



最新

PHP + MySQL + AJAX 網頁程式設計

第 5 章

活用 PHP 函式

本著作僅授權老師於課堂使用，切勿置放在網路上播放或供人下載，除此之外，未經授權不得將全部或局部內容以任何形式重製、轉載、變更、散佈或以其他任何形式、基於任何目的加以利用。

著作權所有 © 旗標出版股份有限公司

本章重點

- 5 - 1 自訂函式
- 5 - 2 引用
- 5 - 3 PHP 函式庫
- 5 - 4 字串處理
- 5 - 5 表單文字處理
- 5 - 6 日期時間

本章重點

- 5 - 7 電子郵件
- 5 - 8 其他重要函式
- 5 - 9 實戰演練 — 橫幅廣告輪播
- 5 - 10 實戰演練 — 網頁聯絡表單

5 - 1 自訂函式

- 隨著程式越寫越龐大之後，您可能會發現很多功能相同的運算式、流程控制...等一再地重複使用。
- 此時便可以將這些重複的程式碼寫成**函式**(function)，即可供程式隨時呼叫，這樣便不需要重複地放置相同的程式碼。
- 此外，寫成函式後，如果需要修改時也會更加容易，只要修改函式即可，不用逐一到程式各個地方去改。

什麼是函式

- 所謂**函式** (function) 就是『一組敘述的集合』，並且以一個函式名稱來代表此敘述集合。
- 當我們每次要用到這組敘述時，只要寫下函式的名稱，就是表示我們要執行這項功能。
- 舉例來說，如果程式中要多次比較最大值，而且不是連續比較，所以不適合使用迴圈，此時就要讓比較最大值的敘述在程式中出現多次。

```
for    (){           //比較最大值的迴圈
    ...
}

...                //其他敘述
```

什麼是函式

```
for    (){           //比較最大值的迴圈
    ...
}

. . .                //其他敘述

for    (){           //比較最大值的迴圈
    ...
}

. . .                //其他敘述
```

- 但如果我們將比較最大值的敘述獨立出來，寫成一個函式，如此程式中要比較變數大小時，只要『呼叫』這個函式進行計算即可，不必每次都要重複撰寫比較最大值的敘述。

什麼是函式

```
function  max()  {           //將比較最大值的敘述放在一個函式中
    ...
}
. . .

max()           //呼叫函式比較最大值
. . .          //其他敘述
max()           //呼叫函式比較最大值
. . .          //其他敘述
max()           //呼叫函式比較最大值
. . .          //其他敘述
```

- 如此一來，比較的動作只需要一行敘述即可，不僅方便，也可以讓程式更加簡潔。

什麼是函式

■ 簡單的說, 使用函式的好處包括：

- 將具有特定功能的敘述獨立成函式, 可提高程式的可讀性。
- 將程式模組化, 讓程式碼可重複使用, 提升寫程式的效率。
- 將程式分解成函式, 發生錯誤時, 可以很容易找出問題在哪一個函式, 提高除錯的效率。
- 將特定功能獨立為函式後, 如果需要修改該功能, 只要更動函式即可, 不用更改主程式, 可提升程式修改的容易度。

自訂函式

- 定義函式的基本語法如下：

```
function 函式名稱( $arg_1, $arg_2, ... $arg_n ) {  
    程式敘述  
    ...  
    return 回傳值;  
}
```

- 函式名稱的命名規則與變數相同，不可與其他函數或變數同名。
- \$arg_1、\$arg_2... 代表要傳入函式的參數。函式的參數可有可無，視程式設計上的需要而定。

自訂函式

- `return` 會結束函式的執行, 然後將回傳值傳回給呼叫者。回傳值可以是任何型別, 例如字串、數字、變數、陣列、物件等。
- 如果函式不需要回傳值, 則可以省略 `return` 敘述。
- 函式可以定義在程式的任何地方, 例如開頭、結尾, 或是中間。
- 下面範例為您說明函式最基本的定義方式：

```
function    hello()    {  
    return   "Hello    World!";  
}
```

自訂函式

- 上面定義了一個名為 `hello` 的函式，所以在程式中只要執行 `"hello()"`，就可以回傳 `"Hello World!"` 字串。
- 如果想要傳遞一些參數給函式，則可如下定義：

```
01 <?php
02 function sum($num1, $num2) {    // 設定了 $num1 與 $num2 參數
03     // 將 $num1 與 $num2 兩個變數值相加後回傳
04     return $num1 + $num2;
05 }
06
```

自訂函式

```
07 // 執行 "sum(3, 6)" 呼叫 sum 函式時, 3 會被指定給 $num1 變數,  
08 // 6 則被指定給 $num2 變數  
09 echo '3+6=' . sum(3,6);  
10 ?>
```

■ 執行結果：



自訂函式的參數預設值

- 自訂函式時，可以先設定參數的預設值，以避免呼叫時沒有給予參數而產生錯誤：

```
function sum($num1=0, $num2=0) {    // 設定 $num1 與 $num2 的預設值為 0
    return $num1 + $num2;          // 傳回兩數的和
}
```

- 因為函式的參數已經有預設值，所以呼叫 sum() 函式時，如果沒有參數，就會使用預設值進行計算：

```
sum();           // 回傳 0
sum(5);         // 回傳 5
sum(NULL, 3);   // 回傳 3
sum(5, 3);      // 回傳 8
```

傳值呼叫與傳址呼叫

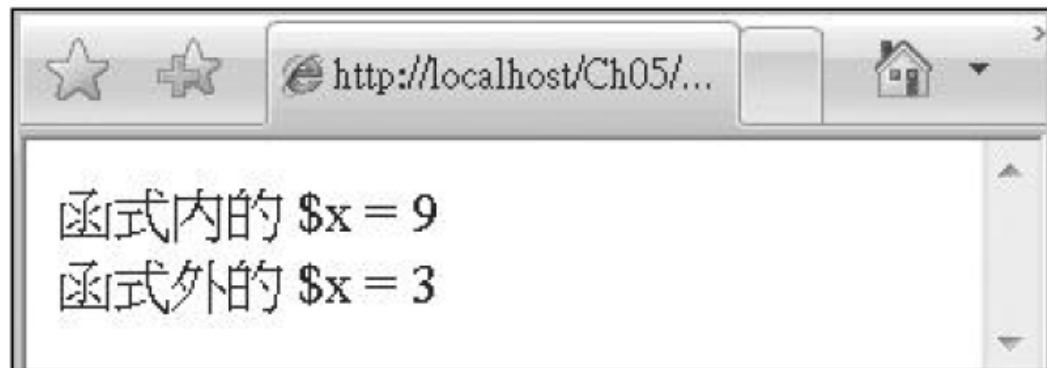
- 如果使用變數做為函式的參數時, PHP 會將變數的值複製給函式的參數。
- 雖然兩者的值相同, 但是在記憶體中卻是不同的實體, 所以即使兩者的名稱相同, 也不會互相影響：

```
01 <?php
02 //header() 設定此頁面使用 UTF8 編碼,
03 //關於 header() 的說明請參考 5-6 節
04 header('Content-Type:          text/html;          charset=utf-8');
05
06 function square($x) {
07     $x = $x * $x;
```

傳值呼叫與傳址呼叫

```
08      echo '函式內的 $x = ' . $x;
09  }
10
11  $x=3;
12
13      square($x);
14
15  echo '<br />函式外的 $x = ' . $x;
16  ?>
```

■ 執行結果：



傳值呼叫與傳址呼叫

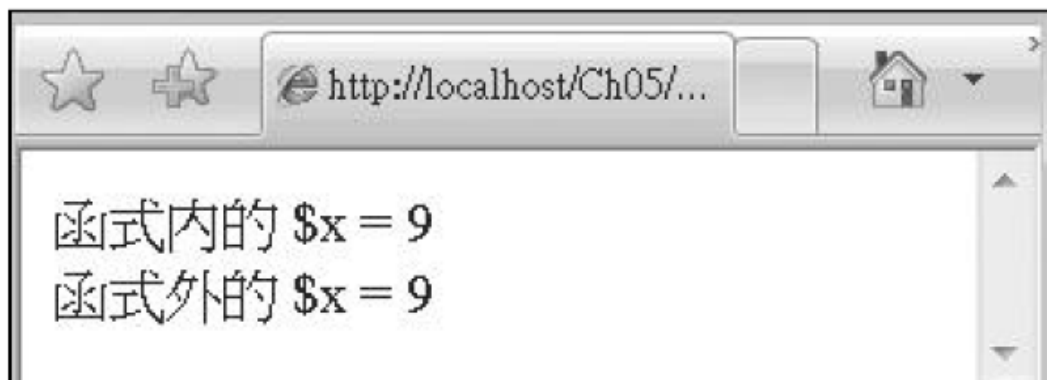
- 因為呼叫函式時傳遞給參數的是『值』，所以稱為**傳值呼叫** (call by value)，這是函式傳遞參數的預設方法。
- 如果您不只是想將值傳遞給函式，而是想將整個變數的實體傳遞給函式，則定義函式時，請如下在參數前面加上 & 符號：

```
01  <?php
02  // 設定此頁面使用 UTF8 編碼
03  header('Content-Type:          text/html;          charset=utf-8');
04
05  function    square(&$x)    {
06      $x  =  $x  *  $x;
```


傳值呼叫與傳址呼叫

```
07      echo '函式內的 $x = ' . $x;
08  }
09
10  $x=3;
11
12      square($x);
13
14  echo '<br />函式外的 $x = ' . $x;
15  ?>
```

■ 執行結果：

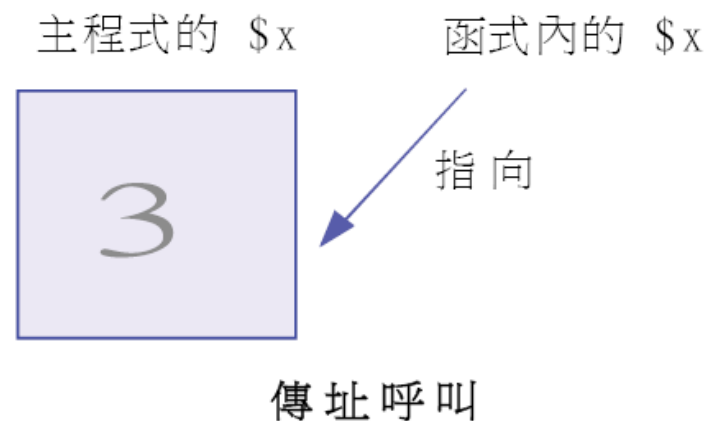
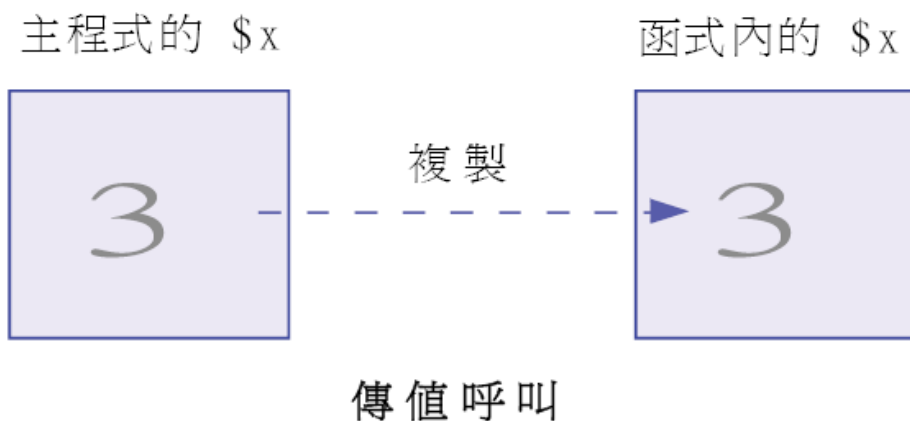


傳值呼叫與傳址呼叫

- 定義函式時在參數前面加上 `&`, 表示會傳遞變數在記憶體的位置, 所以函式的 `$x` 與主程式的 `$x` 都是指向同一位置的變數。
- 因此只要函式內 `$x` 的值有所改變, 主程式的 `$x` 變數值也會跟著變動。
- 即使兩邊變數名稱不同, 其值仍然會同時變動。

