

3.6 表單資料的傳送與接收

表單是網頁瀏覽者將資料傳遞到伺服器處理的介面，在網頁程式的開發過程中，表單佔了相當重要的地位。如何正確佈置表單，程式端又如何接收，就是以下要討論的內容。

3.6.1 表單傳送與接收資料的方法

表單運作的原理是在頁面中建置表單區域，並在區域內放置填寫資料的表單元件，當按下表單的送出鈕時，頁面會將表單區域中元件內填寫的資料，傳送到指定的目標網頁接收處理。

表單的傳送方式與接收方法

表單區域是以 `<form>` 標籤所包圍的區域，在標籤中設定 `action` 屬性指定傳送的目標頁面，而 `method` 屬性即是傳送的方式。表單的傳送方式，也就是 `method` 屬性的值有以下二種：

屬性	說明
GET	表單資料將以字串的方式附加在網址的後面傳送，在網址後會以「？」符號開啟跟著表單中的資料，每個欄位間的值以「&」連接起來。
POST	表單資料將放置在 HTTP 標頭的方式傳送

那目標網頁要如何接收這些方式所傳遞過來的資料呢？

屬性	說明
GET	在結果頁使用 <code>\$_GET["欄位名稱"]</code> 接收其欄位所指定的值。
POST	在結果頁使用 <code>\$_POST["欄位名稱"]</code> 接收其欄位所指定的值。

註 `$_GET` 及 `$_POST` 都必須大寫。

在以下的範例中，我們先佈置一個簡單的表單，並使用 POST 的方式來傳送資料到目標網頁接收顯示：

程式碼：php_form1_post.htm

儲存路徑：C:\htdocs\ch03

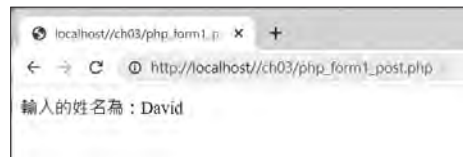
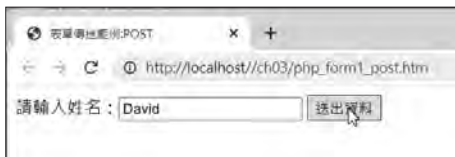
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title> 表單傳送範例:POST</title>
</head>
<body>
<form method="POST" action="php _ form1 _ post.php">
  請輸入姓名:<input type="text" name="username" />
  <input type="submit" value=" 送出資料 " />
</form>
</body>
</html>
```

程式碼：php_form1_post.php

儲存路徑：C:\htdocs\ch03

```
<?php
echo " 輸入的姓名為:";
echo $_POST["username"];
?>
```

執行結果 執行：http://localhost/ch03/php_form1_post.htm、php_form1_post.php



在傳送頁的 <form> 標籤中，以 POST 方式傳送到 <php_form1_post.php>，只要在文字欄位 username 輸入資料，接收頁要用 \$_POST["username"] 接收。

用相同的方式，在以下的範例中，先佈置一個簡單的表單，並使用 GET 的方式來傳送資料到目標網頁接收顯示：

9.4 檔案上傳

所謂檔案上傳，就是將檔案由客戶端的主機，藉由瀏覽器傳送到伺服器的資料夾上。以下我們將詳細介紹使用 PHP 執行檔案上傳的原理與實際操作的方法，並討論系統調校的技巧。

9.4.1 檔案上傳的原理

在網頁中將檔案由客戶端上傳到伺服器中，傳送的過程如下：



- 1 在表單檔案欄位選取要上傳的檔案。
- 2 表單送出，將檔案傳送到伺服器。
- 3 伺服器在接收的過程中，先將接收到的檔案放置在暫存資料夾中。
- 4 傳送完畢後將完整的檔案搬移到指定的網頁資料夾中。

在整個過程中最重要的環節，在於傳送資料的表單與接收資料的程式。檔案上傳的動作，因為是利用表單傳送二進位的資料，有別於一般表單只傳送文字訊息的動作，所以在表單的佈置上就不同於一般的表單。而接收端的程式，也因為要接收二進位資料，所以要處理暫存檔及搬移到指定資料夾的動作。

9.4.2 php.ini 在檔案上傳功能上的調整

在使用檔案上傳的功能前，對於 PHP 的執行環境要進行以下的檢查及調整。

請開啟 <php.ini> 後並搜尋以下的設定值：

是否允許上傳

請開啟允許網頁伺服器允許上傳的設定：

```
file_uploads = On
```

暫存檔資料夾

在檔案上傳到網頁伺服器時會先放在暫存資料夾，完成後才搬到指定資料夾。若沒有設定會造成上傳失敗，若在 Windows 系統中，可以參考以下設定：

```
upload_tmp_dir="C:\Windows\Temp"
```

該資料夾必須真的存在，並且要有寫入的權限，若您是 Linux 作業系統，請參考：

