

Chapter

5

Dreamweaver設計檔案上傳

技能指標

1. 了解流程控制的函數目的
2. 了解函數的各種不同應用
3. 建立檔案上傳頁面

學習總結

完成本章後、您將了解程式語中函數的各種不同運用，讓您可以正確的了解何時該使用函數、如何使用函數協助您達成所需要的功能。

了解流程控制的函數目的

函數在整個程式語言中占有非常重要的角色，因為透過函數的搭配，不僅可以大幅提高程式碼的再利用度，更可以減化程式的重複出現增加程式的可讀性…等，這些都是透過函數所帶來的好處。

「函數」它到底是什麼呢？請看底下的說明，希望您夠了解。

● 劇情

巨匠小學六年一班、班上有15同學，其中風紀股長負責班上秩序，衛生股長負責班上的環境整潔，學藝股長負責班上的作業繳交，服務股長負責班上的營養午餐…等，每一位股長都負責者不同的任務。

● 說明

我們可以假設每一位「股長」就像是每一個「函數」，它們都負責者各種不同的任務，每位同學可以依照不同的需求，尋找不同的股長協助處理相關的工作，就好像我們可以依照每一種不同的需求，分別呼叫不同的函數處理不同的工作內容；這樣不就會容易的將各種不同的工作分類，而且可以重複利用這些函數了呢？

接下來就讓我們來比較一下在沒有使用函數以及使用函數後，在程式設計中會有什麼樣的差異，希望可以讓您了更清楚的了解。

在下面的例子當中，有三位同學要進行分數判斷，分別列出「及格」、「可以補考」、「明年再來」三個等級的檢查；在未使用函數的情況，必須每次重複將檢查的程式碼邏輯再重新寫一次；但是在使用函數的情況下，我們只需要每次呼叫函數即可進行檢查；那麼試想一下，如果班上有50位同學的情況，未使用函數以及使用函數的情況，哪一個比較簡單呢？

未使用函數

```

8 <?php
9 $name="陳桶一";
10 $score=76;
11 if($score >= 60){
12     echo "$name, 您及格了!<br/>";
13 }else if($score >= 50){
14     echo "$name, 您可以補考!<br/>";
15 }else{
16     echo "$name, 明年再來!<br/>";
17 }
18 echo "<br/>";
19 $name="黃光權";
20 $score=49;
21 if($score >= 60){
22     echo "$name, 您及格了!<br/>";
23 }else if($score >= 50){
24     echo "$name, 您可以補考!<br/>";
25 }else{
26     echo "$name, 明年再來!<br/>";
27 }
28 echo "<br/>";
29 $name="李榮富";
30 $score=58;
31 if($score >= 60){
32     echo "$name, 您及格了!<br/>";
33 }else if($score >= 50){
34     echo "$name, 您可以補考!<br/>";
35 }else{
36     echo "$name, 明年再來!<br/>";
37 }
38 echo "<br/>";
39 ?>

```

圖5-1 未使用函數



圖5-2 執行結果

使用函數

```

9 <?php
10 $name="陳桶一";
11 $score=76;
12 scorepass(); //呼叫 scorepass() 函數
13 $name="黃光權";
14 $score=49;
15 scorepass(); //呼叫 scorepass() 函數
16 $name="李榮富";
17 $score=58;
18 scorepass(); //呼叫 scorepass() 函數
19 function scorepass(){
20     global $name,$score;
21     if($score >= 60){
22         echo "$name, 您及格了!(function)<br/>";
23     }else if($score >= 50){
24         echo "$name, 您可以補考!(function)<br/>";
25     }else{
26         echo "$name, 明年再來!(function)<br/>";
27     }
28     echo "<br/>";
29 }
30 }
31 ?>

```

圖5-3 已使用函數



圖5-4 執行結果

聰明的您一定答對了，使用函數的確會簡單的非常多，就像班上有50位同學，要繳交作業，在沒有學藝股長的情況，每一位同學要自己把作業繳交給老師，但是如果有了學藝股長後，同學就只要把作業交給學藝股長就可以了，這樣一來不但省時而且又方便。

