

PHP Programming

Basic Skills (3)

Wen-Hsiang Lu (盧文祥)

Department of Computer Science and Information Engineering,
National Cheng Kung University

2014/12/4

Outline

- SQL Language
- PHP+MySQL

Introduction

- PHP (Hypertext Preprocessor)
 - Open source, server-side, scripting language.
 - Supports databases such as MySQL and Oracle.
 - <http://www.w3schools.com/php/default.asp>
- MySQL (Structured Query Language)
 - Open source, speedy, scalable, reliable database technology.
 - <http://dev.mysql.com/doc/mysql/en/Tutorial.html>

Basic SQL Syntax

- `SHOW DATABASES;`
- `USE database_name;`
- `SHOW TABLES;`
- `DROP TABLE table_name;`

Create/Delete SQL Table

- CREATE TABLE **user** (name varchar(9) NOT NULL,
id int(6) NOT NULL,
PRIMARY KEY (id),
UNIQUE (id)
);
- DROP TABLE **user**;

Add/Delete/Update Table

- INSERT INTO **user** (name, id) VALUES ('tim','001');
- DELETE FROM **user** WHERE id='001';
- UPDATE **user** SET name='Tim' WHERE id='001';

Query Database

- SELECT * FROM **user**;
- SELECT * FROM **user** WHERE name='BOND';
- SELECT DISTINCT name FROM **user**;
- SELECT name, id FROM user **ORDER BY** name;
- SELECT pid, SUM(price*num) as **total** FROM order **GROUP BY** pid HAVING total>1000;

MySQL

- Open Source database server
 - Runs on many platforms (Unix & Windows)
- Networked server – no fancy GUI like MS Access.
 - You can find *clients* that provide a GUI.
- Great for **small to medium-sized applications**

phpMyAdmin

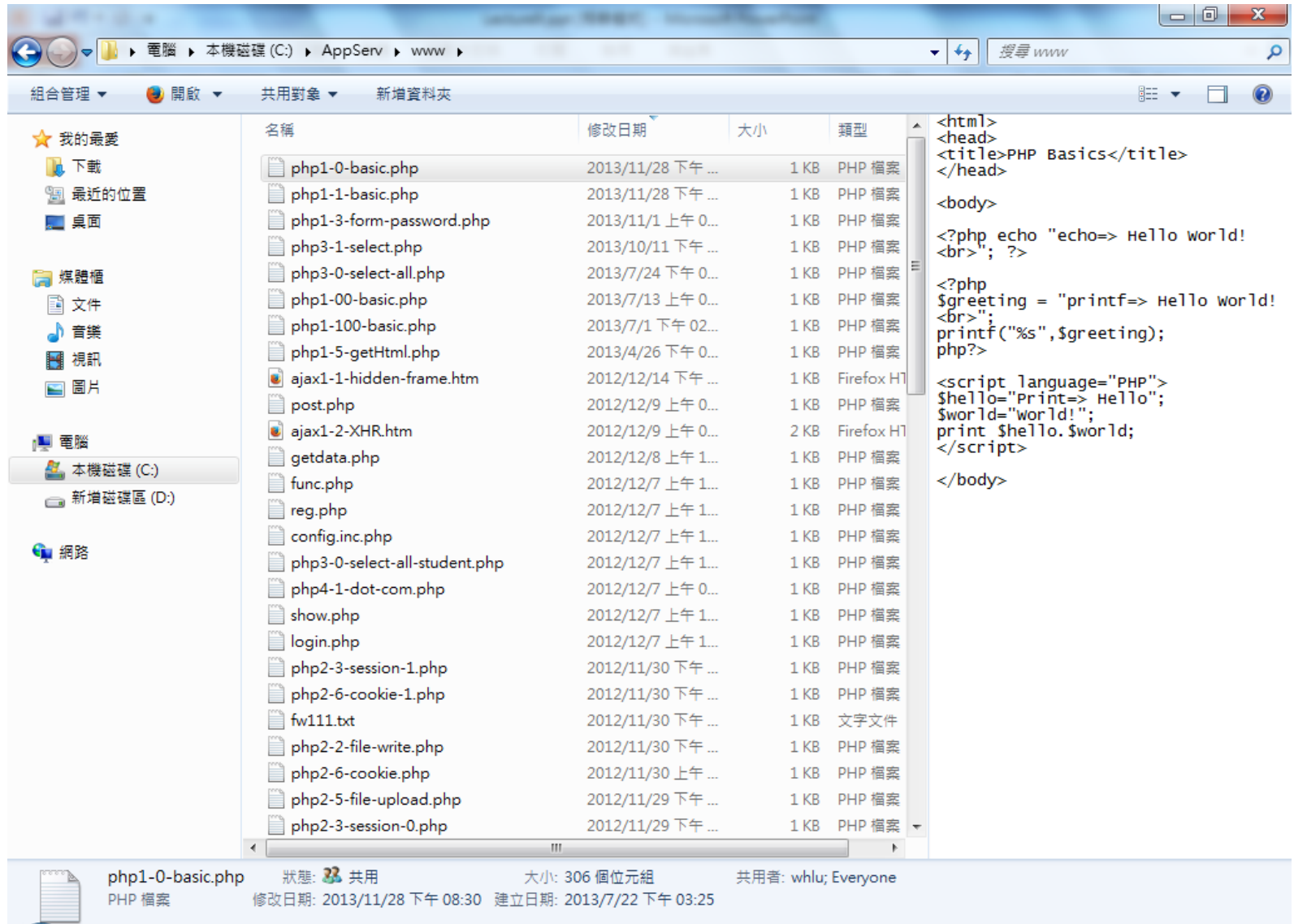
- A MySQL client written in PHP
- Via the web you can :
 - Manage Databases
 - Manage MySQL users
 - Submit queries (SQL)
- A great way to learn SQL!

