

# PHP+MySQL

## 互動式網頁程式設計班

---

[第三章] PHP基礎語法：變數、陣列、  
流程控制

國立台灣大學 資訊工程學系暨研究所  
資訊系統訓練班  
講師：烏明學

# 章節目錄

- 3-1 PHP的歷史與背景技術
- 3-2 註解、變數與常數
- 3-3 陣列的基礎
- 3-4 與使用者互動
- 3-5 運算子
- 3-6 流程控制



## ◦ 3-1 PHP的歷史與背景技術

# PHP 的歷史 (1)

- 在 PHP 出現之前，撰寫網頁程式大部份採用 CGI 完成
  - CGI 是 Common Gateway Interface 的縮寫，它可用 Perl、C/C++ 完成
  - Perl 跟 C/C++ 畢竟是專門拿來撰寫網頁程式的語言，因此不但必須把程式碼跟 HTML 分開來、語言的學習曲線也較長
  - 微軟的 ASP 問市後，其將程式碼嵌入 HTML 的特色非常吸引人
    - 但...只能在微軟的 IIS 伺服器上執行

# PHP 的歷史 (2)

- 1995年，一名叫做 Rasmus Lerdorf 的工程師，因為覺得 Perl 寫網頁程式仍然不夠方便，便用 C 語言寫了一個網頁開發工具稱為：**Personal Home Page**，簡稱 PHP
- 公開後廣受歡迎，並在隔年推出 PHP2
- 1998年後由一家公司 Zend 接手，並重新改寫整個核心，推出 PHP3 後市佔率便一再攀升，直至今日

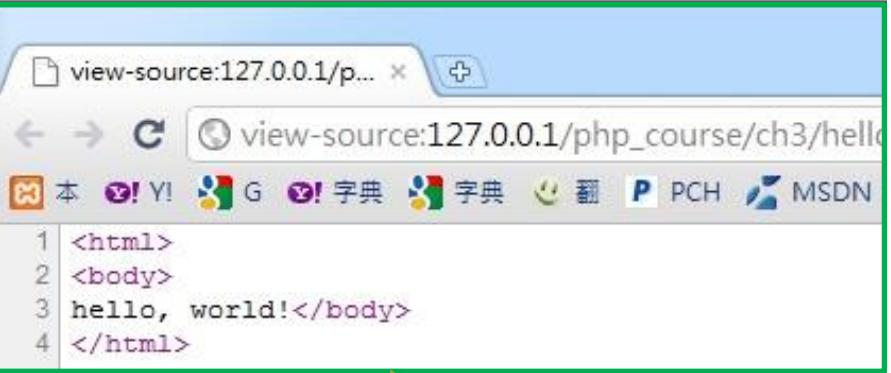
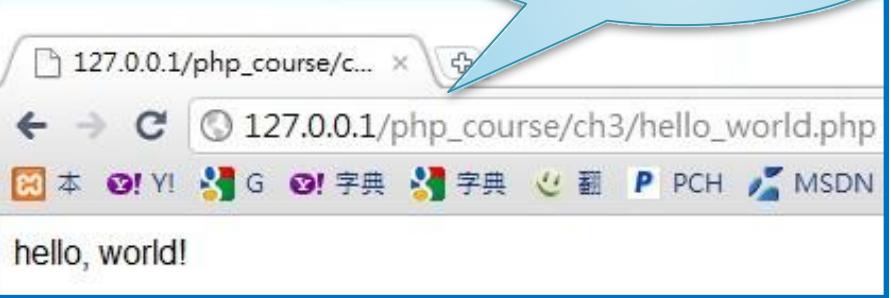
# 伺服器端程式語言

- PHP 是「伺服器端」的程式語言
  - 所有 PHP 的程式碼都會經過 PHP 引擎的 解析後，再將要輸出的部份轉成 HTML 的 內容，因此程式邏輯不會在檢視原始碼
  - HTML 無關邏輯，但 PHP 可以有邏輯判斷在其中
  - 可與資料庫系統或檔案系統結合
    - JavaScript 不行
    - 所以純網頁無法把**大量**資料存起來

# PHP 網頁檢視原始碼

瀏覽器中看到的樣子

```
1 <html>
2 <body>
3 <?php
4 // php 中的註解，不會顯示在畫面中
5 $hello = 'hello, world!';
6 echo $hello;
7 ?>
8 </body>
9 </html>
```

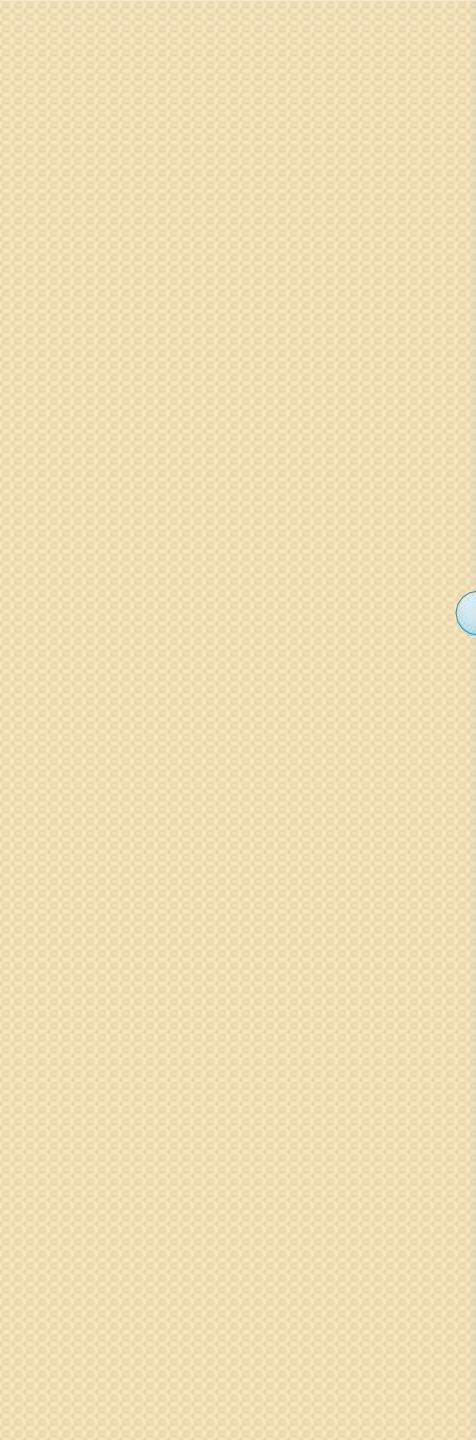


編輯的內容

檢視原始碼後看到  
的樣子

# 準備正式上路

- PHP 看起來不錯吧？
- 那就準備好開始寫 PHP 吧！
- 但...請記得
  - PHP 是伺服器端的程式語言，所以請記得：
    - Apache 一定得開啟
    - Notepad++ 編輯好的 PHP 檔案，需放在 AppServ 中的 www 目錄中
    - 需打開瀏覽器輸入 <http://127.0.0.1:8080/> 及對應的位址中經由 Apache 與 PHP 解讀成對應的 HTML 內容才看的到！



## ◦ 3-2 註解、變數與常數

# 程式註解

- 註解的目的
  - 註解是程式在執行時會被略過不執行的程式碼，目的通常是為了讓自己註記這段程式的用途和或是偵錯
- PHP中的註解：
  - 多行
    - /\* 註解文字 \*/
  - 單行
    - // 註解文字
    - # 註解文字

```
1 <html>
2 <body>
3 <?php
4 /*
5 這是多行的註解
6 */
7
8 // 這是單行的
9
10 # 這也是單行的
11 ?>
12 </body>
13 </html>
```

# 變數(Variable)

- 只要不是固定的數，都可稱為變數
- 原則上，變數的資料型態是沒有限制的，可用「**整數**」、「**浮點數**」或「**字串**」來呈現
- 變數命名有大小寫的分別，即：  
**\$name** 與 **\$Name**是兩個不同的變數

# 資料類型

- 整數：用來表示無小數的普通數字。例如：100
- 浮點數：又稱double或實數，用表示小數或指數的數字。例如：3.14
- 字串：用來表示非數字值，例如：字母、標點符號，甚至數字。例："2 hours"

# 弱型別的變數

- 在PHP裡，變數必須在名稱前面加上'\$'字元
- PHP是一種弱定類型語言，不必事先宣告變數，也**不必聲明它們儲存哪一種資料**，這也意味變數的資料類型可隨時改變。例如：
  - `$name = 'Allan';`
  - `echo $name;`

# 整數資料型態

- 所佔記憶體：4位元組
- 數值範圍：  
-2147483648 ~ +2147483647
- 舉例：高爾夫球的標準桿、投票結果
- 程式碼範例：
  - \$pen = 24; // 鉛筆24支
  - \$egg = 10; // 雞蛋10個

# 浮點數資料型態

- 所佔記憶體：4位元組
- 數值範圍： $\pm(1.7E-308 \sim 1.7E+308)$
- 舉例：學生的平均成績、貨幣
- 程式範例：
  - \$average = 90.8; // 成績平均為90.8分
  - \$money = 7839.81; // 貨幣換算後為  
7839.81元

