第四章 網頁與資料庫-以 PHP & MySQL為例

1

章節大綱

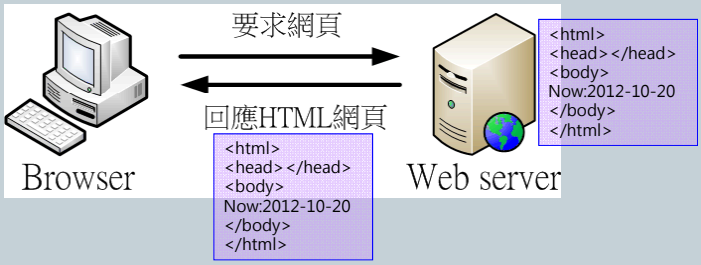
2

- **4-1 PHP動態網頁**
- **4-2 PHP基本語法**
- **4-3 MySQL資料庫**
- **4-4 SQL**

4-1 PHP動態網頁

3

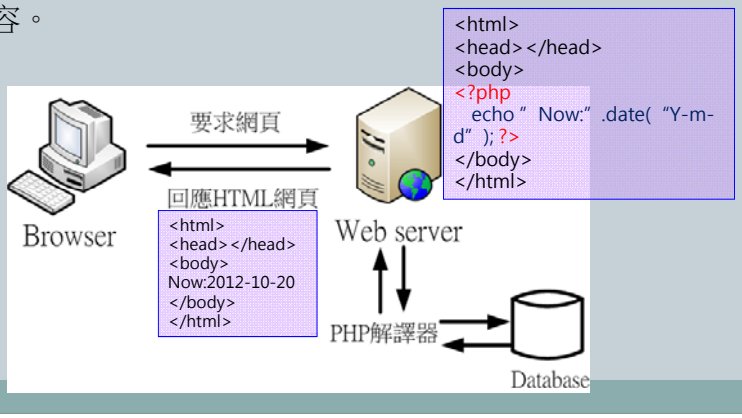
- 傳統網頁的執行方式：
- 要求與傳回的網頁，都是純HTML網頁。
- 網頁內容事先寫好，無法依情況向伺服器要求傳送不同的內容。



PHP動態網頁的執行方式

4

- PHP動態網頁的執行方式：
- 要求PHP網頁，伺服器端處理完後，傳回純HTML網頁。
- 網頁內容事先寫好，無法依情況向伺服器要求傳送不同的內容。



PHP的特色

5

- 跨平台。
- 免費的開放原始碼。
- 支援物件導向。
- 功能強大，支援各種資料庫、網路連結、檔案系統、Java、...等。
- 快速發展週期及支援社群廣泛。

建構 Apache +PHP +MySQL環境

6

- Linux 平台：建議以LAMP安裝包方式安裝。LAMP指一組常用來執行動態網頁的軟體組合。這些開放原始碼程式，並非特意設計成要一同工作。但這是一個普遍且流行的組合。
  - Linux：作業系統。
  - Apache：網頁伺服器軟體。
  - MySQL：資料庫管理系統或資料庫伺服器軟體。
  - PHP：伺服端網頁程式語言。
  - phpMyAdmin：支援MySQL的圖形操作介面。
- Windows 平台：建議以XAMP安裝包（XAMP指一組常用來執行動態網頁的軟體組合）或 AppServ 套件安裝 。
  - XML：可延伸標記語言，可跨平台。
  - Apache：網頁伺服器軟體。
  - MySQL：資料庫管理系統或資料庫伺服器軟體。
  - PHP：伺服端網頁程式語言。
  - phpMyAdmin：支援MySQL的圖形操作介面。

PHP編輯器

7

- PSPad。
- UltraEdit。
- Dreamweaver。
- Topstyle。
- Homesite。
- Zend Studio。
- ...。

PHP基本語法

8

- 第一個 PHP 程式：

```
<html>
<head><title>實作第一個PHP程式</title></head>
<body>
<?php // 表示PHP程式碼開始的指令。
  echo  '實作第一個PHP程式'; // 將echo後面的訊息輸出
                              //每行以「;」做結尾。
?>
</body>
</html>
```

- <?php ... ?> 表示php程式碼開始與結束的標籤。

PHP的標籤寫法

9

- 將PHP程式碼嵌入HTML中的標籤寫法有四種：
  - <?php
    echo “第一種";
    ?>
  - <?
    echo “第二種";
    ?>
  - <script language="php">
    echo “第三種";
    </script>
  - <%
    echo “第四種”;
    %>