傳值呼叫與傳址呼叫

- 這種變數傳遞的方式，便稱為傳址呼叫 (call by reference)：



區域變數與全域變數

- 在 PHP 中，可以在任何需要之處定義變數，但是變數定義之後並非隨處可用。
- PHP 有一套規則規定變數能夠被使用的範圍，稱為變數的**有效範圍** (Scope)。
- 凡是在函式內定義的變數，稱為區域變數 (Local Variable)，而在主程式定義的變數則稱為全域變數 (Global Variable)。
- 區域變數只有在定義該變數的函式內才可以存取，而且因為函式內的變數會被視為區域變數，所以全域變數預設也無法由各函式存取。

區域變數與全域變數

```
01  <?php
02  // 設定此頁面使用 UTF8 編碼
03  header('Content-Type:          text/html;          charset=utf-8');
04
05  $x=5; ← A
06      testVar();
07
08  function  testVar()  {
09      //isset()    可測試變數是否已經定義，
10      // 關於  isset()  的說明請參考  5-6  節
11      if (  isset($x)  ) {
12          echo  '$x  已經定義，變數值為  ' .  $x;
13      }
14      else {
```

區域變數與全域變數

```
15      echo '$x 未定義';  
16  }  
17 }  
18 ?>
```

■ 執行結果：



- 上面程式碼 (A) 處雖然定義了 \$x 變數, 但是 testVar() 函式並無法存取該變數。

區域變數與全域變數

- 如果函式需要存取主程式定義的變數，必須使用 `global` 將該變數宣告為全域變數，才可存取主程式所定義的全域變數：

```
01 <?php
02 // 設定此頁面使用 UTF8 編碼
03     header('Content-Type:      text/html;      charset=utf-8');
04
05     $x=5;
06     testVar();
07
08     function    testVar()    {
09         // 宣告    $x    為全域變數
```

區域變數與全域變數

```
10  global  $x;
11
12  if ( isset($x) ) {
13      echo '$x 已經定義，變數值為 ' . $x;
14  }
15  else {
16      echo '$x 未定義';
17  }
18  }
19  ?>
```


區域變數與全域變數

■ 執行結果：



- 除了使用 `global` 將變數宣告為全域變數以外，也可以使用 `$GLOBALS[]` 陣列來存取全域變數。

區域變數與全域變數

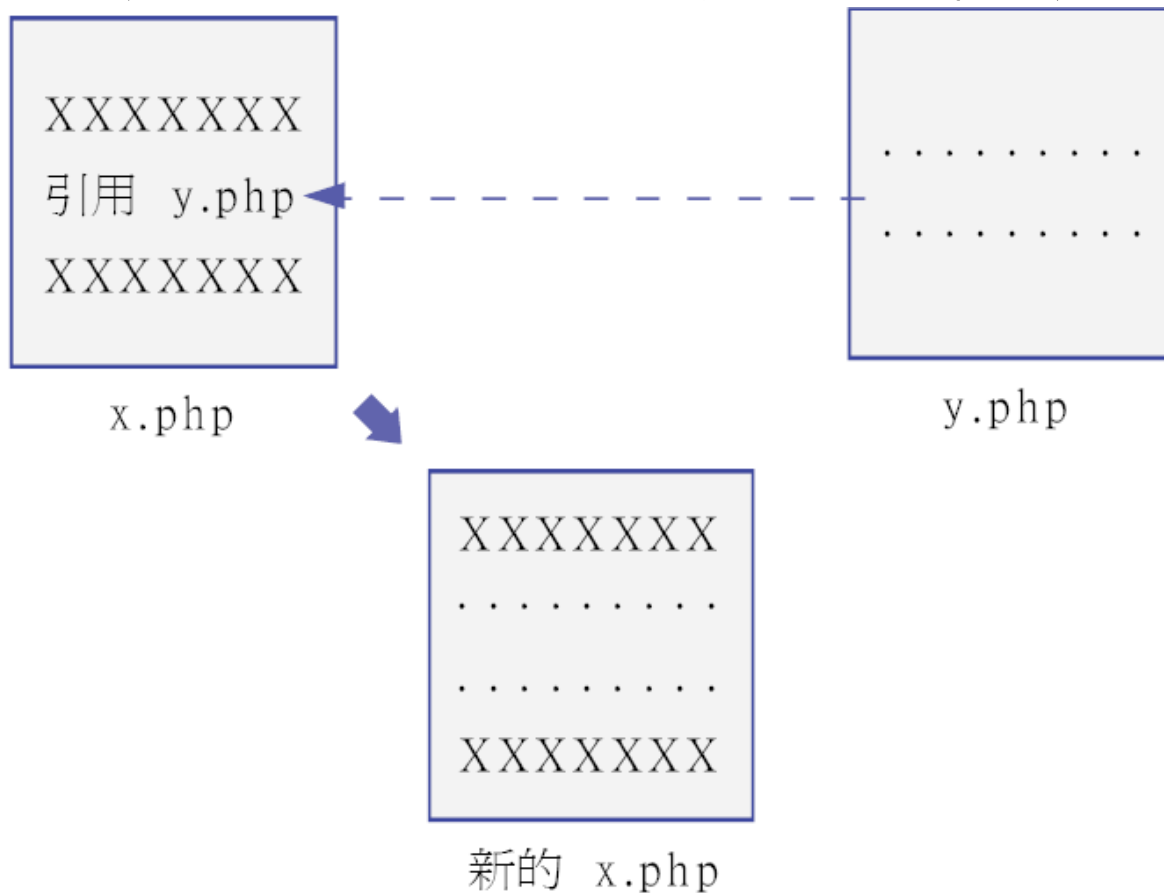
- 例如程式 Ch05-05.php 的 testVar() 函式如下
修改後, 也可存取主程式定義的 \$x 變數：

```
function testVar() {  
    if ( isset($GLOBALS['x']) ) {  
        echo '$GLOBALS[\x\] 已經定義, 變數值為 ' . $GLOBALS['x'];  
    }  
    else {  
        echo '$GLOBALS[\x\] 未定義';  
    }  
}
```

5 - 2 引用

■ 什麼是引用

- 引用是將其他檔案的內容加入目前的程式碼，其示意圖如下：



引用

- 上圖中 x.php 是主要執行的程式, y.php 則是引用檔。
- x.php 會將 y.php 的內容從引用的位置加入程式, 所以您可以將引用檔 y.php 視為 x.php 分割出去的檔案, 必須透過引用, 將兩個檔案合併為一個完整的 x.php。
- 通常我們之所以會使用引用檔, 是為了讓該內容可以給多個檔案共用, 例如共同的函式、變數定義等。

引用

- 這些共同的函式、變數定義如果各別寫在每個檔案中，不僅撰寫程式時會浪費時間在重複的內容，未來維護時要修改的話也很麻煩。
- 必須逐一更改每個檔案，所以藉由引用的方式將共同的內容放在獨立的檔案中，便可以避免這些問題。
- 引用檔的內容並沒有特別的格式限制，可以是 PHP 程式碼或 HTML 碼 / 純文字。

引用

- 如果是 PHP 檔，就會加入並執行其程式碼，若是 HTML 檔 / 純文字，則會在引用處輸出 HTML 碼 / 純文字。
- 此外，引用檔的副檔名也沒有限制，所以可以取名為 xxx.txt、xxx.htm、xxx.inc.php、xxx.inc... 等。
- 使用 include() 引用其他檔案
 - PHP 有許多引用的方法，我們將先介紹如何以 include() 引用其他檔案，下面是 include() 的語法。

引用

- `include(file_name)`
- `include file_name`
- `file_name` 必須包含引用檔的名稱與路徑 (絕對或相對路徑皆可), 否則無法順利引用。
- 下面範例將 `Ch05-05.php` 分割成 `Ch05-06-1.php` (主程式) 與 `Ch05-06-2.php` (引用檔), 為您示範如何以 `include()` 引用其他檔案。

引用

```
01  <?php
02  // 設定此頁面使用 UTF8 編碼
03      header('Content-Type:      text/html;      charset=utf-8');
04
05      include      'Ch05-06-2.php'; ← A
06
07      $x=5;
08      testVar();
09  ?>
```


引用

```
01  <?php
02  function    testVar()    {
03  // 宣告    $x    為全域變數
04  global    $x;
05
06      if (    isset($x)    ) {
07          echo '$x 已經定義，變數值為 ' . $x;
08      }
09      else {
10          echo '$x 未定義';
11      }
12  }
13  ?>
```

引用

- 前面說明引用檔必須包含路徑，不過程式 Ch05-06-1.php 程式碼的 (A) 處卻只有檔名，這是因為 php.ini 預設已經將目前目錄設定在引用的路徑中：

```
...  
include_path      (A)      ".;c:\php\includes;c:\wamp\php\pear"  ← 以 ; 號  
                                                             分隔多  
                                                             個目錄  
...
```

目前目錄

- 所以當 Ch05-06-1.php 與 Ch05-06-2.php 放在同一目錄的情況下，只需要指定檔名。若引用檔不在引用目錄的設定中，便必須明確指定路徑。

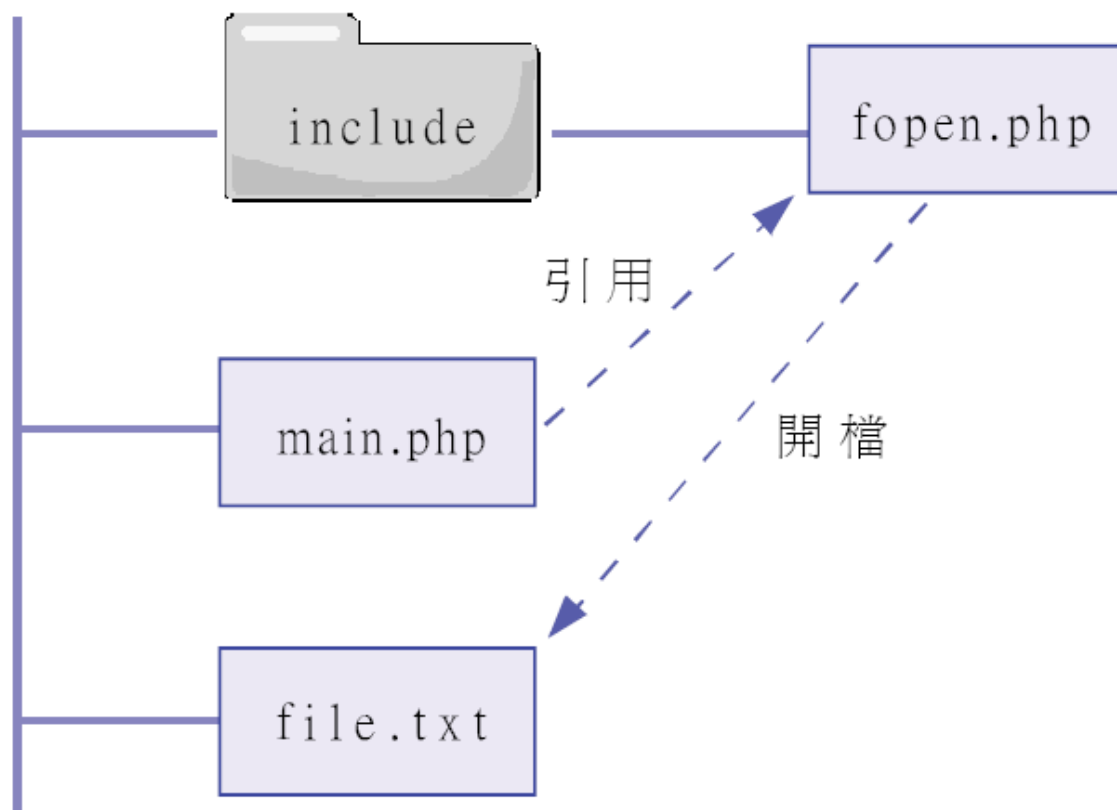
引用

- 當我們使用 `include()` 讀取引用檔時, 如果無法存取引用檔 (如路徑錯誤、檔案不存在、權限不足等), PHP 會顯示警告訊息, 但是仍然會繼續執行。
- 使用 `require()` 引用其他檔案
 - 如果引用檔非常重要, 引用不成功會導致後面的程式執行錯誤, 則不適合使用 `include()`。
 - 因為 `include()` 引用失敗時, 還是會繼續執行後面的程式碼, 所以用其來引用重要的檔案, 將引起一連串的錯誤。

