PHP Programming Basic Skills (1)

Wen-Hsiang Lu (盧文祥)
Department of Computer Science and Information Engineering,
National Cheng Kung University
2015/11/20

Outline

- Introduction
- Resources
- PHP Language Basics
- Built-in Functions

Introduction

- **What is PHP?**
  - PHP stands for "**PHP Hypertext Preprocessor**"
  - The product was originally named *Personal Home Page Tools*
  - PHP is the Web development language written by and for Web developers.
  - An embedded scripting language for HTML like ASP or JSP
  - PHP is a server-side scripting language
  - A language that combines elements of Perl, C, and Java

Introduction

- **History of PHP**
  - Created by Rasmus Lerdorf in 1995 for tracking access to his resume
  - Originally a set of Perl scripts known as the "Personal Home Page" tools
  - Rewritten in C with database functionality
  - Added a forms interpreter and released as PHP/FI: includes Perl-like variables, and HTML embedded syntax

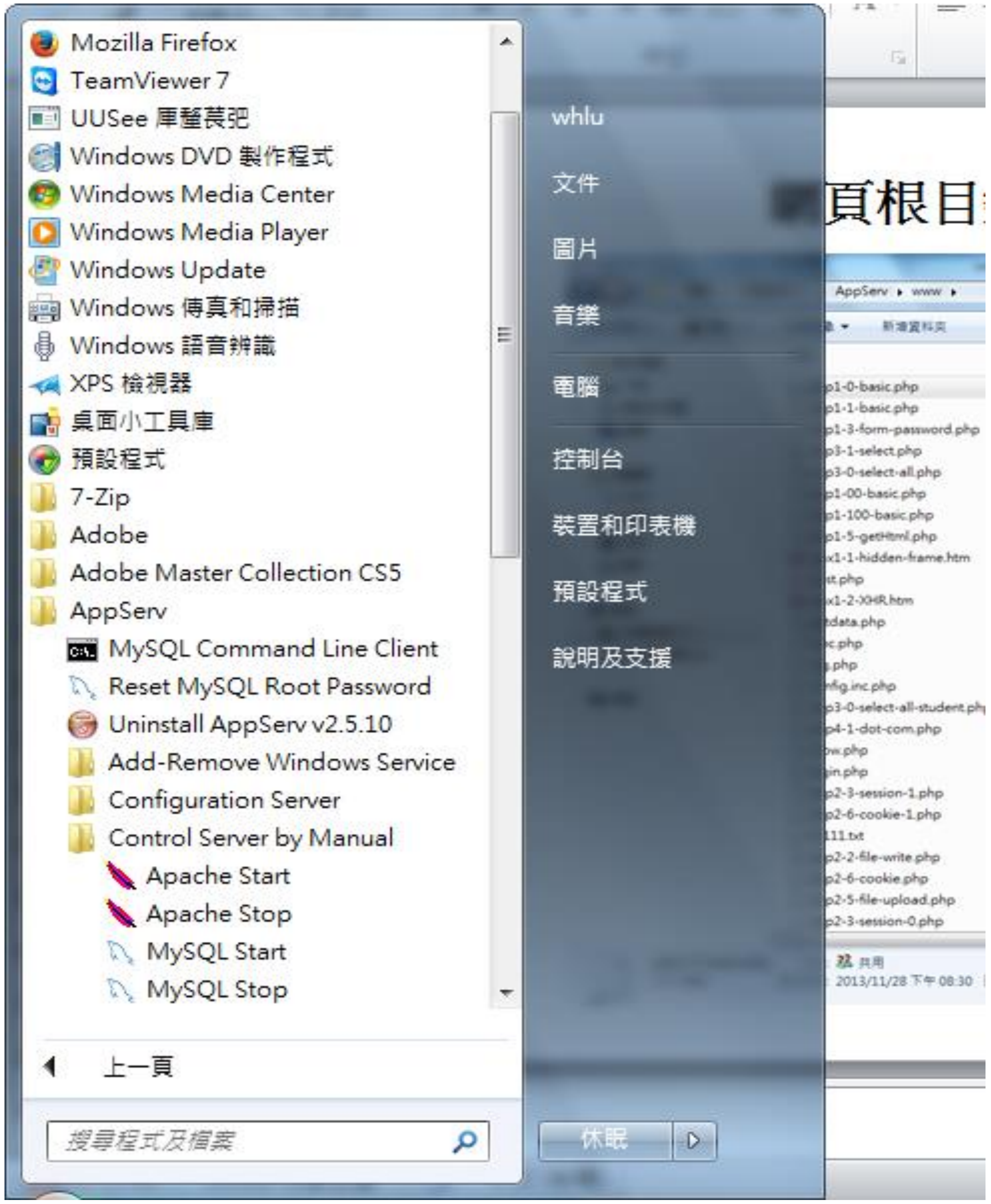
Introduction

- **History of PHP (cont.)**
  - Version 5.0 includes version 2.0 of the Zend Engine
    - New object model is more powerful and intuitive
    - Objects will no longer be passed by value; they now will be passed by reference
    - Increases performance and makes OOP more attractive