第四種會與asp搞混，且必須將**php.ini**中的**asp_tags**設定為On才可以使用。不建議使用。

PHP的註解

10

- 最常用的註解型式同C/C++一樣。有單行註解的「//」，及多行註解的「/* ... */」。另有perl型式的單行註解「#」以及「=開始標籤 ... =結束標籤」的整段註解。
- <?php
  // 單行註解，雙斜線之後該行是註解, 不會被顯示出來
  /*
  多行註解
  ‘在/* 與 */ 之間為註解’;
  */
  ?>

變數與常數

11

- 變數：內容會變動的一小塊記憶體空間。在執行階段可根據情況變動其值。
- 變數的命名：
  - 以「$」開頭，後接變數名稱。
  - 變數名稱以英文字母或「_」底線啟始，之後可接文字字元、數字或底線。
  - 第一個字元不可使用數字。
  - 變數名稱區分大小寫： 如，$a1 與$A1是不同變數。
  - 不能以保留字當作變數名稱。
- 定義變數時不用設定資料型別，編輯時根據所輸入的值自動判斷資料型別。是一種弱資料型態程式語言。
- 常數：內容不會變動的一小塊記憶體空間。
  - define (常數名稱, 常數值);
- 常數的特性：
  - 常數名稱不需要「$」。常數名稱區分大小寫，通常以全大寫代表常數。
  - 只能使用define()來定義。
  - 常數只接受基本資料型態。
  - 常數定義後就不能更改。

PHP 與 HTML

12

嵌入HTML：

```
<html><head>
<title>實作我的PHP程式</title></head><body>
<?php
  echo  “實作我的PHP程式”；
?>
</body>
</html>
```

在HTML標籤外：

```
<?php
  echo “實作我的PHP程式"；
?>
<html><head>
<title>實作我的PHP程式</title></head>
<body>
</body>
</html>
```

單獨存在：

```
<?php
  echo  “實作我的PHP程式";
?>
```

一些資料型態

13

名稱	型態	範例
Integer	整數	$a = 117;
Float	浮點	$a = 3.34;
Boolean	布林	$a = true;
String	字串	$a = “臺南市”;
Array	陣列	$a[0] = “永康區”;
Object	物件	$a = new Class;
Resource	資源	$a = mysql_pcconnect();
Null	Null	$a = Null;

整數型態

14

- 整數範圍如同C語言一樣，取決於電腦的字組大小。以32位元的電腦為例，整數所佔的記憶體為32位元，整數範圍為：-2^{31}~2^{31}-1語法。
- 可使用8進位、10進位、16進位整數。
- 使用超過範圍的整數時，PHP會自動將資料型態轉換為浮點數資料型態。
- 範例：
  - $a = 369; //正整數
  - $b = 065; //八進位整數
  - $c = 0x3A; //十六進位整數

浮點數型態

15

- 浮點數的表示範圍也是由電腦的字組大小所決定。以32位元的電腦為例，浮點數所佔的記憶體為32位元，正浮點數的最大值為：1.8E+308。並有14位10進位數字的精確度。
- 科學符號：E沒有大小寫之分。
- 範例：
  - 1.357
  - 1.35e2 // 科學記號

布林資料型態

16

- 只能表示True（真）、False（偽），無大小寫之分。
  - $a = true;
  - $a = FALSE;
- 當所要表示的資料只有2種狀況時，可使用布林型態。通常用來判斷運算式是否成立。
  - if($a) { // 若$a為true，則執行 }
- 將布林型態的資料轉換成數值型態的資料時，true會變成1，false會變成0。
- 將布林型態的資料轉換成字串型態的資料時，true會變成「”1”」，false會變成空字串「””」。
- 將其他資料型態轉換成布林資料型態時，除以下情況會轉換成false外，其他情況皆會轉換成true。
  - 整數0、浮點數0.0。
  - 空字串””、字串”0”。
  - 無元素的陣列、無成員的物件。
  - NULL。

字串型態

17

- 字串是文字串列的組合，可用雙引號或單引號前後括號起來。

  $a = “我的PHP程式”;
- 在字串中使用到特殊符號需加個反斜線「\」，Escape字元（跳脫字元、命令字元）的用法，如：反斜線、單引號、$符號...，等。因這些符號都有預設的功用，要分辨是否為純文字，則以是否有加Escape字元來判定。