引用

- 此時我們可以使用 `require()` 來進行引用。當 `require()` 無法順利引用檔案時，便會中斷整個程式的執行，避免發生更多錯誤。
- `require()` 與語法和 `include()` 相同，所以此處不重複說明。
- 引用檔的路徑問題
 - 若引用檔內有檔案相關的操作，例如開檔、寫檔，或是引用其他檔案，容易發生路徑混淆的問題。

引用



- 上圖中 main.php 會引用 include / fopen.php, 而 include / fopen.php 會開啟 file.txt 檔案。

引用

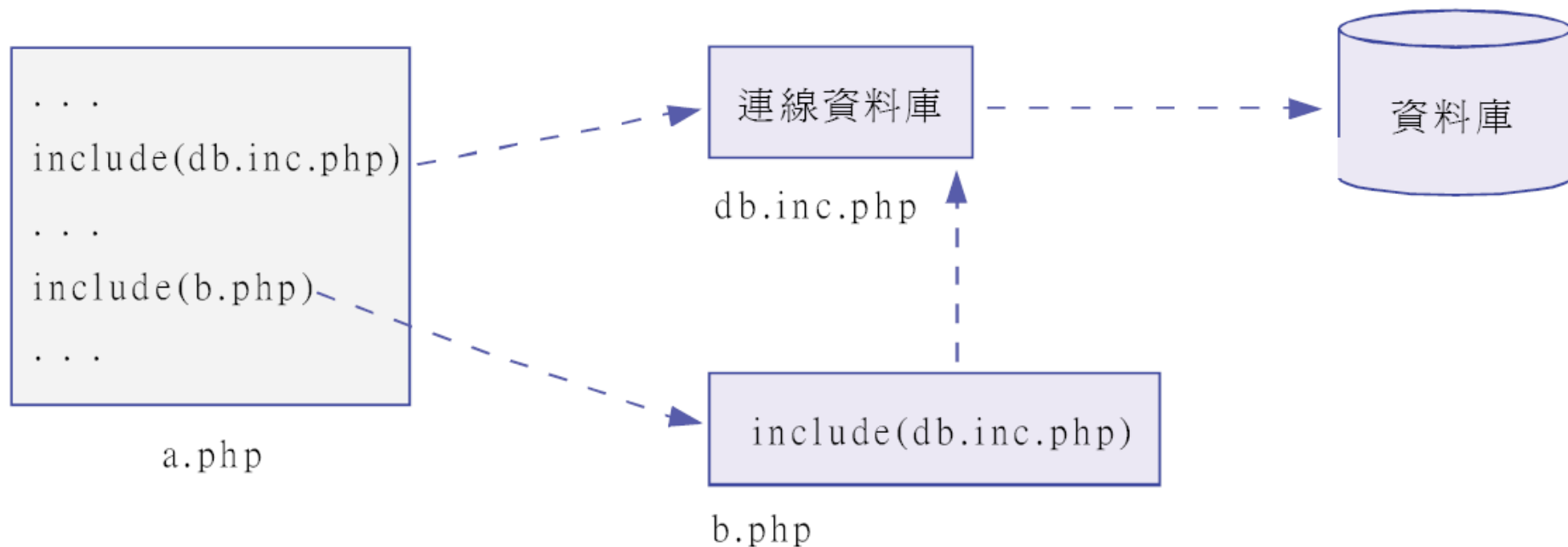
- 此時 `include / fopen.php` 內的程式碼開檔時，路徑應該是 `../file.txt` 還是 `./file.txt` 呢？
- 前面提到，引用檔的程式碼會加入主程式，成為主程式的一部份，所以指定路徑時，自然應該以主程式為主。
- 故以上狀況中，`include / fopen.php` 內的開檔路徑應該是 `./file.txt`。
- 程式的架構越大，分門別類的目錄就越多，引用檔內就越容易弄錯路徑，如果又有巢狀引用，狀況就更形複雜，所以撰寫引用檔時，請特別小心路徑的問題。

引用

■ `include_once()` 和 `require_once()`

- 當我們引用檔案時, 有時候可能會發生重複引用同一檔案的狀況。
- 例如 `a.php` 與 `b.php` 都引用了連線資料庫的引用檔, 但是 `a.php` 又引用了 `b.php`, 如此便會導致連線資料庫的引用檔被引用了兩次, 產生重複的資料庫連線。

引用



- 為了避免這個問題, PHP 提供了 `include_once()` 和 `require_once()` 函式。

引用

- 這兩個函式的使用方法與 `include()` 和 `require()` 相同, 但是以 `include_once()` 和 `require_once()` 引用檔案時, PHP 會檢查該檔案之前是否被引用過, 若否, 才會執行引用的動作。
- `include_once()` 和 `require_once()` 函式的語法與 `include()` 和 `require()` 相同, 在此不再重複說明。

5 - 3 PHP 函式庫

- 除了前面介紹的自訂函式以外, PHP 也內建了許多方便使用的函式, 這些函式可以進行字串處理、顯示日期、寄送電子郵件...等工作。
- 所以只要善用函式, 便不需要重新撰寫程式來處理這些常用的功能, 可節省程式設計的時間。
- 本章隨後會為您介紹各種常用的函式, 不過 PHP 函式庫相當龐大, 裡面包含的函式達數千個, 所能執行的功能包羅萬象, 本書無法一一詳述。

PHP 函式庫

- 未來當您撰寫程式需要某個功能時, 建議先到 PHP 官方網站 <http://www.php.net/manual/en/funcref.php> 查詢, 也許 PHP 已經提供相關的函式可以使用, 如此便不用自己花時間撰寫程式了。
- 因為 PHP 官方網站只有英文版本的函式庫手冊, 所以如果希望有中文參考資料, 可以到以下兩個網站查詢：
 - **PHP 4 手冊**：
<http://linux.tnc.edu.tw/techdoc/banic/>

PHP 函式庫

- **PHP 5 網管實驗室**：<http://www.php5.idv.tw/modules.php?mod=books&act=index&cid=2>
- 這兩個網站為熱心網友翻譯的函式庫中文手冊，雖然閱讀中文較為方便，不過更新的速度以及內容的詳細度皆不如官方網站。
- 因此這兩個網站只能當作參考，建議您還是要以 PHP 官方網站為主。

5 - 4 字串處理

- 我們從網頁表單上讀取的資料皆為字串，因為字串的處理可說是網頁程式設計的重要課題，本節將介紹 PHP 中常用的字串處理函式。
 - 分割字串 — explode()
 - 比對字串 — ereg()
 - 取代字串的內容
 - 傳回格式化字串
 - 清除多餘空白

分割字串 — explode()

- 如果需要將字串分割，例如從表單讀取使用者的生日為 "71 / 12 / 03"，此時如果想要各別取出年、月、日，便需要將這個字串進行分割。
- PHP 分割字串的函式為 `explode()`，其語法如下：

```
explode(delimiter, string)
```

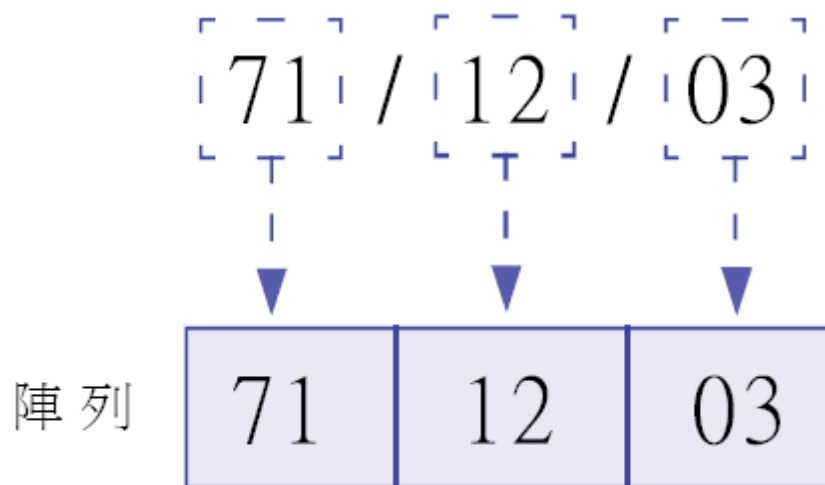
- `delimiter`：分割字元

- `string`：被分割的字串

- `explode()` 會以分割字元切割字串，然後將分割結果以陣列傳回。

分割字串 — explode()

- 以上面生日為例，便需要以 / 做為分割字元，取出字串中的年、月、日，分別放入陣列各元素：



- 所以我們便可使用透過陣列來取得字串分割後的各部分，以下將上面的示意圖寫成程式，為您說明 `explode()` 的使用方法。

分割字串 — explode()

```
01 <?php
02 // 設定此頁面使用 UTF8 編碼
03 header('Content-Type:      text/html;      charset=utf-8');
04
05 $arr    =    explode('/',    '71/12/03');
06 echo '生年為 ' . $arr[0] . '<br />';
07 echo '生月為 ' . $arr[1] . '<br />';
08 echo '生日為 ' . $arr[2];
09 ?>
```

■ 執行結果：



比對字串 — ereg()

- 當我們處理使用者輸入的字串時，會經常需要進行字串比對，例如比對密碼、檢查數字等。
- 比對字串最簡單的方法是使用 if 的判斷條件：

```
if ($age <= 0) echo '年齡不得小於 0';  
if ($password != 'secret') echo '密碼錯誤';
```

- 除了使用 if 的判斷條件外，PHP 也提供了 ereg() 函式，可以使用規則運算式 (Regular Expression) 進行比較複雜的字串比對。

什麼是規則運算式？

- 所謂規則運算式指的是一組描述字元組合與排列的樣式。
- 例如一想到身份證字號，我們心中應該都會浮現一個印象：『最前面是 1 個英文字母，後面再跟著 9 個數字』，這個對於身份證字號的描述便是規則運算式。
- 本章不會說明規則運算式的設計方式，如果您有興趣請自行參閱本書附錄 A。

比對字串 — `ereg()`

- 下面是 `ereg()` 的語法：

`ereg(pattern, string)`

- `pattern`：用來比對的規則運算式
- `string`：要比對的字串

- `ereg()` 會使用規則運算式比對字串，若不符合的話則回傳 **FALSE**，如果相符，則以數字傳回符合的長度，所以一般會使用 `if` 搭配 `ereg()` 來判斷是否符合規則運算式。
- 我們將以檢查電話與身份證字號為例，說明 `ereg()` 的使用方法。

比對字串 — ereg()

- 下面 Ch05-08.html 表單要求使用者輸入電話與身份證字號：

此表單傳送給 Ch05-08.php

請輸入電話 (格式 02-12345678) :

請輸入身分證字號 :

送出

欄位名稱爲 telephone

欄位名稱爲 idcard

比對字串 — ereg()

■ Ch05-08.php 的程式碼如下：

```
01  <?php
02      header('Content-Type:      text/html;      charset=utf-8');
03
04  // 電話號碼的格式為： 2 個數字 + 1 個 "-" 符號 + 8 個數字
05  if (      !      ereg('[0-9]{2}-[0-9]{8}',      $_POST[telephone])      ) ← A
06      echo      '電話號碼的格式錯誤<br      />';
07
08  // 身分證字號的格式為： 1 個 A-Z 之間的英文字（不分大小寫） +
09  // 1 個數字 1 或 2 + 8 個數字
10  if (      !      ereg('[A-Za-z]{1}[1-2]{1}[0-9]{8}',      $_POST[idcard])      ) ← B
11      echo      '身分證字號的格式錯誤';
12  ?>
```

比對字串 — ereg()

■ 執行

結果：

- 1 輸入錯誤的電話號碼與身分證字號



http://localhost/Ch05/...

請輸入電話 (格式 02-12345678) :

(02)12345678

請輸入身分證字號 :

187457578F

送出

- 2 程式顯示錯誤訊息



http://localhost/Ch05/...