```
upload_tmp_dir="/tmp"
```

可接受上傳檔案大小

您可以設定伺服器能夠接受上傳檔案的大小，若設的太大會因為程式執行過長而造成失敗，設的太小可能會影響程式的適用性。在 `<php.ini>` 中預設的大小為 2MB，建議您可以調整為 8MB 到 10MB 之間，適用於大部分的需求。

```
upload_max_filesize = 8M
```

修改完畢之後，請您停止伺服器後再重新啟動，讓網頁伺服器可以適用修改後的環境。

9.4.3 檔案上傳的表單

上傳檔案的表單，有幾個重要的注意事項：

1. 在 `<form>` 標籤中「action」屬性必須設定要接收檔案的 PHP 程式檔。
2. 在 `<form>` 標籤中，傳送方式屬性必須要設定為「method="post"」，因為檔案上傳表單的傳送一定要使用 POST 的方法。
3. 在 `<form>` 標籤中，因為有檔案傳送的動作，所以要設定資料的編碼方式，這裡要加上「enctype="multipart/form-data"」的屬性，才能正確地讓檔案欄位送出。
4. 上傳的檔案欄位為 `<input>` 標籤，屬性必須設定「type="file"」，如此即可在使用時出現 瀏覽 鈕，讓使用者選取要上傳的檔案。

舉例來說，以下我們就佈置一個簡單的上傳檔案表單：

程式碼：php_file9.htm

儲存路徑：C:\htdocs\ch09

```

1  <html>
2  <head>
3    <title>上傳檔案表單 </title>
4  </head>
5  <body>
6    <form action="php_file9.php" method="post" enctype="multipart/
    form-data">
7      請選取要上傳的檔案:<br />
8      <input type="file" name="fileUpload" /><br />
9      <input type="submit" value="送出資料" />
10   </form>
11 </body>
12 </html>
    
```

執行結果

執行網址：http://localhost/ch09/php_file9.htm



程式說明

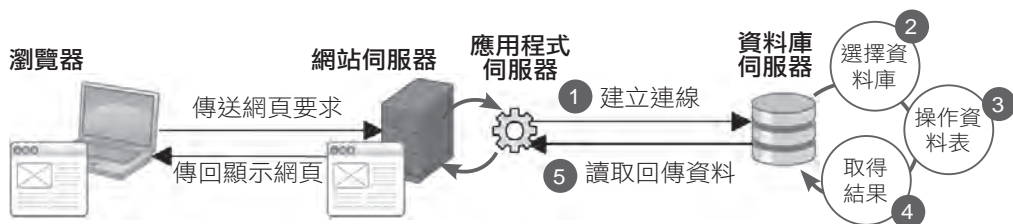
- 6 <form> 標籤中設定傳送方式為 POST，編碼為 multipart/form-data，傳送的目的地為 <php_file9.php>。
- 8 <input> 標籤中設定類別為 file，如此才能顯示選取檔案的按鈕。

如此即可成功的佈置符合規格的表單進行上傳檔案。

14.1 認識 PHP 與 MySQL 的運作

一路認識了 MySQL 的資料庫、資料表及欄位的建置，到使用 SQL 指令來操作資料的新增、查詢、修改與刪除等動作，本章將要說明如何使用 PHP 程式來結合 MySQL 資料庫進行整合運作。

PHP 中提供許多操作 MySQL 資料庫的相關函式，能夠讓使用者存取 MySQL 資料庫中的資料。PHP 在使用 MySQL 資料庫的資源時，必須經過以下的流程：



在瀏覽器的使用者網頁伺服器提出要求後，應用程式根據需求準備對資料庫進行操作：

- 1 **建立連線**：與資料庫伺服器進行連線，必須賦予正確的帳號、密碼，並在權限檢查無誤的狀況下，才能正確連上資料庫伺服器。
- 2 **選擇資料庫**：連上資料庫伺服器之後，即可選擇要使用的資料庫。
- 3 **操作資料表**：從資料庫中對資料表進行查詢、新增、修改、刪除資料…等動作。
- 4 **取得結果**：若上個步驟是進行查詢的動作，即可由此將結果儲存在資源中。
- 5 **資料回傳**：將儲存在資源中的結果讀出化為內容回傳，最後關閉資料庫連線。

在這些步驟中，PHP 都提供了相關的函式，讓使用者可以完成相關的動作與要求。其中 PHP 對於 MySQL 的操作大部分都是應用 SQL 指令，所以若想要快速上手 PHP 與 MySQL 的整合應用，對於 SQL 語法得要有相當的基礎。

PHP 5.3 之後內建了 MySQLi 取代原來的 MySQL 擴充程式進行 MySQL 資料庫的操作。MySQLi 支援物件導向，使用者除了可以使用傳統的函式或是物件導向的方式操作資料庫，MySQLi 除了提供了相對的進階功能，也大大增強了使用的安全性與效能。在本書中的所有範例將會全面介紹使用 MySQLi。

14.2 使用 MySQLi 函式操作資料庫

MySQLi 是原來 PHP 中 MySQL 擴充程式的加強版，PHP 可以使用 MySQLi 函式以原來相同的方式進行資料庫的連結、存取、查詢等動作。

14.2.1 mysqli_connect()：建立資料連結

在 PHP 中可以使用 `mysqli_connect()` 函式建立與 MySQL 之間的連線，其語法格式如下：