  $a = “I\’ll go home”;
- 字串使用雙引號，其內若有變數則會被變數內容所取代。可將變數用大括號{}包住，以方便PHP解析。
  - echo “\$a = $a”;
- 字串使用單引號，其內若有變數則不理變數直接輸出成字串。
  - echo ‘\$a = $a’;

陣列型態

18

- 陣列是一群變數的集合，陣列中的每一個元素各有自己的值，用索引值可參照到陣列中的元素。變數只能儲存一個值，若要使用大量的變數存放資料，是一件十分沒效率的事，此時，應該要使用陣列來處理。其中：
  - 陣列的索引值內定由0開始。
  - 同一陣列元素可為不同的資料型態。
- 範例：
  - $a[0] = 0;
  - $a[1] = 5;
  - $b[“國文”] = 80;

一維陣列

19

- 陣列只用一個名稱就有一段記憶體空間來儲存多個資料。
- 陣列存放的每個資料稱之為「元素」，元素的內容稱之為「元素值」或簡稱為「值」。陣列以索引值（鍵值）來區分及存取。
- 陣列的鍵值必需為整數或字串。
- 建立一維陣列：
  - 直接指派：陣列名稱[索引值] = 元素值;
  - 使用array()函式不指定鍵值：array(元素值1, 元素值2, ...);
  - 使用array()函式指定鍵值：array(鍵值1=>元素值1, 鍵值2=>元素值2, ...);
- 範例：
  - $a[0] = 10;
  - $b[] = “忠”;//未指明索引值，從現在最大值加1開始，若尚無鍵值將從0開始。
  - $c[“國文”] = 80;
  - $d=array(0, 10, 20); // 無指派鍵值，鍵值預設由0開始。
  - $d=array(‘甲’=>0, ‘乙’=>10, ‘丙’=>20); // 三個元素的鍵值分別為甲、乙、丙

二維陣列

20

- 建立二維陣列：
  - 直接指派：陣列名稱[索引值] [索引值] = 元素值;
  - 使用array()函式不指定鍵值：array(array(元素值1, 元素值2, ...), array(元素值1, 元素值2, ...), ...);
  - 使用array()函式指定鍵值： array(鍵值1=>array(鍵值1=>元素值1, 鍵值2=>元素值2, ...), 鍵值2=>array(鍵值1=>元素值1, 鍵值2=>元素值2, ...), ...);
- 範例：
  - $class[1][1] = “一年甲班”;
  - $class[1][2] = “一年乙班”;
  - $class[1][3] = “一年丙班”;
  - $class[2][1] = “二年甲班”;
  - $class[2][2] = “二年乙班”;
  - $class[2][3] = “二年丙班”;
  - $class=array(array(‘一年甲班’,’一年乙班’,’一年丙班’),
array(‘二年甲班’,’二年乙班’,’二年丙班’),
  - $class=array(1=>array(1=>’一年甲班’, 2=>’一年乙班’, 3=>’一年丙班’),
2=>array(1=>’二年甲班’, 2=>’二年乙班’, 3=>’二年丙班’),);

物件型態

21

- 物件可視為是一個容器。就是利用類別(Class)的規範，產生出來的物件實體。包含了與該物件相關的變數及函式。
- 物件的使用：
  - 宣告類別。
  - 產生物件。
  - 存取成員函數。

資源型態

22

- 資源（resource）型態：用來指向外部資源。如：檔案、資料庫、...等。在取存外部資源時自動建立資源型態。
- 如：建立MySQL連線，將資源指派給變數$conn。

Null 型態

23

- Null型態：表示變數只有一種常數值，就是Null值（Null無大小寫之分）。
- 變數未定義、變數被unset()清除後以及變數被設定為Null。此時變數為Null型態。

變數的初始值

24

- 變數初始值指派變數方式：等號右邊運算完後指派給等號左邊。

```
$a = 10;
```

- 不需事先指定變數型態，由初始值來決定初始的資料型態。程式執行中資料型態會自動轉換。

```
$a = 10;
$a = “你好”;
```

變數的種類

25

- 區域變數：
  - 在函式內宣告的變數為區域變數，可視範圍只限於函數中，在函式中若想使用全域變數，需將變數宣告成global。
- 全域變數：
  - 在PHP中函數外所宣告的變數。或宣告為global的變數。
- 靜態變數：
  - 在函式中宣告成為static的變數，在記憶體中佔有固定的空間，不隨函式結束而消失。重複呼叫該函式，其變數會保留上一次呼叫的值。