# 字串資料型態

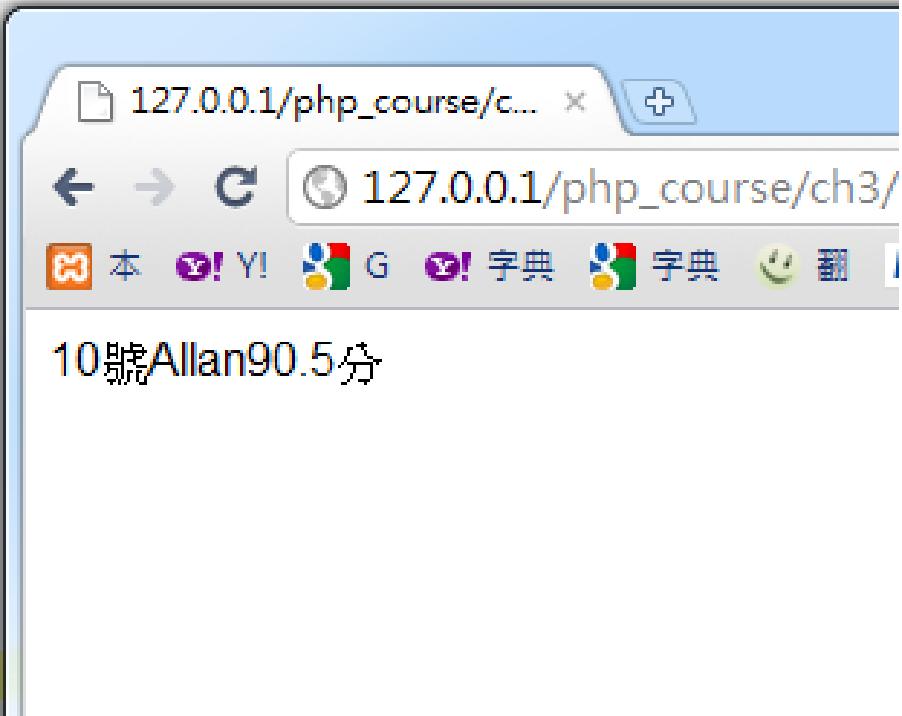
- 字串型態的資料需在前後加上單引號「'」或是雙引號「"」
- 所佔記憶體：未定，看字串的長度大小
- 舉例：姓名、書名、帳戶、帳號
- 程式範例：  
`$name = '王小明';`  
`$account = 'b8505006';`

# 變數的命名原則

- 必須由「**英文字母**」、或底線符號「\_」開頭
- 除了開頭字母之外必須由「**英文字母**」、「**數字**」或底線符號「\_」組成
- 不可使用關鍵字(PHP內建使用的字，例如：echo)

# 啟始化變數與輸出的範例

```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=U
4 </head>
5 <body>
6 <?php
7 $num = 10;
8 $name = 'Allan';
9 $score = 90.5;
10
11 echo $num;
12 echo '號';
13 echo $name;
14 echo $score;
15 echo '分';
16 ?>
17 </body>
18 </html>
```



# 練習一下

- 利用上面的例子，運用串接運算子「.」，將字串印出來
- 利用相同的三個變數：`$num`、`$name` 和`$score`，印出相同的結果
- ```
<?php  
echo $num . '號' . $name . $score . '分';  
?>
```

# 課堂練習 (3-1)

- 寫一個 PHP 程式，裡面需要有：
  - 註解：註解內容包含作者名稱跟時間
  - 一個變數：`$hello = '大家好！';`
  - 再用 `echo` 印出這個變數
  - 別忘了網頁編碼的設定喔！

The screenshot shows two browser windows side-by-side. The top window displays the URL `127.0.0.1/php_course/ch3/pt31.php` in the address bar. The page content is "大家好！". The bottom window shows the source code of the same file:

```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4 </head>
5 <body>
6 大家好！</body>
7 </html>
```

# 跳脫字元

- 「單引號」、「雙引號」是把字串包起來的特別符號，那如果我想在畫面中印出像這樣的內容該怎麼辦呢？



- 這樣嗎？
  - <?php echo '想跟他說聲 'Hi';?>

# 使用跳脫字元

- 這時候我們可以使用跳脫字元
- 常見的跳脫字元：

| 跳脫字元 | 說明          |
|------|-------------|
| \'   | 顯示符號「'」     |
| \"   | 顯示符號「"」     |
| \\"  | 顯示符號「\」     |
| \\$  | 顯示符號「\$」    |
| \n   | 換行字元        |
| \r   | 使游標歸位字元     |
| \t   | 跳至下一個TAB定位處 |

# 使用跳脫字元的範例

```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4 </head>
5 <body>
6 <?php
7 echo '想跟他說聲 \'Hi\'';
8 ?>
9 </body>
10 </html>
```



The screenshot shows two browser windows. The left window displays the rendered HTML output: "想跟他說聲 'Hi'" in black text on a white background. The right window shows the source code of the PHP file, which includes the original code and the rendered output as part of the page content.

```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4 </head>
5 <body>
6 想跟他說聲 'Hi'</body>
7 </html>
```

# 練習一下

- 印出  
    這本書價格為：\$400元
- <?php  
    echo "這本書價格為：\\$400元";  
    ?>

# 該用單引號還是雙引號？

- `$var = '字串';`  
跟  
`$var = "字串";`  
差別在哪呢？
- 差別在於：
  - 被單引號包住的內容，會完全照實的輸出
  - 被雙引號包住的內容，則會解析其中內容
- 範例：
  - `echo 'test\n';` → 會印出「test\n」
  - `echo "test\n";` → 會印出「test」跟一個換行
  - `echo '$test';` → 會印出「\$test」
  - `echo "$test";` → 會印出「\$test這個變數的內容值」

# 常數(constant)

- 「常數」通常指的是不會變的數，如理化課程裡的「重力加速度G」、地科課程的「光速」、數學課程的「 $\pi$ 」等等這些數都是屬於不變的「常數」

`$G=9.8;`

`$Light=299792.458;`

`$PI=3.141592657;`

- 雖然上面的程式碼也可以，但是，無法保障這些常數在程式中不會被變動到
- define()**：PHP內建函式，用來定義常數，以**define()**定義的變數無法更改變數值
- 以**define()**定義的常數，**使用時不用加\$**

# 使用常數的範例

The screenshot shows three windows illustrating the use of PHP constants:

- Code Editor:** Shows the PHP code defining constants G, light, and pi, and echoing them.
- Browser 1:** Shows the raw HTML output of the PHP code, including the constant definitions and their values separated by newlines.
- Browser 2:** Shows the final rendered output where the constants are correctly evaluated and displayed as 9.8, 299792.458, and 3.1415926 respectively.

```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4 </head>
5 <body>
6 <?php
7 define ('G', 9.8);
8 define ('light', 299792.458);
9 define ('pi', 3.1415926);
10
11 echo '重力加速度常數是：' . G . '<br>';
12 echo '光的速度常數是：' . light . '<br>';
13 echo '圓週率的常數是：' . pi . '<br>';
14 ?>
15 </body>
16 </html>
```

view-source:127.0.0.1/p... view-source:127.0.0.1/php\_course/ch3/ex3.php

```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4 </head>
5 <body>
6 重力加速度常數是：9.8<br>光的速度常數是：299792.458<br>圓週率的常數是：3.1415926<br></body>
7 </html>
```

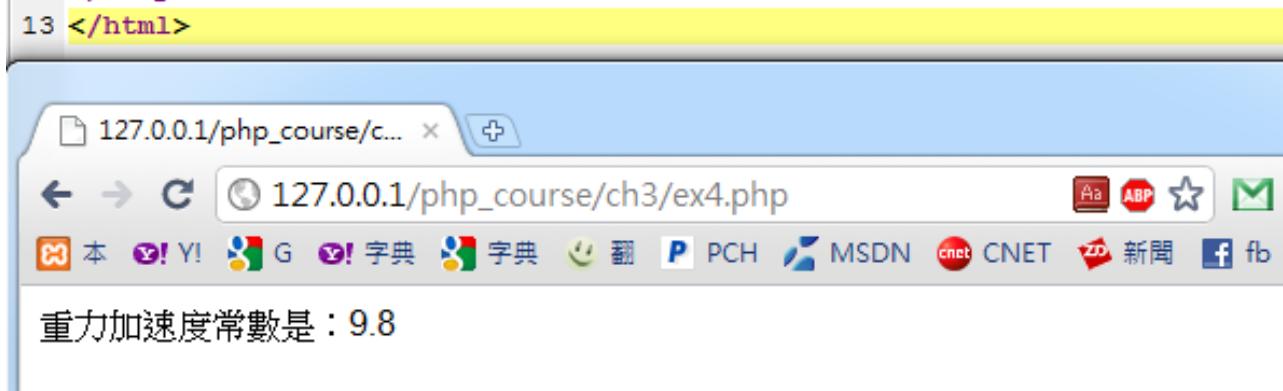
127.0.0.1/php\_course/c... 127.0.0.1/php\_course/ch3/ex3.php

```
重力加速度常數是：9.8
光的速度常數是：299792.458
圓週率的常數是：3.1415926
```

# 常數不可再變

- 已經定義過的常數，就不可再做變更，即使再一次定義也是一樣

```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4 </head>
5 <body>
6 <?php
7 define ('G', 9.8);
8 define ('G', 10.8);
9
10 echo '重力加速度常數是：' . G . '<br>';
11 ?>
12 </body>
13 </html>
```