Start Apache web server



Web pages stored in the web server directory :

C:/AppServ/www/



The screenshot shows a Windows Explorer window with the address bar set to `C:/AppServ/www/`. The left sidebar shows the navigation pane with '我的最愛' (Favorites) and '電腦' (Computer) sections. The main pane displays a list of files and folders. The right pane shows the content of the selected file, `php1-0-basic.php`.

名稱	修改日期	大小	類型
php1-0-basic.php	2013/11/28 下午 ...	1 KB	PHP 檔案
php1-1-basic.php	2013/11/28 下午 ...	1 KB	PHP 檔案
php1-3-form-password.php	2013/11/1 上午 0...	1 KB	PHP 檔案
php3-1-select.php	2013/10/11 下午 ...	1 KB	PHP 檔案
php3-0-select-all.php	2013/7/24 下午 0...	1 KB	PHP 檔案
php1-00-basic.php	2013/7/13 上午 0...	1 KB	PHP 檔案
php1-100-basic.php	2013/7/1 下午 02...	1 KB	PHP 檔案
php1-5-getHtml.php	2013/4/26 下午 0...	1 KB	PHP 檔案
ajax1-1-hidden-frame.htm	2012/12/14 下午 ...	1 KB	Firefox HTML 檔案
post.php	2012/12/9 上午 0...	1 KB	PHP 檔案
ajax1-2-XHR.htm	2012/12/9 上午 0...	2 KB	Firefox HTML 檔案
getdata.php	2012/12/8 上午 1...	1 KB	PHP 檔案