超級全域變數(1)

26

- 超級全域變數：為PHP內建的變數有：$GLOBALS、$_GET、$_POST、$_REQUEST 、$_SESSION、COOKIE、$_FILES、$_SERVER、$_ENV。
- 當操作網頁時，用來記錄與接收的變數
- $_GET：利用網址列來傳遞變數。
  - <?php echo $_GET['name']; ?>
- $_POST利用表單來傳遞變數。
  - <form action="test.php" method="post">
    <input type= "files" name="frm">
    <input type="submit" value="送出">
    </form>
  - <body>
    <?php echo $_POST['frm']; ? >
    </body>
- $_REQUEST:可接收來自GET與POST變數
  -

超級全域變數(2)

27

- 網頁伺服器在處理完要求後便會中斷連線，因此，網頁伺服器並未記錄客戶端資訊。
- 要達成記錄客戶端資訊可透過一些方法，如：
  - 檔案存取。
  - 表單處理。
  - Cookie。
  - Session。
- $_SESSION:用來記錄客戶端資料，每個客戶端都有自己的session，以便區別。Session的生命週期於瀏覽器開始至瀏覽器結束。
- Session包含2部份：
  - Session(ID)。
  - Session變數。

超級全域變數(3)

28

```
<HTML>
  <HEAD>  </HEAD>
  <BODY>
    <?php
     session_start();
     if (!isset($_SESSION['Count']))
       $_SESSION['Count'] = 1;
     else
       $_SESSION['Count']++;
     echo "本網頁第{$_SESSION['Count']}次載入。";
    ?>
  </BODY>
</HTML>
```

超級全域變數(4)

29

- PHP內建的Session相關函式：
  - session_start()
  - session_unset()
  - session_destory()
  - session_id()
  - session_name()
  - session_regenerate_id()
  - session_encode()
  - session_write_close()
  - session_decode()
  - session_save_path()
  - session_set_cookie_params()
  - session_get_cookie_params()
  - session_register()
  - session_unregister()
  - session_is_registered()

超級全域變數(5)

30

- Cookie :由伺服器在客戶端電腦中寫入一些小檔案，用來記錄客戶端一些資料。
- Cookie的優點：
  - 預設生命週期起始於瀏覽器開始，終止於瀏覽器結束。
  - 儲存的資料不會佔用網頁伺服器的資源。
  - 可記錄瀏覽者個人的資料，根據此資料網站可推出個別化的服務。
- Cookie的限制：
  - 每個使用者的瀏覽器只能支援300個Cookie。
  - 大部份瀏覽器對於同一個伺服器只能存取50個以下的Cookie，甚至更少。
  - 每個Cookie的大小約只有4k Bytes的容量 。
  - 使用者可以把瀏覽器的Cookie功能關掉，若關掉後Cookie就不能使用。
  - Cookie會被使用者自行刪除。
  - Cookie會增加個人資料被竊取的風險。

超級全域變數(6)

31

- <?php
  ob_start();
  setcookie(“UserName”, “李一一", time() + 24 * 60 * 60);
  ?>
- $_FILE利用上傳檔案。
  - <form action=“fileup.php “ method="post “ enctype="multipart/form-data" >
    <input type=“file" name=“file_up">
    <input type="submit" value="送出">
    </form>
  - <?php //fileup.php
    echo $_FILES[‘fileup’][‘name’]; //檔案名稱
    echo $_FILES[‘fileup’][‘type’]; // 檔案類型
    echo $_FILES[‘fileup’][‘size’]; //檔案大小
    echo $_FILES[‘fileup’][‘tmp_name’]; //暫存的檔名
    ?>
  -

超級全域變數(7)

32

- $_SERVER：包含標頭訊息、路徑...等系統訊息。
  - // 判斷是否透過proxy，再來取得IP
    if($_SERVER[“HTTP_X_FORWARDED_FOR”] == “”)
    $ip = $_SERVER[‘REMOTE_ADDR’];
    else
    $ip = $_SERVER[‘HTTP_X_FORWARDED_FOR’];

