**Calcolo differenze tra due date :**

```
<?$data1 = mktime(0,0,0, 8, 1, 1962, 0); $ddata1= date("j/n/Y",$data1);
$data2 = mktime(0, 0, 0, 4, 26, 2001, 0); $ddata2= date("j/n/Y",$data2);
echo'<p>';
```

```
echo'Tra il '$ddata1 .' e il ' . $ddata2 .' ci sono : ';
```

```
echo intval (($data2 - $data1) / (60*60*24)); echo ' giorni!! ';>
```

**La differenza tra i due timestamp espressa in secondi, viene convertita in giorni dividendo per 24 ore al giorno, per 60 minuti l'ora, per 60 secondi al minuto. L'ultimo parametro di `mktime()` e' impostato a zero per ignorare l'ora legale.**

**Produce :**

Tra il 1/8/1962 e il 26/4/2001 ci sono : 14148 giorni!!



## Esempi di codice PHP

### • Utilizzo delle date (2)

calcola.php - Script per il calcolo dei giorni a partire da una data (1)

```
<? if(isset($calcola)):
    $data1 = mktime(0,0,0,$mm,$gg,$yy);$ddata1= date("j/n/Y",$data1);
    $today=date("j/n/Y");
    $temp = explode("/", $today);
    $data2 = mktime(0,0,0,$temp[0],$temp[1],$temp[2]);
    echo' Tra la data di nascita : <b>' . $ddata1 . '</b><br> e la data odierna : <b>';
    echo date("j/n/Y") . ' </b><br> ci sono <b> ';
    echo intval (($data2 - $data1) / (60*60*24)); echo '</b> giorni !! ';?>
    <FORM ACTION="calcola.php" METHOD=POST>
    <INPUT TYPE=SUBMIT VALUE="Ritorna">
<? else: ?>
    </center> <br>
    <FORM ACTION="calcola.php" METHOD=POST>
    <?$today=getdate();?><p><b>
    Inserisci la tua data di nascita (posteriore al 1936)<p>
    <table border=0>
        <tr><td align=right><b>Giorno</b><
        td><SELECT NAME=gg size=1 align=left>
        <OPTION>01<OPTION>02<OPTION>03<OPTION>04<
        OPTION>05<OPTION>06<OPTION>07<OPTION>08<
        OPTION>09<OPTION>10<OPTION>11<OPTION>12<OPTION>13
        <OPTION>14<OPTION>15<OPTION>16<OPTION>17<
        OPTION>18<OPTION>19<OPTION>20<OPTION>21<OPTION>22
        <OPTION>23<OPTION>24<OPTION>25<OPTION>26<
        OPTION>27<OPTION>28<OPTION>29<OPTION>30<OPTION>31
        </SELECT> <b>Mese</b> <select NAME=mm size=1 align=left>
        <OPTION>01<OPTION>02<OPTION>03<OPTION>04
        <OPTION>05<OPTION>06<OPTION>07<OPTION>08<OPTION>09<
        OPTION>10<OPTION>11<OPTION>12
        </SELECT> <b>Anno</b>
        <INPUT NAME=yy size=4 maxlength=4 align=left></tr>
    </table><p>
    <INPUT TYPE=SUBMIT NAME="calcola" VALUE="Calcola"> <input type=RESET VALUE="RESET">
    <p>
    </FORM>
<?endif;?>
```





## Esempi di codice PHP

- **Utilizzo delle date**

Script per il calcolo dei giorni a partire da una data (2)

Mandando in esecuzione lo script `calcola.php` si ha la seguente FORM :

Inserisci la tua data di nascita (posteriore al 1936)

Giorno  Mese  Anno

Calcola

RESET

Il risultato del calcolo produce :

Tra la data di nascita : **1/1/1949**

e la data odierna : **26/4/2001**

ci sono **19757** giorni !!

Ritorna



© (2000 - 2004) Daniela Beatrici - CNR/IIA -  
POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





# PHP e Relational DataBase Management System

Php consente di interfacciarsi con numerosi database tra i piu' importanti quali : PostgreSQL, MySql, Oracle, Adabas, filePro, ODBC (permette di interfacciarsi con database tipo Access), Sybase, Informix, Ingres, ecc.