func.php	2012/12/7 上午 1...	1 KB	PHP 檔案
reg.php	2012/12/7 上午 1...	1 KB	PHP 檔案
config.inc.php	2012/12/7 上午 1...	1 KB	PHP 檔案
php3-0-select-all-student.php	2012/12/7 上午 1...	1 KB	PHP 檔案
php4-1-dot-com.php	2012/12/7 上午 0...	1 KB	PHP 檔案
show.php	2012/12/7 上午 1...	1 KB	PHP 檔案
login.php	2012/12/7 上午 1...	1 KB	PHP 檔案
php2-3-session-1.php	2012/11/30 下午 ...	1 KB	PHP 檔案
php2-6-cookie-1.php	2012/11/30 下午 ...	1 KB	PHP 檔案
fw111.txt	2012/11/30 下午 ...	1 KB	文字文件
php2-2-file-write.php	2012/11/30 下午 ...	1 KB	PHP 檔案
php2-6-cookie.php	2012/11/30 上午 ...	1 KB	PHP 檔案
php2-5-file-upload.php	2012/11/29 下午 ...	1 KB	PHP 檔案
php2-3-session-0.php	2012/11/29 下午 ...	1 KB	PHP 檔案

```
<html>
<head>
<title>PHP Basics</title>
</head>

<body>

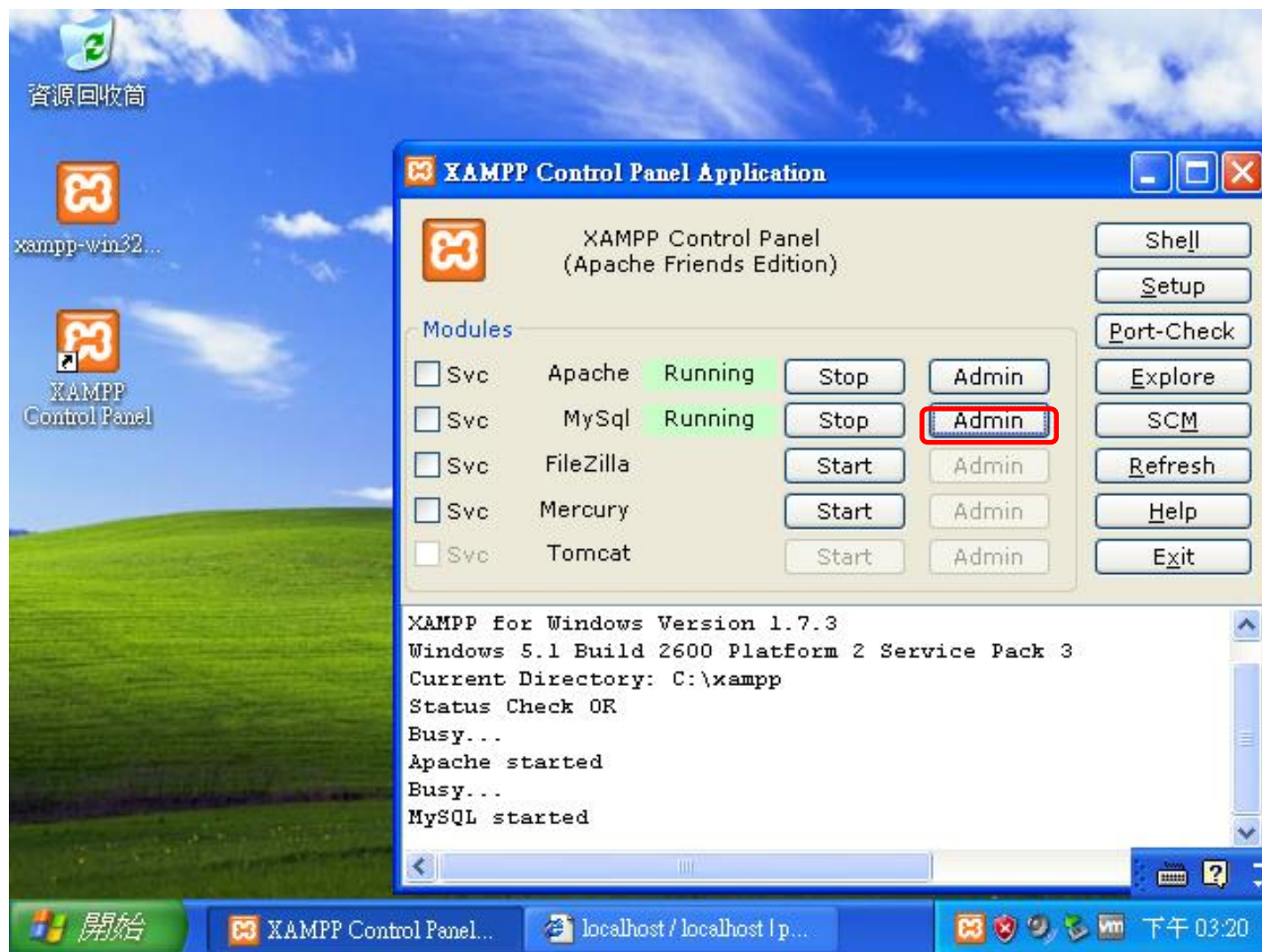
<?php echo "echo=> Hello world!
<br>"; ?>

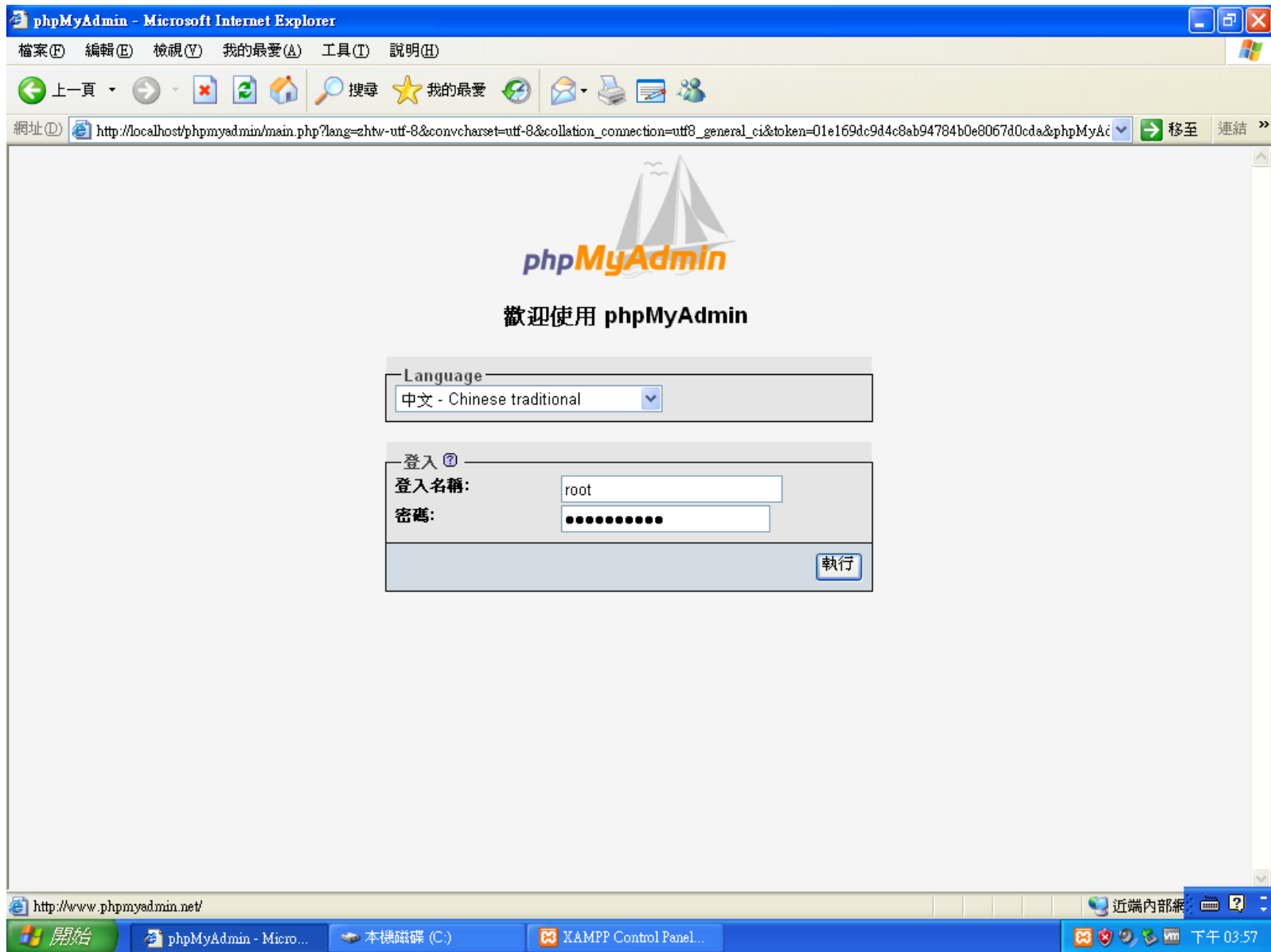
<?php
$greeting = "printf=> Hello world!
<br>";
printf("%s", $greeting);
php?>

<script language="PHP">
$hello="Print=> Hello";
$world="world!";
print $hello.$world;
</script>

</body>
```