算術運算子

33

算術運算子	意義	範例
+	加法	$a + $b
-	減法	$a - $b
*	乘法	$a * $b
/	除法	$a / $b
%	求餘數	$a%$b

遞增/遞減運算子

34

運算子	範例	說明
++(前置)	++ $a	$a 加一
--(前置)	--$a	$a 減一
(後置)++	$a++	$a 加一
(後置)--	$a--	$a 減一

前置：先加減完再運算。後置：先運算完後再加減。

```
$a = 5;
echo  $a++; //結果為 5
echo  $a;           //結果為 6
echo  --$a ;        //結果為 5
echo  $a ;          //結果為 5
```

34

指定運算子與複合運算子

35

運算子	意義	範例	範例說明
=	指定，等號右邊的值指定給等號左邊	$a = 5; $b = $a;	5指定給$a $a的值指定給以b
+=	複合，與本身加	$a+=5;	$a 本身加上5
-=	複合，與本身減	$a-=5;	$a 本身減掉5
*=	複合，與本身乘	$a*=5;	$a 本身乘以5
/=	複合，與本身除	$a /=5;	$a 本身除以5
.=	複合，與本身字串接	$a.=5;	$a 本身接上'5'
尚有%/, <<=, >>=, &=, |=, ^= 等複合運算子。			

比較運算子

36

運算子	意義	範例
<	小於	$a < $b
<=	小於等於	$a <= $b
>	大於	$a > $b
>=	大於等於	$a >= $b
!= 或<>	不等於	$a != $b
==	相等，資料型態會自行轉換。	$a == $b
===	全等且資料型態也相同。	$a === $b
!==	不全等，不相等或資料型態不同。	$a !== $b

邏輯運算子

37

邏輯運算子	意義	範例
!(not)	反相	!$a
&&或and	交集	$a && $b
||或or	聯集	$a || $b
xor	xor，互斥or	$a xor $b

字串運算子

38

字串運算子	意義	範例
.	字串相連接	“Hello”．“ James”；

- $a = ‘今天為：’；
- $b = ’ 101年10月22日’；
- $a = $a . $b;
- echo $a　　// 今天為：101年10月22日

三元運算子

39

運算子	範例	意義
? :	$a ? $b : $c	若$a為true傳回$b 若$a為false傳回$c

- 三元運算子可視為精簡版的if...else...
- 假若$b等於5，$a 就是11，否則$a為12
- $a = ($b == 5) ? 11: 12;

位元運算子

40

運算子	範例	說明
&	$a & $b	$a 與 $b每個位元作AND運算
|	$a | $b	$a 與 $b每個位元作OR運算
^	$a ^ $b	$a與 $b每個位元作XOR運算
~	~ $a	$a 每個位元做NOT運算
>>	$a >> k	$a向右移動k個位元
<<	$a << k	$a向左移動k個位元

其他運算子

41

- 錯誤控制運算子：
  - @：在PHP程式執行有錯誤時，抑制顯示錯誤訊息。
  - 寫在要忽略錯誤訊息的指令之前。只會忽略錯誤訊息，不能修正程式的錯誤。
  - <?php
    $fp=@fopen(1.txt);
    ?>
- 執行外部指令運算子：
  - ``：讓PHP程式執行作業系統的shell命令。
  - 將要執行的shell命令包含住。效果同shell_exec()。
  - <?php
    $ping=`www.google.com`;
    echo nl2br($ping); // nl2br()：自動分行顯示。
    $ping=shell_exec("www.google.com");
    echo nl2br($ping);
    ?>

運算子優先順序

42

優先順序	運算子
高	()
	!、~、++、--、@
	*、/、%
	+、-、.
	<<、>>
	>、>=、<、<=
	==、===、!=、!==
	&、^、|
	&&、||
	=、*=、/=、%=、+=、-=、<<=、>>=、&=、^=、|=、.=
低	and、xor、or、

條件敘述 if

43

```
if(條件判斷式)
{
    敘述句…;
}
```

條件敘述 if

44

```
if(條件判斷式) {
   敘述句…;
}
if(條件判斷式) {
   敘述句…;
}
else {
   敘述句…;
}
if(條件判斷式) {
   敘述句…;
}
elseif (條件判斷式){
   敘述句…;
}
...
else{
   敘述句…;
}
```

