PHP and MySQL

TCNJ Dynamic Web

Spring Break Special

Jean Chu

Appoligize

- I gave you wrong information last time. Sorry!

- database at ipower.com can be controlled by php codes as well, as long as that file are in the tcnjart.com server.
- If you try to use the php scripts from your tcnj.edu/~myid , the codes would not run.
- USE the tcnjart.com server, and you can reach the database through the php codes as well.

Use Our Shared DB and the Server

Our Shared DB

Host:ipower.com

username: interactive

password:
11bestoftimes_2012

Our FTP

Server:ftp.tcnjart.com

Username:interactive

Password:11bestoftimes_2012

PHP functions to talk to the database

1.

mysql_connect()

Connect to a MySQL database

2.

mysql_query()

Issue a query on a MySQL database which often involves storing or retrieving data from a table.

3.

mysql_close()

Close a connection with a MySQL database.

1. mysql_connect()

mysql_connect() Connect to a MySQL database

There are 2 ways here to connect and report error. One is to use the "or die" and the other is to use conditional

---

method 1

Connection Variable

Error Handling

```
$dbc = mysql_connect('host', 'user', 'password', 'databasename') or die('Error connecting to MySQL server.');
```

Or method 2

Conditional

```
if ($dbc = mysql_connect('interactive.ipowermysql.com', 'user', 'password') {print 'successful';}
else{ mysql_error($dbc)}
```

Function for Error report. returns a textual version of the error that the MySQL server returned.

2. mysql_query()

mysql_query() Issue a query on a MySQL database which often involves storing or retrieving data from a table.

Database Connection

mysql_query($dbc, $query) *You could put the inside a variable and apply it, or you could directly type in.*

MySQL Query

---

method using a variable

```
$dbc = mysqli_connect( 'host', 'user', 'password', 'databasename');

$query = 'CREATE TABLE entries (entry_id INT NULL AUTO_INCREMENT
PRIMARY KEY, title VARCHAR(100) NOT NULL, entry TEXT NOT NULL,
date_entered DATETIME NOT NULL )';

$result = mysql_query($dbc, $query) or die('Error querying database.');
```

Result saved in variable

2. mysql_query()

mysql_query() Issue a query on a MySQL database which often involves storing or retrieving data from a table.

Database Connection Variable

mysql_query($dbc, $query) *You could put the inside a variable and apply it, or you could directly type in.*

MySQL Query Variable

Or method typing in directly

mysql_query('interactive.ipowermysql.com', 'user', 'password', "CREATE TABLE entries (entry_id INT NULL AUTO_INCREMENT PRIMARY KEY, title VARCHAR(100) NOT NULL, entry TEXT NOT NULL, date_entered DATETIME NOT NULL)") or die('Error querying database.');

2. mysql_query()

Variable to save the result

$result=mysql_query()

```
$query = 'SELECT * FROM entries ORDER BY date_entered DESC
$result = mysql_query($query, $dbc))
```

** The mysql_fetch_array() function stores a row of data in an array.*

$row = mysqli_fetch_array($result);

The variable $row is an array that initially stores the first row of data from our results

2. mysql_query()

while (*test_condition*) { *action*}

The test condition always results in true or false... keep looping (true) or stop looping (false).

The while loop condition is the return value of the mysqli_fetch_array() function, which is interpreted as true if data is available or false if we're all out of data.

```
while($row = mysql_fetch_array($result)) {
    print "$row['table_column_name'] " ;   }
```

The while loop goes through the table data, row by row. When it runs out of rows of data, it ends.

3. mysql_close()

mysql_close() Close a connection with a MySQL database.

```
mysql_close($dbc);
```

MYSQL

TABLE 4.2 MySQL Data Types

Type	Size	Description
`CHAR[Length]`	Length bytes	A fixed-length field from 0 to 255 characters long
`VARCHAR[Length]`	String length + 1 or 2 bytes	A variable-length field from 0 to 65,535 characters long
`TINYTEXT`	String length + 1 bytes	A string with a maximum length of 255 characters
`TEXT`	String length + 2 bytes	A string with a maximum length of 65,535 characters
`MEDIUMTEXT`	String length + 3 bytes	A string with a maximum length of 16,777,215 characters
`LONGTEXT`	String length + 4 bytes	A string with a maximum length of 4,294,967,295 characters
`TINYINT[Length]`	1 byte	Range of –128 to 127 or 0 to 255 unsigned
`SMALLINT[Length]`	2 bytes	Range of –32,768 to 32,767 or 0 to 65,535 unsigned
`MEDIUMINT[Length]`	3 bytes	Range of –8,388,608 to 8,388,607 or 0 to 16,777,215 unsigned
`INT[Length]`	4 bytes	Range of –2,147,483,648 to 2,147,483,647 or 0 to 4,294,967,295
`BIGINT[Length]`	8 bytes	Range of –9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 or 0 to 18,446,744,073,709,551,615 unsigned
`FLOAT[Length, Decimals]`	4 bytes	A small number with a floating decimal point
`DOUBLE[Length, Decimals]`	8 bytes	A large number with a floating decimal point
`DECIMAL[Length, Decimals]`	Length + 1 or 2 bytes	A `DOUBLE` stored as a string, allowing for a fixed decimal point
`DATE`	3 bytes	In the format of YYYY-MM-DD
`DATETIME`	8 bytes	In the format of YYYY-MM-DD HH:MM:SS
`TIMESTAMP`	4 bytes	In the format of YYYYMMDDHHMMSS; acceptable range starts in 1970 and ends in the year 2038
`TIME`	3 bytes	In the format of HH:MM:SS
`ENUM`	1 or 2 bytes	Short for enumeration, which means that each column can have one of several possible values
`SET`	1, 2, 3, 4, or 8 bytes	Like `ENUM` except that each column can have more than one of several possible values