了解函數的各種不同應用

在了解函數的好處及用途後，底下就讓我們一起來了解一下，函數在PHP當中的各種不同語法的介紹及說明。

語法：	舉例說明：
<pre>function foo(\$arg1,\$arg2,...\$arg_n){ 連續動作; }</pre>	<pre>function 分數檢查(\$姓名,\$分數){ if(\$分數 >= 60){ echo "\$姓名, 及格了! "; }else if(\$分數 >= 60){ echo "\$姓名, 可以補考! "; }else{ echo "\$姓名, 明年再來! "; } }</pre>

應用一》不接收參數不回傳值

說明：透過不斷的呼叫函數可以達到重複執行相同的動作。

```

8  <?php
9  compute();//呼叫函數一次
10 compute();//呼叫函數一次
11 compute();//呼叫函數一次
12 function compute(){
13     $sum = 0;
14     for($i=1; $i<=10; $i++){
15         $sum += $i;
16     }
17     echo "sum= $sum<br/>";
18 }
19 ?>
```

圖5-5 不接收參數不回傳值



圖5-6 執行結果

應用二》接收單一參數不回傳值

說明：當呼叫方法需要傳遞參數時，可以在()當中宣告一個變數，即可以接收呼叫者傳遞參數給函數。

```

8  <?php
9  compute(15); //呼叫函數,傳遞1個整數參數15
10
11  function compute($max){ //被呼叫者
12      $sum = 0;
13      for($i=1; $i<=$max; $i++){
14          $sum += $i;
15      }
16      echo "<B>1+2+3+...+$max= $sum</B><br/>";
17  }
18  ?>

```

圖5-7 接收單一參數不回傳值

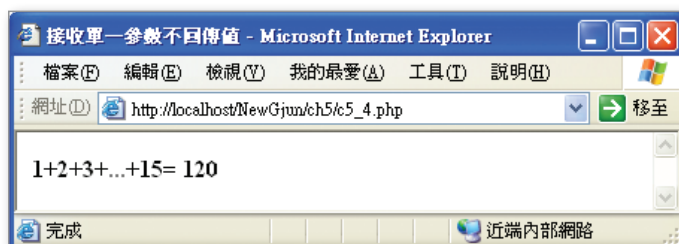


圖5-8 執行結果

應用三》接收多參數不回傳值

說明：當需要傳遞的傳參數超過一個以上時，請依序在()當中宣告即可，當然傳遞的參數也必須要搭配才可正確使用。

```

8  <?php
9  compute(5,15); //呼叫函數,傳遞2個整數參數5以及15
10
11  function compute($start,$max){
12      $sum = 0;
13      for($i=$start; $i<=$max; $i++){
14          $sum += $i;
15      }
16      echo "<B>$start+" . ($start+1) . "+" . ($start+2) . "...+$max= $sum</B><br/>";
17  }
18  ?>

```

圖5-9 接收多參數不回傳值

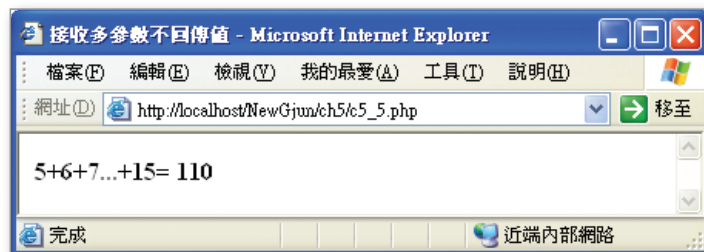


圖5-10 執行結果

應用四》接收多參數回傳一個字串

說明：回傳參數可以是任意類型，不過僅能夠回傳的數量只有一個。

```

8  <?php
9  //呼叫函數,傳遞2個整數參數7以及20, 並且接收回傳字串輸出
10 echo compute(7,20);
11
12 function compute($start,$max){
13     $sum = 0;
14     for($i=$start; $i<=$max; $i++){
15         $sum += $i;
16     }
17     return "<B>$start+" . ($start+1) . "+" . ($start+2) .
18         "...+$max= $sum</B><br/>";
19 }
20 ?>