條件敘述 switch

45

- switch(常數、變數、運算式) {
  case '值1':
  敘述句...;
  break;
  case '值2':
  敘述句...;
  break;
  …
  default:
  敘述句...;
  }

for 、while、do while迴圈

46

- for(起始值; 迴圈終止判斷; 變動量){
  敘述句…;
  }
- while(迴圈終止判斷){
  敘述句…;
  }
- do {
  敘述句…;
  } while (迴圈終止判斷);

foreach迴圈

47

- foreach(array as $value){
  敘述句…;
  }
- foreach(array as $key=>$value){
  敘述句…;
  }

迴圈中的break、continue

48

- break：強制中斷迴圈。
- continue：略過此次迴圈尚未執行的部份，直接執行下一輪迴圈。

函式

49

- 將會重複執行的程式碼，集合成一程式區塊，在程式執行時可重複呼叫使用，此程式區塊稱之為函式。
- 函式的優點：
  - 可重複執行。
  - 使程式碼結構清楚、精簡。
  - 容易除錯及維護。
- 自定函式：

```
function 函式名稱( [ $參數1, $參數2, … ] ){
   敘述句…;
   [return 傳回值;]
}
```

- 使用方式：

```
函數名稱( [ $參數1, $參數2, … ] );
```

內建函式

50

- PHP內建許多常用的函式，內建函式不用事先宣告即可使用。
- 數學函式：
  - abs()。
  - sin()、cos()、tan()、asin()、acos()、atan()。
  - deg2rad()、rad2deg()。
  - bindec()、decbin()。
  - decoct()、octdec()。
  - dechex()、hexdec()。
  - ceil()、floor()。
  - round()。
  - exp()。
  - is_finite()、s_infinite()、is_nan()。
  - log()、log10()。
  - power()。
  - max()。
  - rand()。
  - pi()。
  - ...。

數學常數

51

- M_1_PI：1/π。
- M_2_PI ：2/π。
- M_2_SQRTPI ：2/sqrt(π)。
- M_E ：自然底數e。
- M_EULER ：尤拉常數。
- M_LOG2E ：$\log_2 e$。
- M_LOG10E ： $\log_{10} e$ 。
- M_LN2 ： $\log_e 2$ 。
- M_LN10 ： $\log_e 10$ 。
- M_LNPI ： $\log_e \pi$ 。
- M_PI ：π。
- M_PI_2 ：π/2。
- M_PI_4 ：π/4。
- M_SQRT1_2 ：1/sqrt(2)。
- M_SQRT2 ：sqrt(2)。
- M_SQRT3 ：sqrt(3)。
- M_SQRTPI ：sqrt(π)。

日期、時間函式

52

- date()：將本機的時間、日期格式化。
- getdate()：取得日期、時間資訊。
- checkdate() ：有效的日期/時間。
- gmdate() ：將格林威治的時間、日期格式化。
- time() ：取得目前時間資訊。
- mktime() ：根據參數的時間、日期建立時間戳記。
- ...。

字串函式

53

- strtolower()、 strtoupper()：大小寫轉換。
- trim ()、 ltrim () 、rtrim () ：去除字串開始或結束的空白。
- nl2br ()：將換行符號轉換成HTML的
。
- strlen ()：取得字串長度。
- strpos ()：找出字串第一次出現的位置。
- substr ()：取得部份字串。
- str_replace()、str_ireplace()：取代字串。
- strchr()、 strstr() 、 stristr() 、 strrchr() ：尋找字串。
- strcmp()、 strcasecmp() 、 strncmp() 、 strncasecmp() ：字串比較。
- echo ()：輸出字串。
- implode()：將字串陣列組成字串。
- explode ()：分解成子字串陣列。
- ...。

陣列函式(1)

54

- array() ： 建立陣列。
- range() ：產生一規律的陣列。
- shuffle() ：將陣列內容打亂。
- list() ： 取出陣列。
- count() ： 計算陣列元素個數。
- array_count_values() ： 計算陣列中各種值的出現頻率。
- array_sum() ： 將陣列內容值加總。
- array_key_exists() 、 array_search() ： 搜尋某個 key 是否存在於陣列中，不傳回索引值、傳回索引值。
- asort()、arsort()、ksort()、krsort()、sort()、rsort()、 usort()、 uasort()、uksort()：陣列的各種不同排序函式。
- array_reverse() ： 將陣列內容的順序反轉。

