

Programowanie Aplikacji Internetowych



Funkcje MYSQLIB

Pierwszy parametr umożliwia ustawienie adresu serwera. Konfiguracja możliwa jest poprzez podanie stringa **serwera:port** lub w postaci ścieżki do lokalnego gniazda **:/ścieżka/socket**. Kolejno argumentami są nazwa użytkownika DB oraz jego hasło. W trybie bezpiecznym (mysql.safe_mode) można te parametry pominąć. Będą wówczas brane te, które zdefiniowane zostały w pliku php.ini.

```
resource mysql_connect ([ string $serwer = ini_get("mysql.default_host") [, string  
$nazwa_uzytkownika = ini_get("mysql.default_user") [, string $haslo =  
ini_get("mysql.default_password") [, bool $nowe_polaczenie = false [, int $flagi_klienta =  
0 ]]]] )
```

Funkcje MYSQLIB

Gdy wywołujemy w ramach tego samego skryptu *mysql_connect* z takimi samymi parametrami domyślnie nie jest tworzone nowe połączenie do DB. Zwrócona zostaje referencja do identyfikatora obecnie używanego – otwartego połączenia. Wstawiając parametr **true** można wymusić utworzenie nowego połączenia.

```
resource mysql_connect ([ string $serwer = ini_get("mysql.default_host") [, string  
$nazwa_uzytkownika = ini_get("mysql.default_user") [, string $haslo =  
ini_get("mysql.default_password") [, bool $nowe_połączenie = false [, int $flagi_klienta =  
0 ]]] ] )
```

Funkcje MYSQLIB

Przykład wykorzystania funkcji `mysql_connect`.

Funkcja zwraca identyfikator połączenia w przypadku sukcesu.
W razie porażki zwraca *false*.

```
$link = mysql_connect('localhost', 'uzytkownik_mysql', 'haslo_mysql');  
if (!$link) {  
    die('Nie można się połączyć: ' . mysql_error());  
}  
echo 'Połączenie nawiązane';  
mysql_close($link);
```

Funkcje MYSQLIB

Aby zakończyć nie stałe połączenie z bazą danych (nawiązane przy użyciu *mysql_connect*) należy wywołać funkcję *mysql_close*.

Używanie *mysql_close* nie jest **konieczne**, ponieważ automatycznie po zakończeniu przetwarzania skryptu jest ono zakańczane w ramach **zwalniania zasobów**.

Funkcja zwraca false w przypadku niepowodzenia.

```
bool mysql_close ([ resource $identyfikator_połączenia = NULL ] )
```

Funkcje MYSQLIB

W celu pobrania informacji o bazie danych z którą nawiązano połączenie można wykorzystać funkcję `mysql_get_server_info`.

Można opcjonalnie podać identyfikator połączenia. Domyślnie zostaje przypisany ostatnio utworzony identyfikator.

Zwrócona zostaje informacja o wersji serwera w postaci string.

```
string mysql_get_server_info ( [ resource $identyfikator_połączenia ] )
```

Funkcje MYSQLIB

Przykład

```
mysql_connect('localhost', 'uzytkownik', 'haslo') or  
    die('Nie można się połączyć');  
printf ("MySQL server version: %s\n", mysql_get_server_info());
```

Funkcje MYSQLIB

Aby pobrać listę baz danych dostępnych na serwerze należy skorzystać z funkcji *mysql_list_dbs*.

Wyświetlone zostaną tylko te, do których użytkownik ma uprawnienia. Domyślnie każdy użytkownik ma dostęp do bazy *information_schema*.

```
resource mysql_list_dbs ([ resource $ identyfikator_połączenia ] )
```


Funkcje MYSQLIB

Przykład:

```
$link = mysql_connect ('localhost','uzytkownik','haslo');  
$db_list = mysql_list_dbs($link);  
  
while ($row = mysql_fetch_object($db_list)) {  
    echo $row->Database . "\n";  
}
```

Funkcje MYSQLIB

Aby pobrać listę tabel z konkretnej bazy danych należy skorzystać z funkcji *mysql_list_tables*. Funkcja ta przyjmuje jako argument nazwę bazy danych oraz uchwyt do połączenia.

```
resource mysql_list_tables ( string $baza [, resource $ identyfikator_połączenia ] )
```

Funkcje MYSQLIB

Przykład:

```
mysql_connect('localhost', 'uzytkownik', 'haslo') or  
    die('Nie można się połączyć');  
mysql_select_db('baza');  
  
$result = mysql_list_tables();  
  
while (($row = mysql_fetch_row($result))  
    printf ("Tabela: %s\n", $row[0]);
```

Funkcje MYSQLIB

Aby pobrać kod błędu ostatniego użytego polecenia przy użyciu funkcji `mysql_*` należy skorzystać z funkcji `mysql_errno`. Błędy generowane z DB nie powodują generowania ostrzeżeń w języku PHP. Zamiast tego programista musi sam oprogramować mechanizm informowania o błędach.

```
int mysql_errno ( [ resource $identyfikator_połączenia ] )
```