```

圖5-11 接收多參數回傳一個字串

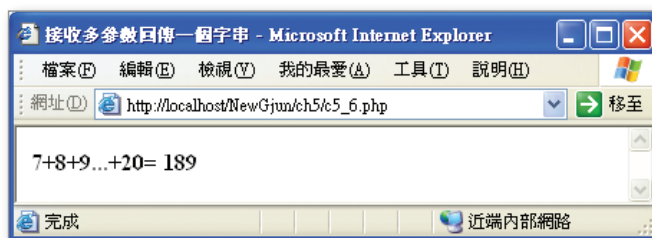


圖5-12 執行結果

應用五》搭配global

說明：當使用到未在function當中所宣告的變數時，就必透過global進行登記。

```

<?php
$sum = 100;
echo compute(1,5); //呼叫函數,傳遞2個整數參數1以及5, 並且接收回傳字串輸出

function compute($start,$max){
    global $sum; //在函中登記一個全域變數, 名稱 $sum
    for($i=$start; $i<=$max; $i++){
        $sum += $i;
    }
    return "<B>$start+" . ($start+1) . "+" . ($start+2) . "...+$max= $sum</B><br/>";
}
?>

```

圖5-13 搭配global



圖5-14 執行結果

應用六 選擇性參數

說明：呼叫函數時，有些參數不一定需要進行傳遞，那麼一來即可透過預設值，來取代呼叫者未傳遞的參數。

```

9  <?php
10 compute(1,15); //呼叫函數,傳遞2個整數參數1以及15
11 compute(1); //呼叫函數,只傳遞1個整數參數1
12
13 function compute($start,$max=20){
14     $sum = 0;
15     for($i=$start; $i<=$max; $i++){
16         $sum += $i;
17     }
18     echo "<B>$start+" . ($start+1) . "+" . ($start+2) .
19         | "...+$max= $sum</B><br/>";
20 }
21 ?>

```

圖5-15 選擇性參數

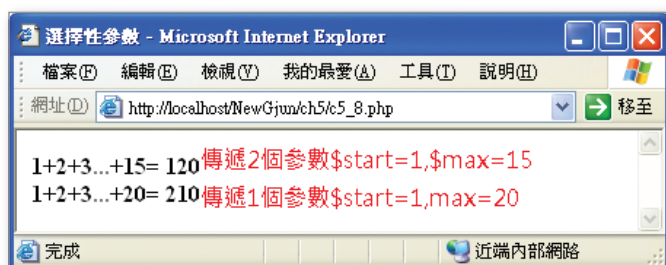



圖5-16 應用六執行結果

■ 單一檔案上傳製作

這個單元，我們要來製作一個具備有檔案上傳功能的頁面，網路上的訪客可以透過這個頁面，能夠提供資料上傳、並且分享。

網頁一共分為2頁，Upload.php接受上傳檔案的選取操作，UploadProc.php則負責用來接著上傳後將檔案複製到正確位置存放。好了接下來讓我們來完成這個單一檔案上傳的製作。

01
STEP

請在「檔案」頁籤「ch5」資料夾中建立2個新的頁面，並且分別將其命名為Upload.php、UploadProc.php。接著開啓Upload.php後在  設計 模式中，並完成底下步驟：

1. 請點選「插入(I)」→「表單(F)」→「表單(F)」。
2. 在表單中輸入提示訊息「請選擇上傳檔案：」。
3. 請點選「插入(I)」→「表單(F)」→「隱藏欄位(H)」、並且設定名稱為MAX_FILE_SIZE、值為2097152Bytes：它必須是表所有欄位的第一個位置。