## 3-3 陣列的基礎

# 陣列 (Array)

- 陣列可以想像成是一串貼有標籤的貨櫃，標籤就是陣列裡的索引(index)，而貨櫃內所裝的東西就是陣列裡的元素(element)
- 在程式裡可以有許多不同的貨櫃(array)，裝著不同種類的物品(element)，並可根據貼上的標籤(index)，歸類物品、尋找物品

標籤皆以  
0號開頭！

1號貨櫃  
(array)

| 0號標籤 | 1號標籤 | 2號標籤 | 3號標籤 | 4號標籤 | 5號標籤 |
|------|------|------|------|------|------|
| 物品A  | 物品B  | 物品C  | 物品D  | 物品E  | 物品F  |

2號貨櫃  
(array)

| 0號標籤 | 1號標籤 | 2號標籤  | 3號標籤 | 4號標籤 | 5號標籤 |
|------|------|-------|------|------|------|
| 物品I  | 物品II | 物品III | 物品IV | 物品V  | 物品VI |

# PHP的宣告與初始化陣列 (1)

- 方式1：使用 **array()**
  - **array()** 是 PHP 內建函式，可以創造一個陣列
  - 基本的使用方法：
    - 設定時：`$array = array('Allan', 'Bill', 'Cindy', 1, 2, 3.0);`
    - 取用時：`echo $array[0]; // 畫面印出 Allan`  
`echo $array[3]; // 畫面印出 1`
  - 或是我們可以自訂索引名稱：
    - 設定時：`$student = array('name'=>'王小明', 'chinese'=>80, 'english'=>70.5);`
    - 取用時：`echo $student['name']; // 畫面印出 王小明`  
`echo $student['chinese']; // 畫面印出 80`

# PHP的宣告與初始化陣列 (2)

- 方式2：直接寫入
  - 紿定傳統數字索引寫入：
    - \$user[0] = '王小寶';
    - \$user[1] = 168;
    - \$user[2] = 50.5;
  - 使用字串當index：
    - \$user['name'] = '王小寶';
    - \$user['height'] = 168;
    - \$user['weight'] = 50.5;
  - 也可以使用自動增加編號：
    - \$user[] = '王小寶'; // 取用時使用 \$user[0]
    - \$user[] = 168; // 取用時使用 \$user[1]
    - \$user[] = 50.5; // 取用時使用 \$user[2]

# 陣列的範例

- 利用陣列實作一個簡單的字典查詢程式：

```
1 <html>
2   <head>
3     <meta http-equiv='content-type' content='text/html; charset=UTF-8'>
4   <title></title>
5   </head>
6   <body>
7   <?php
8     $input = 'dragon';
9     $word['dragon'] = '一種遠古流傳的生物';
10    $word['money'] = '一種人不能沒有的東西';
11    $word['gun'] = '一種人具有強大殺傷力的致命武器';
12    $word['car'] = '一種可以搭載及具有運輸能力的機械裝置';
13    $word['dog'] = '一種哺乳類的四肢動物';
14    echo '使用者輸入的查詢是：' . $input . '<br><br>';
15    echo '查詢結果：' . $word[$input];
16  ?>
17  </body>
18 </html>
```

使用者輸入的查詢是：**dragon**

查詢結果：一種遠古流傳的生物



## 3-4 與使用者互動

# 與使用者互動

- PHP 中與使用者互動的方式有兩種：
- **GET** (用在網址傳值) `$_GET` 陣列
  - 使用者點選某個超連結，PHP 可以藉由網址中 **GET 參數** 的不同，得知使用者的需求
  - 例：相簿(上一張/下一張)、分頁(上一頁/下一頁)
- **POST** (用在表單傳值) `$_POST` 陣列
  - 使用者填寫完表單按下 **submit** 後送出所填寫的表單資訊給 PHP 程式
  - 例：會員註冊、發表文章

# GET 傳值

- 在網址後加入  $?[k1]=[v1]&[k2]=[v2]...$ 
  - 例：get.php?name=rick&sn=1234
  - 表示使用者提供給我們的資訊是：
    - name 是 allan
    - sn 是 1234
  - 如此一來，在 get.php 這支程式中，將有兩個變數可供我們使用：
    - `$_GET['name']` // 將它 echo 出來內容是 allan
    - `$_GET['sn']` // 將它 echo 出來內容是 1234

# GET 實作範例 (1)

## page.php

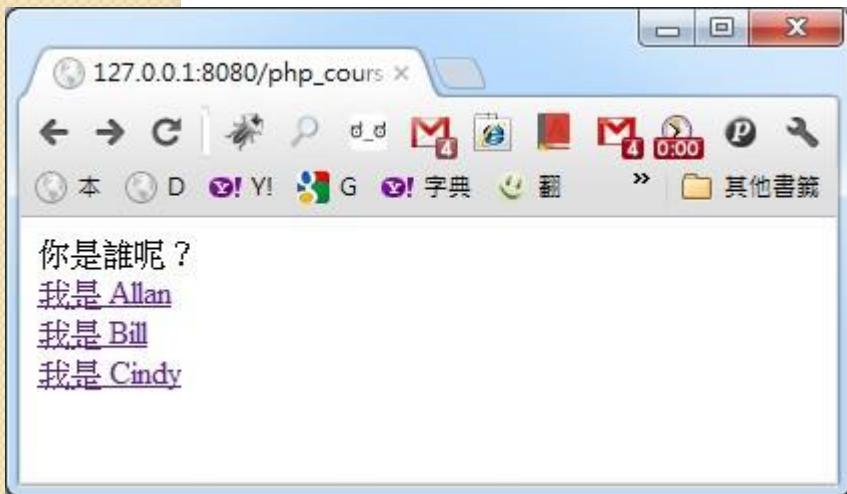
```
1  □<html>
2  □<head>
3  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
4  <title></title>
5  </head>
6  □<body>
7  你是誰呢？<br>
8  <a href="get.php?name>Allan&gender=Male&age=18">我是 Allan<br>
9  <a href="get.php?name=Bill&gender=Male&age=40">我是 Bill<br>
10 <a href="get.php?name=Cindy&gender=Female&age=25">我是 Cindy<br>
11 </body>
12 </html>
```

## get.php

```
1  □<html>
2  □<head>
3  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
4  <title></title>
5  </head>
6  □<body>
7  Hi, <?php echo $_GET['name'] ?><br>
8  我知道你是 <?php echo $_GET['gender'] ?><br>
9  今年 <?php echo $_GET['age'] ?> 歲
10 </body>
11 </html>
```

# GET 實作範例 (2)

page.php



get.php



Hi, Allan  
我知道你是 Male  
今年 18 歲



Hi, Bill  
我知道你是 Male  
今年 40 歲



Hi, Cindy  
我知道你是 Female  
今年 25 歲

# POST 傳值

- HTML 裡的表單元件
  - <form> 元件中其實有兩個很重要的屬性
    - method 屬性代表表單送出後用什麼方法送出
    - action 屬性則是送出後送到哪裡
  - 跟 PHP 有相關的屬性值：
    - method 有：可以是 GET or POST
    - action 中填的則是接受使用者填完表單內容的 php 檔案網址
    - 另外還有一個重要的元件！即表單必備的送出按鈕：`<input type="submit">` (提交)

# 送出表單

- 送出表單

- 當使用者填完表單內容，並按下 submit 的送出鈕時，使用者的瀏覽器會將表單裡的內容包裝成一個包裹，丟給 action 裡填寫的網頁
- 如果 action 裡填寫的是一支 PHP 的網頁，則這支 PHP 的網頁會把這個包裹解開，並把表單裡的資訊，用陣列的方式存起來，讓程式設計師方便使用
  - 若 method="POST" 即得到 `$_POST` 陣列
- 請記住開頭是 `$_`

# POST 範例 (1)

## page.php

```
1 <html>
2  <head>
3      <meta http-equiv="content-type" content="text/html; charset=UTF-8">
4      <title></title>
5  </head>
6  <body>
7      <form method="POST" action="post.php">
8          姓名 : <input type="text" name="name"><br>
9          性別 :
10         <input type="radio" name="gender" value="Male"> 男
11         <input type="radio" name="gender" value="Female"> 女
12         <br>
13         年齡 : <input type="text" name="age"><br>
14         <input type="submit" value="送出">
15     </form>
16  </body>
17 </html>
```

## post.php

```
1 <html>
2  <head>
3      <meta http-equiv="content-type" content="text/html; charset=UTF-8">
4      <title></title>
5  </head>
6  <body>
7      你的名字是 : <?php echo $_POST['name'] ?><br>
8      你的性別是 : <?php echo $_POST['gender'] ?><br>
9      你的年齡是 : <?php echo $_POST['age'] ?>歲
10    </body>
11 </html>
```