電話號碼的格式錯誤
身分證字號的格式錯誤

比對字串 — `ereg()`

- (A) 此處的 `[0-9]{2}` 代表該位置為 2 個 0 - 9 之間的數字, `[0-9]{8}` 則是 8 個 0 - 9 之間的數字。
- (B) 此處的 `[A-Za-z]{1}` 代表該位置為 1 個 A - Z 或是 a - z 之間的英文字母。
- `ereg()` 比對時會區分英文大小寫, 所以上述程式比對身份證字號時, 必須使用 `[A-Za-z]` 來代表不分大小寫的英文字母。

比對字串 — ereg()

- 當遇到不分大小寫的情況時，您也可以使用 `eregi()` 函式來比對，其語法和 `ereg()` 完全相同，所以比對身分證字號可以如下改用 `eregi()` 函式：

```
if ( ! eregi('[a-z]{1}[12][0-9]{8}', $_POST[idcard]) )
```


取代字串的內容

- PHP 提供了許多不同的函式可以用來取代字串的內容, 其中較常用的為 `str_replace()`、`ereg_replace()`。

- `str_replace()` 函式

- 下面是 `str_replace()` 的基本語法：

`str_replace(search, replace, string)`

- `search` : `string` 的部分字串
- `replace` : 要用來取代 `search` 的字串
- `string` : 被取代的字串

取代字串的內容

- `str_replace()` 會在 `string` 中搜尋 `search` 的內容，如果找到的話，就會代換為 `replace` 的內容，然後傳回一個新字串，範例如下：

```
// $string 的值會變成 " 旗標出版公司 "  
$string = str_replace(' 社 ', ' 公司 ', ' 旗標出版社 ');
```

- 上例中，`str_replace()` 會將『社』取代為『公司』，然後傳回『旗標出版公司』給 `$string` 變數。
- `str_replace()` 常見的用途是將字串中的特殊字元代換為 HTML 符號，以使其能夠正確顯示於網頁。

取代字串的內容

- 例如網頁中的連續空白必須使用 " " 才能正確顯示, 否則不論有多少個連續空白字元, 瀏覽器都會將其當成一個空白。
- 所以下面程式為了正確顯示連續空白, 就必須使用 `str_replace()` 將空白取代為 " " :

```
01 <?php
02     header('Content-Type:      text/html;      charset=utf-8');
03
04 // 字串包含 3 個連續空白
05 $string='旗      標      出      版      公      司';
06 echo  '未處理空白:  ' . $string;
07
08 $string  =  str_replace('      ', '&nbsp;&nbsp;&nbsp;', $string);
```

取代字串的內容

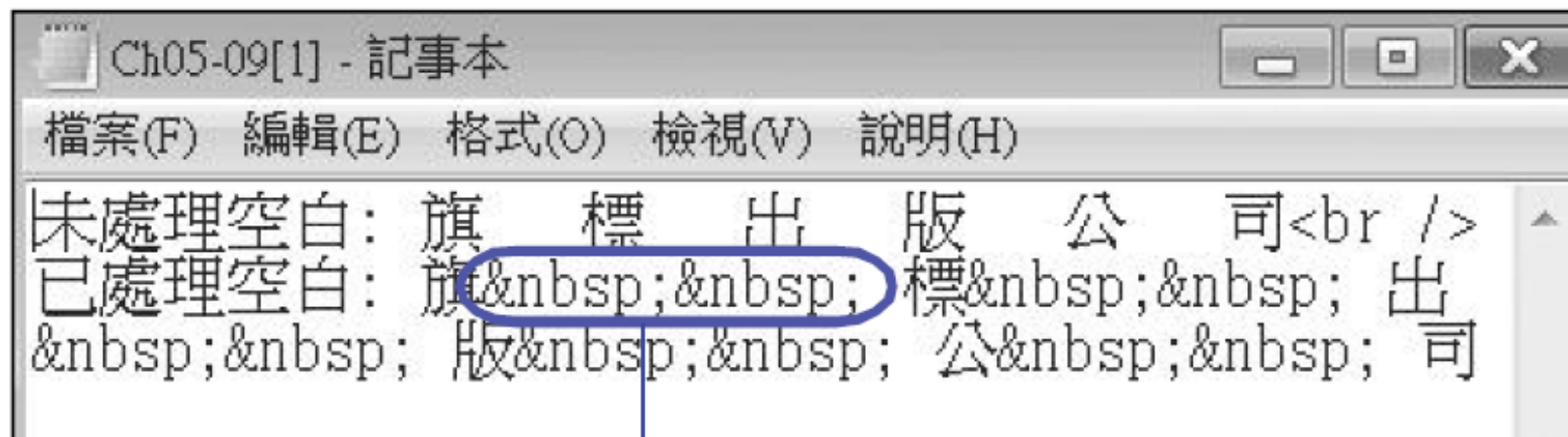
```
09 echo '<br />已處理空白: ' . $string;  
10 ?>
```

- 執行結果：



- 網頁的原始碼如下頁所示。

取代字串的內容



連續兩個空白被代換
為 " "

- 代換空白時, 一般較直覺的想法是將每個空白代換為 " " :

```
str_replace(' ', '&nbsp;', $string);
```

取代字串的內容

- 進行這樣的代換時, 如果字串中包含 HTML 標籤, 因為 HTML 標籤中經常會有空白, 例如 `<a href>`, 所以若使用 `str_replace()` 取代每個空白, 便會破壞 HTML 標籤。
- 所以程式的 (A) 處是將 2 個連續空白字元代換為 `" "`, 這是因為單一的空白字元可以顯示於瀏覽器, 只需要處理連續 2 個以上的空白。

取代字串的內容

- 下表列出各種狀況下，代換兩個連續空白的結果（一個 * 表示一個空白）：

| 代換前 | 代換後 |
|-------|---------------------------------------|
| * | * |
| ** | |
| *** | * (網頁會顯示 3 個空白) |
| **** | |
| ***** | * (會顯示 5 個空白) |

取代字串的內容

■ `ereg_replace()` 函式

- `ereg_replace()` 會使用規則運算式取代字串的內容，其語法如下：

`ereg_replace(pattern, replace, string)`

- `pattern`：要用來比對的規則運算式
 - `replace`：要用來取代 `search` 的字串
 - `string`：被取代的字串
- `ereg_replace()` 會使用規則運算式比對 `string` 的內容，如果找到符合規則運算式的內容，就會將其以 `replace` 取代，然後傳回新的字串。

取代字串的內容

- 若找不到符合規則運算式的內容, 則傳回原本的字串。
- 如下例會將所有英文字母 a - z 取代為空字串, 亦即刪除所有 a - z 英文字母：

```
ereg_replace('[a-z]', '', '旗標 flag 出版社 '); // 輸出 " 旗標出版社 "
```

- 因為 `ereg_replace()` 使用規則運算式比對內容, 所以可以進行比較複雜的字串取代, 不像前面 `str_replace()` 只能單純地取代固定內容。
- 但是相對的, `ereg_replace()` 的速度比 `str_replace()` 慢, 所以如果只需要取代固定內容, 建議使用 `str_replace()` 即可。

傳回格式化字串

- `sprintf()` 可以傳回格式化的字串，下面是 `sprintf()` 的基本語法：

```
sprintf(format_string, arg_1, arg_2 ...)
```

- `format_string`：包含參數的字串
- `arg_1, arg_2...`：各參數的值

- 當我們要在字串中使用變數或陣列的資料時，常常會需要將字串切割：

```
echo '嗨' . $data['user'] . '，您是本站第' . $data['count'] . '位訪客'；
```

傳回格式化字串

- 被切割的字串對於閱讀與除錯都會造成困難，特別是字串內包含多個變數或陣列資料時，不只難以閱讀，更容易因為少一個括號或點號，而導致程式出現語法錯誤。
- 為了讓字串與變數更容易結合與閱讀，我們可以如下使用 `sprintf()` 函式來設定字串：

```
$msg=sprintf(' 嗨      %s,      您是本站第      %d      位訪客 ',      $data['user'],      $data['count']);
```

第 1 個參數 第 2 個參數 參數值 1 參數值 2

echo \$msg;

The diagram illustrates the components of the `sprintf()` function call. Arrows point from labels below to specific parts of the code: '第 1 個參數' points to the first `%s`, '第 2 個參數' points to the second `%s`, '參數值 1' points to `$data['user']`, and '參數值 2' points to `$data['count']`. The code itself is: `$msg=sprintf(' 嗨 %s, 您是本站第 %d 位訪客 ', $data['user'], $data['count']);`

傳回格式化字串

- `sprintf()` 會將參數值 1 代入字串內的第 1 個參數，參數值 2 代入第 2 個參數，其他參數依此類推。
- 處理完成後，會傳回一個代換後的完整字串。
- 上例可以看到，使用 `sprintf()` 設定字串，就不用再切割字串，也不需要再加上許多括號或點號，對於閱讀或是除錯都相當方便。

傳回格式化字串

- `sprintf()` 的參數必須以 `%` 為前置符號, 下表列出 `sprintf()` 常用的參數型別:

| | |
|-----------------|-----|
| <code>%d</code> | 整數 |
| <code>%f</code> | 浮點數 |
| <code>%s</code> | 字串 |

清除多餘空白

- 當使用者在網頁輸入文字時，可能會不小心在開頭或結尾多加了空白。
- 對於人腦而言空白會自動忽略，所以 " abc " 與 "abc" 可說是相同，但是對於一板一眼的電腦，多了空白就是不一樣的字串。
- 為了處理這樣的狀況，PHP 提供了 trim() 函式，可以清除字串開頭與結尾的空白、換行、NULL...等看不見的特殊字元。
- trim() 函式的語法如下：`trim(string)`
 - string：欲處理的字串。

清除多餘空白

- `trim()` 將字串開頭與結尾看不見的特殊字元清除之後，以字串的型別回傳：

```
trim("    Flag Book ");    // 傳回 "Flag Book"
```

- 如果您只想清除字串結尾 (字串右邊) 的空白與特殊字元，可以使用 `rtrim()` 函式。
- 若想要清除字串開頭 (字串左邊) 的空白與特殊字元，則可使用 `ltrim()` 函式：

```
rtrim("    Flag Book ");    // 傳回 "    Flag Book"  
ltrim("    Flag Book ");    // 傳回 "Flag Book "
```

5 - 5 表單文字處理

- 前面 4-6 節介紹過如何以 `$_POST` 或 `$_GET` 陣列取得使用者輸入的文字。
- 取得了資料之後，在許多狀況下都需要處理後才能安全、正常使用，本節將會探討各狀況下應該如何處理使用者輸入的文字。
- 表單文字中，有兩大類必須特別注意並且加以處理，以下將分別為您說明：
 - HTML 特殊字元
 - 括號與反斜線

HTML 特殊字元

- 在 HTML 網頁中, 如果要顯示下面字元, 應該使用 HTML 符號來表示：

| 字元 | HTML 符號或標籤 |
|----|------------|
| < | < |
| > | > |
| & | & |
| " | " |
| ' | ' |
| 空白 | |
| 換行 | |

HTML 特殊字元

- 上表中, < 與 > 是用來定義 HTML 標籤的字元, 所以如果要顯示這兩個字元, 必須以 HTML 符號來表示。
- & 則是 HTML 符號的前置字元, 具有特殊的意義, 因此也必須藉由 HTML 符號來表示。
- 而 ' 與 " 則是 PHP 定義字串的括號, 所以部分程式設計者為了避免影響程式, 也會以 HTML 符號來顯示括號。
- 至於連續空白與換行字元, 都需要使用 HTML 符號與標籤才能正常顯示。

不代換 < 與 > 字元的危險性

- 將 < 與 > 代換為 HTML 符號的目的,除了要讓這兩個字元正常顯示以外,另一方面也可以避免使用者故意輸入 HTML 語法。
- 例如下面程式會顯示使用者輸入的姓名：

```
01  <?php
02      header('Content-Type:      text/html;      charset=utf-8');
03  echo  $_POST['name']    .    ' 您好 ';
04  ?>
```