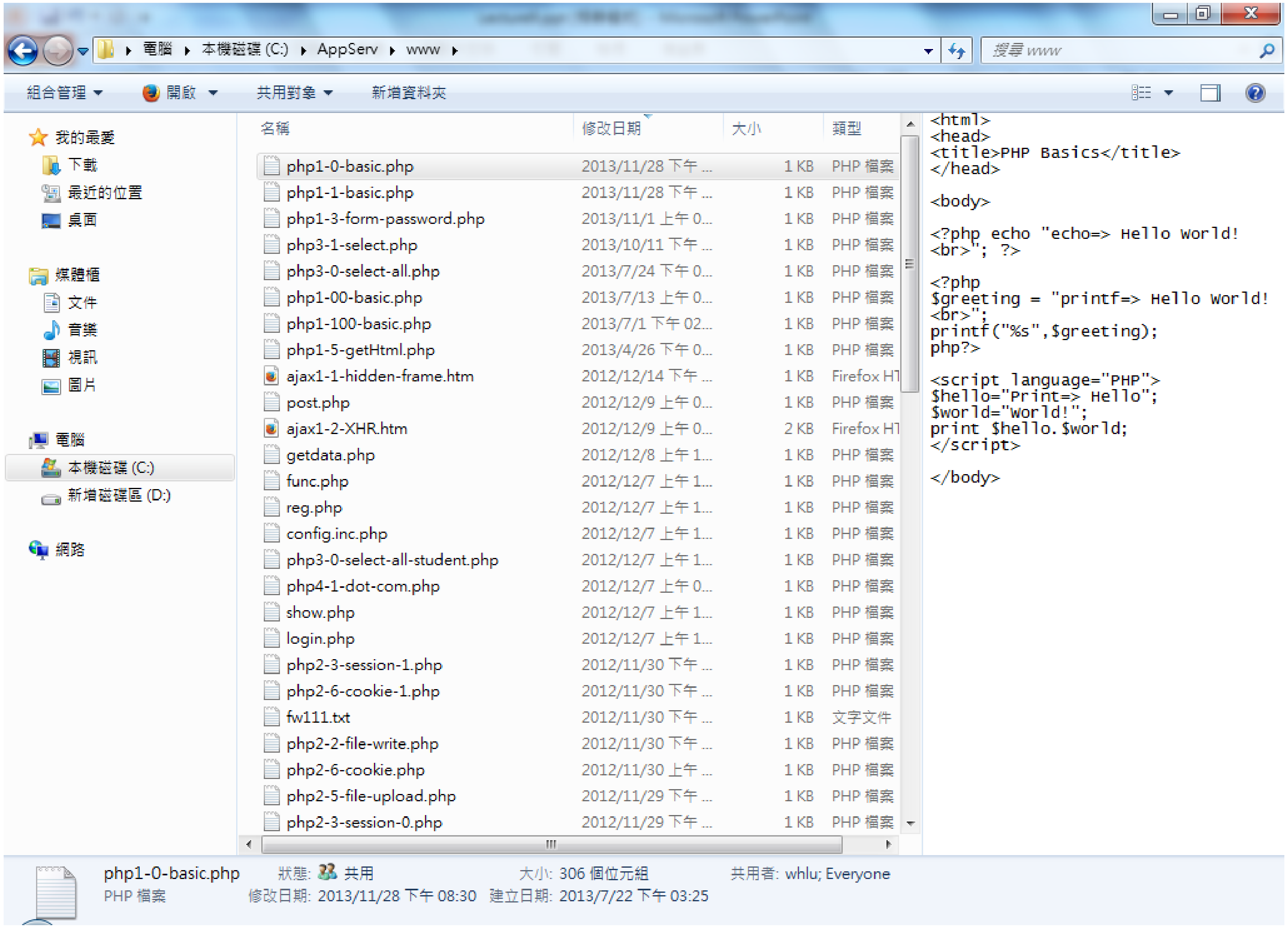
Resources

- **PHP Downloads and Online Documentation**
  - www.php.net
- **Community**
  - www.phpbuilder.com: articles on PHP, discussion forums
  - www.phpresourceindex.com: over 1,000 PHP scripts
  - www.phpvolcano.com: PHP 5 information
- **Newsgroups**
  - comp.lang.php

(1) Start Apache web server via AppServe



Web pages stored in the web server directory：C:/AppServ/www/

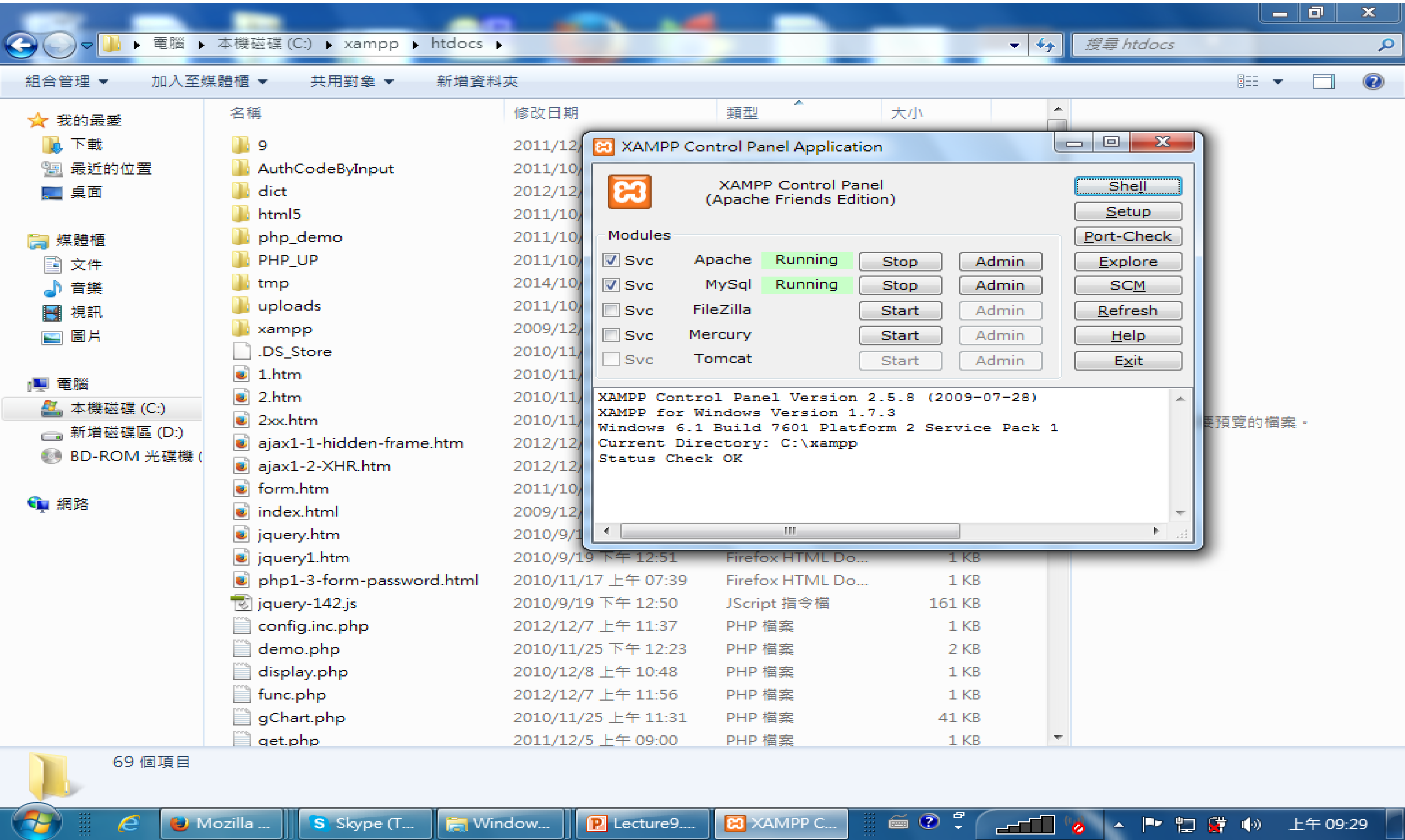


名稱	修改日期	大小	類型
php1-0-basic.php	2013/11/28 下午 ...	1 KB	PHP 檔案
php1-1-basic.php	2013/11/28 下午 ...	1 KB	PHP 檔案
php1-3-form-password.php	2013/11/1 上午 0...	1 KB	PHP 檔案
php3-1-select.php	2013/10/11 下午 ...	1 KB	PHP 檔案
php3-0-select-all.php	2013/7/24 下午 0...	1 KB	PHP 檔案
php1-00-basic.php	2013/7/13 上午 0...	1 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 HT
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 HT
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>
```

php1-0-basic.php 狀態: 共用 大小: 306 個位元組 共用者: whlu; Everyone
PHP 檔案 修改日期: 2013/11/28 下午 08:30 建立日期: 2013/7/22 下午 03:25

(2) Start Apache web server via Xampp
* Web pages stored in the web server directory：
C:/xampp/htdocs/



名稱	修改日期	類型	大小
9	2011/12/		
AuthCodeByInput	2011/10/		
dict	2012/12/		
html5	2011/10/		
php_demo	2011/10/		
PHP_UP	2011/10/		
tmp	2014/10/		
uploads	2011/10/		
xampp	2009/12/		
.DS_Store	2010/11/		
1.htm	2010/11/		
2.htm	2010/11/		
2xx.htm	2010/11/		
ajax1-1-hidden-frame.htm	2012/12/		
ajax1-2-XHR.htm	2012/12/		
form.htm	2011/10/		
index.html	2009/12/		
jquery.htm	2010/9/1		
jquery1.htm	2010/9/19 下午 12:51	Firefox HTML Do...	1 KB
php1-3-form-password.html	2010/11/17 上午 07:39	Firefox HTML Do...	1 KB
jquery-142.js	2010/9/19 下午 12:50	JScript 指令檔	161 KB
config.inc.php	2012/12/7 上午 11:37	PHP 檔案	1 KB
demo.php	2010/11/25 下午 12:23	PHP 檔案	2 KB
display.php	2010/12/8 上午 10:48	PHP 檔案	1 KB
func.php	2012/12/7 上午 11:56	PHP 檔案	1 KB
gChart.php	2010/11/25 上午 11:31	PHP 檔案	41 KB
get.php	2011/12/5 上午 09:00	PHP 檔案	1 KB

XAMPP Control Panel Application

XAMPP Control Panel
(Apache Friends Edition)

Modules

☑ Svc	Apache	Running	Stop	Admin
☑ Svc	MySql	Running	Stop	Admin
☐ Svc	FileZilla		Start	Admin
☐ Svc	Mercury		Start	Admin
☐ Svc	Tomcat		Start	Admin