# POST 範例 (2)

page.php

姓名 : Allan  
性別 :  男  女  
年齡 : 18

post.php

你的名字是 : Allan  
你的性別是 : Male  
你的年齡是 : 18 歲

page.php

姓名 : David  
性別 :  男  女  
年齡 : 32

post.php

你的名字是 : David  
你的性別是 : Male  
你的年齡是 : 32 歲

page.php

姓名 : Evelyn  
性別 :  男  女  
年齡 : 23

post.php

你的名字是 : Evelyn  
你的性別是 : Female  
你的年齡是 : 23 歲

# HTML表單與PHP程式

```
<form name="form_name" method="POST" action="3-2-post.php">
    <p>文字欄位 : <input type="text" name="var_text" size="20" maxlength="20"></p>
    <p>文字區域 : <textarea name="var_textarea" cols="20" rows="5"></textarea></p>
    <p>核取方塊 : <input type="checkbox" name="var_check" value="核取方塊"></p>
    <p>選項按鈕 : <input type="radio" name="var_radio" value="選項按鈕"></p>
    <p>下拉式選單 : <select name="var_select"><option>選單內容</option></select>
    <p>確認按鈕 : <input type="submit" name="var_sub" value="送出"></p>
</form>
```

文字欄位 :

文字區域 :

核取方塊 :

選項按鈕 :

下拉式選單 :

確認按鈕 :

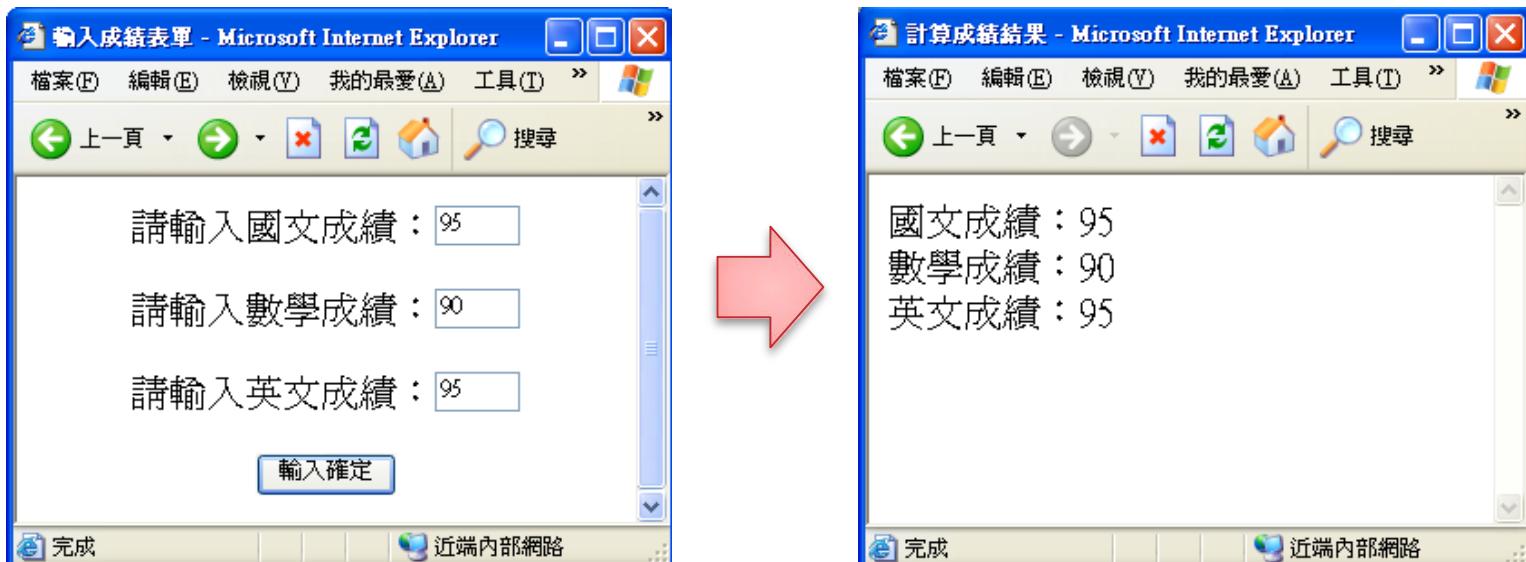
# PHP得到的值

```
<?php  
    echo "文字欄位：" . $_POST['var_text'] . "<br>";  
    echo "文字區域：" . $_POST['var_textarea'] . "<br>";  
    echo "核取方塊：" . $_POST['var_check'] . "<br>";  
    echo "選項按鈕：" . $_POST['var_radio'] . "<br>";  
    echo "下拉式選單：" . $_POST['var_select'] . "<br>";  
    echo "確認按鈕：" . $_POST['var_sub'] . "<br>";  
?  
>
```

文字欄位：我是文字欄位  
文字區域：我是文字區域  
核取方塊：核取方塊  
選項按鈕：選項按鈕  
下拉式選單：選單內容  
確認按鈕：送出

# 課堂練習 (3-2)

- 提供三個文字方塊給使用者輸入：國文、英文、數學的成績
- 按下確認鈕後，在另外一個網頁將三科成績一行一行顯示，POST / GET 不拘





## 3-5 運算子

# PHP中的運算子

- 所謂的運算子(operator)就是像「四則運算」裡的 $+$ (加)、 $-$ (減)、 $*$ (乘)、 $/$ (除)
- 運算子可處理「數字」或「字串」、「變數」或「常數」，運用不同的運算子進行計算、分析，往往可省去許多時間

# PHP的算術運算

- 假設  $\$x = 10$ ;

運算子	說明	範例	結果
+	加法	$\$x+5$	15
-	減法	$\$x-5$	5
*	乘法	$\$x*5$	50
/	除法	$\$x/5$	2
%	取餘數	$\$x \% 3$	1