不代換 < 與 > 字元的危險性



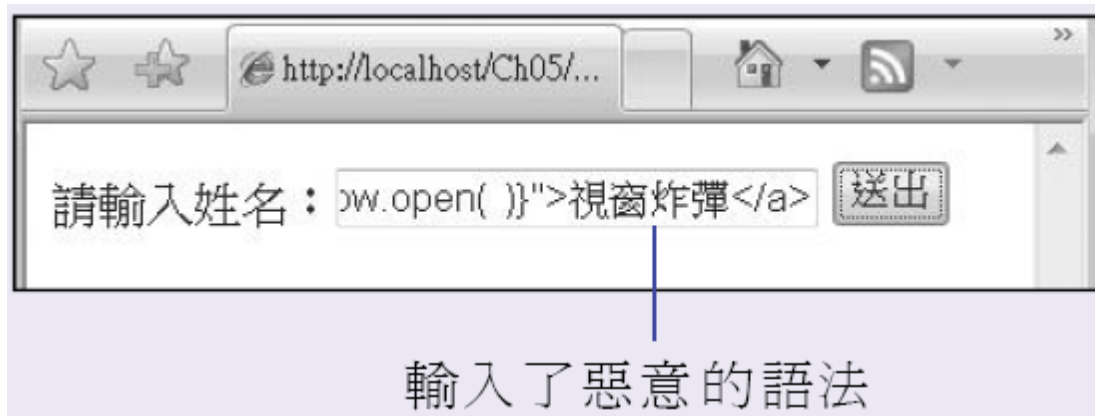
- 因為程式沒有特別處理 < 與 > 字元, 所以使用者可以特意輸入 HTML 標籤:



不代換 < 與 > 字元的危險性

- 如果使用者輸入了惡意的語法，便會讓原本正常的程式輸出帶有破壞性的網頁。
- 例如下面是一段惡意的 Java Script 程式，只要滑鼠經過該連結時，就會導致瀏覽器開啟無數個視窗，更甚者可能會耗盡系統資源後當機：

視窗炸彈



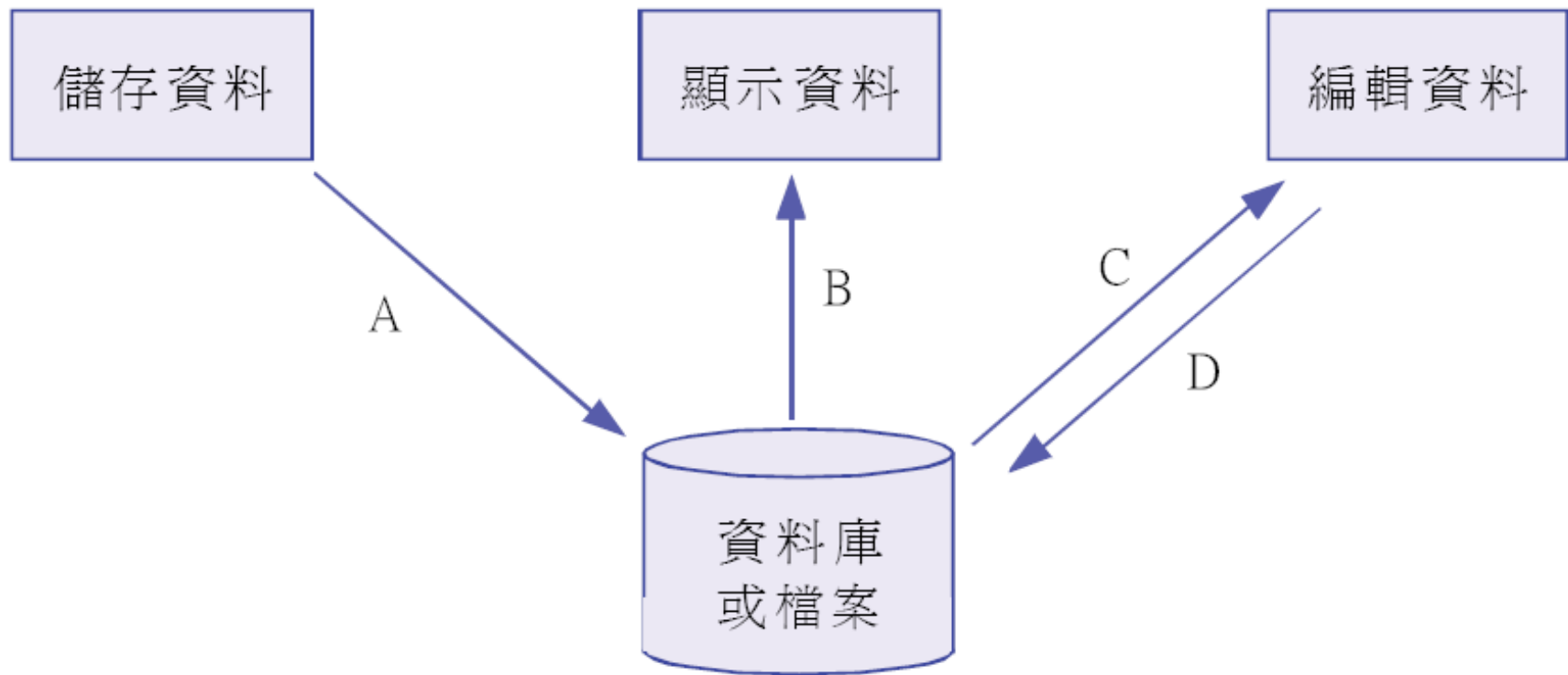
不代換 < 與 > 字元的危險性



- 如果上述程式是一個留言板或論壇程式，其他使用者瀏覽時便可能執行到該惡意程式，所以您的網站就會因為使用者輸入的字串，而變成一個帶有惡意程式的網站。

HTML 特殊字元的建議處理方式

- 一般網頁程式接收表單文字後，可能會有以下步驟：



HTML 特殊字元的建議處理方式

- 若在圖中的狀況 A 就處理 HTML 特殊字元, 那麼存入資料庫的留言就會包含 " " 或 "
" 等 HTML 符號或標籤。
- 如此一來, 雖然顯示資料時可以正確表示空白與換行, 但是編輯資料時, 在狀況 C 必須進行相反的取代, 否則編輯者就會看到 " " 與 "
", 而不是空白與換行。
- 當編輯完畢後, 狀況 D 還需要再次進行取代工作, 所以 4 種狀況中有 3 種需要進行額外的處理。

HTML 特殊字元的建議處理方式

- 其實只要反過來思考，改為在狀況 B 進行取代，其他狀況 A、C、D 都不進行處理，即可避免產生上面麻煩的問題。
- 所以建議您，只有在顯示時，才處理 HTML 特殊字元，其他狀況時則不多加處理。
- 前述應代換為 HTML 符號的字元中，除了連續空白必須使用 `str_replace()` 自行代換外，其他 PHP 皆提供相關的處理函式，不需自行撰寫程式，隨後將分別介紹這些函式。

HTML 特殊字元的建議處理方式

- 處理換行字元 — nl2br() 函式
 - nl2br() 函式是用來將換行字元取代為
 標籤, 其語法如下:
`nl2br(string)`
 - string : 欲處理的字串
 - nl2br() 函式將字串中所有的換行字元取代為
 標籤之後, 會以字串傳回。
 - 下面以範例示範 nl2br() 的使用方法。

```
01 <?php
02     header('Content-Type:      text/html;      charset=utf-8');
03
04 // 定義一個包含換行符號的字串
```

HTML 特殊字元的建議處理方式

```
05 $string = 'Linux 實務應用 \n 旗標出版公司';
06
07 echo '<p>未處理過的字串:</p>' . $string;
08 echo '<hr /><p>nl2br() 處理過的字串:</p>' . nl2br($string);
09 ?>
```

■ 執行結果：



HTML 特殊字元的建議處理方式

- 處理 HTML 特殊字元 — htmlspecialchars() 函式
 - htmlspecialchars() 函式可以將 &、'、"、<、> 等字元取代為相對應的 HTML 符號, 下面是其基本語法：

htmlspecialchars(string, quote_style)

- string：欲處理的字串
- quote_style：指定要取代哪些括號

HTML 特殊字元的建議處理方式

- htmlspecialchars() 函式會依照 5-25 頁對照表，將 &、<、> 字元代換為 HTML 原始碼。
- 至於 '、" 字元是否取代為 HTML 符號，則取決於後面 quote_style 參數的設定。
- quote_style 有 3 種參數：
- **ENT_COMPAT**：只取代"，不取代'。
- **ENT_QUOTES**："與'皆取代。
- **ENT_NOQUOTES**："與'兩個皆不取代。

括號與反斜線

- 在 PHP 中，括號 " 與 ' 和反斜線 \ 都具有特殊的意義，所以若使用者剛好需要輸入特殊字元，例如輸入『It's a cat』，字串中的 ' 符號便有可能會造成程式錯誤。
- 甚至如果遇到惡意者，更可能藉由輸入這些特殊字元，影響程式的邏輯，入侵取得想要的資料。
 - 括號與反斜線的處理

括號與反斜線

- 表單文字中如果包含括號或反斜線，一般會在程式中特別處理，將其再加上一個反斜線，以取消其特殊意義：

| 使用者輸入 | 處理後 |
|------------------|--------------------|
| say "I love you" | say \"I love you\" |
| I'm Tony | I\\'m Tony |
| C:\\Windows | C:\\\\Windows |

■ magic_quotes_gpc 設定

- 在 php.ini 的設定中有一個項目為 **magic_quotes_gpc**，這個設定值預設為開啟的狀態。

括號與反斜線

- 開啟時若表單文字包含 '、"、\ 等特殊字元，PHP 接收後會**自動加上反斜線**『\』，取消其意義，避免這些特殊字元影響 PHP 程式的執行。
- 所以如果開啟 `magic_quotes_gpc` 設定，程式中便不需自行處理表單文字的括號與反斜線，PHP 會自動為我們加上反斜線。
- 如果您需要在程式中知道目前 `magic_quotes_gpc` 是否開啟，可以使用 `get_magic_quotes_gpc()` 函式。

括號與反斜線

- `get_magic_quotes_gpc()` 函式使用時不需參數, 若傳回整數 1, 則表示 `magic_quotes_gpc` 已經開啟, 若傳回整數 0, 則未開啟
`magic_quotes_gpc` :

```
<?php
if (    get_magic_quotes_gpc()    )    {
    echo    'magic_quotes_gpc    功能已經開啓';
}    else    {
    echo    'magic_quotes_gpc    功能未開啓';
}
?>
```

- 自行處理括號與反斜線 — `addslashes()` 與 `stripslashes()`

括號與反斜線

- 除了使用 `magic_quotes_gpc` 功能自動處理括號與反斜線外, PHP 也提供了兩個處理括號與反斜線的函式 — `addslashes()` 與 `stripslashes()`, 可以讓我們自行處理括號與反斜線, 其語法如下:

`addslashes(string)`

// 加上反斜線

`stripslashes(string)`

// 刪除反斜線

■ `string` : 欲處理的字串。

- `addslashes()` 會在字串中的 `\`、`'`、`"` 等字元前面再加上一個 `\`, 讓反斜線與括號不再具備特殊意義; 而相反的, `stripslashes` 則會將字串中所有反斜線刪除。

括號與反斜線

- 這兩個函式會將處理後的資料以字串傳回，範例如下：

```
<?php
// 輸出 『Just say \"WOW\"』
echo addslashes('Just say "WOW");

// 輸出 『Taipei nTaiwan』
echo stripslashes('Taipei \nTaiwan');
?>
```

