檔案處理

檔案路徑及資訊

$\_SERVER預設變數可以顯示伺服器的相關資訊

PHP\_SELF-目前網頁的虛擬路徑

SCRIPT\_FILENAME-目前網頁的實際路徑

也可以用\_ \_FILE\_ \_顯示實際路徑

例：

echo $\_SERVER["PHP\_SELF"]."<br/>";

echo $\_SERVER["SCRIPT\_FILENAME"]."<br/>";

echo \_\_FILE\_\_;

結果：



realpath()：取得實際路徑

語法為：realpath(“相對路徑”)

由路徑中取得資料夾及名稱dirname()與basename()

dirname(檔案路徑) //由完整路徑中去除檔名的路徑

basename(檔案路徑 [,副檔名]) //由完整路徑中取得檔案名稱，設定副檔名還

可以去除副檔名

也可以用\_ \_DIR \_ \_顯示目前頁面的實際路徑

例：

echo "目前檔案所在的路徑為:".realpath(".")."<br/>";

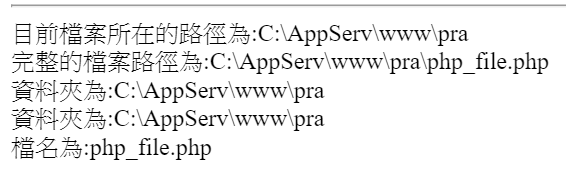
echo "完整的檔案路徑為:".\_\_FILE\_\_."<br/>";

echo "資料夾為:".dirname(\_\_FILE\_\_)."<br/>";

echo "資料夾為:".\_\_DIR\_\_."<br/>";

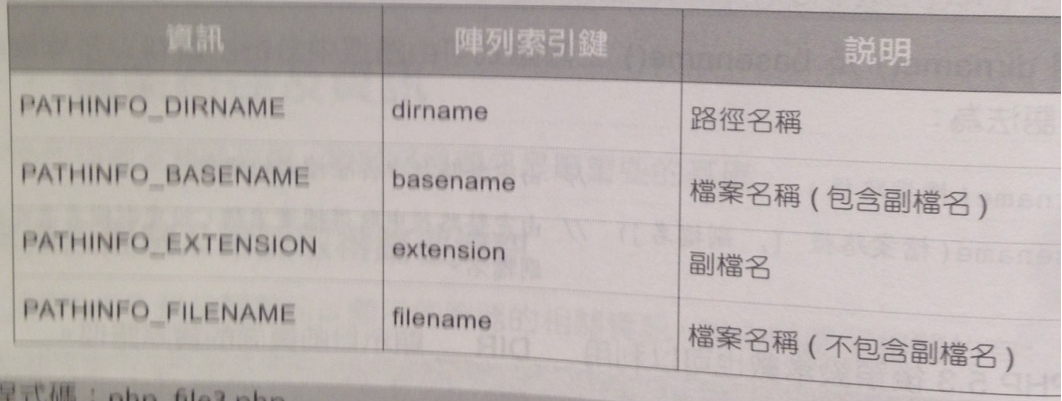
echo "檔名為:".basename(\_\_FILE\_\_)."<br/>";

結果：



pathinfo()取得路徑資訊

pathinfo(檔案路徑 [,指定資訊])



例：

$path=\_\_FILE\_\_;

echo pathinfo($path,PATHINFO\_DIRNAME)."<br/>";

echo pathinfo($path,PATHINFO\_BASENAME)."<br/>";

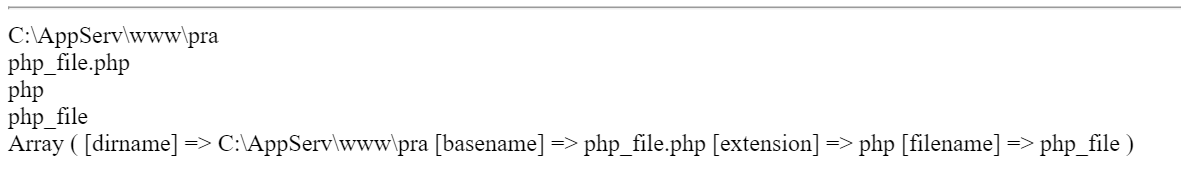
echo pathinfo($path,PATHINFO\_EXTENSION)."<br/>";

echo pathinfo($path,PATHINFO\_FILENAME)."<br/>";