# 連結字串

- 字串連結運算子主要的功能在連結兩個或兩個以上的字串，常配合echo()輸出想要的字串
- 假設 $\$x='abc'$ ， $\$y='def'$

運算子	說明	範例	結果
.	字串連結運算子	$\$z=\$x.\$y$	$\$z='abcdef'$

# 算術運算範例

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4   <title>成績計算</title>
5 </head>
6 <body>
7   <div align="center">
8     <?php
9       $chinese = 90;
10      $english = 75;
11      $math = 69;
12      $sum = $chinese + $english + $math;
13      $avg = $sum / 3;
14      $perfect = 100 * 3;
15      $remainder = $perfect - $sum;
16
17      echo '國文、英文、數學成績分別為：' . $chinese . '分、' . $english . '分、' . $math . '分';
18      echo '<br>';
19      echo '你的總分為：' . $sum . '分、總平均為：' . $avg . '分、滿分' . $perfect . '分，還差' . $remainder . '分！';
20    ?>
21   </div>
22 </body>
23 </html>
```

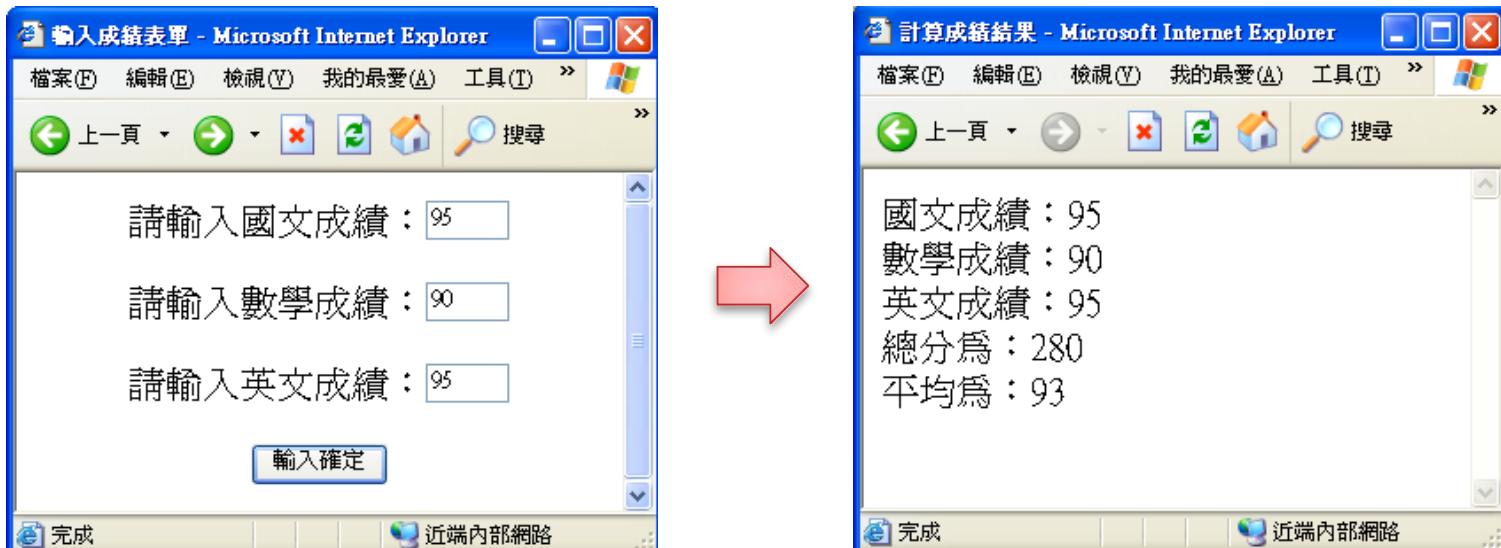
國文、英文、數學成績分別為：90分、75分、69分  
你的總分為：234分、總平均為：78分、滿分300分，還差66分！

# 練習一下

- 利用常數定義圓週率pi=3.14，計算半徑r=10的面積
- 圓面積的公式：pi \* r \* r
- <?php  
define('pi', 3.14);  
\$r = 10;  
echo pi \* \$r \* \$r;  
?>

# 課堂練習 (3-3)

- 請撰寫兩個網頁：
  - input.htm：供使用者輸入國文(chinese)、數學(math)、英文(english)成績
  - output.php：列出各科成績並計算總分及平均



# 遞增/遞減運算子

- `++` 遞增運算子 (每次遞增整數 1)
  - `$price = 100;`
  - `$price++;`
  - `echo $price; // 會印出 101`
- `--` 遞減運算子 (每次遞減整數 1)
  - `$price = 100;`
  - `$price--;`
  - `echo $price; // 會印出 99`

# 先加後加 / 先減後減

- 先加：

- \$price = 100;
- echo **++\$price**; // 會印出 101
- echo \$price; // 會印出 101

- 後加：

- \$price = 100;
- echo **\$price++**; // 會印出 100
- echo \$price; // 會印出 101

- 先減 / 後減因同理就不再贅述

# 指定運算子

- 所謂的指定(assign)變數，就是給定變數一個值，或是改變變數的值
- PHP內建的「指定運算子」如右表，假設變數 \$x 的初始值為「10」：

運算子	說明	範例	結果
=	等於	\$x=1	1
+=	加法	\$x+=1	11
-=	減法	\$x-=1	9
*=	乘法	\$x*=2	20
/=	除法	\$x/=2	5
%=	餘數	\$x%=3	1
.=	字串	\$x.="abc"	10abc

# 練習一下

- 假設一本書的成本為400元，加兩成後當作定價，最後出售時以定價打八折，請問最後的售價是多少元？
- 指定一個變數為400，利用指定運算子計算出最後的售價
- <?php  
\$price = 400;  
\$price \*= 1.2;  
\$price \*= 0.8;  
echo \$price;  
?>

# 比較運算子

- PHP可以比較兩個變數的關係，若關係成立，會傳回true，若關係不成立，會傳回false
- 右表為以\$x=5的例子，其中「全等」運算子還會檢查資料型態是否相等

運算子	說明	範例	結果
<code>==</code>	相等	<code>\$x==5</code>	true
<code>====</code>	全等	<code>\$x====5</code>	true
<code>!=</code>	不等	<code>\$x!=10</code>	true
<code>&gt;</code>	大於	<code>\$x&gt;2</code>	true
<code>&lt;</code>	小於	<code>\$x&lt;10</code>	true
<code>&gt;=</code>	大於或等於	<code>\$x&gt;=5</code>	true
<code>&lt;=</code>	小於或等於	<code>\$x&lt;=5</code>	true

# 邏輯運算

- 在程式設計中，最常在程式判斷「是」、「否」或「對」、「錯」，而掌管這些對錯真偽的運算子，稱之為「邏輯運算子」，以 $\$x=5$ ,  $\$y=10$ ,  $\$z=true$ 為例

運算子	說明	範例	結果
$\&\&$	and(並且)	$\$x==5 \&\& \$y==10$	true
$\ $	or(或)	$\$x==5 \  \$y==5$	true
!	not(不是)	$!\$z$	false

# 邏輯運算的範例

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4   <title>成績計算2</title>
5 </head>
6 <body>
7   <div align="center">
8     <?php
9     $chinese = 70;
10    $english = 60;
11    $math = 50;
12
13    echo '國文或英文是否有一科及格?';
14    echo ($chinese >= 60 || $english >= 60);
15    echo '<br>';
16    echo '英文跟數學是否都及格?';
17    echo ($english >= 60 && $math >= 60);
18    echo '<br>';
19    echo '數學是否不及格?';
20    echo !( $math >= 60 );
21
22  ?>
23 </div>
24 </body>
25 </html>
```

國文或英文是否有一科及格?  
英文跟數學是否都及格?  
數學是否不及格?

PHP中：  
true 等同於 1  
false 等於 {空}



## 3-6 流程控制

# PHP的程式流程控制

- 何謂「流程控制」？
  - 如果合乎條件就執行
  - 若條件判斷成立的話，就執行相關的程式碼，若不成立則執行警告訊息或做不同的處理
- 流程控制的種類
  - 邏輯判斷
  - 迴圈

# 邏輯判斷 – if

- 語法：

if (條件)

{

    條件成立時執行的敘述；

}

- 範例：

if (\$x == 10)

{

    echo "變數x的值是10喔！";

}

# 邏輯判斷 – if / else

- 語法：

if (條件)

{

    條件成立時執行的敘述;

}

else

{

    條件不成立時執行的敘述;

}

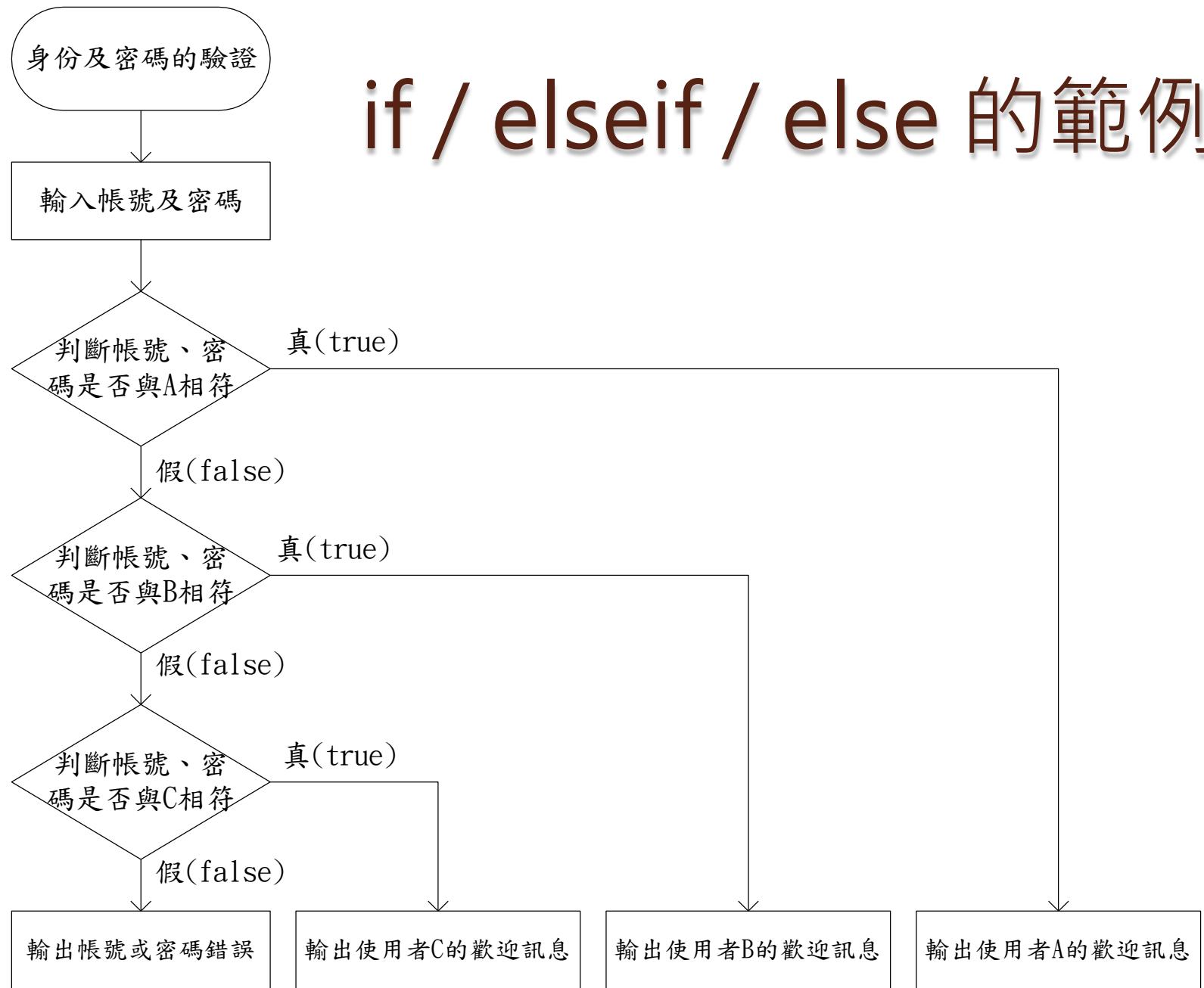
# 邏輯判斷 – if / elseif / else

- 語法：

```
if(條件1)
{
    條件1成立時執行的敘述;
}
elseif(條件2)
{
    條件2成立時執行的敘述;
}

...
elseif(條件n)
{
    條件n成立時執行的敘述;
}
else
{
    上述條件均不成立時執行的敘述;
}
```