```
XAMPP Control Panel Version 2.5.8 (2009-07-28)
XAMPP for Windows Version 1.7.3
Windows 6.1 Build 7601 Platform 2 Service Pack 1
Current Directory: C:\xampp
Status Check OK
```

PHP Language Basics

- **The Script Tags**
  - All PHP code is contained in one of several script tags:

    -
      ```
      <?
      // Some code
      ?>
      ```

    -
      ```
      <?php
      // Some code here
      ?>
      ```

    -
      ```
      <script language=“PHP">
          // Some code here
      </script>
      ```

Print Strings

- Like Perl, there is more than one way to do it
  - <?php **echo** “Hello World!”; ?>

  -
    ```
    <?php
        $greeting = “Hello World!”
        printf(“%s”, $greeting);
    ?>
    ```

  -
    ```
    <script language=“PHP”>
        $hello = “Hello”;
        $world = “World!”;
        print $hello . $world
    </script>
    ```
  - **[Ex]**php1-0-basic.php, php1-1-basic.php

Constants

- Constants define a string or numeric value
- Constants do not begin with a dollar sign
- Examples:
  - define(“**COMPANY**”, “Acme Enterprises”);
  - define(“YELLOW”, “#FFFF00”);
  - define(“PI”, 3.14);
  - define(“NL”, “<br>\n”);
- Using a constant
  - print(“Company name: “ **. COMPANY .** NL);

Data Types

- Integers, doubles and strings
  - isValid = true; // Boolean
  - 25   // Integer
  - 3.14   // Double
  - ‘Four’   // String
  - “Total value”   // Another string

Data Types

- Strings and type conversion
  - $street = 123;
  - $street = $street . “ Main Street”;
  - $city = ‘Naperville’;
  - $state = ‘IL’;
  - $address = $street;
  - $address = $address . NL . “$city, $state”;
  - $number = $address + 1; // $number equals 124

Data Types

- Arrays
  - Perl-like syntax
    - $arr = array("foo" => "bar", 12 => true);
  - same as
    - $arr[“foo”] = “bar”;
    - $arr[12] = true;

Data Types

- Arrays (cont.)

  -
    ```
    <?php
        $arr = array("somearray" => array(6 => 5, 13 => 9,
                                "a" => 42) );

        echo $arr["somearray"][6];     // 5
        echo $arr["somearray"][13];   // 9
        echo $arr["somearray"]["a"];  // 42
    ?>
    ```

Data Types

- Objects
  - Currently not much more advanced than associative arrays
  - Before version 5.0, objects are passed by value
    - Slow
    - Functions can not easily change object variables

Operator & Statement

- Operators
  - Contains all of the operators like in C and Perl (even the ternary)
- Statements
  - if, if/elseif
  - Switch/case
  - for, while, and do/while loops
  - Include and require statements for code reuse

Built-in Functions

- Array Manipulator Functions
  - sort, merge, push, pop, slice, splice, keys, count
- Date and Time Functions
  - getdate, mkdate, date, gettimeofday, localtime, strtotime, time
- String Functions
  - strcmp
  - similar_text
  - md5 => (encoding)
  - **`[Ex]`**[**`php1-2-string.php`**](php1-2-string.php)

Form Handling

- In PHP, form handler is used to deal with HTML form via
  - _GET and _POST arrays
  - **[Ex]**[php1-3-form-password.html]

php1-3-form-password.html	php1-3-form-password.php
<form action="http://localhost/php1-3-form-password.php" method=get> ..... <input type=text name="uname"> <input type=password name="upwd"> <input type=submit value="OK"> ...... </form>	<body> <?php echo "Hello $_GET[uname]!"; ?> ...... <?php echo $_GET['upwd']; ?> </body>

Class & Object

- **[Ex]php1-4-class.php**

Define class

```php
<?php
class cart{
  var $owner;
  var $product;

  function get_owner(){
    return $this->owner;
  }
  function buy_product($item, $number){
    $this->product[$item]= $number;
  }
} //cart
?>
```

Generate object

```php
<?php
  $ucart= new cart;
  $ucart->owner= $_POST['uname'];

  $pitem= $_POST['pname'];
  $pnumber= $_POST['number'];
  $ucart->buy_product($pitem,$pnumber);

  echo $ucart->get_owner();
  echo $ucart->product[$pitem]." ".$pitem."s!";
?>
```

Built-in Functions

- Directory Functions
  - Platform independent
- Error Handling Functions
  - Recover from warnings and errors
- File system Functions
  - Access flat files
  - Check directory, link, and file status information
  - Copy, delete, and rename files

Built-in Functions

- IMAP (Internet Message Access Protocol) Functions
  - Manipulate mail boxes via the IMAP protocol
- LDAP (Lightweight Directory Access Protocol) Functions
  - Works with most LDAP servers
- Mail Functions
  - mail($recipient, $subject, $message)

Built-in Functions

- Database Functions
  - dba: dbm-style abstraction layer
  - dBase
  - Frontbase
  - Informix
  - Ingres II
  - Interbase
  - mSQL

Built-in Functions

- Database Functions (cont.)
  - MySQL
  - Oracle
  - PostgreSQL
  - SQL Server
- MING
  - Macromedia Flash
- PDF
  - Create/manipulate PDF files dynamically

Built-in Functions

- POSIX Functions
  - Manipulate process information
- Regular Expression Functions
  - Uses POSIX regex
- Semaphore and Socket Functions
  - Available only on Unix
- Session Management Functions

Built-in Functions

- Date & Time Functions
  - time(): return time stamp in seconds
    - 1290610056
  - microtime(); return the current time in seconds and microseconds
    - 0.65625300 1290610056
  - date(format, timestamp):format a timestamp
    - date(‘Y-M-d’) => 2010-Nov-24
  - getdate(timestamp): return date information
    - $date1=getdate(time());
    - echo $date1['year'].$date1['month'].$date1['mday'];
  - **[Ex]**php1-5-date.php

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

- ^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.

Table 22-2: Common Perl-Compatible Pattern Constructs

Construct	*Interpretation*
Simple literal character matches	If the character involved is not special, Perl will match characters in sequence. The example pattern `/abc/` matches any string that has the substring '`abc`' in it.
Character class matches: [<*list of characters*>]	Will match a single instance of any of the characters between the brackets. For example, `/[xyz]/` 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 `[0-9]`.
Predefined character class abbreviations	The patterns `\d` will match a single digit (from the character class `[0-9]`), and the pattern `\s` matches any whitespace character.
Multiplier patterns	Any pattern followed by `*` means: "Match this pattern 0 or more times." Any pattern followed by `?` means: "Match this pattern exactly once." Any pattern followed by `+` means: "Match this pattern 1 or more times."
Anchoring characters	The caret character `^` at the beginning of a pattern means that the pattern must start at the beginning of the string; the `$` 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 `[^abc]` means that the set is the complement of the characters listed (that is, any character that is not in the list).
Escape character '\'	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: `. \ + * ? [ ] ^ $ ( ) { } = ! < > | :`
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

- 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.

