

Connessione ad un server MySQL

```
resource mysql_connect ([string server [,  
string username [, string password [, bool new_link [,  
int client_flags]]]])
```

server nome o indirizzo del server

username nome utente MySQL

password password di username

new_link se 1, apre una nuova connessione (PHP 4.2+)

client_flags specifica il tipo di connessione

Es. **MYSQL_CLIENT_COMPRESS** indica l'uso
della compressione dati sulla
connessione

Restituisce un identificatore di connessione

Una `resource` è una variabile speciale che contiene un riferimento ad una risorsa esterna.

```
bool mysql_select_db([string database_name  
                      [,resource link_identifier]])
```

database_name nome del database cui ci si
 vuole conettere

link_identifier identificatore della connessione
 entro la quale ci si connette al
 database *database_name*

Restituisce un valore booleano che indica se la connessione è attiva o meno (cioè se l'apertura del database è andata a buon fine).

Se non si specifica l'identificatore di connessione ne apre una nuova come se si chiamasse `mysql_connect` senza argomenti

```
resource mysql_query ( string query [, resource link_identifier ] )
```

query stringa contenente la query

link_identifier identificatore della connessione
entro la quale ci si connette al
database *database_name*

Restituisce una struttura dati (accessibile solo attraverso opportune funzioni) contenente il risultato della query.

Se non si specifica l'identificatore di connessione ne apre una nuova come se si chiamasse `mysql_connect` senza argomenti

array **mysql_fetch_row** (resource *result*)

result riferimento al risultato di una query

Restituisce una riga appartenente al risultato di una query in forma di array e incrementa di uno il corrispondente puntatore (a *result*).

array **mysql_fetch_assoc** (resource *result*)

Restituisce una riga appartenente al risultato di una query in forma di array associativo e incrementa di uno il corrispondente puntatore. Le chiavi dell'array corrispondono ai nomi degli attributi.

Generalizzate da:

array **mysql_fetch_array** (resource *result* [, int *result_type*])

Restituisce una riga appartenente al risultato in forma di array di tipo specificato da *result_type*, che può assumere valori

MYSQL_ASSOC, **MYSQL_NUM**, **MYSQL_BOTH** (default)

Con **MYSQL_BOTH** si può fare riferimento all'array sia tramite indici che tramite chiavi

```
string mysql_result ( resource result , int row [,  
mixed field ] )
```

result riferimento al risultato di una query

row indice di riga del risultato della query

field indice di colonna (o nome attributo)

Restituisce un elemento (valore) del risultato di una query

field può essere sia un indice (molto più veloce) che una chiave

NB Gli indici iniziano da 0

Usare questa funzione è molto più lento che usare le funzioni
che restituiscono una riga intera alla volta.

object **mysql_fetch_field**(resource *result* [, int *field_offset*])

result riferimento al risultato di una query

field_offset indice di colonna

Restituisce un oggetto contenente informazioni sull'attributo in posizione *field_offset* nel risultato di una query

Informazioni restituite:

- *name* nome attributo
- *table* nome delle tabella cui appartiene l'attributo
- *max_length* massima lunghezza dell'attributo
- *not_null* vale 1 se l'attributo non può assumere valore NULL
- *primary_key* vale 1 se l'attributo è una chiave primaria
- *unique_key* vale 1 se l'attributo è una chiave unica
- *multiple_key* vale 1 se l'attributo è una chiave non unica
- *numeric* vale 1 se l'attributo è numerico
- *blob* vale 1 se l'attributo è un BLOB
- *type* tipo di attributo
- *unsigned* vale 1 se l'attributo è unsigned
- *zerofill* vale 1 se l'attributo è completato con zeri (lung. costante)

NB Per riferirsi a un elemento di un oggetto si usa la notazione `$myobject->element`

```
object mysql_fetch_object ( resource result [, string class_name [, array params ]] )
```

result riferimento al risultato di una query
class_name classe a cui appartiene l'oggetto restituito
params parametri del costruttore della classe

Restituisce una riga del risultato di una query in forma di oggetto di tipo *class_name* costruito con parametri *params*

Esempio:

```
class foo {  
    public $name;}  
mysql_connect("hostname", "user", "password");  
mysql_select_db("mydb");  
$result = mysql_query("select name from mytable limit 1");  
  
$obj = mysql_fetch_object($result, 'foo');  
  
var_dump($obj);
```

```
array mysql_fetch_lengths ( resource $result )
```

result riferimento al risultato di una query

Restituisce un array contenente la lunghezza in caratteri di ciascun elemento della riga estratta.

Esempio:

```
$result = mysql_query("SELECT id,email FROM people  
                        WHERE id = '42'");
```

```
if (!$result) {  
    echo 'Could not run query: ' . mysql_error();  
    exit;  
}
```

```
$row      = mysql_fetch_assoc($result);  
$lengths  = mysql_fetch_lengths($result);
```

```
print_r($row);  
print_r($lengths);
```

```
void var_dump ( mixed $expression [, mixed $... ] )
```

Questa funzione visualizza delle informazioni strutturate riguardanti una espressione, tra cui il suo tipo e il suo valore. Gli array e gli oggetti sono attraversati in maniera ricorsiva e i valori indentati per evidenziare la struttura.

expression La variabile che si desidera visualizzare.