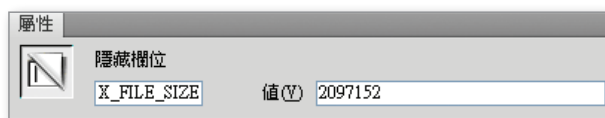


圖5-17 隱藏欄位屬性

4. 請點選「插入(I)」→「表單(F)」→「檔案欄位(F)」、並且設定名稱為UserFile。
5. 請點選「插入(I)」→「表單(F)」→「按鈕(B)」。



圖5-18 檔案上傳

02
STEP

接著點選「<form>」的標記後在開啓屬性視窗，並且依序設定底下2個選項。

1. 將表單「動作」設為「UploadProc.php」。
2. 確認表單「方法」為「POST」。

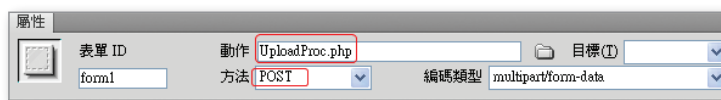



圖5-19 表單屬性

03
STEP

完成了操作介面後，接著來處理後端接收的部份，請開啓UploadProc.php後請切換至  程式碼 模式中。並將鍵盤游標停駐在</html>標記後，請接著加入第一段程式碼，提供顯示上傳詳細訊息的函數。

函數showDetail(\$file)：該函數可以用來將\$_FILE['???']接收前一個頁所送出的資料，並且將接收到的資料顯示在頁面上，在這個範例中我們搭配了一個foreach迴圈。

foreach語法一：

foreach(\$陣列變數 as \$值變數)

功能：可以讀取陣列所有元素內容。

foreach語法二：

foreach(\$陣列變數 as \$鍵值變數 -> \$值變數)

功能：可以讀取陣列中「元素名稱」與「元素內容」。

```

11 </html>
12 <?php
13 /*顯示上傳檔案的詳細資料，包括了「名稱」、「類型」、
14  *「暫存檔名」、「錯誤代碼」、「檔案大小」*/
15 function showDetail($file){
16     foreach($file as $key => $value){
17         echo "<font color=red><B>$key:</B></font> $value<br/>";
18     }
19     echo "<hr/>";
20 }

```

圖5-20 程式片段1

04
STEP

接著再依序加入第2段程式碼，提供複製檔案功能的函數。

函數fileCopy(\$file)：該函數可以用來將上傳後的暫存檔案複製到欲存放的位置。

copy函數說明：

bool copy(string \$source, string \$dest)

功能：可以將\$source(來源)複製到\$dest(目的地)

在程式碼第24列中的目的位置，表示上傳的檔案會放該程式下，「upload/」資料夾當中，該資料夾必須自行建立。

```

22 //將上傳至暫存區中的檔案存至實際欲存放的目錄下
23 function fileCopy($file){
24     if(copy($file['tmp_name'], "upload/userupload.zip")){
25         echo "<B>檔案上傳成功!</B>";
26     }else{
27         echo "<B>錯誤：無法複製檔案...</B>";
28     }
29 }

```

圖5-21 程式片段2

CHAPTER
01

CHAPTER
02

CHAPTER
03

CHAPTER
04

CHAPTER
05

CHAPTER
06

CHAPTER
07

CHAPTER
08

CHAPTER
09

CHAPTER
10

CHAPTER
11

CHAPTER
12

CHAPTER
13

CHAPTER
14

CHAPTER
15

05
STEP

接著再依序加入第3段程式碼，提供將複製到目的地後的暫存檔永久性刪除的函數。

函數deleteFile(\$file)：該函數可以用來將上傳後的暫存檔案進行刪除，這樣才不會留在伺服器中暫空間。

unlink函數說明：

book unlink(string \$filename)

說明：可以將\$filename所指定的檔案進行刪除。

```
31
32 // 將檔案複製至實際目錄後，接著將暫存檔案刪除
33 function deleteFile($file){
34     unlink($file['tmp_name']);
35 }
```

圖5-22 程式片段3

06
STEP

接著回到<body>標記後，加入程式碼片段4，提供處理上傳呼叫函數的先後順序。

```
8 <body>
9 <?php
10 //透過$_FILES['xxx']讀取上傳檔案資料
11 $userfile = $_FILES["UserFile"];
12 showDetail($userfile);
13 fileCopy($userfile);
14 deleteFile($userfile);
15 ?>
```