$pathData=pathinfo($path);

print\_r($pathData);

結果：



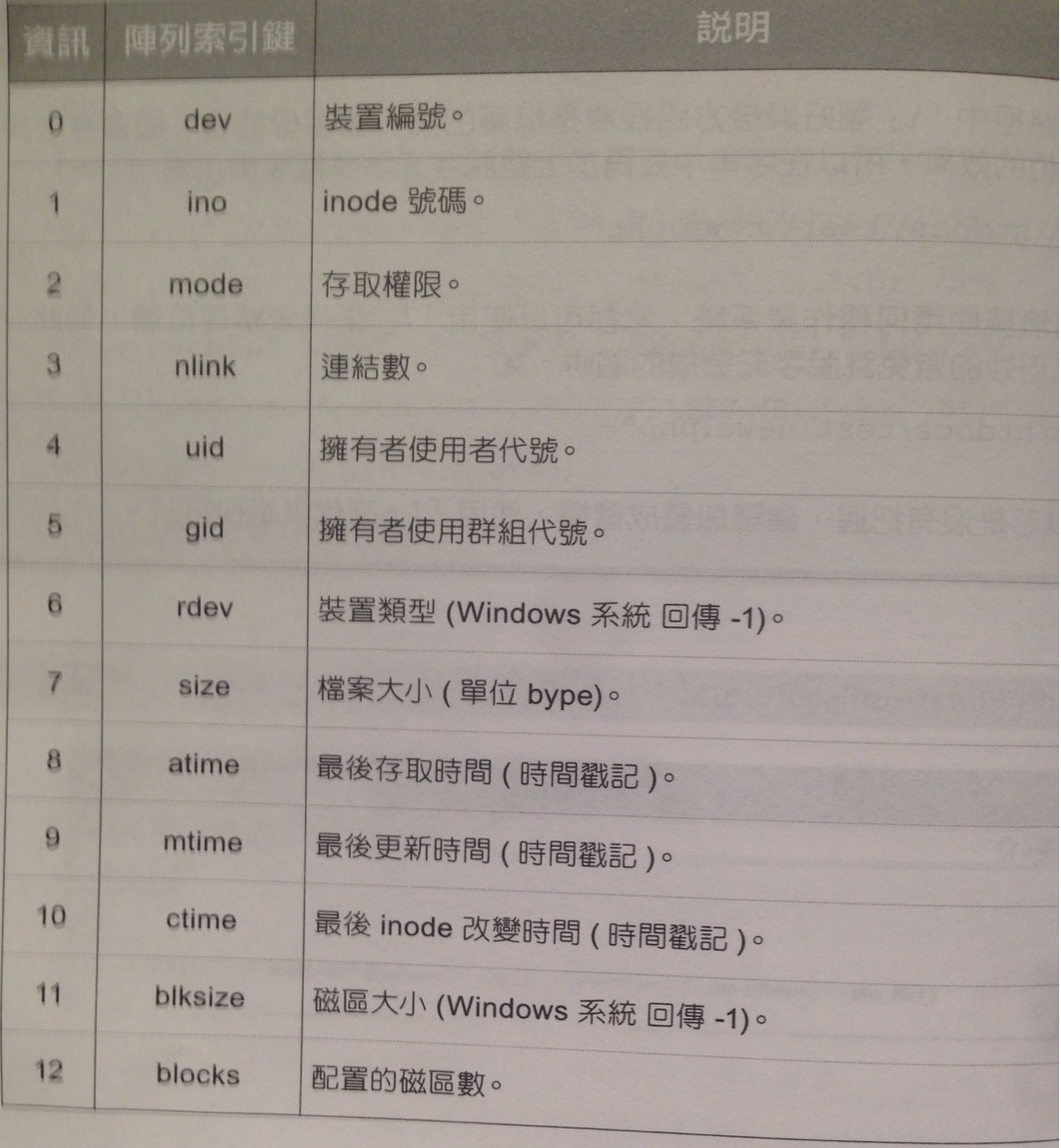
取得檔案的詳細資訊stat()與lstat()

stat(檔案路徑);

lstat(檔案路徑);