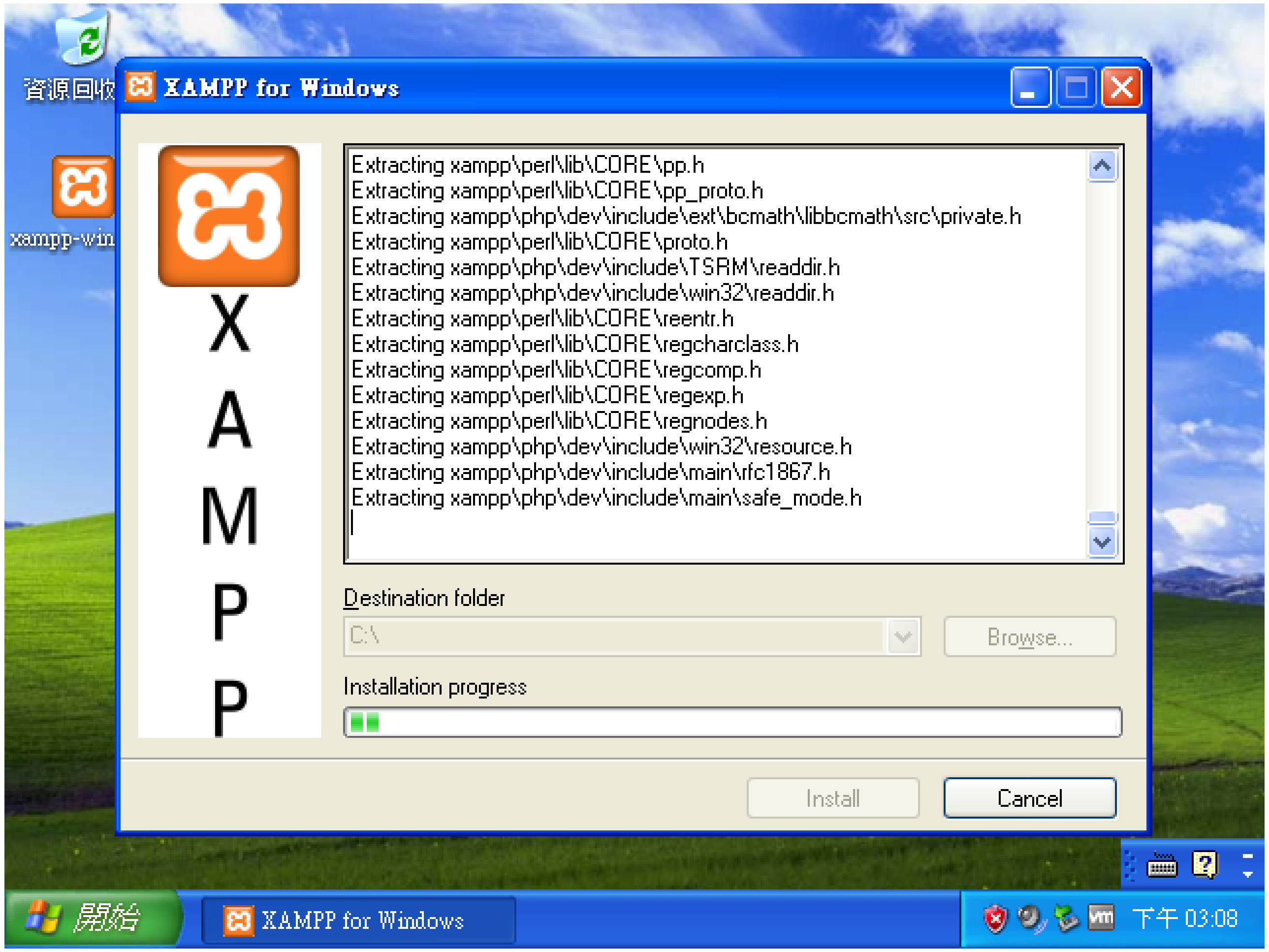
Regular Expression

```
<?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;
  }
 $match_result =  preg_match_all(‘ /<\s*A\s*HREF=”([^\”]+)”\s*>([^>]*)<\/A>/I ’,$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

Function	*Behavior*
`preg_match()`	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 `PREG_OFFSET_CAPTURE` is also available. This flag causes preg match 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.
`preg_match_all()`	Like `preg_match()`, 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 `preg_match()`). The structure of the array returned depends on the optional fourth argument (either the constant `PREG_PATTERN_ORDER`, or `PREG_SET_ORDER`, defaulting to the former). (See further discussion following the table.) `PREG_OFFSET_CAPTURE` is also available with this function.