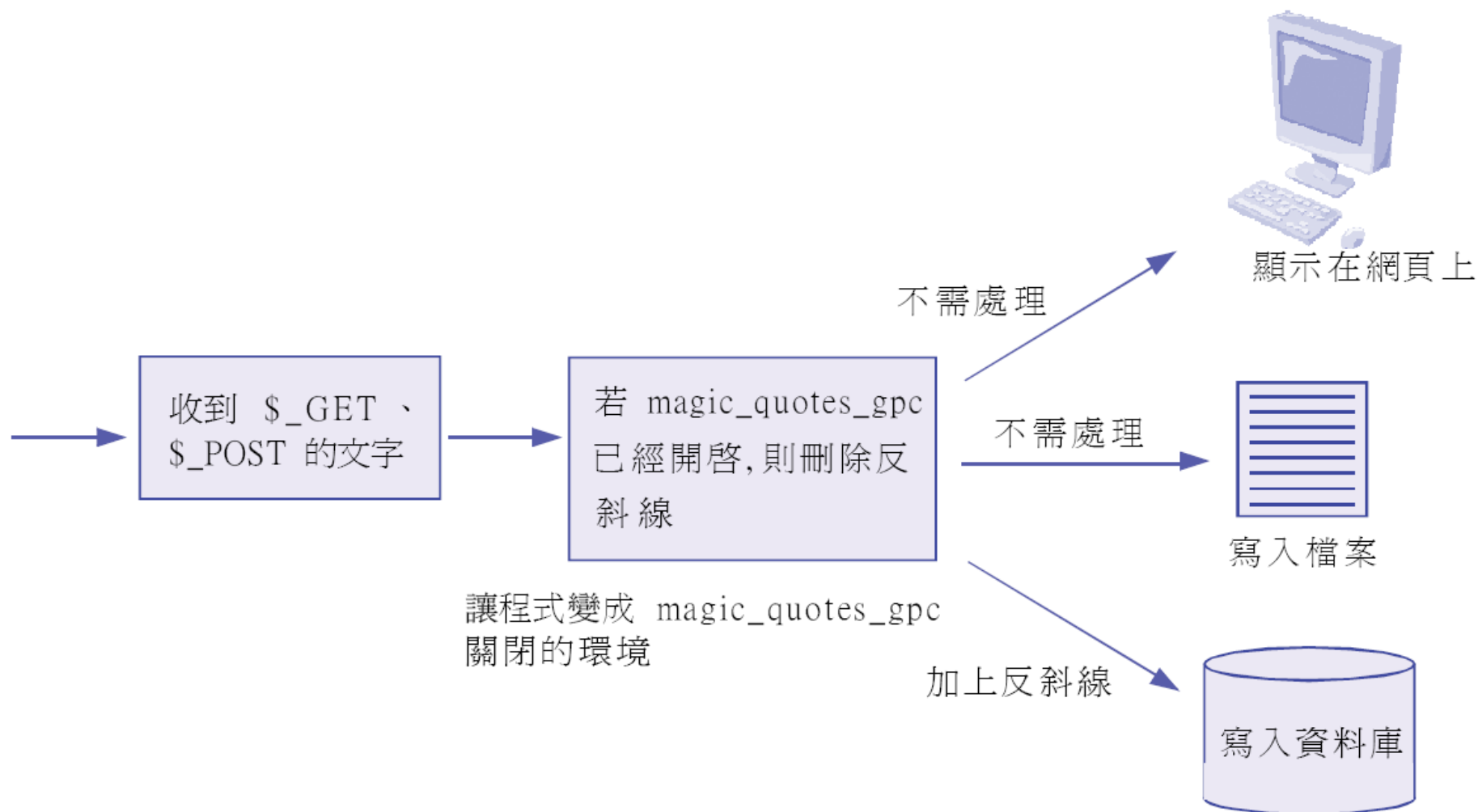
- 括號與反斜線的建議處理方式
 - magic_quotes_gpc 功能雖然方便，但是建議您撰寫程式時，最好不要只依靠 magic_quotes_gpc 的功能。

括號與反斜線

- 否則若遇到系統管理者設定錯誤，或是執行程式的系統原本就關閉 `magic_quotes_gpc`，此時您的程式將立刻變得不安全。
- 因為 `magic_quotes_gpc` 有可能是開啟，也有可能是關閉，而且此功能無法於程式執行時更改設定。
- 所以為了保險與統一起見，筆者建議程式中一律視 `magic_quotes_gpc` 已經關閉，然後在執行有可能被括號與反斜線影響的動作（例如寫入資料庫）前，手動加上反斜線。

括號與反斜線

- 下面是建議的括號與反斜線處理流程圖：



總結表單文字的處理方式

- 本節將綜合以上所述，歸納出一個表單文字的處理原則。
- 對於同時具備輸入、顯示、儲存、編輯的大型程式，嚴格採用這些的原則可以在程式撰寫時有所依據，比較不會發生混亂或出錯的問題。
 - 接收表單文字
 - 接收表單文字時，先刪除 `magic_quotes_gpc` 功能可能加上的反斜線，盡可能地保持資料的原貌。

總結表單文字的處理方式

```
function myStripslashes($value){  
    // 如果 magic_quotes_gpc 功能已經開啓， 便刪除反斜線  
    if (get_magic_quotes_gpc())  
        $value = stripslashes($value);  
  
    return $value;  
}  
  
//isset() 與 empty() 函式請參見本章 5-8 節  
if ( isset($_POST['message']) && !empty($_POST['message']) ) {  
    $message=myStripslashes($_POST['message']);  
}
```

■ 寫入資料庫

- 如果要將表單文字寫入資料庫，先使用 `mysql_real_escape_string()` 函式處理反斜線與括號。

總結表單文字的處理方式

```
$sql=sprintf("INSERT    guestbook    ( ` 姓名 ` ,    ` 留言 ` )    VALUES    ('%s',    '%s')",  
            mysql_real_escape_string($name),  
            mysql_real_escape_string($message) );  
$result=mysql_query($sql);
```

- 如果您使用 PEAR 的 MDB2 類別存取資料庫，應先使用 quote() 方法處理反斜線與括號。

```
$sql=sprintf("INSERT    guestbook    ( ` 姓名 ` ,    ` 留言 ` )    VALUES    ('%s',    '%s')",  
            $mdb2->quote($name, 'text'),  
            $mdb2->quote($message, 'text') );
```

■ 顯示在網頁上

- 要將表單文字，或是從資料庫 / 檔案取出的文字顯示在網頁上時，應先代換 HTML 特殊字元。

總結表單文字的處理方式

```
$message=htmlspecialchars($message, ENT_QUOTES);  
$message=nl2br($message);  
$message=str_replace(' ', '&nbsp;&nbsp;&nbsp;', $message);
```

■ 寫入檔案

- 若要將表單文字寫入檔案，因為反斜線、括號或是 HTML 特殊字元並不會造成影響，所以不需另外處理。

5 - 6 日期時間

- 在網頁程式中, 如留言板、Blog、訂單系統等, 許多都需要取得目前的日期時間, 以記錄事件發生或資料儲存的日期時間。
 - 顯示目前的日期時間
 - 取得目前的時間戳記
 - 取得指定日期時間的時間戳記
 - 設定時區

顯示目前的日期時間

- `date()` 函式可以依照指定的格式，顯示目前時間，其語法如下：

```
date(format_string)
```

- `format_string`：帶有格式參數的字串

- `date()` 函式會依照字串內的格式參數，傳回一個包含目前日期時間的字串，所以您可以自行設定，以顯示 "07 年 11 / 01"、"Wed, Oct 3, 2007" 等各種格式的日期時間。
- 常用的格式參數如下。

顯示目前的日期時間

| 參數 | 說明 | 範例 |
|----|------------------|-------------|
| 年 | | |
| Y | 四位數西元年 | 1997 、 2008 |
| y | 二位數西元年 | 97 、 08 |
| 月 | | |
| F | 月份的英文名稱 | May 、 July |
| M | 月份英文名縮寫 | Jan 、 Aug |
| m | 二位數月份 | 02 、 11 |
| n | 二位數月份 (前面不補零) | 2 、 11 |
| t | 當月份的天數 | 28 ~ 31 |

顯示目前的日期時間

| 日 | | |
|----|---------------------|---------|
| d | 二位數日期 | 07 、 31 |
| j | 二位數日期 (前面不補零) | 7 、 31 |
| 時間 | | |
| a | 上午或下午 (小寫) | am 、 pm |
| A | 上午或下午 (大寫) | AM 、 PM |
| g | 12 小時制的時 (前面不補零) | 1 ~ 12 |
| h | 12 小時制的時 | 01 ~ 12 |
| G | 24 小時制的時 (前面不補零) | 0 ~ 23 |

顯示目前的日期時間

| | | |
|----|--------------|-----------------|
| H | 24 小時制的時 | 00 ~ 23 |
| i | 分 | 00 ~ 59 |
| s | 秒 | 00 ~ 59 |
| 週 | | |
| D | 星期英文縮寫 | Mon 、 Fri |
| l | 星期英文名稱 | Monday 、 Friday |
| N | 星期(數字) | 1 (週日) 、 7 (週六) |
| w | 星期(數字) | 0 (週日) 、 6 (週六) |
| 時區 | | |
| O | 與格林威治標準時間的時差 | +0800 |

顯示目前的日期時間

- 以下例子示範如何使用 `date()` 函式顯示日期時間：

// 顯示各種格式的目前時間

| | |
|--|--|
| <code>echo date('Y-m-d h:i:s a');</code> | <code>//顯示 "2007-09-08 11:05:02 am"</code> |
| <code>echo date('D, M d Y');</code> | <code>//顯示 "Sat, Sep 08 2007"</code> |
| <code>echo date('y 年 n 月 j 日');</code> | <code>//顯示 "07 年 9 月 8 日"</code> |

取得目前的時間戳記

- 執行 `time()` 時不需要任何參數。
- `time()` 會回傳自 1970 / 01 / 01 00:00:00 到現在時間的相差秒數, 這個秒數稱為**時間戳記** (timestamp), 是 PHP 或其他 Unix / Linux 系統用來計算時間的依據。
- `time()` 經常用來與 `date()` 函式搭配, 用來取得 2 小時後、1 天前...等其他時間。
- 下面先介紹 `date()` 函式的延伸語法。

取得目前的時間戳記

`date(format_string, timestamp)`

- `format_string` : 帶有格式參數的字串

- `timestamp` : 要顯示日期時間的時間戳記

- 我們可以使用『`time()` + 與目前時間的相差秒數』來產生 2 小時後、1 天前...等其他時間的時間戳記, 然後放在 `date()` 的第二個參數中, 便可取得 2 小時後、1 天前...的日期時間。

- 其範例如下：

```
// 顯示 2 小時之後的日期時間 (1 小時共有 60 x 60 = 3600 秒)
echo date('Y-m-d H:i:s', time() + 2 * 60*60 );
// 顯示 1 天之前的日期時間 (1 天共有 86400 秒)
echo date('Y-m-d H:i:s', time() - 1 * 86400 );
// 顯示 1 星期之後的日期時間 (1 星期共有 7 天)
echo date('Y-m-d H:i:s', time() + 7 * 86400 );
```

取得指定日期時間的時間戳記

- `mktime()` 函式可以取得指定日期時間的時間戳記，下面是其語法：

```
mktime(hour, minute, second, month, day, year)
```

- "hour, minute, second, month, day, year" 分別是 "時, 分, 秒, 月, 日, 年"。
- 例如我們要取得 2007 / 12 / 25 00:00:00 的時間戳記，可以使用以下程式：

```
mktime(0, 0, 0, 12, 25, 2007)
```

取得指定日期時間的時間戳記

- 如果設定『日』參數為 0, 則可取得『月』參數的上個月最後一天的時間戳記, 例如：

// 顯示 2008 年 2 月最後一天的日期

// 解決閏年 2 月最後一天難以判斷的問題

```
echo date('Y-m-d', mktime(0, 0, 0, 3, 0, 2008) );
```

- mktime() 與 time() 搭配可以取得日期時間的間隔, 下面例子示範搭配 mktime() 與 time(), 取得距離 2007 耶誕節的天數。

取得指定日期時間的時間戳記

```
01  <?php
02      header('Content-Type:      text/html;      charset=utf-8');
03
04  $second=mktime(0,  0,  0,  12,  25,  2007)  -  time();
05  $days=   $second   /   (24*60*60);
06  echo  " 距離  2007  年耶誕節還有  $second  秒, 共  $days  天 ";
07  ?>
```

■ 執行結果：



設定時區

- 一般程式在日期時間的功能上，大多是向執行程式的主機取得日期時間，然後就直接輸出給使用者。
- 這樣的機制對於在本機執行的軟體並無不妥，因為大多數使用者的電腦都會調整為符合自己時區 (timezone) 的時間。
- 不過對於網頁程式來說，因為程式是在遠端主機上執行，所以非常有可能因為頻寬成本考量、或是跨國公司統一管理主機，而發生給台灣使用者的網站，卻架設在美國主機的情況。