兩個函式的使用方法相同，差異在於當檔案路徑是符號連結時，lstat()函式會傳回符號連結本身的資訊

傳回的陣列元素內容：



例：

$path=\_\_FILE\_\_;

clearstatcache();

$pathData = stat($path);

print\_r($pathData);

echo "<hr/>";

$pathData = lstat($path);

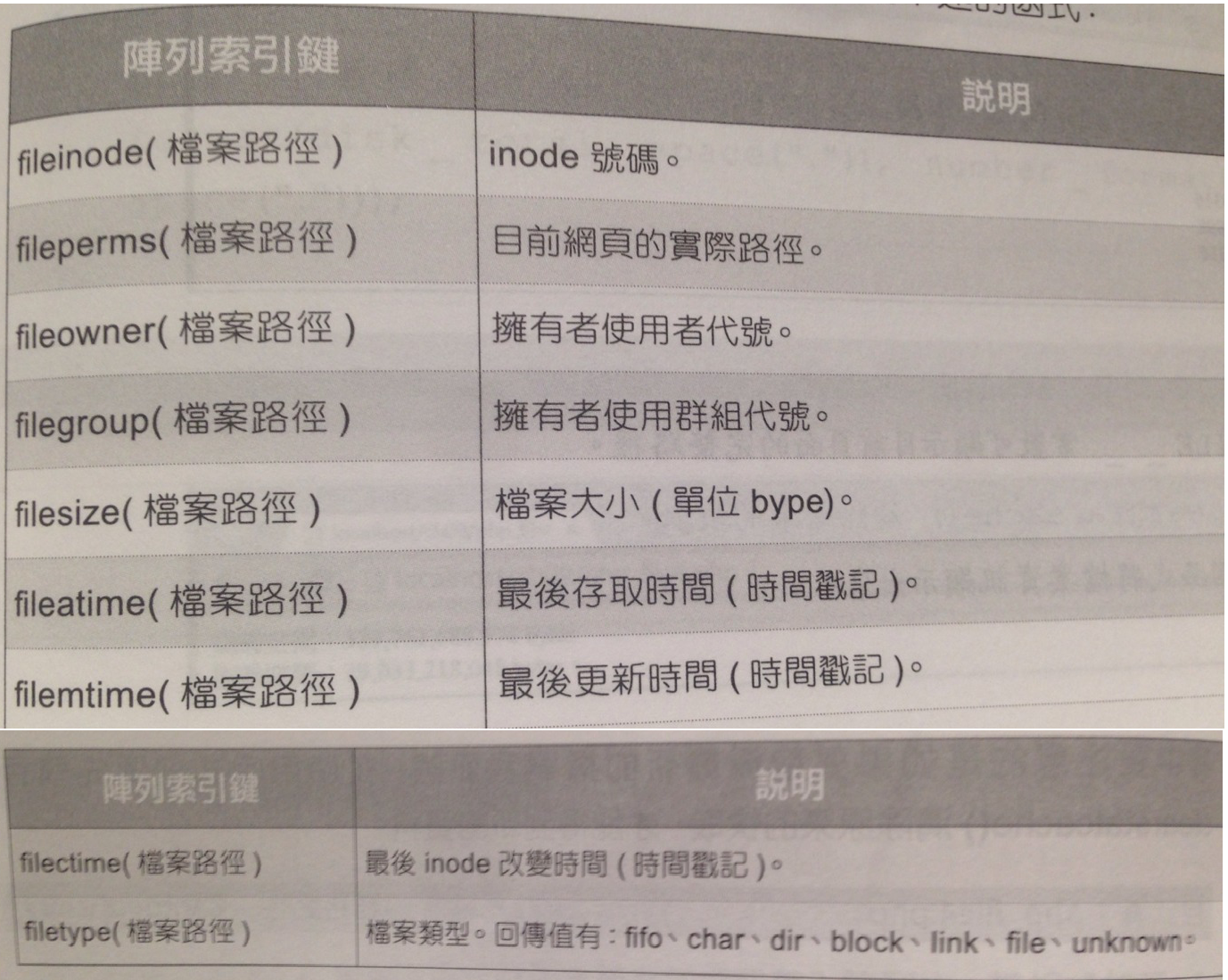
print\_r($pathData);