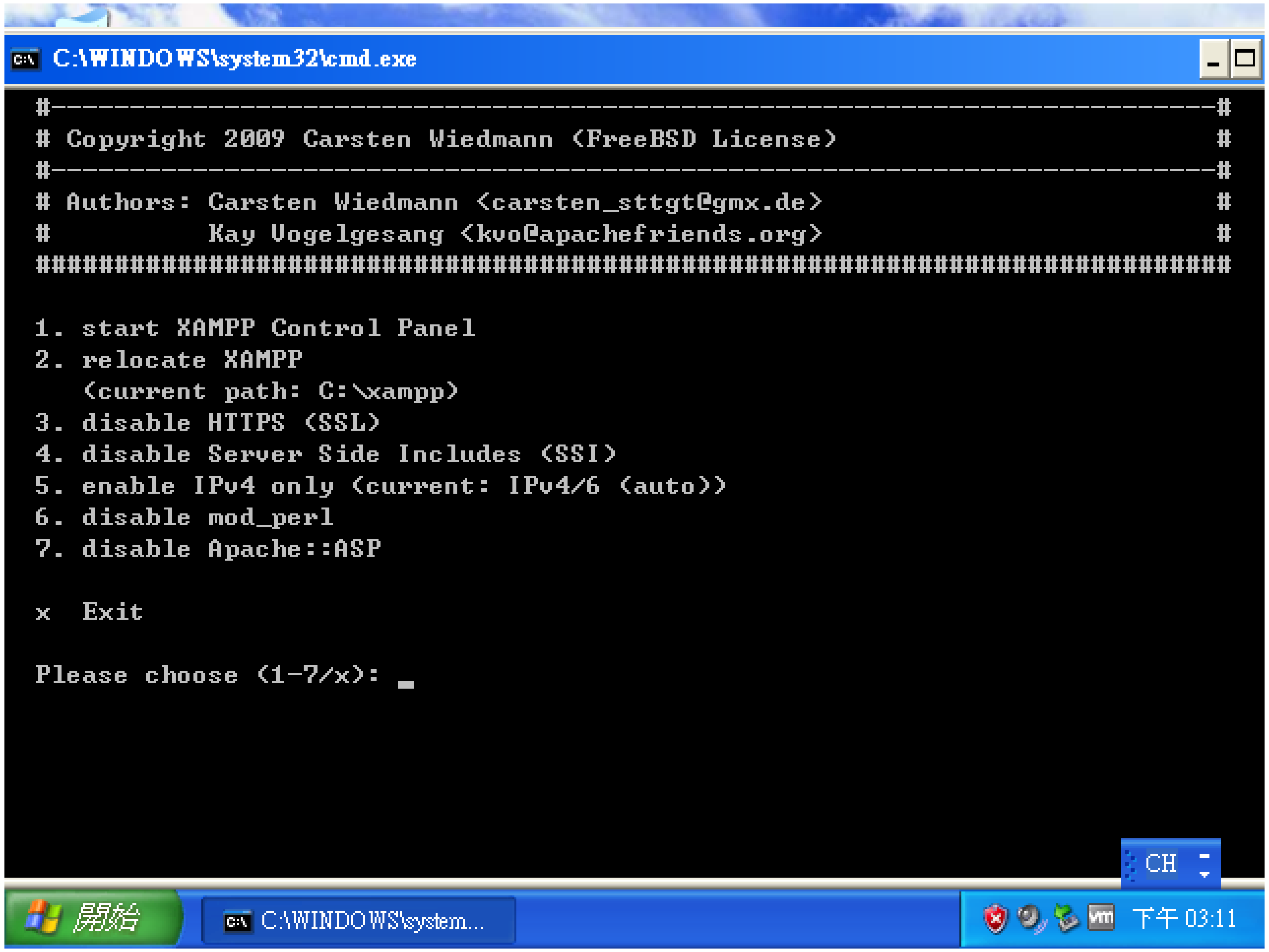
Install XAMPP PhpMyAdmin



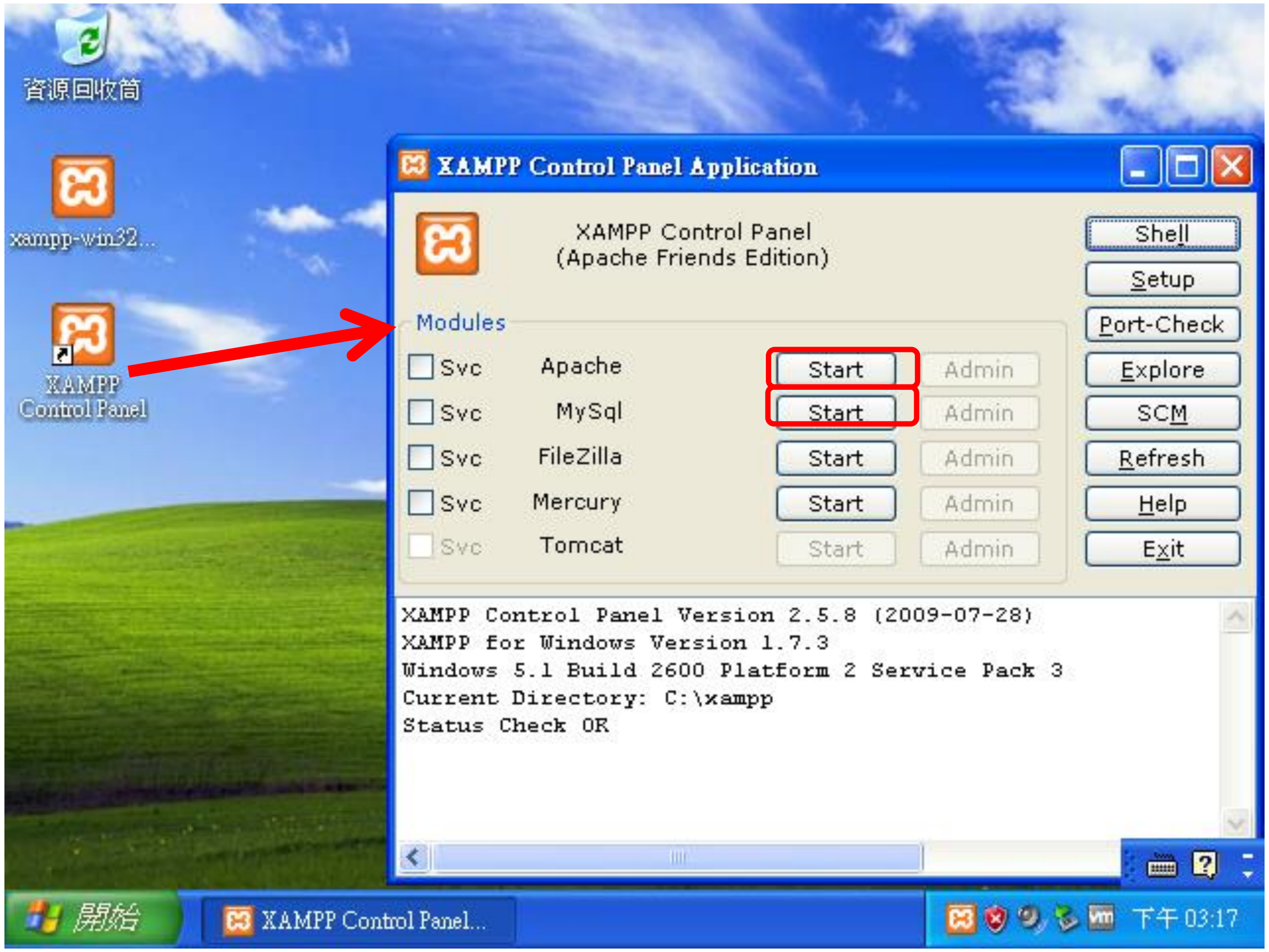
```
Extracting xampp\perl\lib\CORE\pp.h
Extracting xampp\perl\lib\CORE\pp_proto.h
Extracting xampp\php\dev\include\ext\bcmath\libbcmath\src\private.h
Extracting xampp\perl\lib\CORE\proto.h
Extracting xampp\php\dev\include\TSRM\readdir.h
Extracting xampp\php\dev\include\win32\readdir.h
Extracting xampp\perl\lib\CORE\reentr.h
Extracting xampp\perl\lib\CORE\regcharclass.h
Extracting xampp\perl\lib\CORE\regcomp.h
Extracting xampp\perl\lib\CORE\regexp.h
Extracting xampp\perl\lib\CORE\regnodes.h
Extracting xampp\php\dev\include\win32\resource.h
Extracting xampp\php\dev\include\main\rfc1867.h
Extracting xampp\php\dev\include\main\safe_mode.h
```

Destination folder: C:\

數個類似以下畫面，請依預設值
最後來至以下畫面，請按x

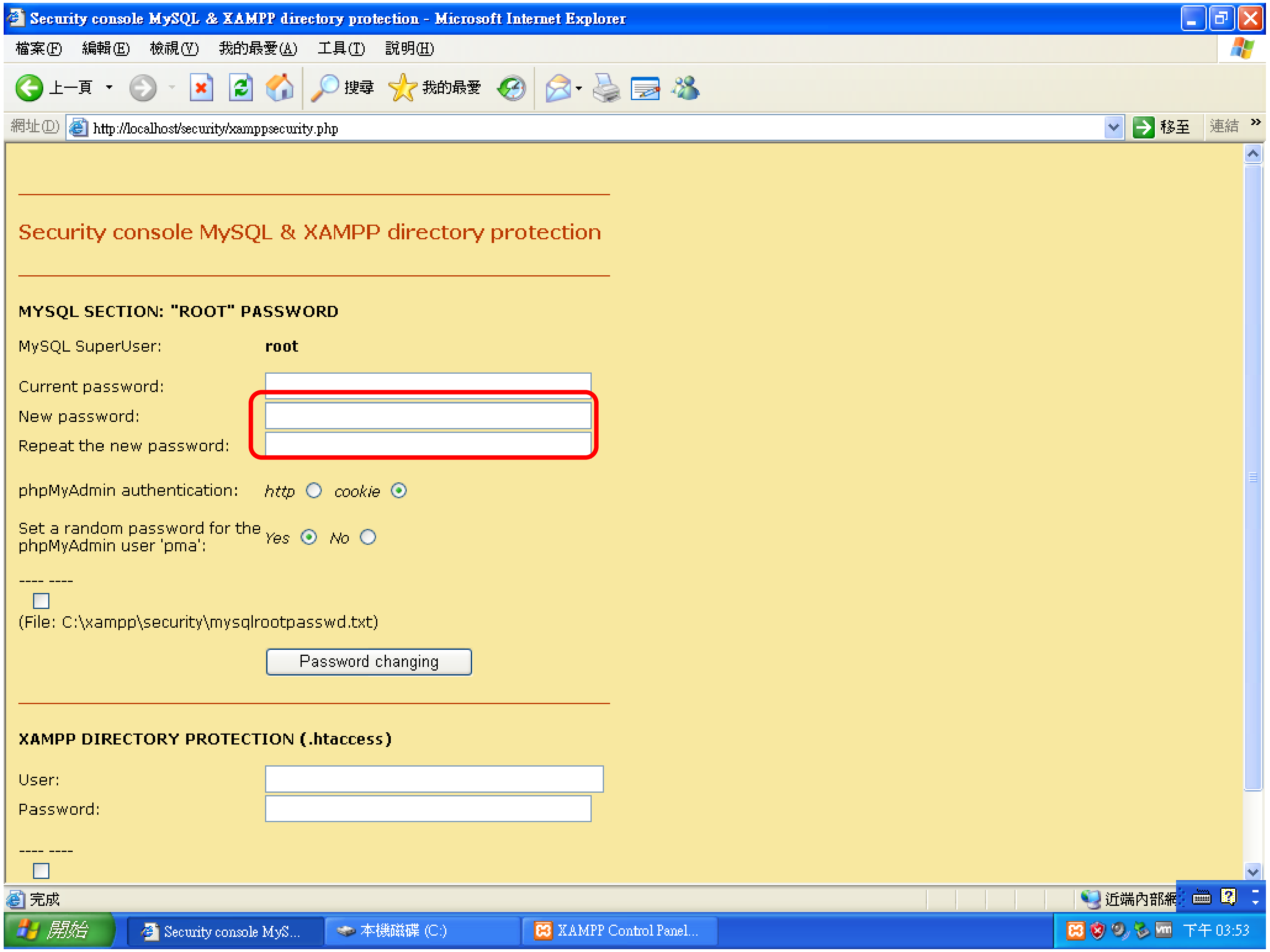