設定時區

- 此時若直接使用程式執行主機的日期時間，便無法符合網頁使用者的時區了。
- 為了讓網頁程式更具彈性，PHP 從 5.1 版本後開始使用時區的觀念，只要在程式指定要顯示的時區，之後使用日期時間函式時，PHP 會自動計算時差，顯示指定時區的日期時間。
- 設定時區的函式為 `date_default_timezone_set()`，下面是其語法：
`date_default_timezone_set(timezone)`
 - `timezone`：指定我們要使用的時區

設定時區

- 台灣所在的時區為『Asia / Taipei』，所以使用日期時間函式之前，應該先如下指定使用『Asia / Taipei』時區：

```
date_default_timezone_set('Asia/Taipei');
```

舊版 PHP 如何指定時區？

- PHP 從 5.1 版開始引入時區的觀念，可以使用 `date_default_timezone_set()` 函式設定日期時間函式的時區。
- 不過如果您使用的 PHP 版本比 5.1 舊，就必須自行撰寫程式碼，才能依照時區顯示日期時間：

```
// 設定      $timezone 變數,用以存放時區的時差
$timezone=8;           // 以台灣為例，時差為 +8 小時

// gmdate()函式可以取得目前的格林威治時間,其語法和 date()完全相同。
// 將目前的格林威治時間加上時差,就是指定時區的時間
echo  gmdate("y-m-d  h:i:s",  time()  +   $\frac{3600}{\text{一小時有 3600 秒}}$  *  $\frac{\$timezone}{\text{時差}}$ );
```


5 - 7 電子郵件

- PHP 發送電子郵件的函式為 `mail()`，本節將介紹其用法與需要注意的安全事項。
 - `php.ini` 的 `mail` 函式設定
 - `mail()` 函式的語法
 - 指定寄件者
 - 設定中文編碼

php.ini 的 mail 函式設定

- 使用 mail() 函式之前，必須先設定好 php.ini 檔案。
- 請開啟 php.ini 檔案，修改下面項目：

```
[mail    function]
; For Win32 only.
SMTP    = 192.168.0.5
```

← 如果網頁主機為 Windows 系統，請在此項目指定 PHP 寄信時要使用的 SMTP 伺服器

- 設定完畢後，請重新啟動 Apache，才能讓新設定生效。

mail() 函式的語法

- 下面是 mail() 函式的語法：

```
mail(to,      subject,      body)
```

- to, subject, body 分別代表電子郵件的收件者，標題，內文。
- 執行後若 mail() 回傳 TRUE，表示執行成功，如果回傳 FALSE，則表示執行失敗，信件無法寄出。

mail() 函式的語法

- 下面是一個簡單的 mail() 範例：

```
if ( mail('tony@php.flag.com.tw', 'testmail', 'This is a test!') )
    echo '成功寄出信件';
else
    echo '無法寄出信件';
```

- 如果希望同時在收件者顯示姓名與郵件位址，
可以使用『Name <user@abc.com>』的格式：

```
<?php
// 設定收件者的姓名為 Tony Chen
mail('Tony Chen <tony@php.flag.com.tw>', 'testmail', 'This is a test!')
?>
```

指定寄件者

■ 電子郵件的表頭

- 在說明如何指定寄件者之前，我們需要先瞭解電子郵件的格式：

From: 寄件者

To: 收件者

Subject: 標題

其他表頭設定



表頭 (header)

信件內文

- 電子郵件的寄件者、收件者、標題等會全部放在表頭中，除了這些資料以外，表頭還會包含編碼、附件收件者等資料。

指定寄件者

- 所以信件是否能夠成功寄出，是否可以讓收件者正確讀取，都仰賴標頭的設定。
- 雖然 `mail()` 函式沒有寄件者的參數，但是我們仍然可以在表頭中自訂寄件者。
- 在表頭中指定寄件者
 - `mail()` 函式設定表頭的語法如下：
`mail(to, subject, body, header)`
 - 下面例子為您示範如何在表頭中指定寄件者。

指定寄件者

```
01  <?php
02      $to='tony@php.flag.com.tw';
03      $subject='Order      confirmation';
04      $body='Hello      Tony,
05
06      We      have      received      your      order.      Our      sales      will      contact      you      soon.
07
08      Thank      you      !      ';
09
10 // 在電子郵件表頭中加上寄件者
11 $header = "From: service@php.flag.com.tw";
12
13 // 寄發信件
14 mail($to,      $subject,      $body,      $header);
15 ?>
```

設定中文編碼

- 當信件的內文包含中文字時，必須在電子郵件**表頭**指定此信件的編碼 (例如 UTF-8)，才能讓收件者正確讀取信件的内容。
- 此外，在電子郵件標準格式中，表頭的部分不允許使用雙位元的文字。
- 所以標題或寄件者若是包含中文，應該使用 `mb_encode_mimeheader()` 函式將其編碼為單位元的字串。

設定中文編碼

- 以下例子示範如何指定內文編碼為 UTF-8, 並且將信件的標題編碼：

```
01  <?php
02  $to='tony@php.flag.com.tw';
03  $subject=' 啓用通知信 ';
04  $body='tony    您好：
05
06  您的帳號 tony 已經可以使用了，請使用您申請時輸入的密碼登入我們網站。
07
08  謝謝您的支持！';
09
10  // 設定信件內文為純文字格式， 編碼為 UTF-8
11  $header='Content-type: text/plain; charset=UTF-8'; ← A
12
```

設定中文編碼

```
13 // 在電子郵件表頭中加上寄件者
14 $header .= "\nFrom: service@php.flag.com.tw"; ← B
15
16 // 設定 mb_encode_mimeheader() 函式的內部編碼為 UTF-8
17 mb_internal_encoding("UTF-8"); ← C
18 // 使用 mb_encode_mimeheader() 函式將標題編碼
19 $subject=mb_encode_mimeheader($subject, 'UTF-8');
20
21 // 寄發信件
22 mail($to, $subject, $body, $header);
23 ?>
```

設定中文編碼

- (A) 電子郵件表頭可以同時包含多項設定, 但是每項設定必須獨立為一行。
- 由於程式中 (A) 處已經將內文編碼項目加入表頭中, 所以 (B) 處附加寄件者的項目時, 必須前面必須有換行符號 "\n", 才能正確設定表頭。
- (C) 因為 `mb_encode_mimeheader()` 預設的字串編碼為西方的 ISO-8859-1。

設定中文編碼

- 但我們使用的是 UTF-8 編碼的中文字，所以程式中 (C) 處要先使用 `mb_internal_encoding()` 將內部預設編碼改成 UTF-8。
- `mail()` 函式只能寄發純文字格式的電子郵件，如果您需要寄送 HTML 格式或是附加檔案的郵件，請使用 `PHPMailer` 或是 `PEAR` 的相關類別，即可發送格式比較複雜的電子郵件。

mail() 函式的安全隱憂 — Mail injection

- 目前網站上常見到聯絡表單，讓使用者填寫表單後就可以詢問問題。
- 設計聯絡表單程式時，會從使用者輸入的表單中取得標題、信件內文的資料，而這些外部資料有可能造成您程式中的 mail() 函式變成廣告垃圾信的發送機。
- 前面提到電子郵件的格式如下：

From: 寄件者
To: 收件者
Subject: 標題
其他表頭設定

信件內文

mail() 函式的安全隱憂 — Mail injection

- 如果是聯絡表單的程式，可能會如下設計：

```
// 從表單中取得寄件者的資料
```

```
$header = "From: $_POST['from'];
```

```
// 將信件寄給 service@flag.com.tw
```

```
mail('service@flag.com.tw', $_POST['subject'], $_POST  
['message'], $header);
```

- 正常的狀況下，上面 mail() 函式寄出的信件如下：

From: tony@php.flag.com.tw

To: service@flag.com.tw

Subject: 請問 Linux 書籍

請問你們有出哪些 Linux 的書？

mail() 函式的安全隱憂 — Mail injection

- 不過如果使用者故意在聯絡表單的 from 欄位輸入
『tony@php.flag.com.tw%0ACc:spam@abc.com』(%0A 是換行字元的編碼)。
- 或是利用特殊的方法, 直接以 POST 方式送出帶有換行字元的欄位值, 就會導致 mail() 函式寄出下面信件：

From: tony@php.flag.com.tw

Cc: spam@abc.com

To: service@flag.com.tw

Subject: 請問 Linux 書籍

請問你們有出哪些 Linux 的書？

mail() 函式的安全隱憂 — Mail injection

- 上面可以看到, From 項目下面已經被使用者加上 Cc 項目, 所以此信件除了寄給 service@flag.com.tw, 還會被寄到使用者指定的其他信箱。
- 為了避免上述問題, 建議您：如果要將使用者輸入字串做為 mail() 函式的參數, 則**除了內文以外的所有參數, 都應該先過濾換行字元**。
- 例如下面範例會過濾收件者中的換行字元：

```
if (eregi("\r\n", $_POST['from'])) {  
    errMsg=" 請勿在寄件人位址輸入換行字元 ";  
}
```


5 - 8 其他重要函式

- 除了前面介紹的函式之外, PHP 還提供了許多其他好用的函式, 本節將為您說明這些函式的使用方法。
 - die() 與 exit() 函式
 - header() 函式
 - 測試變數
 - mt_rand() 函式

die() 與 exit() 函式

- 如同字面上的意思, die() 與 exit() 函式可以讓程式停止執行, 只要遇到 die() 或 exit(), PHP 程式就會立刻停止。
- 一般程式執行檔案開啟、資料庫連線等動作時, 會用 if 判斷是否執行成功, 若不成功, 就會使用 die() 或 exit() 讓程式停止。
- die() 與 exit() 是名稱不同但功能用法完全相同的函式, 兩者的語法如下。

die() 與 exit() 函式

```
die(error_msg)  
exit(error_msg)
```

■ error_msg：要顯示的錯誤訊息字串

- 您可以將程式執行時的錯誤訊息，放入 die() 或 exit() 的參數。
- 如此程式停止執行時，會將參數內的字串顯示在網頁上，讓使用者瞭解程式為什麼會停止執行。

header() 函式

- header() 函式將送出 HTTP 的通訊協定表頭，一般最常見的用途是將網頁轉向，或是設定網頁的編碼：

```
// 將網頁轉向到 http://www.flag.com.tw  
header('Location: http://www.flag.com.tw');
```

```
// 設定網頁使用 UTF8 編碼  
header('Content-Type: text/html; charset=utf-8');
```

- HTTP 的通訊協定表頭必須在輸出網頁內容之前送出，所以 header() 函式執行之前，若有 echo()、或 HTML 標籤等輸出，將會導致 header() 函式執行錯誤。

測試變數

- 我們可以使用以下函式來測試變數：

| 函式 | 用途 |
|----------------|---------------------|
| isset(變數) | 測試變數是否已經設定 |
| empty(變數) | 測試變數是否為0或空字串 |
| is_array(變數) | 測試變數是否為陣列 |
| is_int(變數) | 測試變數是否為整數 |
| is_null(變數) | 測試變數是否為NULL |
| is_numeric(變數) | 測試變數是否為數字、或是數字形式的字串 |
| is_string(變數) | 測試變數是否為字串 |

- 這些函式如果為真，便會回傳 TRUE，若否，則回傳 FALSE。

mt_rand() 函式

- 如果程式中需要亂數，例如抽獎時需要隨機選出得獎者，可以使用 mt_rand() 函式來產生亂數，其語法如下：

```
mt_rand(min,      max)
```

- min 和 max 分別指定最小值與最大值。
- mt_rand() 會產生一個介於最小值與最大值之間的亂數 (包含最小值與最大值)。

mt_rand() 函式

- 如果沒有指定最小值與最大值, 在 Windows 系統上, 最小值預設為 0, 最大值預設為 32768。
- 下面範例會產生 1 ~ 100 之間的亂數。

```
mt_rand(1, 100)
```