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

- apertura/chiusura connessione;
- invio interrogazione SQL;
- lettura risultati.



Per aprire/chiedere una connessione in MySql sono disponibili le funzioni:

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

- Invia una query al database: `mysql_db_query(database, query);`
- Memorizza all'interno della variabile \$risultato tutte le occorrenze della tabella istituti:  
`$risultato=mysql_db_query("localhost", "Select * from istituti");`
- Memorizza numero di righe presenti nel risultato :  
`$nrighe=mysql_num_rows($risultato);`
- Memorizza la prima riga del risultato: `$riga=mysql_fetch_row($result);`
- Visualizza contenuto della prima riga:  
`print($riga[1]. "/" . $riga[2]. "/" . $riga[3]. "/" . $riga[4]. "/");`
- Memorizza il numero di campi presenti nel risultato della query :  
`$ncampi=mysql_num_fields($risultato);`



## PHP e MySQL

Si riporta un esempio per un'operazione di data entry in una tabella MySQL. Per facilitare la lettura del codice, sono stati creati 2 script separati.

- [dataentry.php](#) (script per la visualizzazione della FORM)
- [form](#) (FORM prodotta dallo script dataentry.php)
- [salva.php](#) (script per la generazione del comando SQL INSERT ed inserimento dati nella tabella)
- [comando INSERT](#) (comando SQL generato dallo script salva.php)

### Tabella organo

```
CREATE TABLE organo (
  num SMALLINT NOT NULL,
  sigla VARCHAR(60) NOT NULL,
  nome VARCHAR(120) NOT NULL,
  tipoorg VARCHAR(10),
  indirizzo VARCHAR(120),
  cap CHAR(5),
  citta VARCHAR(40),
  provincia CHAR(2),
  url VARCHAR(80),
  note VARCHAR(40),
  modifica DATE );
CREATE UNIQUE INDEX xnum ON organo (num);
CREATE UNIQUE INDEX xsigla ON organo (sigla);
```

### Tabella tiporg

```
CREATE TABLE tiporg (
  num SMALLINT NOT NULL,
  tipo VARCHAR(25) NOT NULL
);
CREATE UNIQUE INDEX xtiporg ON tiporg (tipo);
```





## PHP e MySQL

### Codice dello script dataentry.php

```

<html>
<head>
    <title>Data entry della tabella ORGANO</title>
</head>
<body>
<font size=4>Data Entry :<i><b> " ORGANO "
<br>
<FORM ACTION="salva.php" METHOD=POST>
<INPUT TYPE=HIDDEN NAME="tabella" VALUE="<?echo"$table"?">">
<center>
<? $today=getdate(); $mday=$today[mday]; $month=$today[month]; $year=$today[year];
$data="$mday - $month - $year"; echo$data;
switch ($month) {
    case "January"; $mon="01"; break;
    case "February"; $mon="02"; break;
    case "March"; $mon="03"; break;
    case "April"; $mon="04"; break;
    case "May"; $mon="05"; break;
    case "June"; $mon="06"; break;
    case "July"; $mon="07"; break;
    case "August"; $mon="08"; break;
    case "September"; $mon="09"; break;
    case "October"; $mon="10"; break;
    case "November"; $mon="11"; break;
    case "December"; $mon="12"; break;
}
$modifica=$year."-".$mon."-".$mday; ?>
<table border=0> <tr><td colspan=2><center></tr><tr>
<td align=right><b> Sigla <br><font size=1> in maiuscolo
    <td><INPUT NAME=sigla size=60 maxlength=60 align=left></tr>
<tr><td align=right><b> Organismo
    <td><TEXTAREA NAME=nome COLS=60 ROWS=2 align=left wrap="virtual"></TEXTAREA>
</tr><tr><td align=left> Tipo Ente<td><select NAME=tipoorg size=1 align=left>
<OPTION selected>
<? mysql_connect("localhost");
$query="select tipo from tiporg order by tipo ";
$result= mysql("stspa",$query);
    if($result == -1):
    echo "Error from table TIPORG !! " ;
    exit;
    endif;
$rows=mysql_numrows($result);
$rows=0;
while ($rows < $nrows):          /* estraee risultato della query da array 1 colonna
per n rows righe */
    $a=mysql_fetch_row($result); ?>