狀態: 共用
大小: 306 個位元組
共用者: whlu; Everyone
修改日期: 2013/11/28 下午 08:30 建立日期: 2013/7/22 下午 03:25





localhost / localhost | phpMyAdmin 3.2.4 - Mozilla Firefox

檔案 (F) 編輯 (E) 檢視 (V) 歷史 (S) 書籤 (B) 工具 (T) 說明 (H)

localhost/phpmyadmin/index.php?db=homework&token=b41c269147b95fb01015657c5b0737c9

phpMyAdmin

- cdcol (1)
- information_schema (28)
- mysql (23)
- phpmyadmin (8)
- products (4)
- test (1)
- whlu (2)

請選擇資料庫

伺服器: localhost

資料庫 SQL 狀態 資訊 文字編碼 引擎 權限 處理 輸出 載入

Actions

- 更改密碼
- 登出系統

MySQL localhost

建立新資料庫 ?

student 校對 建立

MySQL 連線校對: utf8_general_ci

Interface

Language: 中文 - Chinese

款式: Original

Custom color: 重置

Font size: 82%

utf8_czech_ci
utf8_danish_ci
utf8_esperanto_ci
utf8_estonian_ci
utf8_general_ci
utf8_hungarian_ci
utf8_icelandic_ci
utf8_latvian_ci
utf8_lithuanian_ci
utf8_persian_ci
utf8_polish_ci
utf8_romanian_ci
utf8_romanian_ci
utf8_slovak_ci
utf8_slovenian_ci
utf8_spanish2_ci
utf8_spanish_ci
utf8_swedish_ci
utf8_turkish_ci
utf8_unicode_ci

MySQL

- 伺服器: localhost via TCP/IP
- 伺服器版本: 5.1.41
 - 通訊協定版本: 10
 - 使用者: root@localhost
- MySQL 文字編碼: UTF-8 Unicode (utf8)

Web server

- Apache/2.2.14 (Win32) DAV/2
 - mod_ssl/2.2.14 OpenSSL/0.9.8l
 - mod_autoindex_color PHP/5.3.1
 - mod_apreq2-20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
- MySQL 客戶端版本: 5.1.41
- PHP extension: mysqli

phpMyAdmin

- 版本資訊: 3.2.4
- 說明文件
- Wiki
- phpMyAdmin 官方網站
- [ChangeLog] [Subversion] [Lists]

設定檔內有關設定 (root 登入及沒有密碼) 與預設的 MySQL 權限戶口相同。MySQL 伺服器在這預設的設定運行的話會很容易被入侵，您應更改有關設定去防止安全漏洞。

localhost / localhost / student | phpMyAdmin 3.2.4 - Mozilla Firefox

檔案 (F) 編輯 (E) 檢視 (V) 歷史 (S) 書籤 (B) 工具 (T) 說明 (H)

localhost/phpmyadmin/index.php?db=student&token=b41c269147b95fb01015657c5b0737c9

phpMyAdmin

資料庫

student

student (0)

資料庫中沒有資料表

伺服器: localhost ▶ 資料庫: student

結構 SQL 搜索 依範例查詢 (QBE) 輸出 載入 Designer 管理 權限 刪除

資料庫中沒有資料表

建立新資料表於資料庫 student

名稱: user 欄位數目: 3

執行

開啟新 phpMyAdmin 視窗

localhost / localhost / student / user | phpMyAdmin 3.2.4 - Mozilla Firefox

檔案 (F) 編輯 (E) 檢視 (V) 歷史 (S) 書籤 (B) 工具 (I) 說明 (H)

localhost/phpmyadmin/index.php?db=student&token=b41c269147b95fb01015657c5b0737c9

phpMyAdmin

資料庫

student

student (0)