# if / elseif / else 的範例



# 寫成程式看看

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4   <title>邏輯判斷</title>
5 </head>
6 <body>
7   <div align="center">
8     <?php
9       $userA = 'tom';
10      $userB = 'john';
11      $pwdA = '1234';
12      $pwdB = 'hello';
13      $login_name = 'john';
14      $login_pwd = '1111';
15
16      if($login_name == $userA && $login_pwd == $pwdA)
17      {
18        echo '嗨！'. $login_name . '，歡迎再次蒞臨本站~';
19      }
20      elseif($login_name == $userB && $login_pwd == $pwdB)
21      {
22        echo '嗨！'. $login_name . '，歡迎再次蒞臨本站~';
23      }
24      else
25      {
26        echo '很抱歉喔，你輸入的帳號或密碼可能有誤，請檢查！';
27      }
28    ?>
29   </div>
30 </body>
31 </html>
```

很抱歉喔，你輸入的帳號或密碼可能有誤，請檢查！

# 課堂練習 (3-4)

- 設計簡單的登入表單：可以讓使用者自行輸入帳號和密碼欄位
- 有2組正確的帳號/密碼 (記在變數裡)：
  - allan / 1234
  - bill / 5678
- 若使用者在表單內填的帳號密碼為正確的就印出「登入成功」的訊息
- 反之則印出「登入失敗」的訊息

# 邏輯控制 – switch

- 語法：

```
switch (條件變數)  
{
```

case 變數1:

    執行敘述;  
    break;

case 變數2:

    執行敘述;  
    break;

.....

case 變數n:

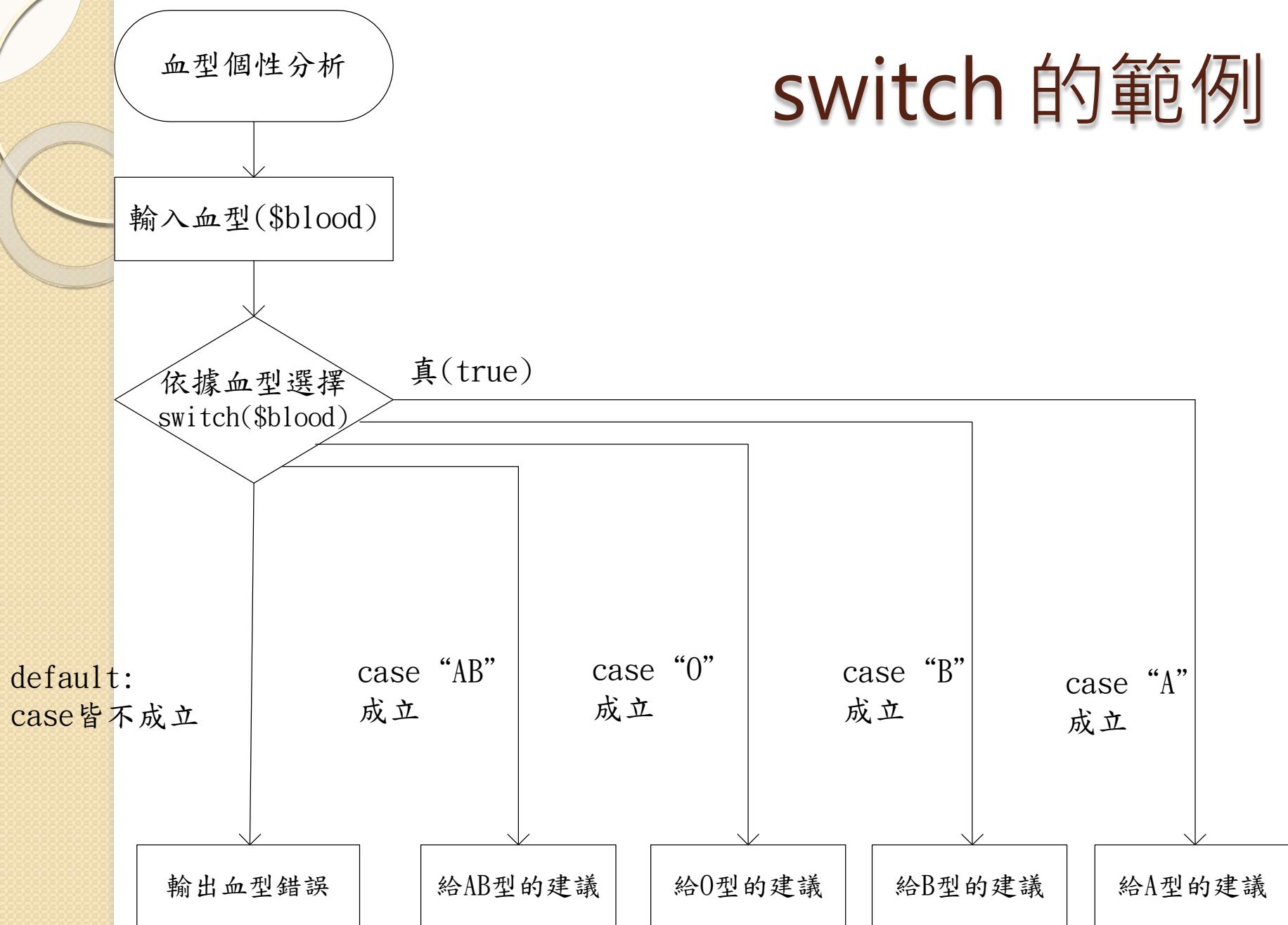
    執行敘述;  
    break;

default:

    所有條件變數均不符合時執行的敘述;

```
}
```

# switch 的範例



# 寫成程式看看

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4   <title>switch 範例</title>
5 </head>
6 <body>
7   <div align="center">
8     <?php
9     $blood = 'AB';
10    echo '你的血型是：' . $blood . ' 型<br>';
11    switch ($blood)
12    {
13      case 'A':
14        echo '絕頂聰明但不喜改變的你，<br>';
15        echo '偶爾變化一下，生活將更多采多姿。';
16        break;
17      case 'B':
18        echo '在擇善固執、勇於改變的外表下，<br>';
19        echo '加點寬容與圓滑您會活得更快樂。';
20        break;
21      case 'O':
22        echo '直爽而充滿魅力的你，<br>';
23        echo '要小心不要得罪小人喔。';
24        break;
25      case 'AB':
26        echo '浪漫與愛幻想是你的特性，<br>';
27        echo '但麵包也要兼顧喔。';
28        break;
29      default:
30        echo '關心自己也要關心別人！';
31    }
32  ?>
33 </div>
34 </body>
35 </html>
```

你的血型是：AB 型

浪漫與愛幻想是你的特性，  
但麵包也要兼顧喔。

# 迴圈

- 「迴圈」的使用可幫助我們重複執行相同的工作
- 在PHP提供的迴圈機制：
  - while 迴圈
  - do while 迴圈
  - for 迴圈

# 迴圈控制 – while

- 語法：

while (迴圈條件)

{

敘述本體;

}

- 若「迴圈條件」成立，則執行「敘述本體」，執行完後，再檢查「迴圈條件」，若成立，再執行「敘述本體」，依此步驟重複執行，直到「迴圈條件」不符合為止

# while 迴圈範例

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4   <title>while 範例</title>
5 </head>
6 <body>
7   <div align="center">
8     <?php
9       $sum = 0;
10      $i = 1;
11      while ($i <= 10)
12      {
13        $sum = $sum + $i;
14        $i++;
15      }
16
17      echo '1 加到 10 的總合是: ' . $sum;
18    ?>
19  </div>
20 </body>
21 </html>
```

1 加到 10 的總合是: 55

# 迴圈控制 – do while

- 語法：

do

{

敘述本體;

} while (迴圈條件);

- 與while迴圈類似，唯一的差別是，do while迴圈一定會做一次(先做了再說)，而while迴圈若一開始「迴圈條件」不成立，就不會執行

# do while 迴圈範例

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4   <title>do while 範例</title>
5 </head>
6 <body>
7   <div align="center">
8     <?php
9       $sum = 0;
10      $i = 11;
11      do
12      {
13        $sum = $sum + $i;
14        $i++;
15      } while ($i <= 10);
16
17      echo '1 加到 10 的總合是: ' . $sum . '<br>';
18      echo 'i 的值是: ' . $i;
19    ?>
20   </div>
21 </body>
22 </html>
```