圖5-23 程式片段4

07
STEP

按下「F12」執行，讓我們來測試結果是否正確，首先按下 **瀏覽...** 選擇上傳檔案。



圖5-24 執行結果

挑選到欲上傳檔案後，請按下「開啓(O)」繼續。

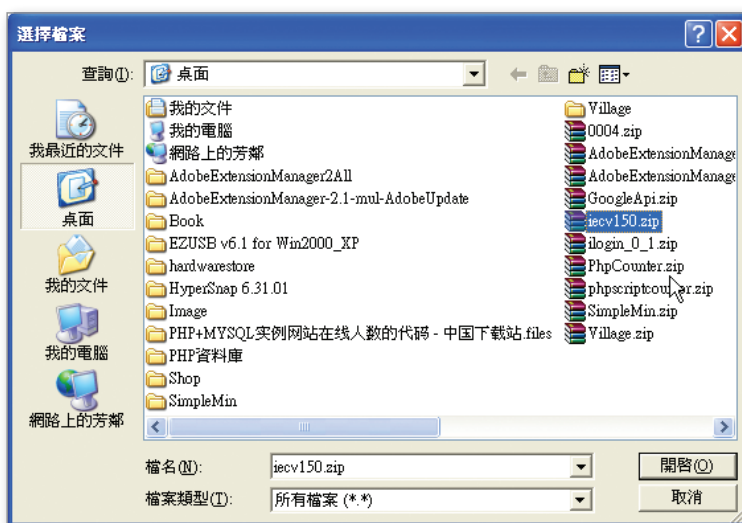


圖5-25 選取上傳檔案

按下「送出」按鈕後，完成檔案上傳動作。



圖5-26 執行結果

多檔案上傳製作

在設網頁時，可能會遇到一次想要上傳多檔案，那麼在PHP中該如何實現呢？接著下來、我們要來製作一個具備有多檔案上傳功能的頁面。

網頁一共分為2頁，MultiUpload.php接受上傳檔案的選取操作，MultiUploadProc.php則負責用來接著上傳後將檔案複製到正確位置存放。好了接下來讓我們來完成這個多檔案上傳的製作。

01
STEP

請在「檔案」頁籤「ch5」資料夾中建立2個新的頁面，並且分別將其命名為MultiUploadProc.php、MultiUploadProc.php。接著開啓MultiUpload.php後在

 設計 模式中，並完成底下步驟：

1. 請點選「插入(I)」→「表格(T)」、設定「4列2欄」。
2. 請點選「插入(I)」→「表單(F)」→「表單(F)」。
3. 在將第1列合併，並且在該欄中輸入提示訊息「請選擇上傳檔案：」。
4. 請點選「插入(I)」→「表單(F)」→「隱藏欄位(H)」、並且設定名稱為MAX_FILE_SIZE、值為2097152Bytes：它必須是表所有欄位的第一個位置。

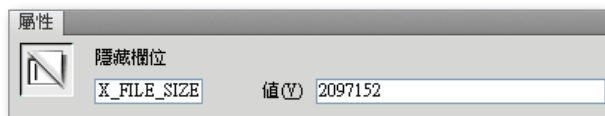


圖5-27 隱藏欄位屬性

5. 接著在表格第2列第1欄輸入「檔案一：」，接著點選第2列第2儲存格，接著點選「插入(I)」→「表單(F)」→「檔案欄位(F)」、並且設定名稱為UserFile[]，採用陣列格式傳遞資料。
6. 接著在表格第3列重複前一個步驟。
7. 請點選「插入(I)」→「表單(F)」→「按鈕(B)」。