資料庫中沒有資料表

伺服器: localhost 資料庫: student 資料表: user

欄位	id	name	pw
型態 ①	INT	VARCHAR	VARCHAR
長度/集合* ¹		12	
預設值 ²	None		None
校對			
屬性			
Null	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
索引鍵 INDEX	---	---	---
AUTO_INCREMENT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
註解			
MIME 類型			
瀏覽器轉換方式			
轉換方式選項 ³			

資料表註解文字:

PARTITION definition: ?

儲存引擎: MyISAM

校對:

INT
VARCHAR
TEXT
DATE
NUMERIC
TINYINT
SMALLINT
MEDIUMINT
INT
BIGINT
-
DECIMAL
FLOAT
DOUBLE
REAL
-
BIT
BOOL
SERIAL
DATE and TIME

localhost / localhost / student / user | phpMyAdmin 3.2.4 - Mozilla Firefox

檔案 (F) 編輯 (E) 檢視 (V) 歷史 (S) 書籤 (B) 工具 (I) 說明 (H)

localhost/phpmyadmin/index.php?db=student&token=b41c269147b95fb01015657c5b0737c9

phpMyAdmin

資料庫

student

student (0)

資料庫中沒有資料表

伺服器: localhost 資料庫: student 資料表: user

欄位	id	name	pw
型態	INT	VARCHAR	VARCHAR
長度/集合 ^{*1}			
預設值 ²	None	None	None
校對			
屬性			
Null	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
索引鍵 INDEX	---	---	---
AUTO_INCREMENT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
註解			
MIME 類型			
瀏覽器轉換方式			
轉換方式選項 ³			

資料表註解文字:

儲存引擎: MyISAM

校對:

PARTITION definition:

localhost / localhost / student / user | phpMyAdmin 3.2.4 - Mozilla Firefox

檔案 (F) 編輯 (E) 檢視 (V) 歷史 (S) 書籤 (B) 工具 (T) 說明 (H)

localhost/phpmyadmin/index.php?db=student&token=b41c269147b95fb01015657c5b0737c9

phpMyAdmin

資料庫

student (1)

student (1)

user



















伺服器: localhost ▶ 資料庫: student ▶ 資料表: user

瀏覽 結構 SQL 搜索 新增 輸出 載入 管理 清空 刪除

✓ Table 'student'. 'user' has been created.

```
CREATE TABLE `student`.`user` (  
  `id` INT NOT NULL ,  
  `name` VARCHAR( 15 ) NOT NULL ,  
  `pw` VARCHAR( 12 ) NOT NULL  
) ENGINE = MYISAM ;
```

[編輯] [建立 PHP 程式碼]

	欄位	型態	校對	屬性	Null	預設值	附加	執行
<input type="checkbox"/>	id	int(11)			否	None		     
<input type="checkbox"/>	name	varchar(15)	latin1_swedish_ci		否	None		     
<input type="checkbox"/>	pw	varchar(12)	latin1_swedish_ci		否	None		     

全選 / 全部取消 選擇的資料表

列印檢視 關聯檢視 分析資料表結構

新增 1 個欄位 於資料表尾端 於資料表開頭 在之後 id 執行

⚠ 沒有已定義的索引!

新增 1 組索引欄 執行

已使用空間			資料列統計數值	
型態	使用		敘述	值
資料	0	Bytes	格式	動態
索引鍵 INDEX	1,024	Bytes	資料列列數	0
總共	1,024	Bytes	建立	Dec 07, 2012, 02:30
			最後更新	Dec 07, 2012, 02:30

localhost / localhost / student / user | phpMyAdmin 3.2.4 - Mozilla Firefox

檔案 (F) 編輯 (E) 檢視 (V) 歷史 (S) 書籤 (B) 工具 (I) 說明 (H)

ll.php http://...n.php 體育 詹姆斯最... Web Appli... localhost / ... http://...n.php http://...t.php http://...m.php localhost / ...

localhost/phpmyadmin/index.php?db=student&token=b41c269147b95fb01015657c5b0737c9

phpMyAdmin

資料庫

student (1)

student (1)

user

伺服器: localhost 資料庫: student 資料表: user

瀏覽 結構 SQL 搜索 新增 輸出 載入 管理 清空 刪除

欄位	型態	函數	Null	值
id	int(11)			1
name	varchar(15)			Tim
pw	varchar(12)			123

執行

☐ 忽略

欄位	型態	函數	Null	值
id	int(11)			2
name	varchar(15)			Danny
pw	varchar(12)			1234

執行

儲存為新記錄 然後 返回

1 執行 重置

Restart insertion with 2 rows

1 按 TAB 鍵跳到下一個數值, 或 CTRL+方向鍵 作隨意移動

開啟新 phpMyAdmin 視窗

http://localhost/phpmyadmin/tbl_chang...ken=b41c269147b95fb01015657c5b0737c9

localhost / localhost / student / user | phpMyAdmin 3.2.4 - Mozilla Firefox

檔案 (F) 編輯 (E) 檢視 (V) 歷史 (S) 書籤 (B) 工具 (I) 說明 (H)