1 加到 10 的總合是: 11  
i 的值是: 12

# 迴圈控制 – for

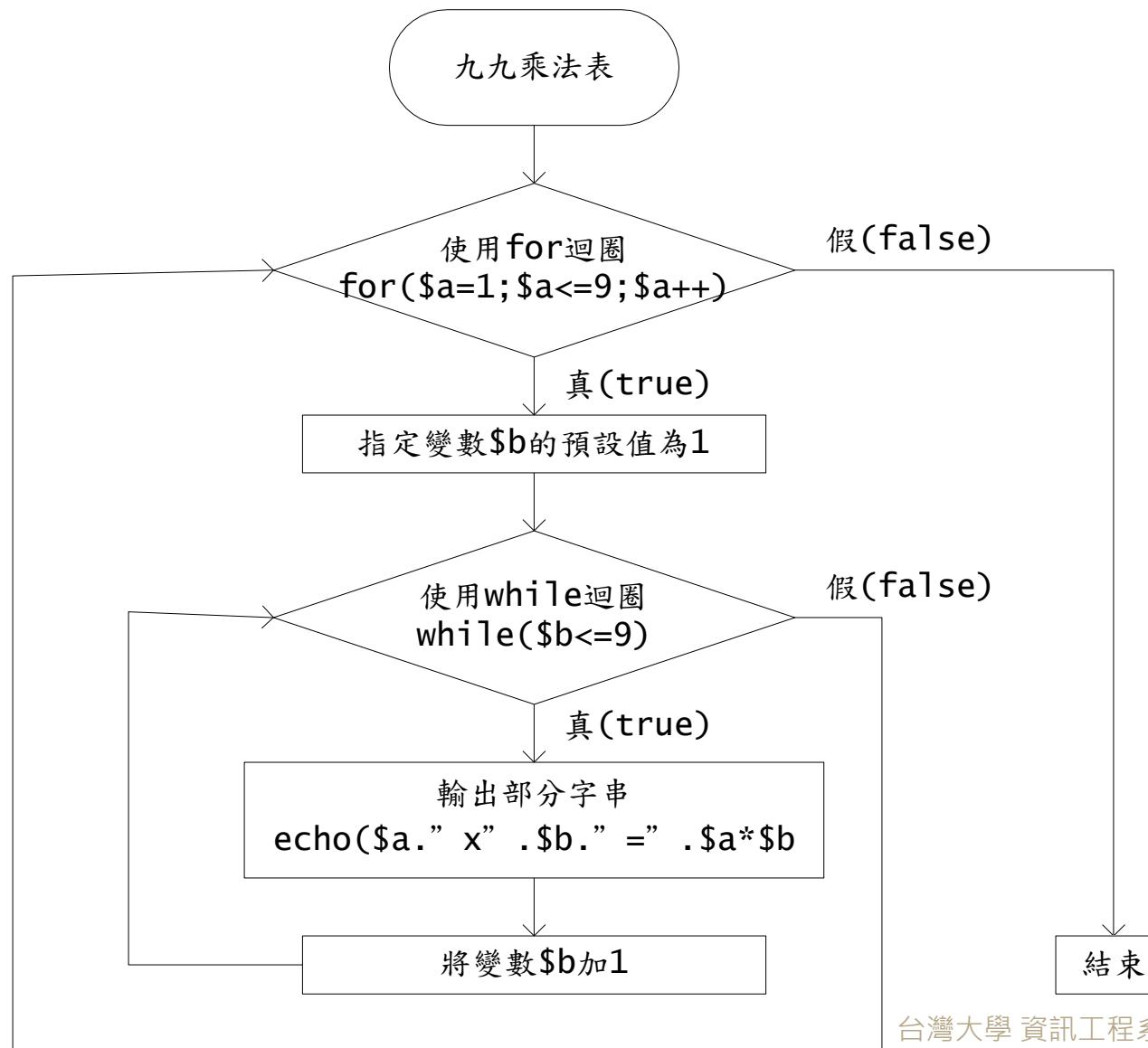
- 語法：  
`for (啟始條件; 迴圈條件; 遞增規則)  
{  
 敘述本體;  
}  
• 以「啟始條件」檢查是否符合「迴圈條件」，若符合，則執行「敘述本體」，不符合就結束迴圈  
• 執行完「敘述本體」，執行「遞增規則」，再檢查「迴圈條件」，若符合，則執行「敘述本體」，不符合就結束迴圈  
• 重複上步驟，直到不符合「迴圈條件」，結束迴圈為止`

# for 迴圈範例

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4   <title>do while 範例</title>
5 </head>
6 <body>
7   <div align="center">
8     <?php
9       $sum = 0;
10      for ($i = 1; $i <= 10; $i++)
11      {
12        $sum = $sum + $i;
13      }
14      echo '1 加到 10 的總合是: ' . $sum . '<br>';
15    ?>
16   </div>
17 </body>
18 </html>
```

1 加到 10 的總合是: 55

# 迴圈中的經典範例



# 寫成程式看看

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4   <title>99乘法表範例</title>
5 </head>
6 <body>
7   <table border="1">
8     <?php
9     for($a = 1; $a <= 9; $a++)
10    {
11      //<tr>為表格的一列
12      echo '<tr>';
13      $b = 1;
14      while ($b <= 9)
15      {
16        //<td>為儲存格
17        echo '<td align = "left">';
18        echo $a. '*' . $b. '=' . $a * $b;
19        echo '</td>';
20        ++$b;
21      }
22      echo '</tr>';
23    }
24  ?>
25  </table>
26 </body>
27 </html>
```

1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32	4*9=36
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

# 中斷指令

- **exit** 敘述：
  - 結束程式執行
- **break** 敘述：
  - 結束最接近的迴圈
- **continue** 敘述：
  - 略過本次迴圈的執行，繼續檢查是否還需要執行下一次的迴圈

# break 敘述 (1)

「break」敘述用於終止迴圈的執行，並強迫跳離迴圈區段，例：

```
7  <?php
8  $i = 1;
9  $j = 5;
10 echo 'do-while 迴圈開始執行前：i=' . $i . ' ; j=' . $j . '<br>';
11
12 do
13 {
14     if ($i > $j)
15     {
16         break;           // 離開 do-while 迴圈
17     }
18     $j--;
19     $i++;
20 } while($i < 100);
21
22 echo 'do-while 迴圈執行完之後：i=' . $i . ' ; j=' . $j;
23 ?>
```

# break 敘述 (2)

## 執行結果

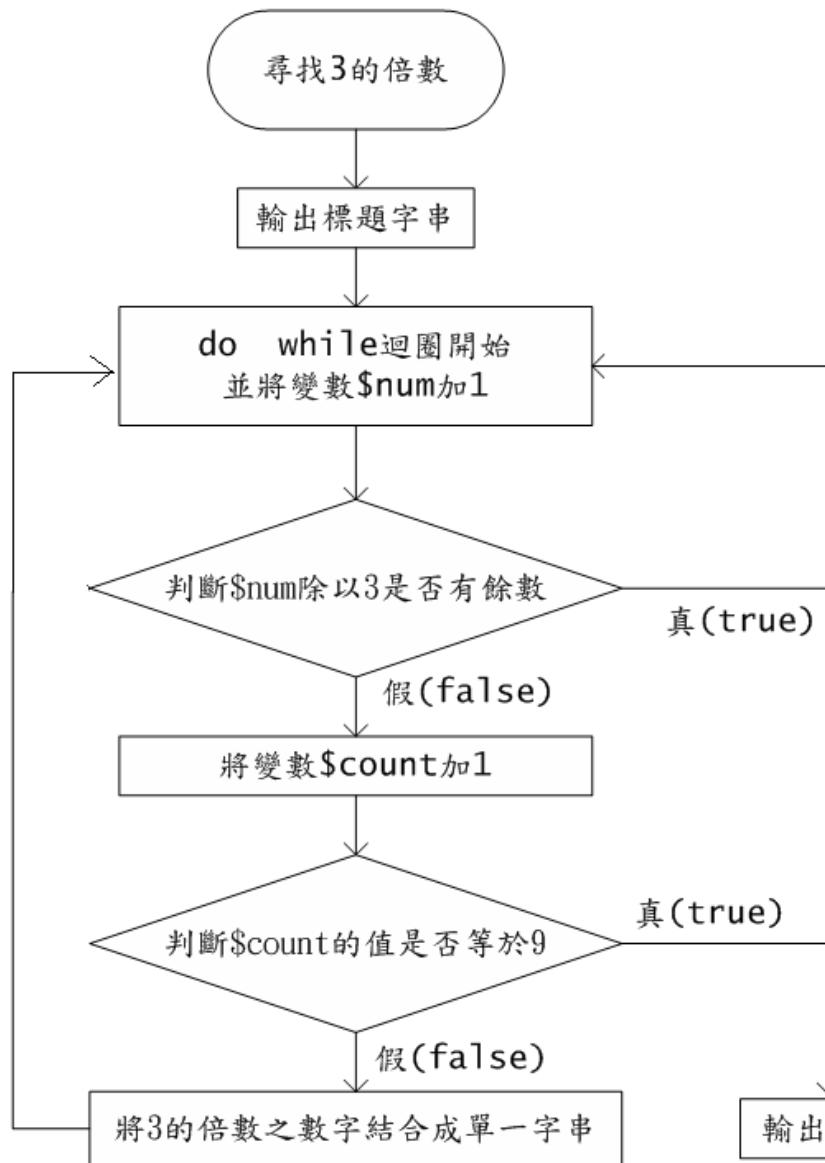
do-while 迴圈開始執行前： $i=1 ; j=5$

do-while 迴圈執行完之後： $i=4 ; j=2$

	執行if區塊條件判斷時		執行if區塊後的增值		是否可以繼續執行迴圈
do第一次執行	\$i = 1	\$j = 5	\$i = 2	\$j = 4	是
do第二次執行	\$i = 2	\$j = 4	\$i = 3	\$j = 3	是
do第三次執行	\$i = 3	\$j = 3	\$i = 4	\$j = 2	是
do第四次執行	\$i = 4	\$j = 2	---	---	由if中直接跳離迴圈

由上表中可以很明顯的看出，「do-while」迴圈在執行第四次時，因為「\$i」的值大於「\$j」的值，程式會執行「if」區塊中的內容，並直接由第16行「break」敘述跳離「do-while」迴圈區塊