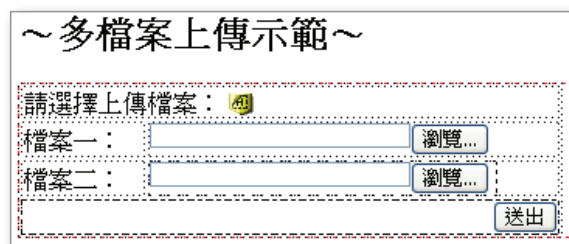


圖5-28 檔案上傳

02 STEP 接著點選「<form>」的標記後在開啓屬性視窗，並且依序設定底下2個選項。

1. 將表單「動作」設為「MultiUploadProc.php」。
2. 確認表單「方法」為「POST」。

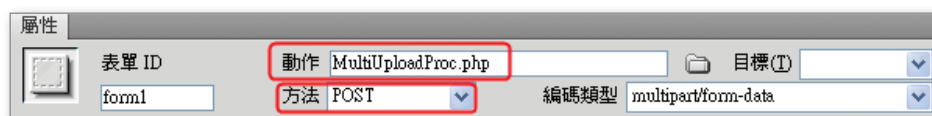


圖5-29 表單屬性

03 STEP 完成了操作介面後，接著來處理後端接收的部份，請開啓MultiUploadProc.php後請切換至 **程式碼** 模式中。並將鍵盤游標停駐在</html>標記後，請接著加入第一段程式碼，提供顯示所有上傳檔案詳細訊息的函數。

因為上傳頁面採用陣列資料格式，因此在透過\$_FILES['???']接收資料時，將會形成二維陣列資料格式。

參數說明：\$msg動態表示檔案上傳的詳細資訊5大資訊。

\$value顯示際的詳細資訊內容。

```

11 </html>
12 <?php
13 /*顯示上傳檔案的詳細資料，包括了「名稱」、「類型」、
14    「暫存檔名」、「錯誤代碼」、「檔案大小」*/
15 function showDetail($msg,$file){
16     foreach($file as $key => $value){
17         echo "<font color=red><B>$msg $key:</B>";
18         echo "</font> $value<br/>";
19     }
20     echo "<hr/>";
21 }

```

圖5-30 程式碼片段1

04 STEP 接著再依序加入第2段程式碼，提供複製檔案功能的函數。

參數說明：\$key用來表示上傳序號（預設值0）。

\$file表示上傳的暫存檔案名稱所在位置。

```

23 //將上傳至暫存區中的檔案存至實際欲存放的目錄下
24 //$key 表示檔案的序號， $file 表示暫存檔的來源名稱
25 //寫至目的地的檔名為「userupload?.zip」
26 function fileCopy($key,$file){
27     if(copy($file,"upload/userupload$key.zip")){
28         echo "<B>檔案編號$key 上傳成功!</B><BR/>";
29     }else{
30         echo "<B>錯誤：無法複製檔案...</B><BR/>";
31     }
32 }

```

圖5-31 程式碼片段2

05 STEP 接著再依序加入第3段程式碼，提供將複製到目的地後的暫存檔永久性刪除的函數。

參數說明：**\$file**表示原始上傳

```
34 // 將檔案複製至實際目錄後，接著將暫存檔案刪除
35 function deleteFile($file){
36     unlink($file);
37 }
```

圖5-32 程式碼片段3

06 STEP 接著回到<body>標記後，加入程式碼片段4，提供處理上傳呼叫函數的先後順序。

```
8 <body>
9 <?php
10 //透過$_FILES['xxx']讀取上傳檔案資料(二維陣列格式)
11 $userfile = $_FILES['userFile'];
12
13 //顯示詳細資訊
14 foreach($userfile as $key => $value){
15     showDetail($key,$userfile[$key]);
16 }
17
18 //複製至目的地
19 foreach($userfile['tmp_name'] as $key => $value){
20     if($value!=NULL){
21         fileCopy($key,$value);
22     }
23 }
24 //刪除暫存檔
25 foreach($userfile['tmp_name'] as $key => $value){
26     if($value!=NULL){
27         deleteFile($value);
28     }
29 }
30 ?>
```