localhost/phpmyadmin/index.php?db=student&token=b41c269147b95fb01015657c5b0737c9

伺服器: localhost ▶ 資料庫: student ▶ 資料表: user

瀏覽 結構 SQL 搜索 新增 輸出 載入 管理 清空 刪除

顯示記錄 0 - 1 (2 總計, 查詢需時 0.0003 秒)

```
SELECT *
FROM `user`
LIMIT 0 , 30
```

☐ Profiling [編輯] [說明 SQL] [建立 PHP 程式碼] [更新]

顯示: 30 筆記錄, 開始列數: 0
顯示為 水平 方式及 每隔 100 行顯示欄名

+ Options

	id	name	pw
<input type="checkbox"/>	1	Tim	123
<input type="checkbox"/>	2	Danny	1234

全選 / 全部取消 選擇的資料表: ☐ ☐

顯示: 30 筆記錄, 開始列數: 0
顯示為 水平 方式及 每隔 100 行顯示欄名

查詢結果操作

☐ 列印檢視 ☐ 列印檢視 (顯示完整文字) ☐ 輸出 ☐ CREATE VIEW

將此 SQL 語法加入書籤

書籤名稱: ☐ 所有用者可讀取此書籤

將此 SQL 語法加入書籤

開啟新 phpMyAdmin 視窗

http://localhost/phpmyadmin/sql.php?...1c269147b95fb01015657c5b0737c9&pos=0

Basic Manipulication Steps

- Create MySQL database(s) with tables as required.
- Create PHP powered webpage for adding, deleting, updating and viewing database information.

Opening a MySQL database

```
$host="localhost";  
$username="whlu";  
$password="123";  
$database="student";  
  
mysql_connect($host,$username,$password);  
mysql_select_db($database) or die( "Unable to select database");
```

- Assumes that the machine running the server is “localhost”
- You can easily use a MySQL server that is running on a remote machine.

Fetching Functions

- the fetching functions (mysql_fetch) are as follows:
 - mysql_fetch_row: Returns row as an enumerated array
 - mysql_fetch_object: Returns row as an object
 - mysql_fetch_array: Returns row as an **associative array**

```
$query = "SELECT ID, Name, Password FROM user WHERE id = 1";
$result = mysql_query($query);
while ($row = mysql_fetch_row($result)) {
    print(" $row[0], $row[1], $row[2] <BR>\n");
}
while ($row = mysql_fetch_object($result)) {
    echo " $row->ID, $row->LastName, $row->FirstName <BR>\n";
}
while ($row = mysql_fetch_array($result)) {
    echo " $row[ID], $row[LastName], $row[FirstName] <BR>\n";
}
```

Session: php3-0-select-all.php

(localhost: php3-0-select-all.php)

```
<?php
```

```
$host="localhost";
```

```
$user="whlu";
```

```
$upwd="whlu";
```

```
$db="whlu";
```

```
$link=mysql_connect($host,$user,$upwd) or die ("Unable to connect!");
```

```
mysql_select_db($db, $link) or die ("Unable to select database!");
```

```
$query="SELECT * FROM `user` ";
```

```
$result= mysql_query($query,$link) or  
    die ("Error in query: $query.". mysql_error());;
```

```
while ($rows=mysql_fetch_array($result)){
```

```
    echo "$rows[0] $rows[1] $rows[2] <br>";
```

```
}
```

```
?>
```


Session: php3-1-select.php

(localhost: php3-1-select.php)

```
<?php
```

```
$host="localhost";
```

```
$user="whlu";
```

```
$upwd="123";
```

```
$db="student";
```

```
$link=mysql_connect($host,$user,$upwd) or die ("Unable to connect!");
```

```
mysql_select_db($db, $link) or die ("Unable to select database!");
```

```
$query="SELECT * FROM `user` where user='Danny' ";
```

```
$result= mysql_query($query,$link) or  
    die ("Error in query: $query. " . mysql_error());;
```

```
if($rows= mysql_fetch_array($result)){
```

```
    echo "$rows[0] $rows[1] $rows[2]";
```

```
}
```

```
else{  print "fetch error";  }
```

```
?>
```

PHP Configuration File

- Use a securely positioned 'config' file to store variables.
- Other PHP pages can link to it and use the variables as their own.