陣列函式(2)

55

- is_array()：是否為陣列。
- in_array()：某值是否在陣列中。
- array_flip() ： 將陣列中的鍵與值對調。
- array_unique() ： 去除陣列中重複的元素。
- array_shift()、 array_unshift() ： 將元素由陣列前端移出、前端塞入。
- key()、 current()、each() ：將陣列指標目前所指的 鍵值(key)值(value)、兩者都傳回來 。
- reset() 、prev() 、 next() 、end() ：將陣列指標至開頭、往前、往後、至最後。
- …。

MySQL有關函式

56

- mysql_connect() 、 mysql_pconnect() ：與Mysql資料庫建立連線、建立永久連線。
- mysql_select_db() 、 mysql_create_db() 、 mysql_drop_db() ：開啟資料庫、建立資料庫、刪除資料庫。
- mysql_close() ：關閉資料庫連線。
- mysql_query() ：執行SQL指令。
- mysql_result() ：取得查詢結果。
- mysql_num_rows() 、 mysql_num_field() ：查詢總筆數、總欄位數。
- mysql_fetch_array() ：將查詢結果以陣列表示。
- mysql_fetch_row() ：將查詢結果以變數表示。
- mysql_insert_id() ：前一筆紀錄新增時的ID值。
- mysql_error() 、 mysql_errno() ： Mysql錯誤訊息、錯誤訊息編號。
- ...。

引入檔

57

- 將常用的程式區塊儲存成檔案，在需要時才引入程式中。簡化網頁程式的維護、設計上更有彈性、更有效率。可稱之為模組式開發。
- 引入外部檔：
  - include()。
  - require()。
- require()在檔案不存在時會終止程式，include()顯示警告訊息後仍繼續執行。
- 同一網頁中require()只需要讀取評估一次。include()每用到一次都要讀取評估一次。
- require()不能傳回值。include()可傳回值。
- 只引入一次：
  - include_once()
  - require_once()
- 會先檢查是否已經載入過，避免重複載入問題。

4-3 MySQL資料庫

58

- 資料庫的操作分成以下幾部份。
- 建立連線：
  - mysql_connect()。
  - mysql_pconnect()。
  - 建議使用引入檔建立連線。
- 選擇資料庫：
  - mysql_select_db()。
- 資料表操作：
  - mysql_query()。
  - mysql_db_query()。
- 取得查詢結果：
  - mysql_fetch_row()。
  - mysql_fetch_array()。
  - mysql_fetch_assoc()。
  - ...。

59

- 資料回傳：
- 關閉連線：
  - mysql_close()。

MySQL的欄位資料型態

60

型態	大小	範圍說明
Char(n)	固定長度	n=1~255
varchar(n)	變動長度	n=1~255
tinytext text mediumtext longtext	變動長度	255 bytes 65535 bytes 16777215 bytes 4294967295 bytes
enum(value1, value2, …)	列舉型態	可設定65535個列舉值
set(value1, value2, …)	集合型態	數目最多64個

MySQL的欄位資料型態

61

類型	大小	範圍說明
tinyint	1byte	-128~127 unsigned： 0~255
smallint	2bytes	-32768~32767 unsigned：0~65535
mediumint	3bytes	-8388608~8388607 unsigned：0~16777215
integer	4bytes	-2147483648~2147483647 unsigned：0~4294967295
bigint	8bytes	-9.2*10^18~9.2*10^18 unsigned：0~1.8*10^19
float	4bytes	-3.4E+38~3.4E+38
double	8bytes	-1.79E+308~1.79E+308
decimal(m, n)	視參數而定	m:數值總位數。n小數位數。