```

```

<OPTION>
<?$tipoorg=$a[0];echo$a[0];      /* $tipoorg = organismo passata dalla select
*/
    $rows++;
endwhile;?>
</SELECT>
</tr><tr><td align=right><b> Indirizzo
    <td><INPUT NAME=indirizzo size=75 maxlength=75 align=left> </tr>
<tr><td align=right><b> cap <td> <INPUT NAME=cap size=5 maxlength=5 align=left>
<b> Citta' <INPUT NAME=citta size=60 maxlength=60 align=left>
<tr><td align=right><b>Provincia<td> <INPUT NAME=provincia size=2 maxlength=2>
    <b>Nazione<</b>
<select NAME=nazione size=1 align=left>
<OPTION selected>I<OPTION>B<OPTION>CH<OPTION>F<OPTION>P
<OPTION>USA
</SELECT>    <b> Tel.</b><INPUT NAME=tel size=15 maxlength=15 align=left>
    </tr><tr><td align=right><b> URL <td><INPUT NAME=url size=80 maxlength=80
align=left>
</tr><tr><td align=right><b>Note <td><INPUT NAME=note size=40 maxlength=40
align=left>
</tr></table>
<INPUT TYPE=SUBMIT VALUE="SALVA"> <input type=RESET VALUE="RESET">
<INPUT TYPE=HIDDEN NAME="modifica" VALUE =<?echo"$modifica"?>>
</FORM>
</body>
</html>

```





## PHP e MySQL

FORM generata dallo script dataentry.php

Data Entry : " **ORGANO** "

*23 - April - 2001*

Sigla   
in maiuscolo

Organismo

Tipo Ente

Indirizzo

cap  Citta'

Provincia  Tel.

URL

Note

### Dettaglio della FORM sul campo Tipo Ente

ALTRO
AMBASCIATA
CNR
DITTA
ENEA
FORZE ARMA
GOVERNO/PU
ICRAM
INGV
INTERNAZIO
IUN
MUSEO
NYSMAC
OGS
PRIVATO
SCUOLA
UNIV

Tipo Ente



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## PHP e MySQL - Codice dello script salva.php

```

<html>
<head><title>Memorizza record nella tabella ORGANO</title></head>
<body>
<?mysql_connect("localhost");$database="stspa";$table='organo';
    $queryresult=mysql_db_query($database,"select * from $table")
        or die ("Query fallita - Errore: '$php_errormsg' "); /*    +++ ERROR
CONTROL */
    $nfields=mysql_num_fields($queryresult);
    $sqlcmd = "INSERT INTO $table (";
    $field=0;
    while ($field < $nfields):
        $sqlcmd = $sqlcmd . mysql_field_name($queryresult,$field) . ",";
        $field++;
    endwhile;
    $len = strlen($sqlcmd); $len = $len-1; /* prepara query per INSERT */
    $sqlcmd = substr($sqlcmd,0,$len); $sqlcmd=$sqlcmd .") VALUES (";
    $database="stspa"; $queryresult="";
    $queryresult=mysql_db_query($database,"select * from $table order by num
desc");
    $nfields=mysql_num_fields($queryresult);
        /* determina ultimo numero libero da incrementare per insert
*/
    $result=mysql_db_query($database,"Select num from $table order by num desc");
    $nrighe=mysql_num_rows($result);$a=mysql_fetch_row($result); $nrec=$a[0];
    $tdate=$yyyy."-".$mon."-".$dd;
    $sqlcmd.= $nrec+1 . ",". "' . $sigla . "' . ",". "' . $nome . "' . ",";
    $sqlcmd.= "' . $tipoorg . "' . ",". "' . $indirizzo . "' . ",";
    $sqlcmd.= "' . $cap . "' . ",". "' . $citta . "' . ",". "' . $provincia . "' .
    ",";
    $sqlcmd.= "' . $nazione . "' . ",". "' . $tel ."' . ",". "' . $url . "' . ",";
    $sqlcmd.= "' . $note . "' . ",". "' . $modifica . "' . ")";
    echo $sqlcmd;echo '<p>';
    mysql_db_query($database,$sqlcmd);
    $err=mysql_error();echo '<<p>';echo$err;
    if ($err != ""): echo ' Nessun record inserito ERRORE !!!!!';
    else:??> Record ok !!!!
    <?endif; ??>
</body></html>