config.inc.php

```
<?
// database configuration parameters
$host="localhost";
$user="whlu";
$upwd="123";
$db="whlu";

//database connection
$link=mysql_connect($host,$user,$upwd)
    or die ("Unable to connect!");
mysql_select_db($db, $link) or die
    ("Unable to select database!");
?>
```

reg.php

```
<?php
if($_REQUEST[user]){
    include "./config.inc.php";
    $result= mysql_query("SELECT * FROM `user` WHERE user='$_REQUEST[user]'",
    $link);
    if($rows= mysql_fetch_array($result,MYSQL_ASSOC)){
        print "此帳號有人使用";
    }else{
        print "此帳號註冊成功";
        $r=md5($_POST[pw]);
        $result=mysql_query("INSERT INTO `user` (`id` ,`user` ,`pw`) VALUES
        ( NULL , '$_REQUEST[user]', '$r');", $link);
        print "<script>location.href=('show.php');</script>";
    }
}else{
    print '<form action="reg.php" method="post">
    Username: <input type="text" name="user" /><br/>
    password: <input type="password" name="pw" />
    <input type="submit" value="註冊" />      </form>';
}
?>
```

show.php

```
<?php
include_once "../config.inc.php";
include_once "../func.php";
if(vaild($_COOKIE["user"],$_COOKIE["pw"])){
    print " you are ".$_COOKIE["user"]."    ";
    <a href="\login.php?action=logout\">logout</a><br/>";
}else{ login("show.php"); }
$result= mysql_query("SELECT * FROM `score` WHERE id=(SELECT id FROM user
    WHERE user='".$_COOKIE["user"]."')",$link);
?>
<table border="1">
<tr><td>id</td> <td>english</td> <td>chinese</td> <td>math</td>
</tr>
<?php
while($rows= mysql_fetch_array($result,MYSQL_ASSOC)){
    print '<tr><td>'.$rows[id]."</td><td>".$rows[eng]."</td><td>".$rows[ch].
        "</td><td>".$rows[math]."</td></tr>";
}
?>
```

func.php

```
<?php
function vaild($users,$pws){
    if($users and $pws){
        include "./config.inc.php";
        $result= mysql_query("SELECT * FROM `user` WHERE user='$users'",
$link);
        if($rows= mysql_fetch_array($result,MYSQL_ASSOC)){
            if($rows[pw] == $pws){    return 1;    }
        }
    }
    return 0;
}

function login($url){
    print '<form action="login.php" method="post">
    Username: <input type="text" name="user" /><br/>
    password: <input type="password" name="pw" />
    <input type="hidden" name="url" value="' . $url . '" />
    <input type="submit" value="登入" />
    </form>';
    exit;
}
?>
```

login.php

```
<?php
include_once "../func.php";
if($_REQUEST[action]=="logout"){
    setcookie("user","");
    setcookie("pw","");
    print "登出OK";
    exit;
}
if($_REQUEST[user]){
    $r=md5($_POST[pw]);
    // echo "$_REQUEST[user],$r<br/>";
    if(vaild($_REQUEST[user],$r)){
        setcookie("user",$_REQUEST[user]);
        setcookie("pw",$r);
        print "登入OK";
        print "<script>location.href='show.php';</script>";
        exit;
    }else{ print "登入失敗"; }
}
login($_REQUEST[url]);
?>
```

PHP Delete

```
Include "./config.inc.php";
```

```
// form not yet submitted, display initial form with values pre-filled
```

```
$id=$_GET['id'];
```

```
{
```

```
// open database connection
```

```
$connection = mysql_connect($host, $user, $pass) or die ("Unable to connect!");
```

```
// select database
```

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

```
// generate and execute query
```

```
$query = "DELETE FROM score WHERE id = '$id'";
```

```
$result = mysql_query($query) or die ("Error in query: $query. " . mysql_error());
```

```
// close database connection
```

```
mysql_close($connection);
```

```
// print result
```

```
echo "<font size=-1>Deletion successful. <br><br><a href=papers.php>Go back to the main page </a></font>";
```

```
}
```

PHP Update

```
<?php
Include "./config.inc.php";
    // form submitted so start processing it
    $title = $_POST["title"];
    $authors = $_POST["authors"];
    $id = $_POST["id"];
    $count = 0;
    // validate text input fields
    if (!$contact) { $contact = $def_contact; }
    // check for errors, if none found...
        $connection = mysql_connect($host, $user, $pass) or die ("Unable to connect!");
        mysql_select_db($db) or die ("Unable to select database!");

        $query = "UPDATE papers SET title = '$title', authors = '$authors',
url = '$url' WHERE id = '$id'";

    $result = mysql_query($query) or die ("Error in query: $query. " . mysql_error());
    mysql_close($connection);

?>
```


Regular Expression

- function **ereg()**: Regular expression match
 - Searches a string for matches to the regular expression given in pattern in a case-sensitive way.
 - `int ereg (string $pattern , string $string)`
 - [Ex]

```
function simple_dot_com ($url){  
    return(ereg('^www\\.[a-z]+\\.com$', $url));  
}
```
- Confusingly, we have to put two backslashes in the pattern string, because PHP treats the first slash as an escape character for the second backslash. (You can get away with just one backslash, but that behavior is not guaranteed to continue in future versions of PHP.) The second backslash (escaped by the first), in turn, is a regex escape character for the following character.
- {Ex]: [php4-1-dot-com.php](#)