```
#---------------------------------------------------------------------------#
# Copyright 2009 Carsten Wiedmann (FreeBSD License)                         #
#---------------------------------------------------------------------------#
# Authors: Carsten Wiedmann <carsten_sttgt@gmx.de>                          #
#          Kay Vogelgesang <kvo@apachefriends.org>                          #
#############################################################################

1. start XAMPP Control Panel
2. relocate XAMPP
   (current path: C:\xampp)
3. disable HTTPS (SSL)
4. disable Server Side Includes (SSI)
5. enable IPv4 only (current: IPv4/6 (auto))
6. disable mod_perl
7. disable Apache::ASP

x  Exit

Please choose (1-7/x): _
```



```
XAMPP Control Panel Version 2.5.8 (2009-07-28)
XAMPP for Windows Version 1.7.3
Windows 5.1 Build 2600 Platform 2 Service Pack 3
Current Directory: C:\xampp
Status Check OK
```

http://localhost/security/xamppsecurity.php
只需填紅框中的資料


Security console MySQL & XAMPP directory protection - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛
網址(D) http://localhost/security/xamppsecurity.php
移至 連結
完成
近端內部網路
開始 Security console MyS... 本機磁碟 (C:) XAMPP Control Panel...
下午 03:53

Security console MySQL & XAMPP directory protection

MYSQL SECTION: "ROOT" PASSWORD

MySQL SuperUser:	**root**
Current password:	
New password:	
Repeat the new password:	
phpMyAdmin authentication:	*http* ○ [✓ *cookie*]
Set a random password for the phpMyAdmin user 'pma':	[✓ *Yes*] *No* ○

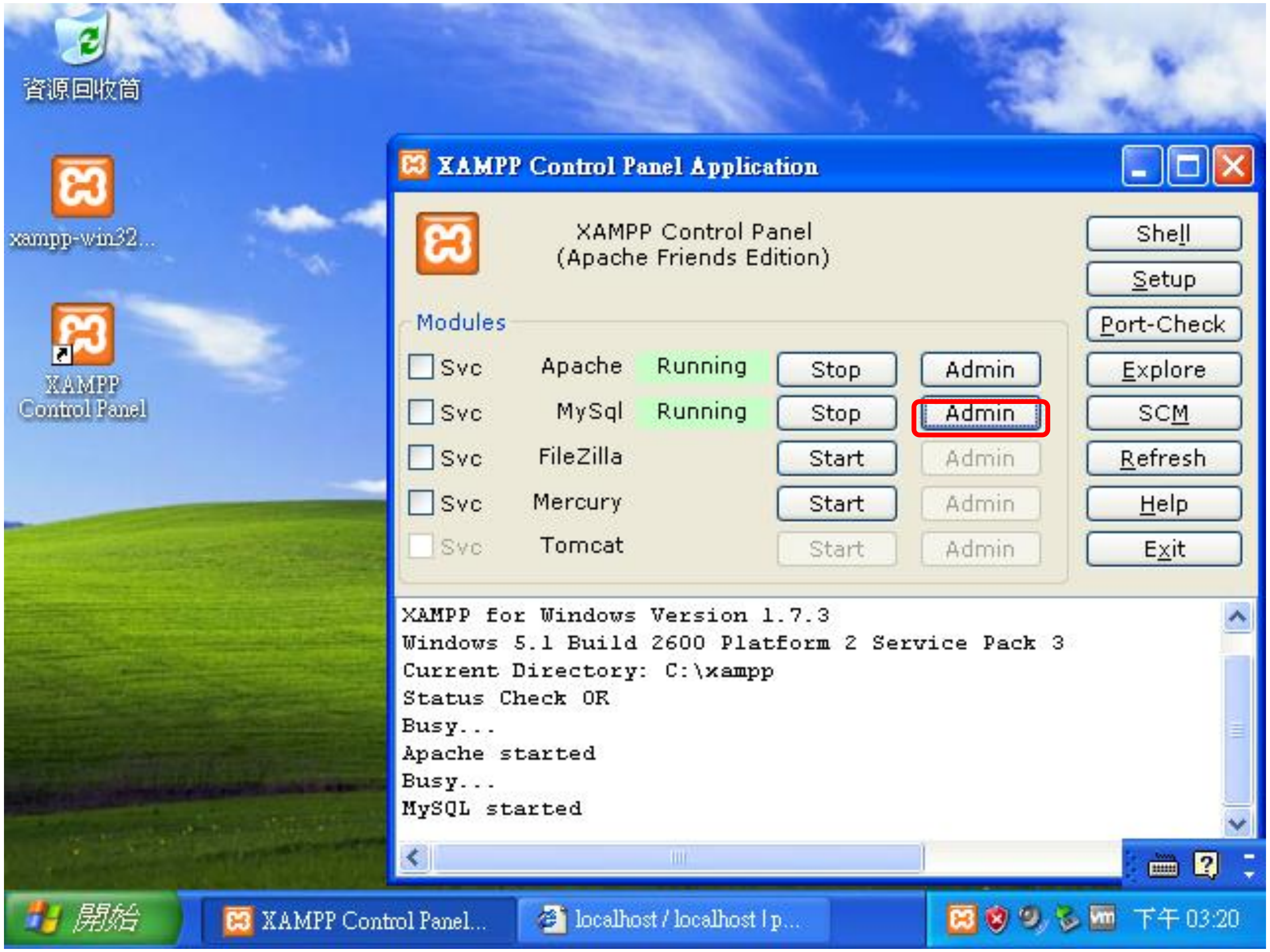
---- ----
☐
(File: C:\xampp\security\mysqlrootpasswd.txt)

Password changing

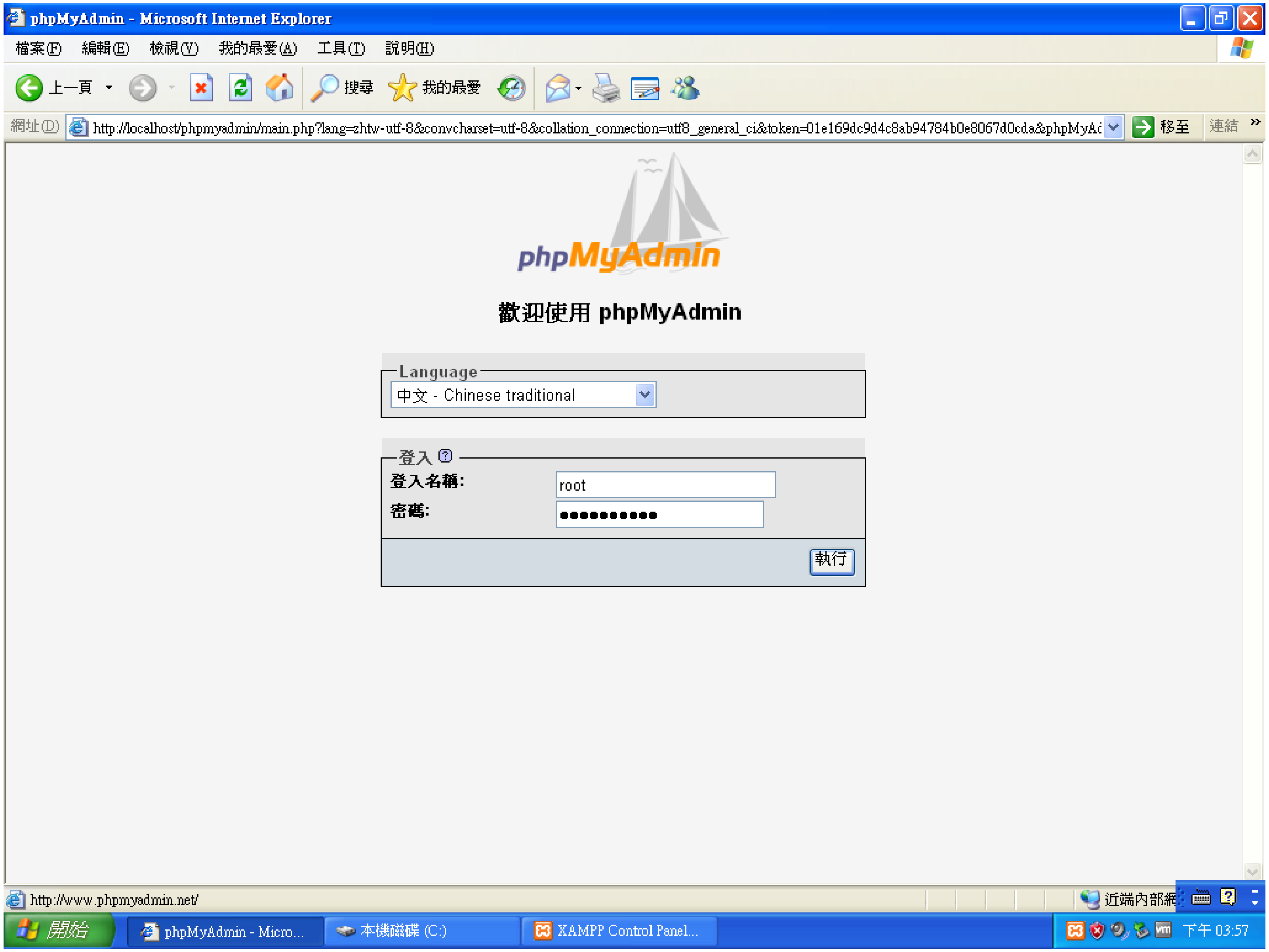
XAMPP DIRECTORY PROTECTION (.htaccess)

User:	
Password:	

---- ----
☐



