第 １３ 章

檔案上傳與

資料庫圖片存取

13-1 檔案上傳

在 PHP 的世界中，檔案上傳已不是件困難的事囉！但請您千萬記得一件事：**表單的資料編碼型態（EncType）一定要設爲『Multipart/Form-data』。**

在表單欄位元件中有一個名爲「File」的元件，它是由「Text」文字欄位與「Button」所組成，當我們按下「瀏覽」按鈕時，就會出現「選擇檔案」視窗讓我們選取本機中的檔案，例如：範例 file.php

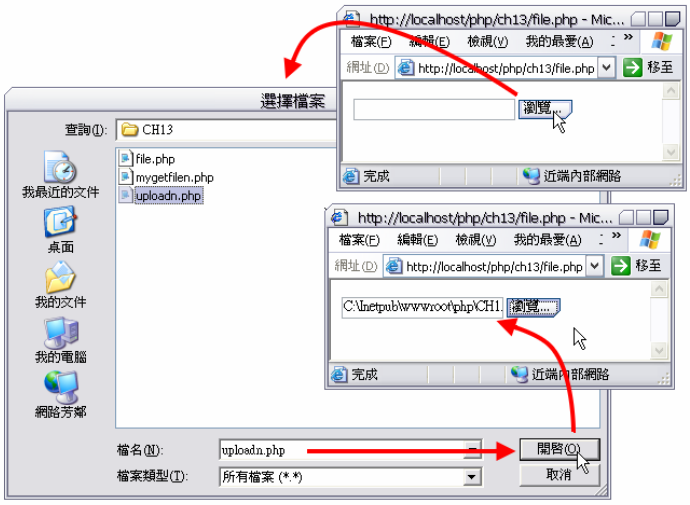

圖13-1　範例file.php。

這個「File」欄位元件就是主角之一，在「選擇檔案」市窗中出現的檔案與目路都是使用者端機器的內容，而非我們的伺服器檔案內容

喔！廢話不多說，我們先來完成基礎的上傳網頁及表單元件佈置，完成畫面如下圖：

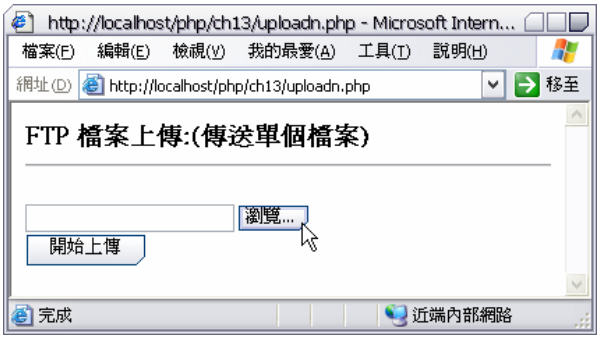

圖13-2 範例uploadn.php。

uploadn.php

```
1.  <HTML><BODY>
2.  <H3>FTP  檔案上傳:(傳送單個檔案)<HR></H3>
3.
4.  <Form Action="mygetfilen.php" Method="POST"
5.  Enctype="multipart/form-data">
6.  <Input Type="File" Name="upfile" ><br>
7.  <Input Type="Submit" value="  開始上傳  ">
8.  </Form>
9.
10. </BODY></HTM>
```