```
mysqli_connect(MySQL 伺服器位址, 帳號, 密碼, 資料表, 連線埠位)
```

其中資料表及連線埠位非必填，可以使用預設值。

我們曾說明過，MySQL 在連線時必須有三個關卡，一是檢查連線來源的 IP 是否合法，再來是帳號與密碼是否正確，最後是該帳號是否有適當的權限。

舉例來說如果要連結到目前本機的 MySQL 資料庫伺服器，其帳號、密碼為：「root」、「1234」，程式碼的內容為：

```
mysqli_connect("localhost", "root", "1234");
```

要注意的是在 `mysqli_connect()` 函式中使用的參數，都是以字串的方式來填入。

若要連結到非本機的 MySQL 資料庫伺服器，可以設定伺服器的位址或是網域名稱。例如要連結到主機 IP 為「192.168.0.50」的 MySQL 資料庫伺服器，其帳號、密碼為：「root」、「1234」，程式碼內容為：

```
mysqli_connect("192.168.0.50", "root", "1234");
```

使用 `mysqli_connect()` 連線後會回傳一個連接識別碼 (link identifier)，否則會回傳 FALSE 值。

以下的範例我們將依連結後的結果來顯示訊息：

程式碼：php_mysql1.php

儲存路徑：C:\htdocs\ch14

```
1 <?php
2 $db_link = mysqli_connect("localhost", "root", "1234");
3 if (!$db_link) {
```

```
4     die(" 資料連結失敗 ");
5 }else{
6     echo " 資料連結成功 ";
7 }
8 ?>
```

執行結果

執行網址：http://localhost/ch14/php_mysql1.php



程式說明

2 使用 `mysqli_connect()` 函式連結本機 MySQL 資料庫伺服器，並將連接識別碼回傳至 `$db_link` 中。

3~7 如果 `$db_link` 為 `FALSE` 則顯示連結失敗的訊息，若成功顯示成功的訊息。

使用 `mysqli_connect()` 函式建立連線後，在預設狀況下是在程式碼執行完畢之後即會自動關閉。若有需要在程式中先行關閉連線，可以使用 `mysqli_close()` 函式進行關閉的動作，其語法格式如下：

```
mysqli_close([ 連接識別碼 ])
```

使用 `mysqli_connect()` 函式連接資料庫伺服器成功後會產生連接識別碼，使用 `mysqli_close()` 函式時可以直接指定要關閉的連線，若沒有指定即會關閉最新的一個連線。

14.2.2 設定字元集與編碼

在第 12 章中曾深入討論了 MySQL 字元集與編碼的問題，現在我們要利用 PHP 的函式來使用 MySQL 資料庫資源也必須遵守一樣的規則。如果 MySQL 資料庫本身的字元集與編碼是使用 big5 繁體中文，而 PHP 程式中卻是使用 utf-8 的編碼進行連線，那顯示的結果就必然會是亂碼了！

在本書中我們統一使用 utf-8 的字元集與編碼資料庫進行存取的動作，所以在建立資料連線之後就必須馬上宣告，如此一來就不會發生 PHP 存取編碼與資料庫設定編碼不同步的狀況發生。

14.5 使用 PDO 物件操作MySQL

除了 MySQLi，現在 PHP 可以使用 PDO 物件的方式操作 MySQL 資料庫，除了提高使用的效率與安全性，也能輕易整合其他不同類型的資料庫。

14.5.1 使用 PDO 類別建立資料庫物件

使用 PDO 的好處

過去許多人在 PHP 中利用 MySQL 或 MySQLi 函式來進行資料庫的連結，並依照標準步驟取得資料庫中的內容進行應用。但若要使用的資料庫平台更多元、更複雜，往往就會必須視 PHP 是否有支援該資料庫的擴充程式或對應的語法，對於開發者來說會是相當頭痛的問題。

PHP 引入了一個新的資料庫物件擴充程式：PDO (PHP Data Objects)，試途使用更簡單的方式，但更安全有效率的連接資料庫，以下將介紹 PDO 的使用方式。

建立資料庫連結

這個方式是利用 PDO 類別來新增資料庫連結物件，無論使用何種資料庫驅動程式，只要指定資料連結的資料來源名稱 DSN (Data Source Name)，再依需求提供連結時要使用的帳號、密碼，即可建立資料庫的連結並進行使用。DSN 中描述了要連結資料庫的結構，包含了資料庫類型、主機位址埠位，以及預設要使用的資料表。以 MySQL 為例，PDO 的語法格式為：