```
XAMPP for Windows Version 1.7.3
Windows 5.1 Build 2600 Platform 2 Service Pack 3
Current Directory: C:\xampp
Status Check OK
Busy...
Apache started
Busy...
MySQL started
```



phpMyAdmin - Microsoft Internet Explorer

http://localhost/phpmyadmin/main.php?lang=zhtw-utf-8&convcharset=utf-8&collation_connection=utf8_general_ci&token=01e169dc9d4c8ab94784b0e8067d0cda&phpMyAd

歡迎使用 phpMyAdmin

Language: 中文 - Chinese traditional

登入

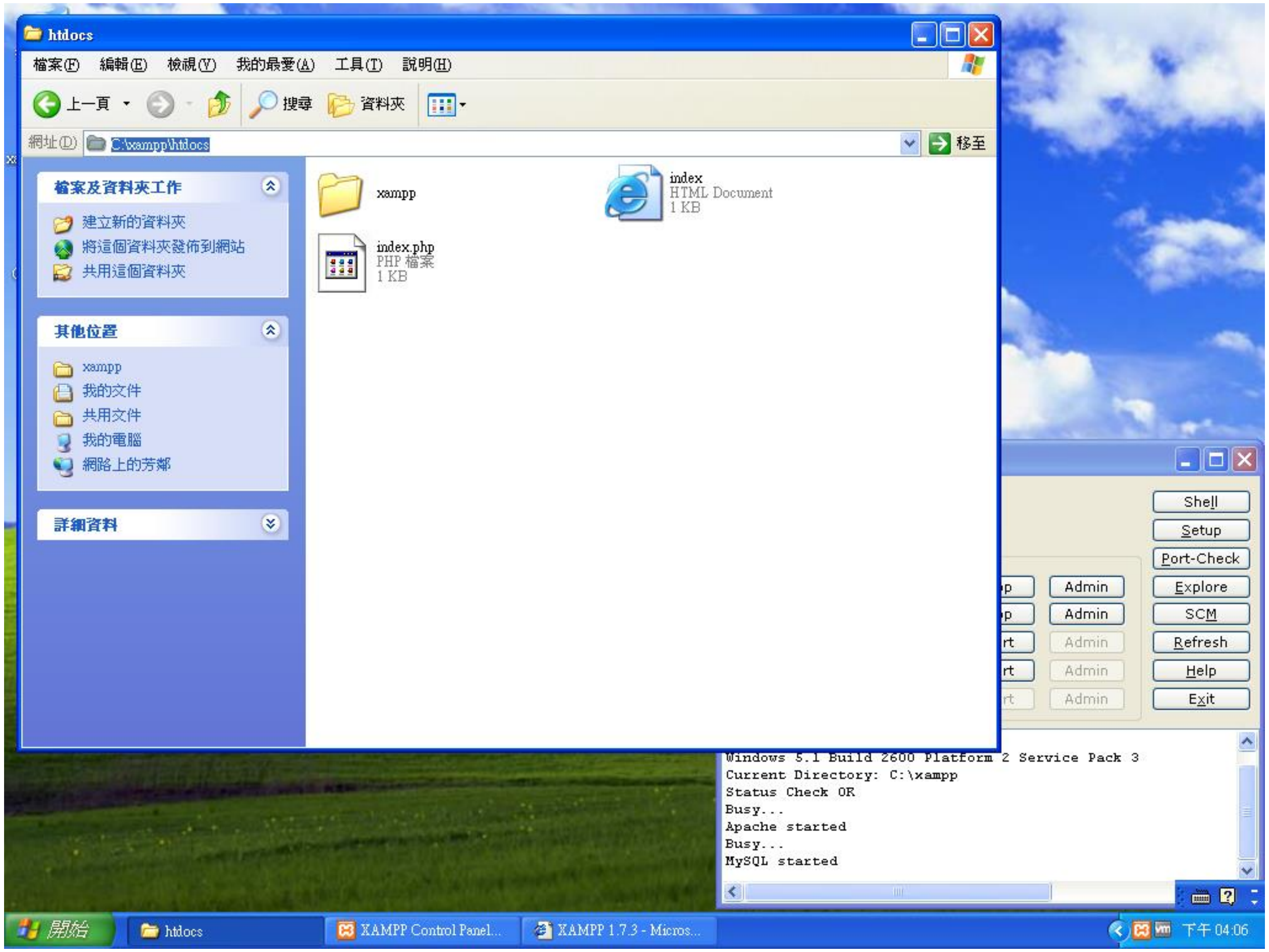
登入名稱: root

密碼: ●●●●●●●●●●●

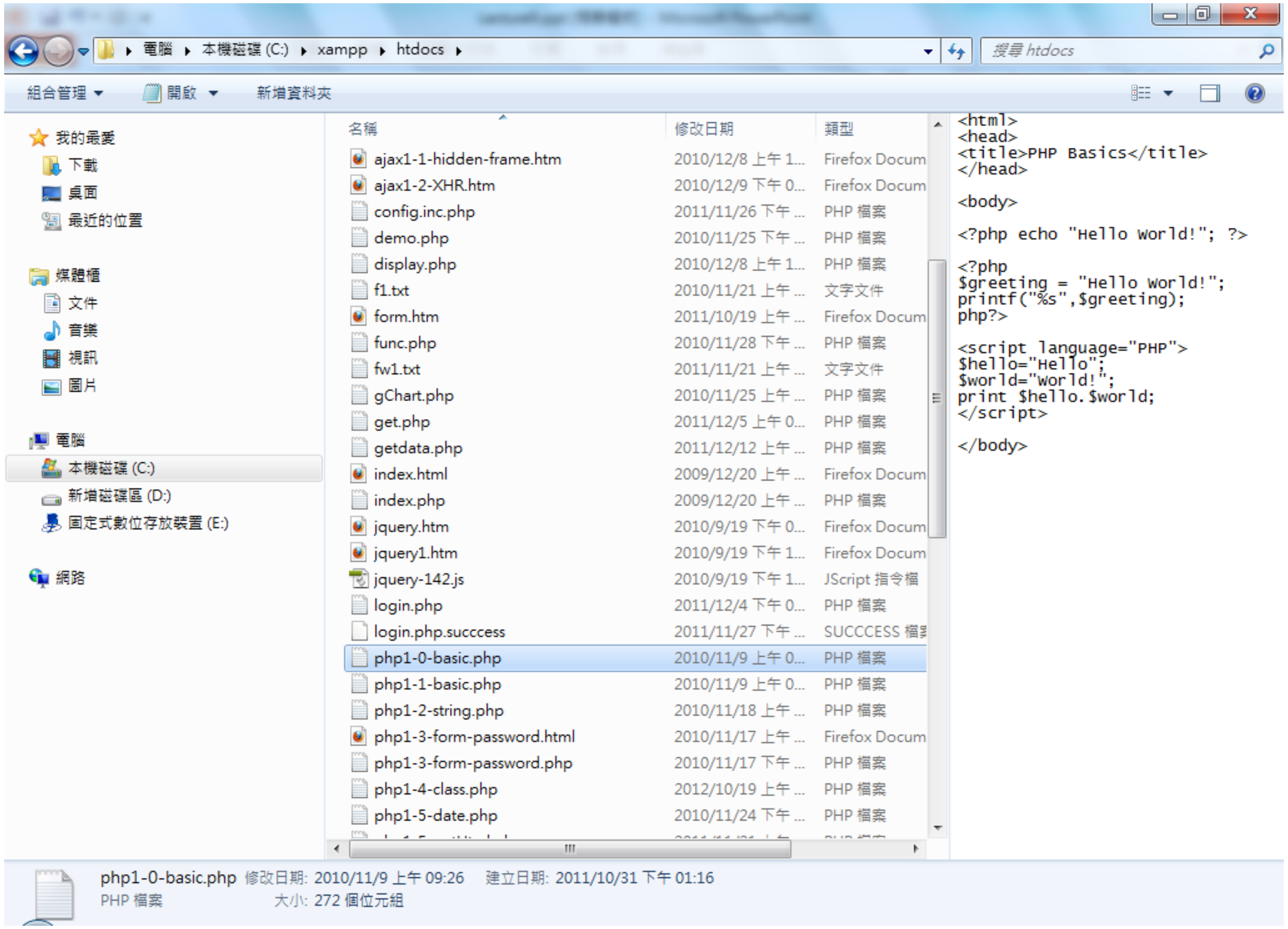
執行

http://www.phpmyadmin.net/

網頁根目錄：C:\xampp\htdocs