真正處理檔案上傳工作的是「mygetfilen.php」而非上傳表單「uploadn.php」：

mygetfilen.php

```
1.  <HTML>
2.  <head>
3.  <meta http-equiv="Content-Type" content="text/html; charset=big5">
4.  <title>單檔案上傳</title>
```

```
5.  </head>
6.  <BODY><H3>上傳檔案相關資訊：<HR></H3>
7.
8.  <?
9.          echo "<BLOCKQUOTE>";
10.         echo "檔案名稱：" . $_FILES["upfile"]["name"] . "<BR>";
11.         echo "檔案大小：" . $_FILES["upfile"]["size"] . "<BR>";
12.         echo "檔案類型：" . $_FILES["upfile"]["type"] . "<BR>";
13.         echo "暫存檔名：" . $_FILES["upfile"]["tmp_name"] . "<BR>";
14.                 move_uploaded_file($_FILES["upfile"]["tmp_name"], "file\\" .
    $_FILES["upfile"]["name"]);
15.             echo "您所上傳的檔案已儲存爲 " . $_FILES["upfile"]["name"];
16.         echo "</BLOCKQUOTE>";
17. ?>
18.
19. <HR></BODY></HTML>
```

✓ 要取得檔案相關資訊，必須利用「$_FILES」變數陣列：

$_FILE[表單中的 FILE 欄位元件名][上傳的檔案屬性]

上傳的檔案屬性有下列數種：

1. name：上傳檔案的原始名稱。
2. size：上傳檔案的空間大小（檔案大小）。
3. type：上傳檔案的資料類型。
4. tmp_name：PHP 程式接收上傳檔案後，未上傳檔案所訂定的暫時名稱。

✓ move_upload_file()函數：

Move_upload_file() 函數
bool **move_uploaded_file** (string filename, string destination)

因爲檔案上傳後，該檔案將被放至於暫存目錄中，當程式執行完畢後，這個上傳檔案將會被刪除，因此，必須使用 move_upload_file()函數將此上傳檔移到我們指定的存放位置去。

上傳檔案的存放位置，也就是放置上傳檔案的目錄資料夾，其寫入的權限必須打開，這樣 move_upload_file()函數才能順利將此上傳檔移到我們指定的存放位置去。

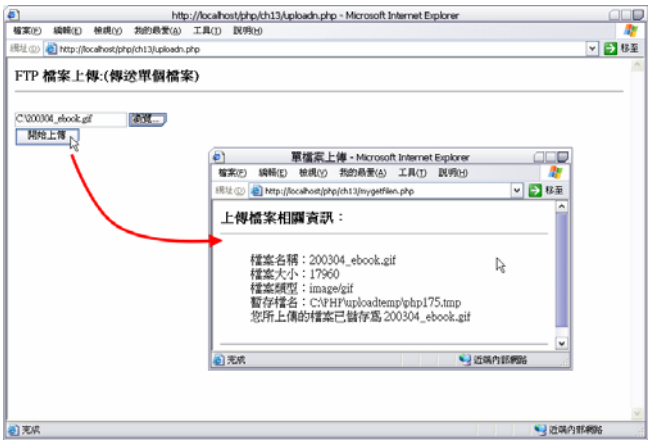

圖13-3 上傳檔案成功。

特別小心：如果上傳檔案的大小是 0，則 move_upload_file()函數在搬移暫存檔到指定位置時將會產生錯誤，所以在 move_upload_file()函數執行之前，，我們應該透過檔案大小的檢驗來預防錯誤：範例 mygetfilen2.php

mygetfilen2.php

```
1. <HTML>
2. <head>
3. <meta http-equiv="Content-Type" content="text/html; charset=big5">
4. <title>單檔案上傳</title>
```

```
5.  </head>
6.  <BODY><H3>上傳檔案相關資訊：<HR></H3>
7.
8.  <?
9.          echo "<BLOCKQUOTE>";
10.         echo "檔案名稱：" . $_FILES["upfile"]["name"] . "<BR>";
11.         echo "檔案大小：" . $_FILES["upfile"]["size"] . "<BR>";
12.         echo "檔案類型：" . $_FILES["upfile"]["type"] . "<BR>";
13.         echo "暫存檔名：" . $_FILES["upfile"]["tmp_name"] . "<BR>";
14.       //檢驗檔案大小是否大於 0
15.       if ( $_FILES["upfile"]["size"] > 0 )
16.         {
17.             move_uploaded_file($_FILES["upfile"]["tmp_name"],   "file\\"   .
    $_FILES["upfile"]["name"]);
18.             echo "您所上傳的檔案已儲存爲  " . $_FILES["upfile"]["name"];
19.         }
20.       else
21.         {
22.           echo "上傳檔案錯誤!您傳送的是空檔案!!";
23.         }
24.         echo "</BLOCKQUOTE>";
25. ?>
26.
27. <HR></BODY></HTML>
```