執行前要先執行clearstatcache()清除原來的快取，才能取得新的資料

結果：



個別取得檔案資訊



例：

$path=\_\_FILE\_\_;

clearstatcache();

echo fileinode($path)."<br/>";

echo fileperms($path)."<br/>";

echo fileowner($path)."<br/>";

echo filegroup($path)."<br/>";

echo filesize($path)."<br/>";

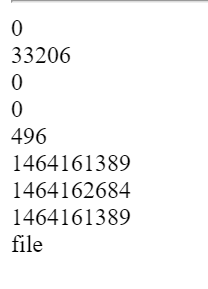
echo fileatime($path)."<br/>";

echo filemtime($path)."<br/>";

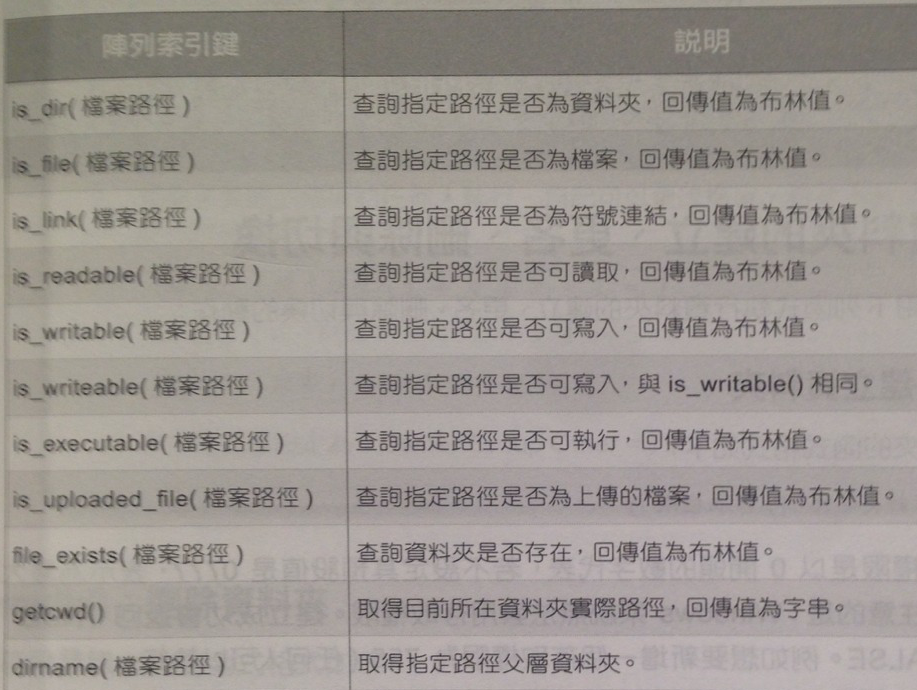
echo filectime($path)."<br/>";

echo filetype($path)."<br/>";

結果：



查詢檔案及資料夾屬性



查詢磁碟空間及剩餘空間 disk\_total\_space()

例：

printf("磁碟空間 : %s bytes <br/> 剩餘空間 : %s bytes",number\_format(disk\_total\_space("."))