Funkcje MYSQLIB

W parze z kodem błędu idzie również tekst zawierający pełniejszy komunikat o występującym błędzie w ostatnio użytym poleceniu z funkcji *mysql_**.

```
string mysql_error ([ resource $ identyfikator_połączenia ] )
```

Funkcje MYSQLIB

Przykład:

```
mysql_select_db("kossu", $link);  
if (!mysql_query("SELECT * FROM nieistniejaca tabela", $link)) {  
    echo mysql_errno($link) . ": " . mysql_error($link) . "\n";  
}
```

Funkcje MYSQLIB

Aby wybrać konkretną bazę danych należy skorzystać z polecenia *mysql_select_db*. Jako obowiązkowy argument należy podać nazwę bazy z której będziemy chcieli korzystać podczas aktywnego połączenia. Kolejne wywołania będą dotyczyły tej bazy.

```
bool mysql_select_db ( string $nazwa_bazy [, resource $identyfikator_połączenia ] )
```

Funkcje MYSQLIB

Przykład:

```
// Ustaw foo jako aktualną bazę danych
$db_selected = mysql_select_db('foo', $link);
if (!$db_selected) {
    die ('Nie można ustawić foo : ' . mysql_error());
}
```


Funkcje MYSQLIB

Gdy mamy nawiązane połączenie oraz wybraliśmy bazę danych na której będziemy pracować można przystąpić do wykonywania poleceń SQL. Aby przekazać je do serwera należy skorzystać z funkcji *mysql_query*.

Jeśli nie zostaną podane żadne argumenty, nastąpi odwołanie do ostatniego otwartego połączenia. W przypadku gdy nie ma żadnych aktywnych, wywoływana jest funkcja *mysql_connect* bezargumentowa.

```
resource mysql_query ( string $zapytanie [, resource $identyfikator_połączenia [, int  
$typ_wyniku ]] )
```

Funkcje MYSQLIB

Opcjonalny argument `result_mode` może przyjmować wartość `MYSQL_USE_RESULT` lub `MYSQL_STORE_RESULT`. Domyślnie używana jest wartość `MYSQL_STORE_RESULT` dlatego wynik jest buforowany. Odpowiednikiem użycia drugiej opcji jest funkcja `mysql_unbuffered_query()`

```
resource mysql_query ( string $zapytanie [, resource $identyfikator_połączenia [, int  
$typ_wyniku ]] )
```

Funkcje MYSQLIB

mysql_query() zwraca identyfikator wyniku (lub **FALSE** w przypadku niepowodzenia) jedynie dla zapytań typu SELECT, SHOW, EXPLAIN i DESCRIBE. Dla innych zapytań SQL **mysql_query()** zwraca **TRUE** lub **FALSE** informując czy zapytanie zakończyło się sukcesem czy też nie. Jeśli nie została zwrócona wartość **FALSE** to znaczy, że zapytanie było prawidłowe i może być wykonane przez serwer. Nie mówi natomiast nic o liczbie przetworzonych lub zwróconych wierszy. Jest również możliwe, że zapytanie zostanie wykonane poprawnie, nie przetwarzając lub zwracając żadnych wierszy.

Funkcje MYSQLIB

Przykład:

UWAGA !!!!

Zapytanie SQL w `mysql_query` nie powinno kończyć się średnikiem

```
$result = mysql_query("SELECT kolumna FROM tabela")  
or die("Zapytanie niepoprawne");
```

Funkcje MYSQLIB

Pobranie wyniku zapytania odbywać się może na wiele sposobów. Służy do tego kilka funkcji `mysql_fetch_*`.

`mysql_fetch_array` — Zapisuje wiersz wyniku w tablicy asocjacyjnej, numerycznej lub w obu. Zwraca tablicę zawierającą pobrany wiersz, lub **`FALSE`** jeżeli nie ma więcej wierszy w wynik.

```
array mysql_fetch_array ( resource $wynik [, int $ typ_wyniku ] )
```

Funkcje MYSQLIB

Pobranie wyniku zapytania odbywać się może na wiele sposobów. Służy do tego kilka funkcji `mysql_fetch_*`.

`mysql_fetch_array` — Zapisuje wiersz wyniku w tablicy asocjacyjnej, numerycznej lub w obu. Zwraca tablicę zawierającą pobrany wiersz, lub **`FALSE`** jeżeli nie ma więcej wierszy w wynik.

```
array mysql_fetch_array ( resource $wynik [, int $ typ_wyniku ] )
```

Funkcje MYSQLIB

Opcjonalny drugi argument `result_type` w funkcji **`mysql_fetch_array()`** jest stałą i może przyjmować następujące wartości:

`MYSQL_ASSOC`,
`MYSQL_NUM`
`MYSQL_BOTH`.