```

-----  
**+++ ERROR CONTROL** : mettendo il simbolo @ si possono intercettare gli errori prodotti da un'espressione; il messaggio d'errore viene conservato nella variabile globale \$php\_errormsg. (Abilitare la variabile track\_errors On nel file di configurazione php.ini o php.ini)





## PHP e MySQL

Lo script `salva.php` genera il comando SQL INSERT a partire dai dati inseriti nella FORM.

```
.....
.....
$sqlcmd.= "'" . $nazione . "'". "," . "'" . $tel . "'" . "," . "'" . $url . "'" . "," .
$sqlcmd.= "'" . $note . "'" . "," . "'" . $modifica . "'" . ")";
    echo $sqlcmd;echo '<p>';
    mysql_db_query($database,$sqlcmd);
.....
.....
```

```
INSERT INTO organo (num,sigla,nome,tipooorg,indirizzo,cap,citta,provincia,nazione,tel,url,note,modifica)
VALUES (327,'IAS','Istituto di Astrofisica Spaziale','CNR','Area della Ricerca di Roma Tor Vergata Via
Fosso del Cavaliere, 100','00133','Roma','RM','I','06-49930000','www.ias.rm.cnr.it','-','2001-04-23')
```



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)





## PHP e MySQL

Si riporta, qui di seguito, un esempio di pubblicazione su Web del contenuto di una tabella MySQL. L'uso di una singola tabella dovrebbe agevolare la comprensione del codice utilizzato.

La tabella in questione, potrebbe essere una tabella tipo, utilizzata per schedare parte delle informazioni relative alle unita' di ricerca di un progetto scientifico.

- 1) Il file, creato con un editor qualsiasi, contiene le direttive SQL per la creazione di una tabella. Il file si potrebbe chiamare `uo.create` nella directory `/usr/local/mysql/bin`

```
# MYSQL
# Creazione di una tabella
# Host: localhost Database: progetto
#-----
# Table structure for table 'uo' 17 fields
#
USE progetto;

CREATE TABLE uo (
  codice CHAR(3) NOT NULL,
  cognome CHAR(15) NOT NULL,
  nome CHAR(25) NOT NULL,
  email CHAR(60),
  anno YEAR(4) NOT NULL,
  sigla CHAR(10),
  inst CHAR(85),
  org CHAR(4),
  indirizzo VARCHAR(80),
  cap CHAR(5),
  citta CHAR(20),
  url CHAR(50),
  tel CHAR(15),
  fax CHAR(15),
  fondi INT,
  titolo VARCHAR(220) NOT NULL,
  note VARCHAR(100)
);

CREATE UNIQUE INDEX icodice ON uo (codice);
CREATE UNIQUE INDEX leader (cognome,nome);
```

- 2) Creazione della tabella utilizzando il comando `mysql` :

```
/usr/local/mysql/bin/mysql -vvv < uo.create > uo.out
```

MySQL salva ogni tabella creata nel database, come un insieme di tre files; ad esempio per la tabella `uo` si hanno:

- `uo.ISD` (contiene i dati effettivi)
- `uo.ISM` (contiene informazioni sui caratteri ed altri dati di indicizzazione)
- `uo.frm` (contiene la struttura della tabella).

- 3) Script per la visualizzazione del contenuto della tabella MySQL. L'esempio e' ridotto al minimo

essenziale per agevolare una facile lettura ed interpretazione del codice.