,number\_format(disk\_free\_space(".")));

結果：



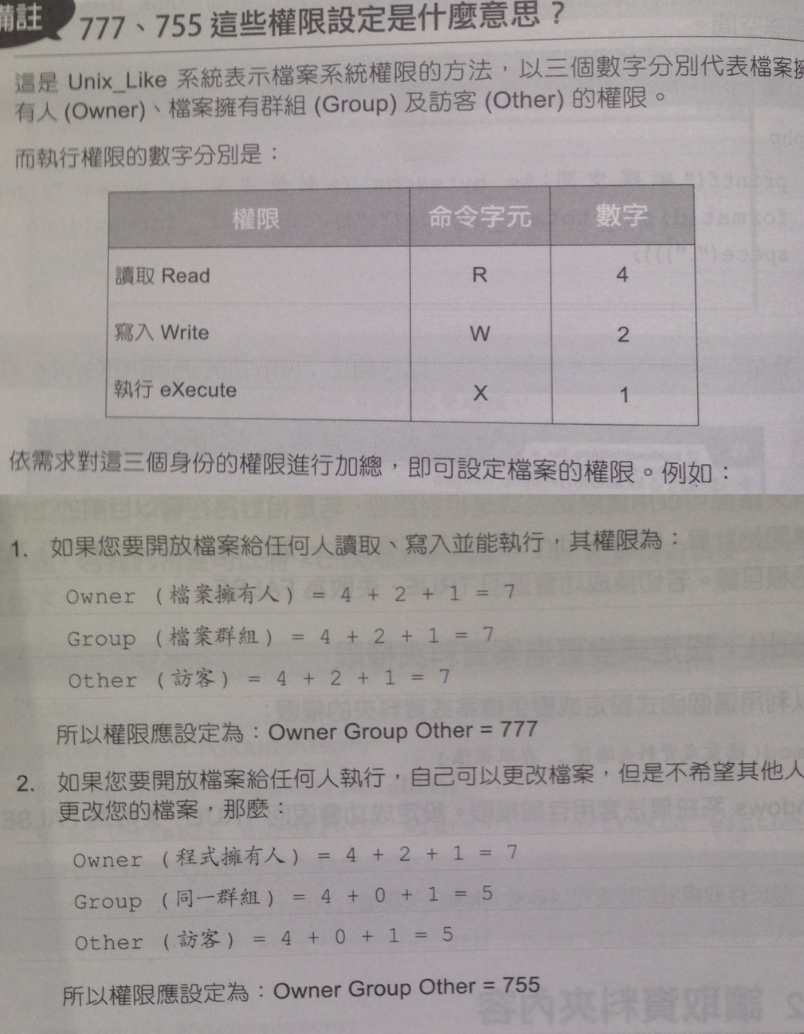
資料夾的處理

建立資料夾-mkdir()

mkdir(資料夾路徑 [,存取權限])

若不設定權限則預設值為777表示為最大權限

Windows系統無法套用存取權限



刪除資料夾-rmdir()

rmdir(資料夾路徑);

rmdir()函式只能刪除空的資料夾，所以在刪除資料夾前要先把其中的檔案刪除

資料夾更名-rename()

rename(舊的資料夾路徑名稱,新的資料夾路徑名稱)

rename()不僅可以對資料夾更名，對檔案也可以

切換資料夾-chdir()

chdir(資料夾路徑)

設定或變更資料夾權限-chmod()

chmod(檔案或資料夾路徑,存取權限)

Windows系統無法套用存取權限

讀取資料夾內容

使用opendir() readdir() 與closedir()函式

1. opendir()開啟資料夾

資源變數 = opendir(資料夾路徑)

1. readdir()讀取資料夾內容

readdir(資源變數)

1. closedir()關閉資料夾資源

closedir(資源變數)

當讀取指標到底部時如還需要使用同一資源，可在closedir()關閉資源前使用rewinddir()將讀取指標移到最前方

Rewinddir(資源變數)

例：

<?php

echo "<hr/>以下為Appserv下的資料夾<hr/>";

$fileDir = "C:\\AppServ";

$fileResource=opendir($fileDir);

while($fileList=readdir($fileResource)){

if(is\_dir($fileDir.'\\'.$fileList)) echo $fileList."<br/>";

}

echo "<hr/>以下為Appserv下的檔案<hr/>";