```
Windows 5.1 Build 2600 Platform 2 Service Pack 3
Current Directory: C:\xampp
Status Check OK
Busy...
Apache started
Busy...
MySQL started
```



名稱	修改日期	類型
ajax1-1-hidden-frame.htm	2010/12/8 上午 1...	Firefox Docum
ajax1-2-XHR.htm	2010/12/9 下午 0...	Firefox Docum
config.inc.php	2011/11/26 下午 ...	PHP 檔案
demo.php	2010/11/25 下午 ...	PHP 檔案
display.php	2010/12/8 上午 1...	PHP 檔案
f1.txt	2010/11/21 上午 ...	文字文件
form.htm	2011/10/19 上午 ...	Firefox Docum
func.php	2010/11/28 下午 ...	PHP 檔案
fw1.txt	2011/11/21 上午 ...	文字文件
gChart.php	2010/11/25 上午 ...	PHP 檔案
get.php	2011/12/5 上午 0...	PHP 檔案
getdata.php	2011/12/12 上午 ...	PHP 檔案
index.html	2009/12/20 上午 ...	Firefox Docum
index.php	2009/12/20 上午 ...	PHP 檔案
jquery.htm	2010/9/19 下午 0...	Firefox Docum
jquery1.htm	2010/9/19 下午 1...	Firefox Docum
jquery-142.js	2010/9/19 下午 1...	JScript 指令檔
login.php	2011/12/4 下午 0...	PHP 檔案
login.php.succcess	2011/11/27 下午 ...	SUCCCESS 檔
php1-0-basic.php	2010/11/9 上午 0...	PHP 檔案
php1-1-basic.php	2010/11/9 上午 0...	PHP 檔案
php1-2-string.php	2010/11/18 上午 ...	PHP 檔案
php1-3-form-password.html	2010/11/17 上午 ...	Firefox Docum
php1-3-form-password.php	2010/11/17 下午 ...	PHP 檔案
php1-4-class.php	2012/10/19 上午 ...	PHP 檔案
php1-5-date.php	2010/11/24 下午 ...	PHP 檔案

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

<body>

<?php echo "Hello World!"; ?>

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

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

</body>
```

php1-0-basic.php 修改日期: 2010/11/9 上午 09:26 建立日期: 2011/10/31 下午 01:16
PHP 檔案 大小: 272 個位元組