5 - 9 實戰演練 — 橫幅廣告輪播

- 為了達到宣傳與廣告的效果，許多網站會擺放橫幅圖片，希望能在使用者一進入就吸引其目光：

吸引使用者目光的橫幅廣告



實戰演練 — 橫幅廣告輪播

- 通常橫幅廣告都會有多張圖片，每個不同連線隨機送出不同圖片，不但能同時進行多項宣傳，也能讓網站比較不那麼單調。
- 此外，橫幅廣告各圖片會連結到不同網址，這樣當使用者看到有興趣的廣告時，便可以直接點選進入相關網頁。
- 要製作這樣的橫幅廣告輪播機制並不難，只要使用 PHP 撰寫程式，隨機送出不同圖片的 `` 與 `<a>` 標籤即可。

實戰演練 — 橫幅廣告輪播

■ 以下是程式碼：

```
01  <?php
02  // 定義 $banner 二維陣列，用以存放橫幅廣告的路徑與連結網址
03      $banner=array(
04          array('ban_top1.gif',
05                'http://www.flag.com.tw/book/5105.asp?bokno=FS124'),
06          array('ban_top2.gif',
07                'http://www.flag.com.tw/book/5105.asp?bokno=FS691'),
08          array('ban_top3.gif',
09                'http://www.flag.com.tw/book/5105.asp?bokno=F6603')
10      );
11
```

實戰演練 — 橫幅廣告輪播

```
12 //count() 函式可以取得陣列中元素的數量，
13 // 此處使用 count() 計算 $banner 陣列中有多少廣告
14     $amount=count($banner);
15
16 // 利用亂數在 $amount 的範圍內選出一個廣告
17     $index=mt_rand(0,     $amount-1);
18
19 // 以 <a> <img> </a> 標籤顯示橫幅廣告
20     echo     sprintf('<a     href="%s"></a>',
21                     $banner[$index][1], $banner[$index][0]);
22     ?>
```

實戰演練 — 橫幅廣告輪播

■ 執行結果：



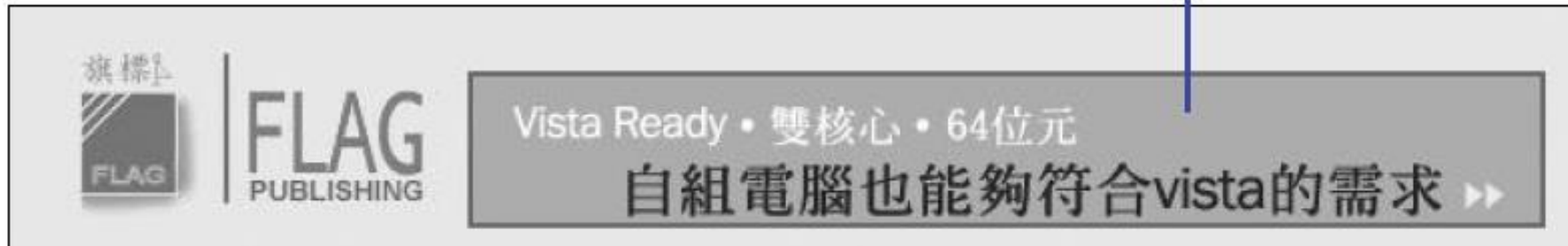
- 日後當其他 PHP 網頁要放置橫幅廣告，只要設定好 \$banner 陣列，然後如下引用此程式即可：

```
include('Ch05-15.php');
```

5 - 10 實戰演練 — 網頁聯絡表單

- 網頁聯絡表單可以讓使用者更方便地與網站管理者聯絡，不用先開啟 Outlook 等郵件軟體寄信，只要在網頁表單上輸入資料，就可以直接傳送給網站管理者。
- 我們將製作的網頁聯絡表單如下：

這是上一節製作的
橫幅廣告輪播



實戰演練 — 網頁聯絡表單

這些欄位的名稱由上而下依序為 `name`、`e m a i l`、`telephone`、`message`

此表單會傳送給 `Ch05-16-2.php`

您可以使用下面表單與我們連絡

您的姓名：

您的電話：

您的電子郵件：

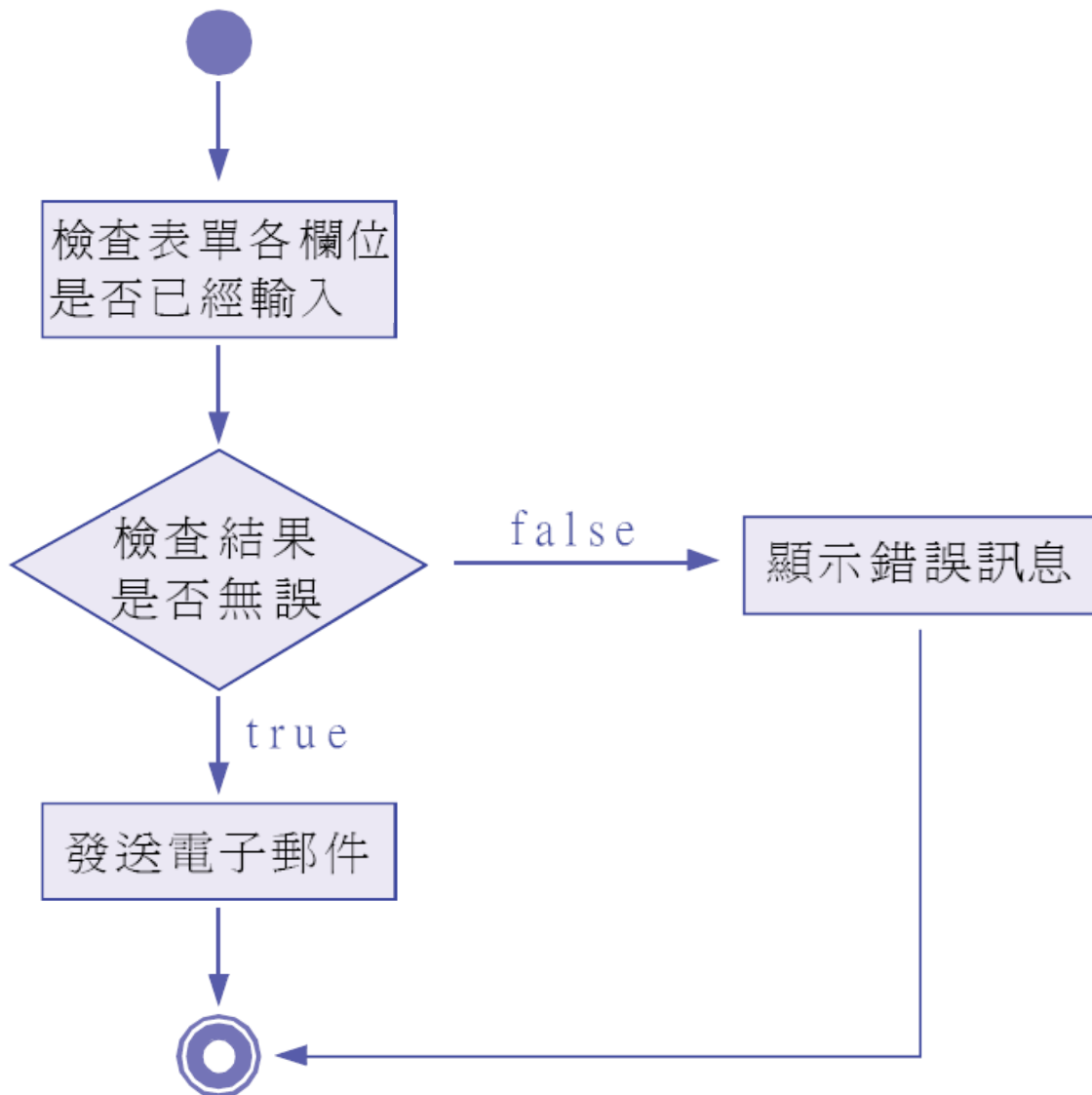
聯絡的內容：

送出

重設

實戰演練 — 網頁聯絡表單

- 上面 Ch05-16-1.php 的表單資料會傳送給 Ch05-16-2.php, Ch05-16-2.php 收到後會進行檢查, 若無誤則寄送電子郵件給管理者。
- 程式運作流程圖如下：



實戰演練 — 網頁聯絡表單

■ 下面是 Ch05-16-2.php 的原始碼：

```
01  <?php
02      header('Content-Type:          text/html;          charset=utf-8');
03
04  //-----          設定相關變數          -----
05  // 定義 $errMsg 變數， 用以存放錯誤訊息
06  $err_msg = '';
07
08  // 設定收件者、標題
09  $to = 'service@php.flag.com.tw';
10  $subject = ' 使用者來函 ';
11
```


實戰演練 — 網頁聯絡表單

```
1 2 // 將標題以 UTF-8 編碼
1 3 mb_internal_encoding("UTF-8");
1 4 $subject=mb_encode_mimeheader($subject, 'UTF-8');
1 5
1 6
1 7 //----- 檢查表單各欄位是否已經輸入 -----
1 8 // 檢查表單的 name 欄位是否已經輸入
1 9 if ( isset($_POST['name']) && ! empty($_POST['name']) ) {
2 0     $name=$_POST['name'];
2 1 } else {
2 2     $err_msg.=' 您忘記輸入名字<br />';
2 3 }
2 4
2 5 // 檢查表單的 telephone 欄位是否已經輸入
2 6 if ( isset($_POST['telephone']) &&
```

實戰演練 — 網頁聯絡表單

```
    ! empty($_POST['telephone']) ) {
27         $telephone=$_POST['telephone'];
28     } else {
29         $err_msg.=' 您忘記輸入電話<br />';
30     }
31
32 // 檢查表單的 email 欄位是否已經輸入
33 if ( isset($_POST['email']) && ! empty($_POST['email']) ) {
34     $email=$_POST['email'];
35 } else {
36     $err_msg.=' 您忘記輸入電子郵件<br />';
37 }
38
39 // 檢查表單的 message 欄位是否已經輸入
40 if ( isset($_POST['message']) && ! empty($_POST['message']) ) {
41     $message=$_POST['message'];
```

實戰演練 — 網頁聯絡表單

```
42 } else {
43     $err_msg.=' 您忘記輸入留言<br />';
44 }
... //HTML 部分省略

53 <body bgcolor="#636563">
54     <div id="title" style="width: 600px; height: 100px;
        background-color: #efefe7; z-index: 101; margin-left:auto;
        margin-right:auto; padding: 10px">
55         <?php include('Ch05-15.php'); ?>
56     </div>

...//HTML 部分省略

59 <?php
60 //----- 發送電子郵件 -----
61 // 若 $err_msg 是空字串，表示沒有錯誤發生，可以開始寄送電子郵件的動作
```

引用上一節製作的
Ch05-15.php

實戰演練 — 網頁聯絡表單

```
6 2 if ($err_msg == ""){
6 3
6 4 // 設定信件的內文
6 5 $body=" 姓名:      $name\n 電話:      $telephone\n 留言:      $message";
6 6
6 7 // 因為程式只有採用使用者輸入的 email 欄位做為參數，
6 8 // 所以只需檢查 email 欄位中是否包含換行字元
6 9 if (eregi("\r\n", $email)) {
7 0     die(" 請勿輸入換行字元 ");
7 1 }
7 2
7 3 // 在電子郵件表頭設定信件內文為純文字格式，編碼為 UTF-8
7 4 $header='Content-type:      text/plain;      charset=UTF-8';
7 5
```

實戰演練 — 網頁聯絡表單

```
76 // 在電子郵件表頭中加上寄件者
77 $header .= "\nFrom: $email";
78
79 if ( mail($to, $subject, $body, $header) ) {
80     echo '信件傳送成功，我們會儘快回覆您，謝謝';
81 }
82 else {
83     echo '傳送失敗!';
84 }
85 }
86 else {
87     echo $err_msg . '<p>請按瀏覽器的上一頁鈕重新輸入</p>';
88 }
89 ?>
```

實戰演練 — 網頁聯絡表單

■ 執行結果：

提醒使用者
哪些欄位沒
有輸入

您忘記輸入名字
您忘記輸入電話
您忘記輸入電子郵件
您忘記輸入留言

請按瀏覽器的上一頁鈕重新輸入

信件成
功傳送

信件傳送成功, 我們會儘快回覆您, 謝謝

實戰演練 — 網頁聯絡表單

■ 網站管理者收到的信件：