Regular Expression

```
function simple_dot_com ($url){  
    return(ereg('^www\\.[a-z]+\\.com$', $url));  
}
```

- **`^www\\.[a-z]+\\.com$`**
 - **`^www`** : In this expression we have the **`^`** symbol, which says that the three letters **`www`** portion must start at the beginning of the string.
 - **`\\.`** : Then comes a dot (**`.`**), preceded by a backslash that says we really want a dot, not the special **`.`** wildcard character.
 - **`[a-z]`** :Then we have a bracket-enclosed range of all the **lowercase alphabetic** letters.
 - **`+`** : The following **`+`** indicates that we want to match any number of these lowercase letters in a row, as long as we have at least one of them.
 - **`\\.com$`** : Then another literal **`.`**, the three letters **`com`**, and the special symbol **`$`** that says that com is the end of it.

Regular Expression

- The special character
 - ^ matches the beginning of a string
 - \$ matches the end of a string
 - . matches any character
 - ? matches exactly once character
 - * matches zero or more instances of the previous regular expression
 - + matches one or more instances of the previous expression.
- A set of characters enclosed in square brackets matches any of those characters
 - the pattern [ab] matches either a or b
 - the pattern [a-c] matches a, b, or c
- **Special characters** that are escaped with a backslash (\) lose their special meaning and are matched literally, e.g., \d, \s.

```
function simple_dot_com ($url){  
    return(ereg('^www\\.[a-z]+\\.com$', $url));  
}
```

Table 22-2: Common Perl-Compatible Pattern Constructs

<i>Construct</i>	<i>Interpretation</i>
Simple literal character matches	If the character involved is not special, Perl will match characters in sequence. The example pattern <code>/abc/</code> matches any string that has the substring 'abc' in it.
Character class matches: <code>[<list of characters>]</code>	Will match a single instance of any of the characters between the brackets. For example, <code>/[xyz]/</code> matches a single character, as long as that character is either x, y, or z. A sequence of characters (in ASCII order) is indicated by a hyphen, so that a class matching all digits is <code>[0-9]</code> .
Predefined character class abbreviations	The patterns <code>\d</code> will match a single digit (from the character class <code>[0-9]</code>), and the pattern <code>\s</code> matches any whitespace character.
Multiplier patterns	Any pattern followed by <code>*</code> means: "Match this pattern 0 or more times." Any pattern followed by <code>?</code> means: "Match this pattern exactly once." Any pattern followed by <code>+</code> means: "Match this pattern 1 or more times."
Anchoring characters	The caret character <code>^</code> at the beginning of a pattern means that the pattern must start at the beginning of the string; the <code>\$</code> character at the end of a pattern means that the pattern must end at the end of the string. The caret character at the beginning of a character class <code>[^abc]</code> means that the set is the complement of the characters listed (that is, any character that is not in the list).
Escape character <code>'\'</code>	Any character that has a special meaning to regex can be treated as a simple matching character by preceding it with a backslash. The special characters that might need this treatment are: <code>. \ + * ? [] ^ \$ () { } = ! < > :</code>
Parentheses	A parenthesis grouping around a portion of any pattern means: "Add the substring that matches this pattern to the list of substring matches."

Regular Expression

```
<?php
function print_links ($url){
    $fp = fopen($url, "r")
    or die("Could not contact $url");
    $page_contents = "";
    while ($new_text = fread($fp, 100)) {
        $page_contents .= $new_text;
    }
    $match_result = preg_match_all(' /<\s*A\s*HREF="([^\"]+)"\s*>([^\>]*)<\s*\s*>I ', $page_contents, $match_array,
                                    PREG_SET_ORDER);
    foreach ($match_array as $entry) {
        $href = $entry[1];
        $anchortext = $entry[2];
        print("<B>HREF</B>: $href;<B>ANCHORTEXT</B>: $anchortext<BR>");
    }
}

?>
```

Table 22-3: Perl-Compatible Regular Expression Functions

<i>Function</i>	<i>Behavior</i>
<code>preg_match()</code>	Takes a regex pattern as first argument, a string to match against as second argument, and an optional array variable for returned matches. Returns 0 if no matches are found, and 1 if a match is found. If a match is successful, the array variable contains the entire matching substring as its first element, and subsequent elements contain portions matching parenthesized portions of the pattern. As of PHP 4.3.0, an optional flag of <code>PREG_OFFSET_CAPTURE</code> is also available. This flag causes <code>preg match</code> to return into the specified array a two-element array for each match, consisting of the match itself and the offset where the match occurs.
<code>preg_match_all()</code>	Like <code>preg_match()</code> , except that it makes all possible successive matches of the pattern in the string, rather than just the first. The return value is the number of matches successfully made. The array of matches is not optional (If you want a true/false answer, use <code>preg_match()</code>). The structure of the array returned depends on the optional fourth argument (either the constant <code>PREG_PATTERN_ORDER</code> , or <code>PREG_SET_ORDER</code> , defaulting to the former). (See further discussion following the table.) <code>PREG_OFFSET_CAPTURE</code> is also available with this function.