```
PDO 物件名稱 = new PDO("mysql:host=MySQL 伺服器位址 ;dbname= 資料表名稱 ; charset= 預設字元集編碼 " , 帳號 , 密碼 , )
```

當連接發生錯誤時會產生錯誤的 PDOException 物件，您必須將連接步驟包裝在一個 try / catch 區域中進行錯誤補捉，並進行相關處理。以下我們將之前的資料庫範例以物件的方式進行連結，程式碼如下：

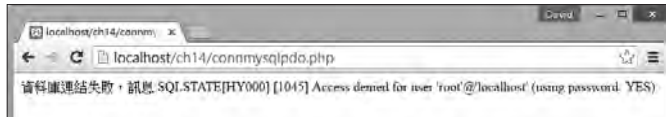
程式碼：connMySQLPdo.php

儲存路徑：C:\htdocs\ch14

```
1 <?php
2 // 資料庫主機設定
3 $db_host = "localhost";
4 $db_username = "root";
```

```
5     $db_password = "1234";
6     $db_name = "class";
7     // 錯誤處理
8     try{
9         // 連線資料庫
10        $db_link = new PDO("mysql:host={$db_host};dbname={$db_name};charset=utf8", $db_username, $db_password);
11    } catch (PDOException $e) {
12        print " 資料庫連結失敗，訊息 :{$e->getMessage()}<br/>";
13        die();
14    }
15    ?>
```

執行結果

執行網址：http://localhost/ch14/php_mysql11.php

程式說明

- 2 使用 `header()` 函式宣告本頁的編碼。
- 3~6 設定資料庫連結的主機、帳號、密碼及選用資料庫資料。
- 8~14 利用 `try/catch` 區域進行資料庫連結與錯誤捕捉。
- 9~10 利用 `PDO()` 類別新增 `$db_link` 資料庫物件。其中應用之前設定好的資料庫主機、帳號、密碼及預設資料庫，最後再設定要使用的字元集編碼即可完成。
- 11~13 當有錯誤發生時在 `catch` 中會捕捉到 `PDOException` 物件並進行錯誤處理。

在這個範例中，如果 DSN 的資料都設定正確，並不會顯示任何訊息。請刻意輸入錯誤的連線密碼，在執行時因為產生錯誤，所以會 `catch` 到 `PDOException` 物件，就會顯示到畫面中。

這個範例也可以成為 `PDO` 資料庫連結的引用檔，未來當有需要資料庫資源的程式頁面，在一開始只要匯入該檔，即可馬上完成建立資料庫伺服器連線到設定字元集與編碼的必要步驟。

14.5.2 執行資料表查詢並顯示

利用 PDO 類別新增資料庫物件後可以使用 `query()` 方法執行 SQL 指令，其語法格式如下：

```
PDO 物件名稱 -> query( SQL 指令字串 )
```

執行成功時會返回一個 `mysqli_result` 物件，失敗時會返回 `FALSE`。此時可以使用 `fetch()` 方法來讀取查詢結果中的一筆資料，並將記錄指標往下移動，若到達記錄的底部就會回傳 `FALSE` 值，其語法格式如下：

```
mysqli _ result 物件名稱 -> fetch()
```

例如我們若想要選取 `class` 資料庫中 `students` 資料表所有資料，其語法如下：

程式碼：php_mysql20.php

儲存路徑：C:\htdocs\ch14

```
1  <?php
2  header("Content-Type: text/html; charset=utf-8");
3  include("connMySQLPdo.php");
4
5  $sql_query = "SELECT * FROM students";
6  $result = $db_link->query($sql_query);
7  while($row_result=$result->fetch()){
8      foreach($row_result as $item=>$value){
9          echo $item."=".$value."<br>";
10     }
11     echo "<hr>";
12 }
13 ?>
```

執行結果

執行網址：http://localhost/ch14/php_mysql20.php



程式說明

- 2 使用 `header()` 函式宣告本頁的編碼。
- 3 使用 `include()` 函式匯入連線引入檔完成 MySQL 資料庫物件建立，預設連結到 `class` 資料庫，並設定字元集與編碼的動作。
- 5 將要執行的 SQL 指令字串設定在 `$sql_query` 中，此處要選取 `students` 資料表中所有欄位。
- 6 使用 `query()` 方法執行 `$sql_query` 中的 SQL 指令並將取得的資料放置在 `$result` 物件中。
- 7~12 使用 `fetch` 方法由 `$result` 物件依次取出每個學生的資料為 `$row_result` 陣列到資料底部為止。
- 8~11 使用 `foreach` 迴圈導出每一欄的主索引 (`$item`) 與資料 (`$value`) 來顯示。每顯示完一個學生的所有資料後，使用「`<hr>`」進行分行動作。

14.5.3 PDO 的預備語法

PDO 也支援預備語法，以下將先說明使用流程後再以實例說明。

預備 SQL 敘述

在使用預備語法前，首先要將使用的 SQL 敘述化為字串，這裡使用 PDO 物件的 `prepare` 方法將這個字串進行預存產生一個 `pdo_stmt` 物件，其語法格式如下：