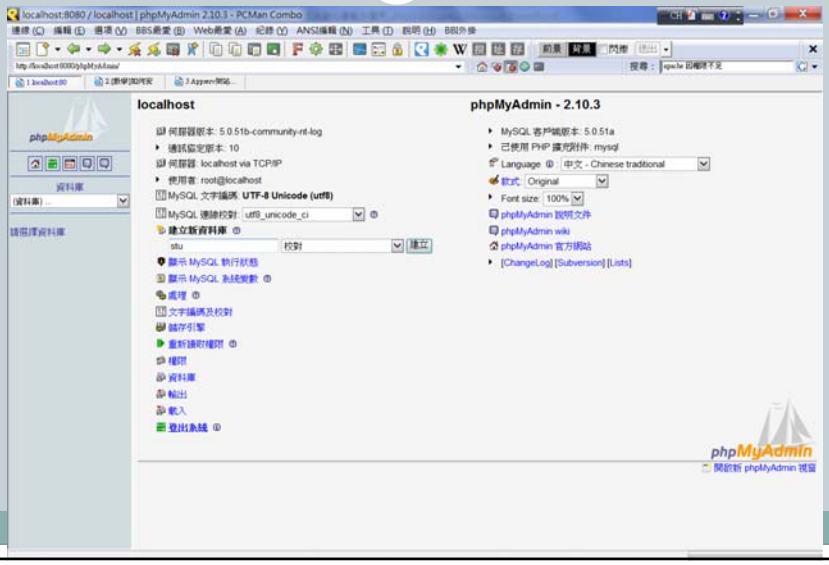
MySQL的欄位資料型態

62

類型	大小	說明
date	3bytes	格式：yyyy-mm-dd
time	3bytes	格式：hh:mm:ss
datetime	8bytes	格式：yyyy-mm-dd hh:mm:ss
Timestamp(n)	4bytes n=2,4,6,8,10,12,14	格式： 14：yyyymmddhhmmss(default) 12：yyyymmddhhmm 10：yyyymmddhh 8：yyyymmdd 6：yymmdd 4：yymm 2：yy

phpMyAdmin

63



4-4 SQL

64

- SQL是「結構化查詢語言」（Structured query language）的簡稱，為關聯式資料庫的標準語言。1970 年代初期所開發。
- 利用 SQL 可以用來對資料庫新增、異動或查詢資料…等操作。
- SQL1（SQL-86）。
- SQL2（SQL-92）。
- SQL3（SQL-99）。
- SQL4（SQL-2003）。

Select

65

- Select 語法
  select 欄位1, …
  from 資料表1, ...
  [where 條件]
  [group by 欄位1, ...];
- select * from student
  - 選擇student資料表所有紀錄。
- select name, birth from student order by name
  - 選擇student資料表所有紀錄的(name, birth)兩個欄位，並按姓名升冪排序(asc 升冪、desc 降冪)。
- select * from student where id > 4 and id< 16
  - 選擇id欄位介於4至16的紀錄。
- select * from student limit 4,16
  - 選擇student資料表第4筆到第16筆紀錄。

Insert、Delete、Update

66

- Insert ：
  insert [into] [資料庫.]資料表 (欄位1, …)
  values (值1, …);
- Delete ：
  delete [資料庫.]資料表
  [set 欄位1 = 值1, …]
  where 條件;
- Update ：
  update [資料庫.]資料表
  where 條件;

參考文獻(1/2)

67

- 吳弘凱。PHP + MySQL快速入門。松崗。臺北市。2006。
- 陳惠貞、陳俊榮。PHP & MySQL 程式設計實例講座。學貫臺北市。2009。
- 蔡憲維、陳朝均、辛曼榕、黃俊銘、李衍儀。PHP+MySQL網站系開發講座。博碩文化。臺北縣。2010。
- 鄧文淵。挑戰PHP5 MySQL程式設計樂活學。第2版。碁峰。台北市。2011。
- 西沢夢路。MySQL + PHP資料庫網頁程式設計實例入門。博碩文化。臺北縣。2011。
- 陳惠貞、陳俊榮。PHP&MySQL案例開發實戰手冊。碁峰。台北市。2012。

參考文獻(2/2)

68

- 張亞飛。徹底研究PHP6+MySQL全能權威指南。上奇。臺北市。2012。
- 恩光技術團隊。跟我學PHP&MySQL。http://followme.gotop.com.tw/PHP5MySQL/index.htm。2012/8/1讀取。
- 維基百科。PHP。http://zh.wikipedia.org/wiki/PHP。2012/8/1讀取。
- PHP官方網站。http://www.php.net。2012/9/1讀取。
- MySQL官方網站。http://www.mysql.com。2012/9/2讀取。
- Jacch。PHP5網管實驗室。http://www.php5.idv.tw。2012/9/1讀取。
- PHP初探。http://chensh.loxa.edu.tw/php/A_1.php。2012/9/18讀取。