Esempio

```
<?php
$a = array(1, 2, array("a", "b", "c"));
var_dump($a);
?>
```

```
array(3) {
  [0]=>
  int(1)
  [1]=>
  int(2)
  [2]=> array(3) {
    [0]=>
    string(1) "a"
    [1]=>
    string(1) "b"
    [2]=>
    string(1) "c"
  }
}
```

```
bool print_r (mixed $expression [, bool $return ] )
```

Stampa informazioni sul contenuto di una variabile in un formato leggibile. Se la variabile contiene una stringa, un intero o un numero decimale, ne visualizza il valore. Se la variabile contiene un array le chiavi ed i relativi elementi vengono evidenziati. Una notazione simile viene utilizzata per gli oggetti.

NB **print_r()** posiziona il puntatore dell'array alla fine.
Utilizzare **reset()** per riportarsi all'inizio.

Per catturare l'output di **print_r()**, utilizzare il parametro `return`.
Se `TRUE`, **print_r()** restituirà l'output anziché visualizzarlo.

Esempio

```
<pre>
```

```
<?php
```

```
    $a = array ('a' => 'apple', 'b' => 'banana', 'c' => array  
('x','y','z'));
```

```
    print_r ($a);
```

```
?>
```

```
</pre>
```

```
<pre>
```

```
Array
```

```
( [a] => apple
```

```
  [b] => banana
```

```
  [c] => Array
```

```
    ( [0] => x
```

```
      [1] => y
```

```
      [2] => z
```

```
    )
```

```
)
```

```
</pre>
```

```
bool mysql_close ([ resource link_identifier ] )
```

link_identifier identificatore della connessione

Chiude la connessione indicata da *link_identifier* e restituisce un valore booleano: 1 se l'operazione si è chiusa con successo, 0 altrimenti.

@ sopprime il messaggio di errore generato da ciò che segue. Se seguito da `or` esegue in alternativa ciò che segue `or`.

```
@expr1 or expr2
```

Se `expr1` è una chiamata di funzione equivale a

```
if <expr1 does not generate an error> expr1  
else expr2
```

Es.

```
@mysql_select_db($database)  
or die( "Unable to select database" );
```

Se `expr1` ed `expr2` sono valori da assegnare a una variabile restituisce `expr1` **se** `expr1` non è vuoto, altrimenti restituisce `expr2`

Es.

```
$bar= @$foo or $bar= "undefined";
```

```
<html> <body>
<?php
$user="root"; $password=""; $host="127.0.0.1"; $database="prova";

mysql_connect($host,$user,$password);
@mysql_select_db($database) or die( "Unable to select database");

$query="SELECT * from utenti";
$result=mysql_query($query);

mysql_close();

$num=mysql_numrows($result);
echo "<b><center>Database Output</center></b><br><br>";

$i=0;
while ($i < $num) {
    $nome=mysql_result($result,$i,"nome");
    $cognome=mysql_result($result,$i,"cognome");
    echo "Nome: $nome<br>Cognome: $cognome<br><hr><br>";
    $i++;}

?> </body> </html>
```

```
<html> <body>
<?php
$user="root"; $password=""; $host="127.0.0.1"; $database="prova";
mysql_connect($host,$user,$password);
@mysql_select_db($database) or die("Unable to select database");
$query="SELECT * from utenti";
$res=mysql_query($query);
mysql_close();

$row = mysql_fetch_assoc($res);
print '<table><tr>';
foreach($row as $name => $value) {
print "<th>$name</th>";}
print '</tr>';

while($row) {
print '<tr>';
foreach($row as $value) {
print "<td>$value</td>";}
print '</tr>';
$row = mysql_fetch_assoc($res);}
print '</table>';
?> </body> </html>
```

//upload di un file

```
<html>
```

```
<body>
```

```
<!-- The data encoding type, enctype, MUST be specified as below -->
```

```
<form enctype="multipart/form-data" action="finalize_upload.php"  
method="POST">
```

```
    <!-- MAX_FILE_SIZE must precede the file input field -->
```

```
    <input type="hidden" name="MAX_FILE_SIZE" value="900000" />
```

```
    <!-- Name of input element determines name in $_FILES array -->
```

```
Fai l'upload del file: <input name="userfile" type="file" /> <br>
```

```
Nella directory: <br>
```

```
<input type="text" name="path" /> <br>
```

```
<input type="submit" value="Invia il File" />
```

```
</form>
```

```
</body>
```

```
</html>
```

```
<?php
// In PHP versions earlier than 4.1.0, $HTTP_POST_FILES should be used
// instead of $_FILES.

// controlla che non si voglia fare l'upload nella root directory
if ($_POST['path']!=" && $_POST['path']!=".)
{
    $uploaddir = "/var/www/studenti/" . $_POST['path'] . "/";
    $uploadfile = $uploaddir . basename($_FILES['userfile']['name']);

    echo '<pre>';

    if (move_uploaded_file($_FILES['userfile']['tmp_name'], $uploadfile))
        echo "Trasferimento avvenuto con successo.\n";
    else
        echo "Errore\n";
}
else
echo "Non puoi trasferire il file nella root directory del server.<br>";
}

echo "</pre>";

?>
```