```
pdo_stmt 物件名稱 = pdo 物件 -> prepare(SQL 敘述字串)
```

以剛才的範例來說，如果想要取得所有同學的資料，要產生預備 SQL 敘述方式如下：

```
$sql_query = "SELECT * FROM students";  
$stmt = $db_link -> prepare($sql_query);
```

使用 SQL 敘述時大部份會利用不同的條件進行資料的篩選，此時會應用到相關的參數。在預備語法的 SQL 敘述中，若要加入參數可以使用問號 (?) 取代。例如我們要取得座號為「1」的同學資料，其預備 SQL 敘述方式如下：

```
$sql_query = "SELECT * FROM students WHERE cid = ?";  
$stmt = $db_link -> prepare($sql_query);
```

16

CHAPTER

專題：會員系統的製作

在完整的網站中加入會員系統是十分重要的，因為會員的收集與資料使用，不僅可以讓網站累積人脈，善用這些會員的資料，也可能為網站帶來無窮的商機。

會員系統可以結合許多其他的作品，讓其他的程式能夠加上群組或是分眾化的特性，對於網站經營或是客戶服務都很有幫助，所以應用層面是十分廣泛的。

- ◎ 專題說明及準備工作
- ◎ Password Hasing 加密函式
- ◎ 資料連線引入檔的製作
- ◎ 會員系統主頁面的製作
- ◎ 會員系統加入會員頁面的製作
- ◎ 會員系統會員中心頁面的製作
- ◎ 會員系統修改資料頁面的製作
- ◎ 會員系統管理主頁面的製作
- ◎ 會員系統管理員修改會員資料頁面
- ◎ 會員系統補寄密碼信頁面的製作

16.1

專題說明及準備工作

本章將以一個完整的範例來介紹網站會員系統的程式，讓會員可以依本身不同的權限前往不同頁面，執行不同功能。

16.1.1 認識會員系統及學習重點

什麼是會員系統？

在完整的網站中加入會員系統是十分重要的，因為會員的收集與資料使用，不僅可以讓網站累積人脈，善用這些會員的資料，也可能為網站帶來無窮的商機。



▲ 會員系統完成圖

本章學習重點

在會員系統的程式設計中，有幾個學習重點：

1. 使用 `Session` 記錄會員帳號與權限，依權限瀏覽不同的頁面，執行不同功能。
2. 會員的密碼以 `password_hash()` 進行加密，重新以亂數隨機產生密碼。
3. 使用 `Cookie` 記錄會員的登入資訊。

16.1.2 程式環境及資料庫分析

程式環境

本書把每個不同的程式以資料夾的方式完整地整理在 <C:\htdocs> 裡，我們已經將作品完成檔放置在本章範例資料夾 <phpmember> 中，您可以將它整個複製到 <C:\htdocs> 裡，就可以開始進行網站的規劃。

註 強烈建議您可以先按照下述步驟，將資料庫匯入到 MySQL 中，再調整完成檔中連線的設定，如此即可將程式的完成檔安裝起來執行並測試功能，再依書中的說明對照每一頁程式的原始碼，了解整個程式的運作與執行結果。

匯入程式資料庫

本書範例中，一律將程式使用資料庫備份檔 <*.sql> 放置在各章範例資料夾的根目錄中，在這裡請將 <phpmember> 資料夾中資料庫的備份檔 <phpmember.sql> 匯入，其中包含了一個資料表：「memberdata」。



1 首先要新增一個資料庫，請輸入資料庫名稱：「phpmember」，並設定校對為：「utf8_unicode_ci」，最後按 **建立** 鈕。



2 在左側選取「phpmember」資料庫後，請選按 **匯入** 連結執行備份匯入的動作，按 **瀏覽** 鈕來選取本章範例資料庫檔案 <phpmember.sql>，最後按 **執行** 鈕。



3 畫面上顯示了執行成功的訊息，即可在左列看到新增的資料表。



4 請按 **結構** 文字連結來看看這資料表的結構。

請開啟瀏覽器進入 phpMyAdmin 的管理介面，使用設定的帳號、密碼登入。

資料表分析

請選按資料表後方的 **屬性** 圖片連結，觀看資料表內容。

	m_id	m_name	m_username	m_passwd	m_sex	m_birthday
<input type="checkbox"/>	1	系統管理員	admin	\$2y\$10\$FO70lc.3VTe0Vaf7O3Jes.UAryizLnmxZfTF741...	男	NULL
<input type="checkbox"/>	2	老張玲	elven	\$2y\$10\$YdUhOvUTvwK5oWpf3LafOd2imwsEi85YmmoY2knsx...	女	1987-04-05