圖5-33 程式碼片段4

07 STEP 按下「F12」執行，讓我們來測試結果是否正確，首先按下 **瀏覽...** 選擇上傳檔案。



圖5-34 執行結果

挑選到欲上傳檔案一，接著請按下「開啓(O)」繼續同樣的步驟。

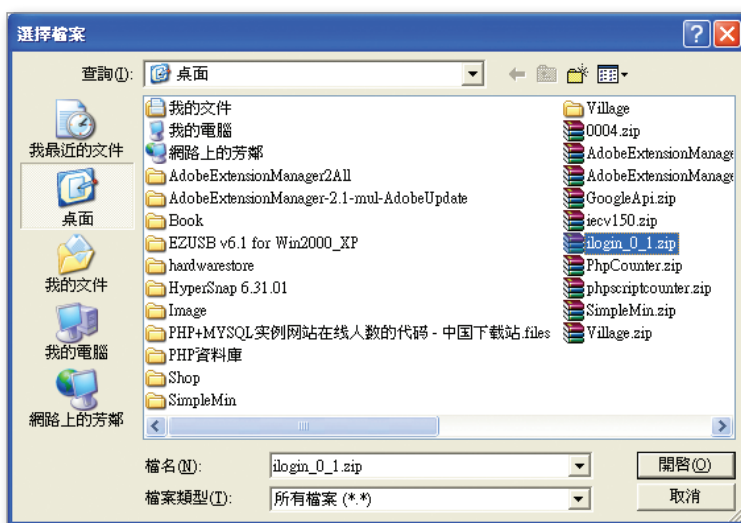


圖5-35 挑選檔案

按下「送出」按鈕後，完成檔案上傳動作。

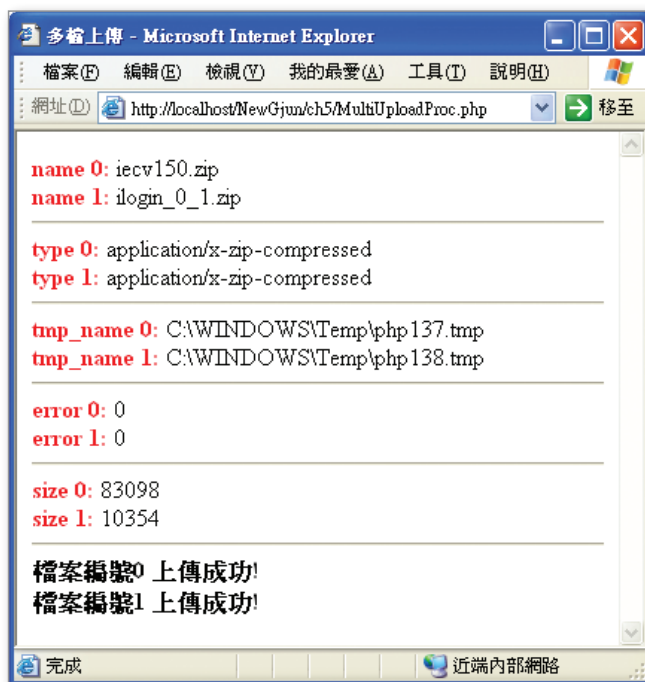


圖5-36 執行結果

在這個範例當中，PHP藉由陣列的輔助提供了多檔上傳的功能，因此若是當我們需要上傳4個檔案時，我們只需要修改前端的操作介面，後端的部份完全不需要修改，所以只要增加4個檔案欄位即可完成4個檔案上傳。

自我測驗 & 能力檢核

完成每一項測驗後，請在□中打勾、代表完成您已了解。

☆函數控制—函數綜合練習、請試著寫出底下每道練習結果 ☐我已了解

1. 撰寫一程式，可以接受2個參數，分別為「最大值」與「最小值」，並且從最大值累加至最小值。

建議事項：請參考圖5-5之範例程式碼

2. 撰寫一程式，可以接受3個參數，分別為「最大值」與「最小值」以及「辨識奇/偶數累加、並且同設定該變數預設值」，最後依照傳遞之參數、決定奇數或偶數還是連序數字、由最大值累加至最小值。

建議事項：請參考圖5-9之範例程式碼