# 尋找第9個3的倍數，並把前8個印出



# 寫成程式看看

```
7 <?php
8 $num = 0;      // 從 1~50
9 $count = 0;    // 第幾個3的倍數
10 echo '1至50的數字中第9個3的倍數是：<br>';
11 do
12 {
13     if (++$num % 3 == 0)
14     {
15         if (++$count == 9) // 被高亮
16         {
17             echo '我就是第9個3的倍數：' . $num . '<br>';
18             echo '結束程式！';
19             exit;
20         }
21         else
22         {
23             echo '第' . $count . '個3的倍數是：' . $num . '<br>';
24         }
25     }
26 } while ($num <= 50);
27
28 echo '此行測試當使用exit之後是否仍會顯示出來！';
29 ?>
```

## 執行結果

1至50的數字中第9個3的倍數是：  
第1個3的倍數是：3  
第2個3的倍數是：6  
第3個3的倍數是：9  
第4個3的倍數是：12  
第5個3的倍數是：15  
第6個3的倍數是：18  
第7個3的倍數是：21  
第8個3的倍數是：24  
我就是第9個3的倍數：27  
結束程式！

# 尋找 1~100 中所有4的倍數

- 利用while迴圈 + continue指令，計算1~100中，所有4的倍數和。
  - Hint: 檢查是否為4的倍數
    - 不是：直接continue掉
    - 是：將該數加到總和內

# 寫成程式看看

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4   <title>尋找 1~100 中 4 的倍數總和</title>
5 </head>
6 <body>
7 <?php
8 $total = 0;
9
10 for ($i = 1; $i <= 100; $i++)
11 {
12   if ($i % 4 != 0)
13   {
14     continue;
15   }
16
17   $total += $i;
18 }
19
20 echo '1~100 中 4 的總和為: ' . $total;
21 ?>
22 </body>
23 </html>
```

1~100 中 4 的總和為: 1300

# 陣列與迴圈的應用

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4   <title>break 範例</title>
5 </head>
6 <body>
7   <?php
8   $price[0] = 25000;
9   $price[1] = 20000;
10  $price[2] = 15000;
11  $product[0] = 'iPhone';
12  $product[1] = 'HTC Desire';
13  $product[2] = 'Samsung i9000';
14  for ($i = 0; $i < 3; $i++)
15  {
16    echo $product[$i] . '的價格：' . $price[$i] . '元<br>';
17  }
18 ?>
19 </body>
20 </html>
```

iPhone的價格：25000元  
HTC Desire的價格：20000元  
Samsung i9000的價格：15000元

# 怎麼得知陣列中有幾個元素？

- 像剛剛的例子中，是我們已經知道陣列中有3個元素，因此可以把迴圈設成跑3次就停止，但一般情況下是無法得知 陣列中有多少元素的
- 此時可以使用 **sizeof(\$array)** 或 **count(\$array)** 兩個方式取得陣列大小
  - `$array = array(0=>'A', 1=>'B', 2=>'C');`
  - `echo sizeof($array);` → 印出 3

# 改寫成 sizeof() 方式

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4   <title>break 範例</title>
5 </head>
6 <body>
7   <?php
8   $price[0] = 25000;
9   $price[1] = 20000;
10  $price[2] = 15000;
11  $product[0] = 'iPhone';
12  $product[1] = 'HTC Desire';
13  $product[2] = 'Samsung i9000';
14  for ($i = 0; $i < sizeof($price); $i++)
15  {
16    echo $product[$i] . '的價格：' . $price[$i] . '元<br>';
17  }
18 ?>
19 </body>
20 </html>
```

iPhone的價格：25000元  
HTC Desire的價格：20000元  
Samsung i9000的價格：15000元

# 若用字串當索引

- 發現了什麼事嗎？
  - 無論用 while 迴圈或是 for 迴圈，都沒辦法把用「字串」當索引的陣列正常取出
- 解決方式：
  - while 迴圈 + each() 與 list()
  - foreach 迴圈

# list + each

- list() 與 each() 都是 PHP 內建的函數
- list():
  - 將陣列的元素值指定給特定的變數
- each():
  - 依序取出陣列裡的每一個元素的索引及內容值，並傳給list()所設定的變數中
- 例如：
  - **list(\$index, \$element) = each(\$word);**  
就是將 \$word 陣列中的索引與內容值一一取出，存到 \$index 與 \$element 中

# list + each 的範例

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4   <title>list + each 範例</title>
5 </head>
6 <body>
7 <?php
8 $word['dragon'] = '一種遠古流傳的生物';
9 $word['money'] = '一種人不能沒有的東西';
10 $word['gun'] = '一種人具有強大殺傷力的致命武器';
11 $word['car'] = '一種可以搭載及具有運輸能力的機械裝置';
12 $word['dog'] = '一種哺乳類的四肢動物';
13
14 while (list($index, $key) = each($word))
15 {
16   echo $index . '<br>' . $key . '<hr>';
17 }
18
19 ?>
20 </body>
21 </html>
```

dragon  
一種遠古流傳的生物

---

money  
一種人不能沒有的東西

---

gun  
一種人具有強大殺傷力的致命武器

---

car  
一種可以搭載及具有運輸能力的機械裝置

---

dog  
一種哺乳類的四肢動物

---

# foreach

- 另一種寫法是變化型的 for 迴圈，稱做 foreach
- **foreach** 顧名思義也是將陣列中的資料用類似 **for** 迴圈的方式，一個一個 (**each**) 的取出來
- 其中有另一個關鍵字稱做 **as**
- 完整語法如：
  - **foreach** (\$陣列 **as** \$索引 => \$值)  
{  
}

# foreach 的範例

```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4   <title>foreach 範例</title>
5 </head>
6 <body>
7 <?php
8 $word['dragon'] = '一種遠古流傳的生物';
9 $word['money'] = '一種人不能沒有的東西';
10 $word['gun'] = '一種人具有強大殺傷力的致命武器';
11 $word['car'] = '一種可以搭載及具有運輸能力的機械裝置';
12 $word['dog'] = '一種哺乳類的四肢動物';
13
14 foreach ($word as $index => $key)
15 {
16   echo $index . '<br>' . $key . '<hr>';
17 }
18
19 ?>
20 </body>
21 </html>
```

dragon  
一種遠古流傳的生物

---

money  
一種人不能沒有的東西

---

gun  
一種人具有強大殺傷力的致命武器

---

car  
一種可以搭載及具有運輸能力的機械裝置

---

dog  
一種哺乳類的四肢動物

---

# 陣列的 index 搭配變數 (1)

```
1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2  └<html>
3  └<head>
4  <meta http-equiv="content-type" content="text/html; charset=UTF-8">
5  <title></title>
6  </head>
7  └<body>
8  └<form method="POST" action="query.php">
9    輸入你的姓名：
10   <input type="text" name="student">
11   <input type="submit" value="送出">
12   </form>
13   </body>
14   </html>
```



輸入你的姓名：

# 陣列的 index 搭配變數 (2)

```
1 曰<?php
2   $student = array(
3     'allan' => 'allan@csie.org',
4     'bill' => 'bill@ntu.edu.tw',
5     'cindy' => 'cindy@gmail.com'
6   );
7  ?>
8  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
9  <html>
10 <head>
11   <meta http-equiv="content-type" content="text/html; charset=UTF-8">
12   <title></title>
13 </head>
14 <body>
15   Hi, 你的 email 是：
16  <?php
17   echo $student[$_POST['student']];
18  ?>
19 </body>
20 </html>
```



# 回家作業 – 簡易水果字典

- 用HTML表單做一個文字輸入框，讓使用者可以自行輸入單字，送出後檢查使用者輸入的字是否在水果字典列表中：
  - apple: 蘋果
  - orange: 橘子
  - watermelon: 西瓜
  - strawberry: 草莓
  - pineapple: 鳳梨
  - 若使用者輸入的單字在上述列表中 → 則印出其中英對照
  - 若不存在 → 則印出「水果字典中沒有xxx這個單字」
  - Hit: 要不要用迴圈或只採用 if/else 或 switch 檢查都可以

# 範例畫面

## 簡易字典查詢系統

請輸入要查詢的英文單字 :



英文	中文
apple	蘋果

## 簡易字典查詢系統

請輸入要查詢的英文單字 :



水果字典中沒有 tomato 這個單字

# 繳交回家作業

- 把「簡易水果字典」完成後存檔成 hw3.html 與 hw3.php
- 把兩個檔案壓縮成 **姓名.zip** 或 **姓名.rar** 檔，把該檔案附件 email 中寄到：
  - 主旨：「**[PHP265][hw][3]姓名**」
  - 收件人：[m10112101@mail.ntust.edu.tw](mailto:m10112101@mail.ntust.edu.tw)
- Deadline: 下週同一時間上課前

# 繳交課堂作業

- 把課堂練習(3-1)~課堂練習(3-n) 分別儲存在：
  - pt3-1 ~ pt3-n 的資料夾中，資料夾裡存放所有你該次練習的所有檔案 (如果有圖片、html 或 php)都存在裡面
  - 把這 n 個資料夾都壓縮成 **姓名.zip** 或 **姓名.rar** 檔，把該檔案附件 email 中寄到：
    - 主旨：「**[PHP265][pt][3]姓名**」
    - 收件人：[m10112101@mail.ntust.edu.tw](mailto:m10112101@mail.ntust.edu.tw)
    - Deadline: 下週同一時間上課前