I dati, visualizzati come risultato, sono un sottoinsieme del contenuto previsto inizialmente nella tabella `uo`.

```
<html>
<head>
<title>Progetto di Ricerca - Lista delle Unità Operative< /title>
<? /* procedura per la visualizzazione unità operative - Settembre 2000 D.B. */ ?>
</head>
<body bgcolor="FFFFFF" text="0000FF" link="0000FF" vlink="000080">
<font face="arial, sans serif, helvetica" size=2>
<?mysql_connect('localhost');
$dbase='progetto';
$table='uo';
$query='select * from $table';
    $queryresult=mysql($dbase,$query);
    $nrows=mysql_numrows($queryresult);
    $rows=0;?>
    <center>
<font face="arial" size=3><i> Trovate: <b> <?echo"$nrows"?> Unità operative <br>
<table border=2 ><tr><th>Titolo <th>Responsabile<th>Istituto <th>Org.</tr>
<? while($rows<$nrows):
    $nome =mysql_result($queryresult,$rows,"nome");
    $cognome =mysql_result($queryresult,$rows,"cognome");
    $sigla =mysql_result($queryresult,$rows,"sigla");
    $inst =mysql_result($queryresult,$rows,"inst");
    $org =mysql_result($queryresult,$rows,"org");
    $url =mysql_result($queryresult,$rows,"url");
    $titolo =mysql_result($queryresult,$rows,"titolo");
    if ($url== " "):
        $www=$inst;
    else:
        $www= ("< a href=\"http://\" . "$url" . "\">\" . "$inst" . "</a>");
    endif;?>
<tr><td><p align=justify><font face="arial" size=2><?echo"$titolo"?>
<td><font face="arial" size=2><?echo"$nome"?> <?echo"$cognome"?>
<td><font face="arial" size=2> <?echo"$www"?> (<?echo"$sigla"?>)
        <td><font face="arial" size=1> <?echo"$org"?></tr>
<? $rows++;
endwhile;?> </table>
</body>
</html>
```

**Risultato visualizzato al client**

Progetto di Ricerca - Lista delle Unità Operative - Netscape

File Edit View Go Communicator Help

Bookmarks Go to: <http://localhost/php/uo.php3> What's Related

Back Forward Reload Home Search Netscape Print Security Stop

Trovate: **98 Unità operative**

Titolo	Responsabile	Istituto	Org.
Study of atmospheric processes leading to the formation and evolution of Arctic aerosols	Ivo ALLEGRINI	<a href="#">Istituto Inquinamento Atmosferico (IIA)</a>	CNR
Size distribution of metals in arctic troposphere by means of a combination of multistage impactor sampling and neutron activation/ICP-MS analysis	Mario GALLORINI	<a href="#">Centro di Radiochimica e Analisi per Attivazione (CRAA)</a>	CNR
Parameterisation of turbulent heat and momentum fluxes profiles by acoustic remote sensing at the ice/air interface	Stefania ARGENTINI	<a href="#">Istituto di Fisica dell'Atmosfera (IFA)</a>	CNR
Measurements of radiation balance and turbulence at the surface during intense melting periods	Teodoro GEORGIADIS	<a href="#">Istituto per lo Studio dei Fenomeni Chimici della Bassa ed Alta Atmosfera (FISBAT)</a>	CNR
Stable boundary layer structure and its interaction with horizontal inhomogeneities	Umberto GIOSTRA	<a href="#">Istituto per lo Studio dell'Inquinamento atmosferico e dell'Agrometeorologia (ISIAA)</a>	CNR
Co-ordinated studies on polar stratospheric clouds in the Arctic by a quasi-lagrangian balloon experiment and Ny-Alesund ground based Lidar (COSPOC-ARCIC)	Francesco CAIRO	<a href="#">Istituto di Fisica dell'Atmosfera (IFA)</a>	CNR

Document: Done



© (2000 - 2004) Daniela Beatrice - CNR/IIA - POLARNET  
 Daniela.Beatrice@polarnet.cnr.it