除了檔案大小不能爲 0 外，我們也應該限制上傳檔案的最大體積，免得我們的伺服器被大型的檔案給塞暴了！範例 mygetfilen3.php

Mygetfilen3.php(節錄)

```
1.        //檢驗檔案大小是否大於 0
2.        if ( $_FILES["upfile"]["size"] <= 0)
3.          {
4.              echo "上傳檔案錯誤!您傳送的是空檔案!!";
5.          }
6.        //檢驗檔案體積是否過大
7.        else if ( $_FILES["upfile"]["size"] > 50000)
```

```
8.          {
9.              echo "上傳檔案錯誤!您傳送的檔案大於 50k!!";
10.         }
11.       else
12.         {
13.             move_uploaded_file($_FILES["upfile"]["tmp_name"],      "file\\"      .
    $_FILES["upfile"]["name"]);
14.             echo "您所上傳的檔案已儲存爲  " . $_FILES["upfile"]["name"];
15.         }
```

13-1-1 多檔案上傳

要做多檔案上傳的程式，應該使用迴圈敘述來簡化程式，如此一來，不管我們任意增加幾個上傳欄位都不必變更程式敘述！

假設我們現在要一次上傳三個檔案，以範例 uploadn.php 爲設計基礎，將表單內容佈置如下圖：範例 uploadm.php

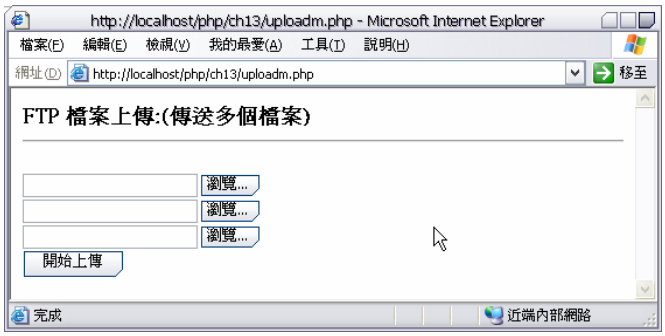

圖13-4 範例uploadm.php。

uploadm.php

```
1.      <HTML><BODY>
2.  <H3>FTP  檔案上傳:(傳送多個檔案)<HR></H3>
3.
```

```
4.  <Form Action="mygetfilem.php" Method="POST"
5.  Enctype="multipart/form-data">
6.  <Input Type="File" Name="upfile[]" ><br>
7.  <Input Type="File" Name="upfile[]" ><br>
8.  <Input Type="File" Name="upfile[]" ><br>
9.  <Input Type="Submit" value="  開始上傳  ">
10. </Form>
11.
12. </BODY></HTM>
```