<input type="checkbox"/>	3	彭建志	jingjun	\$2y\$10\$WqB2bnMSOwgBiHSOBV2iuLbrUCsp8VmNjdK2AyiW6l...	男	1987-07-01
<input type="checkbox"/>	4	謝啟鴻	suglin	\$2y\$10\$6uWtdYAT3bWMRk.Aaqiei852Plf.WjeKm8X.Asi0V...	男	1987-08-11
<input type="checkbox"/>	5	蔣志明	shane	\$2y\$10\$PwefN9kxXOKCx59GF6ZJuSGNnIFBY4qDCmCvAwOFI...	男	1984-06-20
<input type="checkbox"/>	6	王佩珊	ivy	\$2y\$10\$RPht3YfaSs0d82inYIK8he.JaPqOrisWMqASuxN5g62...	女	1988-02-15
<input type="checkbox"/>	7	林志宇	zhong	\$2y\$10\$pee.jvO8M4eSKahlc4cLLO9RUMyx8baphyqkSUDwHTNS...	男	1967-05-05
<input type="checkbox"/>	8	李麗茹	lala	\$2y\$10\$oiC9CaGiOdWu.6w5b3.b/Ora8FSuh8Lrbj8Kg5BUPT1...	女	1985-08-30
<input type="checkbox"/>	9	蘇秀英	crystal	\$2y\$10\$8Q0.JEGLRM91qAlMmWnB.wpcY.rjEbgNgV5ntllqZm...	女	1986-12-10
<input type="checkbox"/>	10	張雅琪	peggy	\$2y\$10\$SRNqpxDNHkcT12ZHzbKTKnOeszUFLXhinhT8ZL8OHoMe...	女	1988-12-01

memberdata 資料表：儲存網站會員的資料，欄位的命名都以「m_」為前置字元。本資料表以「m_id」（計數編號）為主索引，並設定為「UNSIGNED」（正數）、「auto_increment」（自動編號），如此即能在新增資料時加上一個單獨的編號而不重複。

1. 會員性別「m_sex」欄位，是單選的列舉資料型態，在這裡設定值為「男」及「女」。
2. 會員等級「m_level」欄位，是單選的列舉資料型態，在這裡設定值為「admin」及「member」，預設值為「member」。
3. 會員生日「m_birthday」欄位因為我們只需要日期的資料，所以設定其欄位的型態為「date」。
4. 會員登入次數「m_login」欄位記錄登入次數，設定其欄位的型態為「int」，預設值為 0。

- 會員本次登入時間「m_logintime」欄位及會員加入時間「m_jointime」欄位，包含了日期與時間的資訊，所以設定其欄位的型態為「datetime」。

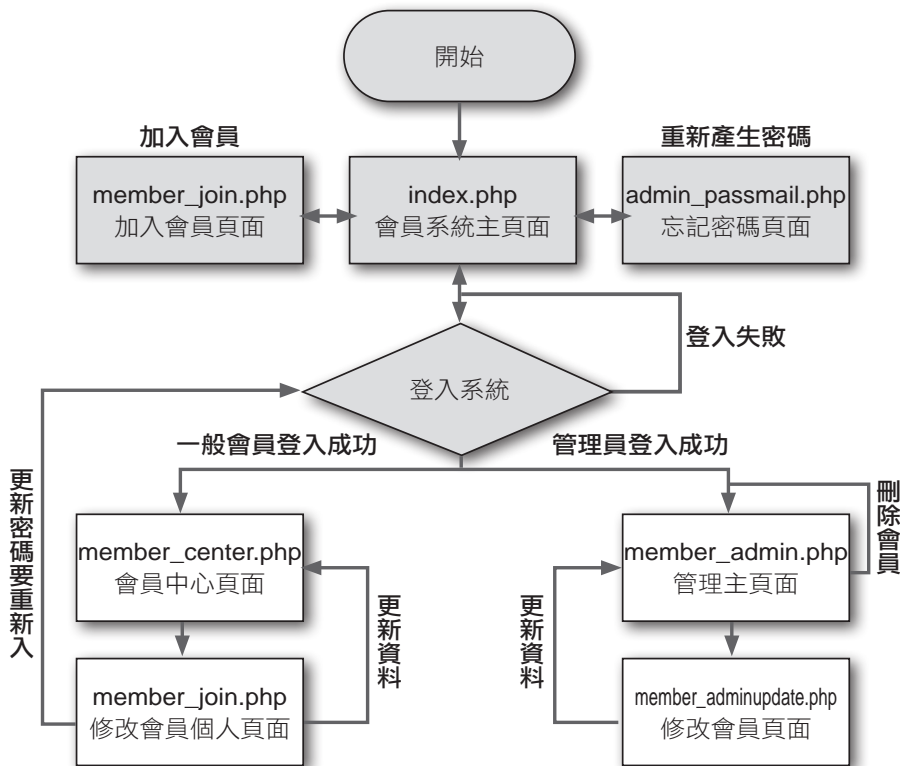
相關資訊整理

最後我們將相關的資訊再做一次整理，讓您在開發時參考：

- 本機伺服器網站主資料夾是 <C:\htdocs>，程式儲存的資料夾為 <C:\htdocs\phpmember>，本章的測試網址會變為：<http://localhost/phpmember/>。
- 目前 MySQL 是架設在本機上，其伺服器位址為：「localhost」，為了安全性考量，我們修改了它的管理帳號：「root」的密碼為：「1234」。
- 目前欲使用的「phpmember」資料庫，並匯入了資料表：「memberdata」。

16.1.3 會員系統程式流程圖分析

以下是整個會員系統程式運作的流程圖：



16.2 Password Hasing 加密函式

面臨越來越多的資安問題，對於程式中重要資訊的加密就相當重要。在 PHP 中雖然已經提供許多加密方式，這裡將要介紹一個強大但相當簡單而實用的函式進行加密動作。

password_hash() 函式

PHP 在過去常使用 MD5 或 SHA1 演算法進行加密的動作，但隨著電腦的運算速度越來越快，這二種方式都很容易遭到破解。新一代的 PHP 推出更理想的加密函式：`password_hash()`，可以輕易產生更難以破解的加密結果。

`password_hash()` 使用強度足夠的單向散列演算法進行資料的加密動作。它的功能是指定字串以選擇的演算法進行加密，成功即回傳加密結果，否則回傳 `False`，格式為：