$fileDir = "C:\\AppServ";

$fileResource=opendir($fileDir);

while($fileList=readdir($fileResource)){

if(is\_file($fileDir.'\\'.$fileList)) echo $fileList."<br/>";

}

?>

結果：



使用scandir()函式讀取資料夾內容

scandir(資料夾路徑[，排序方式]

預設為遞升排序，若值為1則為遞減

函式會以字串陣列回傳

例：

<?php

echo "<hr/>使用scandir--以下為Appserv下的檔案<hr/>";

$fileDir = "C:\\AppServ";

$fileResource=scandir($fileDir);

foreach($fileResource as $fileName){

if(is\_dir($fileDir.'\\'.$fileName)) echo $fileName."<br/>";

}

echo "<hr/>使用scandir--以下為Appserv下的檔案<hr/>";

$fileDir = "C:\\AppServ";

$fileResource=scandir($fileDir);

foreach($fileResource as $fileName){

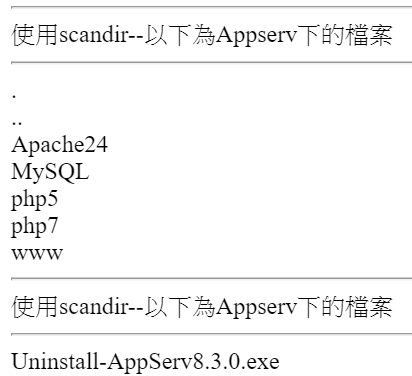
if(is\_file($fileDir.'\\'.$fileName)) echo $fileName."<br/>";

}

print\_r ($fileResource);

?>

結果：



unlink()-刪除檔案

unink(檔案路徑)

注意這個指令不能刪除資料夾

例：

If (file\_exists(“testfile.txt“)){ //檢查檔案是否存在

unlink(“testfile.txt”); //若有則刪除

}

copy()-複製檔案

copy(複製的檔案路徑和來源，要複製的檔案路徑&複製後的檔案名稱)

若複製指定的位置已經有同名的檔案，copy()函式會直接將檔案覆蓋而不會顯示任何警告訊息

例：

<?php

If (!file\_exists("../endfile.txt")){ //若指定的路徑檔案不存在

copy("start.txt","../endfile.txt"); //就複製start.txt檔案到上

一層並更名為endfile.txt

}

?>

更改檔案修改 存取時間--touch()

touch (檔案路徑 [,檔案最後更新時間][,檔案最後存取時間])

若無指定時間則會以當時的系統時間為準，若指定檔案不存在則會自動新增一個空白檔案

檔案上傳

開啟php.ini進行設定

1. 是否允許上傳：file\_uploads = on

2.暫存檔案資料夾：upload\_tmp\_dir=”C:\Windows\Temp”(設定路徑)

若沒有設定會上傳失敗

該資料夾必須要真的存在，並要有寫入的權限

1. 可接受上傳檔案大小

upload\_max\_filesize = 8M

檔案上傳的表單

1. <form>標籤中 action屬性必須設定要接收檔案的php程式檔
2. <form>標籤中 傳送方法必須為method=”POST”因為檔案上傳表單一定要用POST方法
3. <form>標籤中，因為傳送檔案要設定編碼方式，所以要加上

enctype=”multipart/form-data”屬性才能讓檔案欄位送出

1. 上傳的檔案欄位為input標籤，屬性必須設為type=”file”

如此才能在使用時出現(瀏覽鈕)讓使用者選取要上傳的檔案

檔案傳送

例：

<html>

<head>

<title>上傳表單</title>

</head>

<body>

<form action="upload.php" method="POST" enctype="multipart/form-data">

請選取要上傳的檔案:<br/>

<input type="file" name="fileUpload" /> <br/>

<input type="submit" value="送出資料" />

</form>

</body>

</html>

<?php

if ($\_FILES["fileUpload"]["error"]==0){

if(move\_uploaded\_file($\_FILES["fileUpload"]["tmp\_name"],"./".$\_FILES["fileUpload"]["name"])){

echo "上傳成功";

echo "檔案名稱:".$\_FILES["fileUpload"]["name"]."<br/>";

echo "檔案類型:".$\_FILES["fileUpload"]["type"]."<br/>";

echo "檔案大小:".$\_FILES["fileUpload"]["size"]."<br/>";