注意看到表單 FILE 元件的 Name 屬性設定值，她們的名稱都一樣，但是要改變成陣列形式！

現在來看一下處理多檔案上傳的 PHP 程式 mygetfilem.php：

mygetfilem.php

```
1.  <HTML>
2.  <head>
3.  <meta http-equiv="Content-Type" content="text/html; charset=big5">
4.  <title>多檔案上傳</title>
5.  </head>
6.  <BODY><H3>上傳多檔案相關資訊：<HR></H3>
7.
8.  <?
9.  for ( $I=0; $I < (count($_FILES["upfile"])-2); $I++ )
10. {
11.      if ( $_FILES["upfile"]["name"][$I] <> "" )
12.         {
13.           echo "<BLOCKQUOTE>";
14.           echo "檔案名稱：" . $_FILES["upfile"]["name"][$I] . "<BR>";
15.           echo "檔案大小：" . $_FILES["upfile"]["size"][$I] . "<BR>";
16.           echo "檔案類型：" . $_FILES["upfile"]["type"][$I] . "<BR>";
17.           echo "暫存檔名：" . $_FILES["upfile"]["tmp_name"][$I] . "<BR>";
18.         //檢驗檔案大小是否大於 0
19.         if ( $_FILES["upfile"]["size"][$I] <= 0)
20.           {
```

```
21.                 echo "上傳檔案錯誤!您傳送的是空檔案!!";
22.                 echo "</BLOCKQUOTE>";
23.             }
24.         //檢驗檔案體積是否過大
25.         else if ( $_FILES["upfile"]["size"][$I] > 50000)
26.             {
27.                 echo "上傳檔案錯誤!您傳送的檔案大於 50k!!";
28.                 echo "</BLOCKQUOTE>";
29.             }
30.         else
31.             {
32.                 move_uploaded_file($_FILES["upfile"]["tmp_name"][$I],  "file\\"  .
    $_FILES["upfile"]["name"][$I]);
33.                 echo  "您所上傳的檔案已儲存爲  "  .
    $_FILES["upfile"]["name"][$I];
34.                 echo "</BLOCKQUOTE>";
35.             }
36.         }
37. }
38. ?>
39.
40. <HR></BODY></HTML>
```

在上列程式碼中，我們利用$_FILES 陣列變數取得每一個欄位內的資料，而我們佈置的三個「FILE」表單元件就成了陣列集合，索引編號由 0 開始，所以$_FILES[“upfile”][][0]、$_FILES[“upfile”][][1]、$_FILES[“upfile”][][2]，就等於 upfile[0]、upfile[1]、upfile[2]。

而迴圈敘述的終止值，我們是利用 count() 函數取得「$_FILES["upfile"]」檔案陣列集合數目，由程式自動判斷何時該離開迴圈，就因爲這樣，所以不管如何的增減表單中的「FILE」表單元件數量，我們的程式敘述都是不需變動的！

在單上傳檔案的處理時，上傳檔案的相關資訊就以二維陣列的方式來處理，例如檔案大小：「$_FILES["upfile"]["size"]」，那現在有多個檔案要處理，因此，上傳檔案的相關資訊就以三維陣列的方式來處理了！

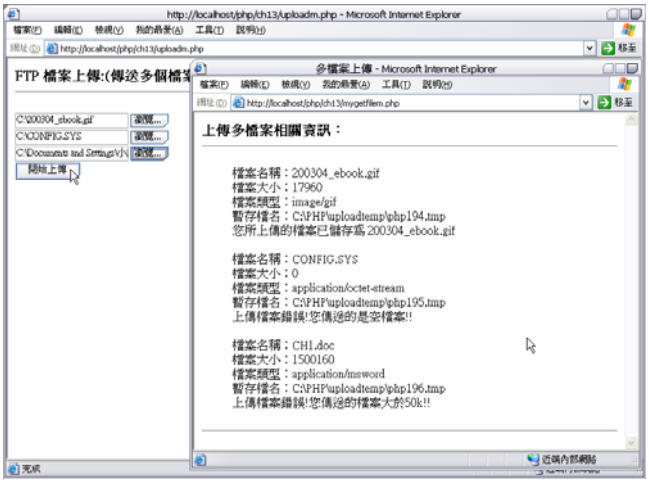

圖13-5　多檔案上傳。

13-2　資料庫圖片存取

不知道爲什麼，大家似乎對於如何將如片檔案塞到資料庫很感興趣！因爲一旦將圖片檔放入資料庫後，不僅資料庫的負擔變的很重，而且，若要對圖片檔加以編修是很困難的。

既然大家想知道，小誌就介紹如何將圖片檔存入資料庫吧！