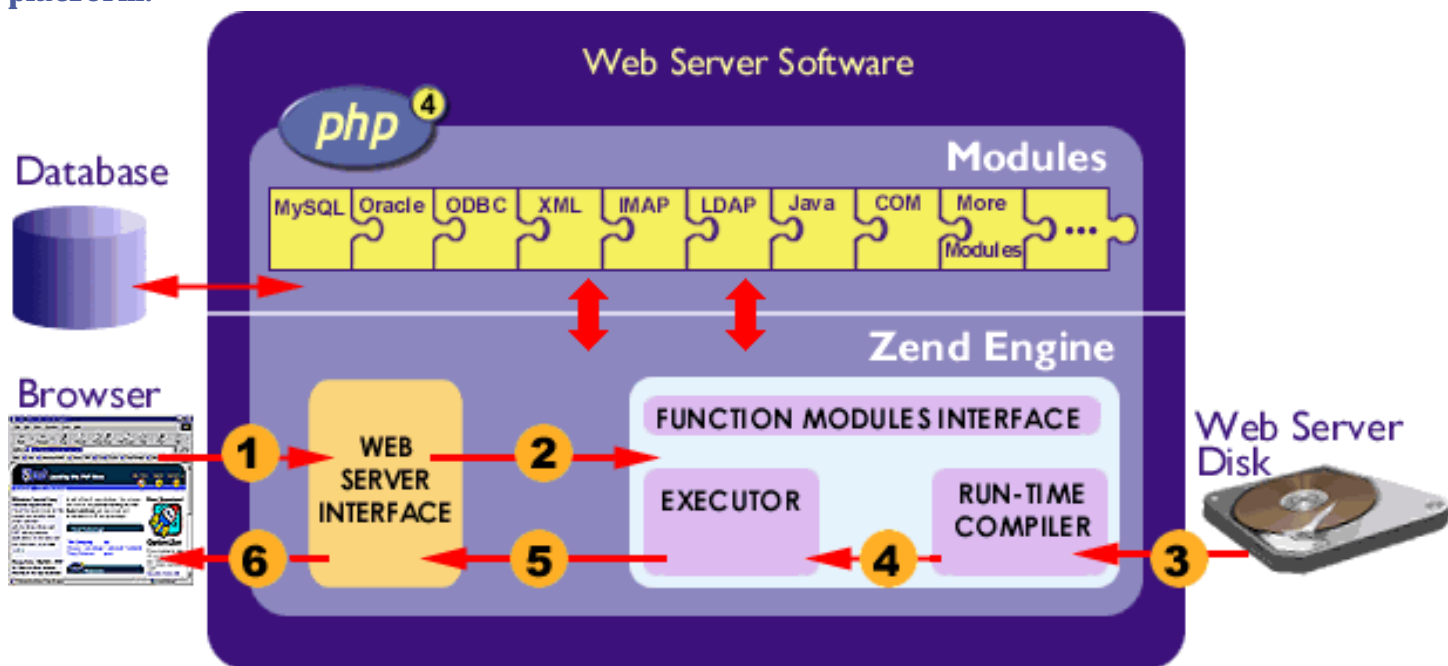
## PHP4 - ZEND Technology

Zend e' il cuore centrale, il nuovo engine della versione 4.

Il nome deriva dall'unione dei suoi autori, Andi Gutmans e Zeev Suraski, e racchiude al suo interno diversi prodotti e tecnologie quali, primo fra tutti, Zend Engine ed ancora Optimizer e Zend Cache, atti a migliorare le prestazioni e le capacita' del PHP ma anche a dotarlo di qualita' commerciali.

Caratteristiche fondamentali

- piu' veloce in esecuzione,
- mantiene la compatibilita' con le precedenti versioni,
- modus operandi diverso. (PHP presenta un compilatore compile-then-execute. Avviene una sorta di compilazione al volo del sorgente internamente al parser)
- Zend Engine: software di proprieta' della Zend Technologies rilasciato come opensource sotto licenza Qpl ( Q Public License della Troll Tech.). Prodotto di natura cross-platform.





## Bibliografia - Sitografia

- Rasmus Lendorf : "PHP - Pocket Reference"; O'Reilly Associates, HOPS Libri - 2000;
- HTML.it home page : " Guide a PHP" - <http://www.html.it/php/index.html>
- [Inter.Net n. 49](#)
- LatoServer.it : "Corso PHP" - <http://www.LatoServer.it/php/corso/>
- The Open Source Initiative (OSI) : [Open Source](#)
- PHP home page: <http://www.php.net>
- [PHP Italia](#)
- PHP Reference Manual : <http://www.php.net/docs.php/>
- ZEND Technologies : <http://www.zend.com/>



© (2000 - 2004) Daniela Beatrici - CNR/IIA - POLARNET  
[Daniela.Beatrici@polarnet.cnr.it](mailto:Daniela.Beatrici@polarnet.cnr.it)