}else{

echo "上傳失敗";

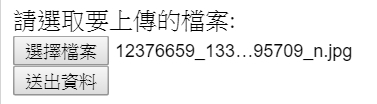
echo "<a href='javascript:window.history.back();'>回上一頁<a/>";

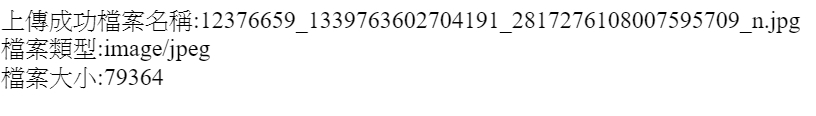
}

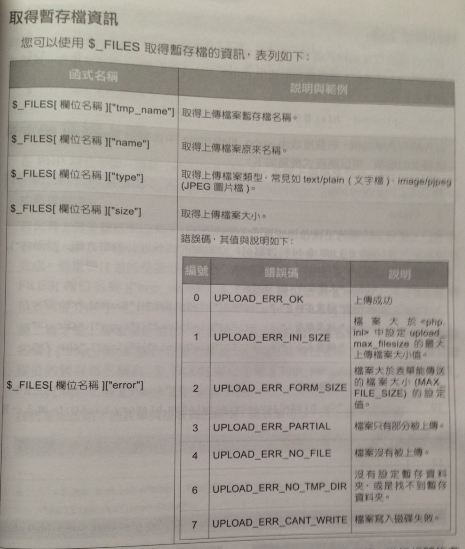
}

?>

結果：







多檔傳送

例：

<html>

<head>

<title>多檔上傳</title>

</head>

<body>

<form action="uploads.php" method="POST" enctype="multipart/form-data">

請選取要上傳的檔案:<br/>

檔案1:<input type="file" name="fileUpload[]" /> <br/>

檔案2:<input type="file" name="fileUpload[]" /> <br/>

檔案3:<input type="file" name="fileUpload[]" /> <br/>

<input type="submit" value="送出資料" />

</form>

</body>

</html>

<?php

$i=count($\_FILES["fileUpload"]["name"]);

for($j=0;$j<$i;$j++){

if ($\_FILES["fileUpload"]["error"][$j]==0){

if(move\_uploaded\_file($\_FILES["fileUpload"]["tmp\_name"][$j],"./".$\_FILES["fileUpload"]["name"][$j])){

echo $\_FILES["fileUpload"]["name"][$j]."上傳成功<br/>";

}else{

echo "上傳失敗";

echo "<a href='javascript:window.history.back();'>回上一頁<a/>";

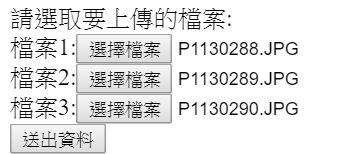
}

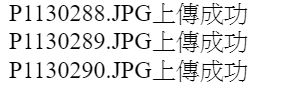
}

}

?>

結果：





讀取及寫入檔案的內容

file():讀入檔案的內容儲存為陣列

file(檔案路徑 [,狀態指標])

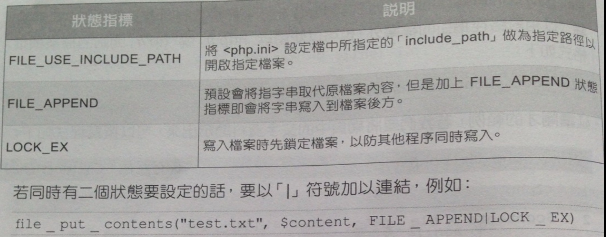
當狀態指標設為1時會將php.ini設定檔中的include\_path做為指定路徑

file\_get\_contens():讀入檔案的內容儲存為字串

file\_get\_contens(檔案路徑[,狀態指標])

file\_put\_contents():寫入檔案

file\_put\_contents(檔案路徑,寫入字串[,狀態指標])



readline():讀取檔案並輸出

readline(檔案路徑[,狀態指標])