```
password_hash( 加密字串, 加密演算法 )
```

目前提供了二種加密方式：

1. **PASSWORD_BCRYPT**：使用 BCRYPT 演算法進行加密，會產生 60 個字元長度的密碼，所以在設定儲存欄位長度時要特別注意。
2. **PASSWORD_DEFAULT**：當沒有提供演算法時會以預設的方式進行加密，這是較為推薦的方式，因為這個選項會隨著 PHP 擴充程式的升級調用更好的演算方式進行加密。但目前預設還是使用 BCRYPT 演算法，請注意儲存資料欄位的長度。

password_verify() 函式

使用 `password_hash()` 函式加密的結果，可以利用 `password_verify()` 函式進行比對，成功即回傳 `True`，否則回傳 `False`，格式為：

```
password_verify( 比對字串, 加密字串 )
```

無論加密的方法再好，使用的密碼過於簡單還是容易遭到破解。透過 `password_hash()` 加密後的密碼，因為每次生成的密碼都不一樣，想破解只能採用暴力演算，不易成功，真的是很推薦的解決方案。

16.4 會員系統主頁面的製作

會員系統主頁面第一個功能是負責會員登入，依不同權限能進入不同的頁面，執行不同功能；第二個功能是負責會員申請加入的動作。

在這裡有幾個重要的功能：第一是負責會員登入的首頁，在這個會員系統中我們只有一個登入的窗口，在會員登入後即可以依照其權限進入不同的頁面、使用不同的功能。第二是讓新會員申請加入的頁面，所有瀏覽人都可以藉由這個循環加入到會員系統中。

16.4.1 檢查登入狀態與執行會員登入

在進入會員系統的首頁時，第一件事就是要檢查目前使用者的登入狀態，若是已經登入則會將頁面導向會員中心或是管理會員的頁面。

在這個範例中，我們使用 Session 的 `$_SESSION["loginMember"]` 來記錄登入者的帳號與 `$_SESSION["memberLevel"]` 會員等級，一旦這二個值都不存在即會往下執行。

登入檢查狀態

程式碼：index.php

儲存路徑：C:\htdocs\phpmember

```
1  <?php
2  require _once("connMysql.php");
3  session _start();
4  // 檢查是否經過登入,若有登入則重新導向
5  if(isset($_SESSION["loginMember"]) && ($_SESSION["loginMember"]
   != "")){
6      // 若帳號等級為 member 則導向會員中心
7      if($_SESSION["memberLevel"]=="member"){
8          header("Location: member_center.php");
9      // 否則則導向管理中心
10     }else{
11         header("Location: member_admin.php");
```

```
12 }
13 } ...
```

程式說明

- 2 設定使用資料連線引入檔。
- 3 啟動 Session。
- 5 我們以 `$_SESSION["loginMember"]` 來記錄登入者的帳號，如果沒有登入 Session 值或是 Session 值為空則跳過此區，若有則往下執行程式。
- 7~8 我們以 `$_SESSION["memberLevel"]` 來記錄登入者的會員等級，若會員等級等於「member」則將頁面導向會員中心 `<member_center.php>`。
- 10~11 否則就將頁面導向管理中心 `<member_admin.php>`。

執行會員登入動作

接著要執行會員登入的動作，這裡與其他範例不同的地方在於：登入系統時不僅要檢查帳號、密碼，還要視會員權限等級不同，導向不同的頁面去執行功能。

在資料表中儲存密碼的欄位，其值是經由 `password_hash()` 函式加密後才存入，所以使用表單送出的密碼值也要先經過 `password_verify()` 函式才能進行比對。以下是程式碼的內容：

程式碼：index.php

儲存路徑：C:\htdocs\phpmember

(接續前程式碼)...