Wartością domyślną jest `MYSQL_BOTH`.

```
array mysql_fetch_array ( resource $wynik [, int $ typ_wyniku ] )
```

Funkcje MYSQLIB

Przykład:

```
$result = mysql_query("SELECT id, nazwa FROM tabela");  
  
while ($row = mysql_fetch_array($result, MYSQL_BOTH)) {  
    printf ("ID: %s  Nazwa: %s", $row[0], $row["nazwa"]);  
}
```


Funkcje MYSQLIB

`mysql_fetch_assoc` — Zapisuje wiersz wyniku w tablicy asocjacyjnej. Zwraca tablicę zawierającą pobrany wiersz, lub `FALSE` jeżeli nie ma więcej wierszy w wynik. Użycie `mysql_fetch_assoc()` jest równoznaczne z wywołaniem `mysql_fetch_array()` podając jako drugi argument `MYSQL_ASSOC`. Zwraca jedynie tablicę asocjacyjną. Początkowo tak właśnie zachowywała się funkcja `mysql_fetch_array()`.

```
array mysql_fetch_assoc ( resource $wynik )
```

Funkcje MYSQLIB

`mysql_fetch_assoc` — Zapisuje wiersz wyniku w tablicy asocjacyjnej. Zwraca tablicę zawierającą pobrany wiersz, lub `FALSE` jeżeli nie ma więcej wierszy w wynik. Użycie `mysql_fetch_assoc()` jest równoznaczne z wywołaniem `mysql_fetch_array()` podając jako drugi argument `MYSQL_ASSOC`. Zwraca jedynie tablicę asocjacyjną. Początkowo tak właśnie zachowywała się funkcja `mysql_fetch_array()`.

```
array mysql_fetch_assoc ( resource $wynik )
```

Funkcje MYSQLIB

Przykład:

```
mysql_connect('localhost','uzytkownik','haslo');  
mysql_select_db('baza');  
$result = mysql_query('select * from tabela');  
while ($row = mysql_fetch_assoc($result)) {  
    echo $row["id_uzytkownika"];  
    echo $row["pelna_nazwa"];  
}
```

Funkcje MYSQLIB

mysql_fetch_object — Zapisuje wiersz wyniku jako obiekt

```
$result = mysql_query('select * from tabela');  
while ($row = mysql_fetch_object($result)) {  
    echo $row->id_uzytkownika;  
    echo $row->pelna_nazwa;  
}
```

```
object mysql_fetch_object ( resource $wynik )
```

Funkcje MYSQLIB

`mysql_num_rows` — zwraca liczbę wierszy w wyniku. Tę funkcję stosuje się tylko do operacji SELECT

```
$result = mysql_query('SELECT * FROM tabela', $link);  
$num_rows = mysql_num_rows($result);
```

```
int mysql_num_rows ( resource $wynik )
```

Funkcje MYSQLIB

`mysql_affected_rows` — Zwraca liczbę wierszy przetworzonych w poprzedniej operacji MySQL i -1 jeśli ostatnie zapytanie zawiedzie.

Pobiera liczbę wierszy przetworzonych w ostatnim zapytaniu INSERT, UPDATE, REPLACE lub DELETE skojarzonym zidentyfikator_połączenia.

```
int mysql_affected_rows ([ resource $identyfikator_połączenia ] )
```

Funkcje MYSQLIB

Przykład:

```
mysql_query('DELETE FROM mytable WHERE id < 10');  
printf("Usuniętych rekordów: %d\n", mysql_affected_rows());
```

Funkcje MYSQLIB

`mysql_fetch_field` — Pobiera z wyniku informacje o kolumnie i zwraca jako obiekt

`mysql_fetch_field()` pozwala uzyskać informacje o polach w danym wyniku zapytania. Jeżeli offset pola nie został podany, zwracane zostanie następne pole nie pobrane jeszcze przez `mysql_fetch_field()`.

```
object mysql_fetch_field ( resource $wynik [, int $ offset_pola ] )
```



```

$result = mysql_query('select * from tabela')
    or die('Zapytanie zakończone niepowodzeniem');
/* pobieramy dane kolumny */
$i = 0;
while ($i < mysql_num_fields($result)) {
    echo "Informacje o kolumnie $i:<br />\n";
    $meta = mysql_fetch_field($result);
    if (!$meta) {
        echo "Informacje niedostępne<br />\n";
    }
    echo "<pre>
blob:          $meta->blob
max_length:    $meta->max_length
multiple_key:  $meta->multiple_key
name:          $meta->name
not_null:      $meta->not_null
numeric:       $meta->numeric
primary_key:   $meta->primary_key
table:         $meta->table
type:          $meta->type
unique_key:    $meta->unique_key
unsigned:      $meta->unsigned
zerofill:      $meta->zerofill
</pre>";
    $i++;
}

```

Funkcje

```
$link = mysql_connect('localhost', 'użytkownik_mysql', 'hasło_mysql');  
if (!$link) {  
    die('Nie można się połączyć: ' . mysql_error());  
}  
echo 'Połączenie nawiązane';  
mysql_close($link);
```

Funkcje MYSQLIB

Aby wygodnie tworzyć parametryzowane zapytania SQL zalecane jest skorzystanie z funkcji `sprintf`, która zwraca nam string sformatowany.

```
string sprintf ( string $format [, mixed $args [, mixed $... ] ] )
```

```
$query = sprintf("SELECT * FROM uzytkownicy WHERE uzytkownik='%s' AND haslo='%s'",  
    mysql_real_escape_string($uzytkownik),  
    mysql_real_escape_string($haslo));
```