當狀態指標設為1時會將php.ini設定檔中的include\_path做為指定路徑

檔案開啟 讀取 寫入與關閉的操作

fopen():開啟檔案

fopen(檔案路徑,開啟模式,[,狀態指標])

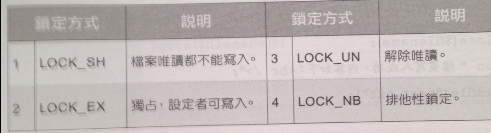
當狀態指標設為1時會將php.ini設定檔中的include\_path做為指定路徑

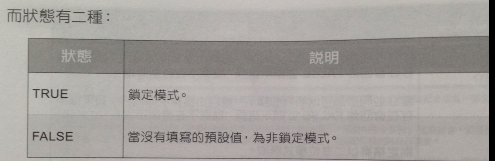
fclose():關閉檔案

fclose(資源)

flock():鎖定開啟的檔案

flock(資源,鎖定方式[,狀態])





fputs()與fwrite():寫入檔案

fputs(資源,字串[,檔案大小 bytes])

fwrite(資源,字串[,檔案大小 bytes])

fgetc()與fgets():讀取檔案

fgets()是每次讀取一行資料回傳,並將讀取指標往下一行移動,若到達檔案底部會回傳FALSE並停止讀取

格式如下：

fgets(資源,[,檔案大小bytes]);

因為fgetc()每次只讀取一個字元，而中文字都是由兩個字元組成，使用時要注意

格式：

fgetc(資源);

fpassthur()與stream\_get\_contents()函式一次讀取檔案

fpassthur()函式可以將fopen()函式開啟的檔案由開始讀取到檔案底部並顯示在頁面上，回傳值為檔案大小bytes

fpassthur(資源)

stream\_get\_contents()函式也可以將fopen()函式開啟的檔案由開始讀取到檔案底部並顯示在頁面上，但回傳值為字串

stream\_get\_contents(資源,[,檔案大小 bytes])

fread()函式讀取二進位檔案

fread()函式可以將fopen()函式開啟的檔案到指定的大小bytes為止，回傳值為字串，最大的特色在於可讀取二進位的檔案，例如圖片檔

格式：

fread(資源,檔案大小bytes)

若要讀取二進位檔案前，要注意的是在fopen()開啟檔案時，就要加上 b的開啟模式選項，讀取才會正確

例：

<?php

$filename=fopen("21.gif","rb");

$contents=fread($filename,filesize("21.gif"));

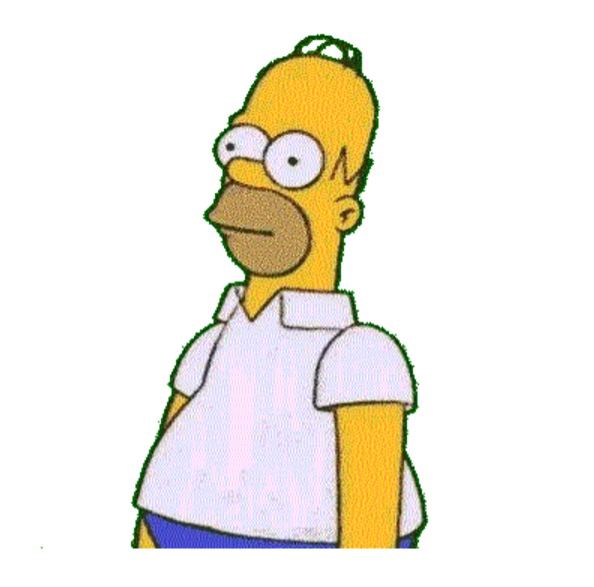
fclose($filename);

header('Content-Type:image/gif;');

echo $contents;

?>

結果：



控制檔案的讀取指標