```
14 // 執行會員登入
15 if(isset($_POST["username"]) && isset($_POST["passwd"])){
16     // 繫結登入會員資料
17     $query_RecLogin = "SELECT m_username, m_passwd, m_level
    FROM memberdata WHERE m_username=?";
18     $stmt=$db_link->prepare($query_RecLogin);
19     $stmt->bind_param("s", $_POST["username"]);
20     $stmt->execute();
21     // 取出帳號密碼的值
22     $stmt->bind_result($username, $passwd, $level);
23     $stmt->fetch();
24     $stmt->close();
25     // 比對密碼，若登入成功則呈現登入狀態
26     if(password_verify($_POST["passwd"],$passwd)){
```

16.6 會員系統會員中心頁面的製作

會員專區的頁面是提供給所有登入會員使用的，在這個頁面中您可以看到會員的登入資訊，進入修改頁面更改自己的基本資料。

16.6.1 登入檢查及登出管理的製作

程式碼：member_center.php

儲存路徑：C:\htdocs\phpmember

```
1 <?php
2 require _once("connMysql.php");
3 session_start();
4 // 檢查是否經過登入
5 if(!isset($_SESSION["loginMember"])|($_SESSION["loginMember"]
  == "")){
6     header("Location: index.php");
7 }
8 // 執行登出動作
9 if(isset($_GET["logout"]) && ($_GET["logout"]=="true")){
10     unset($_SESSION["loginMember"]);
11     unset($_SESSION["memberLevel"]);
12     header("Location: index.php");
13 } ...
```

程式說明

- 2 設定使用資料連線引入檔。
- 3 啟動 Session。
- 4~7 `$_SESSION["loginMember"]` 是用來記錄登入者的帳號，如果沒有登入 Session 值或是 Session 值為空則代表目前的使用者並沒有經過登入的過程，如此一來就將目前的頁面導向瀏覽的主頁面 `<index.php>`。
- 8~12 當我們接收到 URL 的參數「logout」，而且其值為「true」時就執行登出的動作。這裡使用 `unset()` 函式將 Session 值刪除，再將目前的頁面導向瀏覽的主頁面 `<index.php>`，完成登出的動作。

16.6.2 繫結登入會員資料及顯示

接下來要將繫結登入會員的資料及在頁面上顯示：

```

程式碼：member_center.php          儲存路徑：C:\htdocs\phpmember
( 接續前程式碼 )...
14 // 繫結登入會員資料
15 $query_RecMember = "SELECT * FROM memberdata WHERE m_
    username = '{$_SESSION["loginMember"]}";
16 $RecMember = $db_link->query($query_RecMember);
17 $row_RecMember=$RecMember->fetch_assoc();
18 ?>
略 ...
55          <p class="heading"><strong> 會員系統 </strong></p>
56
57          <p><strong><?php echo $row_RecMember["m_
    name"];?></strong> 您好。</p>
58          <p>您總共登入了 <?php echo $row_RecMember["m_
    login"];?> 次。<br>
59          本次登入的時間為：<br>
60          <?php echo $row_RecMember["m_logintime"];?></p>
61          <p align="center"><a href="member_update.php"> 修改
    資料 </a> | <a href="?logout=true"> 登出系統 </a></p>
62 </div> ... 後略

```

程式說明

- 14** 將登入本頁的會員資料繫結進來，這裡佈置要用的 SQL 指令：選取「memberdata」資料表中所有欄位，並設定篩選條件為欄位「m_username」的值要等於 \$_SESSION["loginMember"] 的值。
- 15~17** 執行 \$db_link->query 方法取得資料放置在 \$RecMember 中。使用 fetch_assoc() 方法取出該會員的資料儲存在 \$row_RecMember 陣列中。
- 57~58** 顯示登入者姓名「m_name」欄位與顯示登入次數「m_login」欄位。
- 60** 顯示本次登入時間「m_logintime」欄位。
- 61** 設定前往 <member_update.php> 的文字連結及設定登出管理頁面的連結：「?logout=true」，當重新進入本頁時會因為 URL 參數與值執行登出管理的動作。

如此即完成會員中心頁面的製作。