

PHP 基本語法

引用自：王勝雄，[台中市網【PHP 程式語言編寫】研習課程](http://km.tceb.edu.tw/~wsx/php/index.htm)

網址：<http://km.tceb.edu.tw/~wsx/php/index.htm>

目錄

PHP 基本語法	1
程式碼概論	1
註解	2
變數	3
常數	4
資料型態	5
運算子 (Operator)	8
函數(或稱函式)	13

程式碼概論

一、PHP 程式碼執行方式：

透過 Web Server 方式：例如利用 Apache HTTP Server 來執行 .php(或 .php3)副檔名的網頁(可參考 PHP 簡介的一個簡單介紹的範例)。

文字模式下執行程式：即在文字命令模式下透過 php.exe(for Windows)或 php(for Unix)來執行內含 PHP 指令的程式碼。(示範影片)

ps. 如果把一般網頁的副檔名(.html)改為.php時，如果其中都沒有 PHP 程式碼，那只有輸出原來的 html 內容。

二、PHP 程式碼寫法：

PHP 程式碼可單獨編寫，也可以「嵌入」HTML 格式中，但為了讓 PHP 能夠被處理，會有幾種被設定的「標籤」符號來表示其中的區塊是 PHP 程式，避免與一般 HTML 網頁標準語法混淆，例如：

```

01: <html>
02: <head>
03: <title>php 特殊定義符號</title>
04: </head>
05: <body>
06:   <?
07:     echo "方法一";
08:   ?>
09:   <?php
10:     echo "方法二";
11:   ?>
12:   <SCRIPT LANGUAGE="php">
13:     echo "方法三";
14:   </SCRIPT>
15:   <%
16:     echo "方法四";
17:   %>
18: </body>
19: </html>

```

執行結果>>

第 06~08 行，方法一：<? 程式碼 ?> 最簡單表達方式；SGML 的處理指令。

第 09~11 行，方法二：<?php 程式碼 ?> 最正統表達方式；XML 的處理指令。

第 12~14 行，方法三：<SCRIPT LANGUAGE="php"> 程式碼 </SCRIPT>，
為 HTML 的 script 表達格式。

第 15~17 行，方法四：<% 程式碼 %>，與 ASP 語法相同表達格式，但容易與 asp
混淆，且須在 php.ini 的設定檔中，設定 asp_tags 為 On 才
可使用。

註解

在編寫 php 或其他程式時，「註解」應該是必要的項目，因為註解有利於了解整個程式的功能，或是程式段落變數的定義或功用的說明，便於日後維護、修改，甚至在移轉、分享時令人容易掌握，所以，「程式註解」應該是程式設計者基本的要求。

PHP 支持 'C'，'C++' 和 Unix Shell 風格的程式註解。例如：

```

01: <?php
02:     echo "這是個測試!"; // 這是 c++ 註解的風格
03:     /* 這是多行
04:        註解的方式 */
05:     echo "這是其他的測試!";
06:     echo "最後的測試"; # This is shell-style style comment
07:   ?>

```

執行結果>>

第 02 行用的是「 // 」符號，後面是註解文字；

第 03~04 行是用「`/* */`」來註解大量說明文字；
第 06 行用的是「`#`」符號，與第 02 行的用法相同。

使用多行註解時，應注意避免「巢狀」方式使用，例如以下就會出錯：

```
01:  <?php
02:      /* 這是多行註解的錯誤示範
03:          /* echo "這是其他的測試！"; */
04:      */
05:  ?>
```

變數

基礎概念

在 php 中的變數與大多數的的程式語言變數一樣，是某些資料的容器；所以，您可以自行命名變數名稱，並以「`$`」來識別，再將資料指定給變數，提供程式後續的引用其內存的資料。

但在使用變數之前，須先注意幾件事：

- 變數名稱有大小寫之分，所以 `$abc` 與 `$ABC`，甚至是 `$Abc` 都各自表示不同的變數。
- 變數名稱的「開頭字母」必須是英文字母「a-z 或 A-Z」或底線「`_`」，不可用其他字元，如數字。
- 變數名稱必須由英文字母、數字及底線組成，中間不可空白。
- 不可使用關鍵字及識別字。

例如：

```
01: <?php
02:  $var = "小叮噹";
03:  $Var = "大雄";
04:  echo "$var, $Var"; // 會輸出為 "小叮噹, 大雄"
05:
06:  $4site = '胖虎'; // 錯誤; 因為開頭是數字
07:  $_4site = '小夫'; // 可以; 開頭是底線
08: ?>
```

上述程式第 06 行的錯誤訊息是：

```
Parse error: parse error, unexpected T_LNUMBER, expecting T_VARIABLE or '$' in
ch3-3-1.php on line 6
```

變數的初啟化(或稱設值)

就是直接將一個數值(文字或數字)直接設定給變數，然而 PHP 與一般 C 語言不同的

是變數的名稱不一定要事先宣告，如果給予字串數值，該變數就是字串變數，給予數字，就是數字變數，可做為數字運算用，所以有人稱 PHP 的變數型態是一種「鬆散」的變數型態，例如：

```
01:  <?php
02:    $a = "這是測試程式！<BR>";
03:    echo $a;
04:
05:    $a = "12345";
06:    echo ($a * 10);
07:  ?>
```

執行結果>>

第 02 行：\$a 是字串變數；
是 html 格式換行的意思。

第 05 行：\$a 變為數字變數，所以可進行第 06 行的計算。

變數的變數(或稱參照)

變數的變數就是將變數的「值」當做別一個變數的名稱。例如：

```
01:  <?php
02:    $a = "test";
03:    echo "$a <BR>";
04:
05:    $$a = "12345";
06:    echo $test;
07:  ?>
```

執行結果>>

第 02 行：\$a 變數內容是「test」；

第 03 行：輸出變數 \$a 的內容；

第 05 行：「\$\$a」就是所謂「變數的變數」，它的變數名稱就是「\$test」；

第 06 行：輸出變數 \$test 的內容是否就是 \$\$a 的內容。除了寫成 \$\$a 外，也可寫成 \${\$a} (用大括號包起來)，其意義是相同的。但請注意，其運用規則也須遵照變數名稱的命名規則。

常數

常數也是存放資料的變數，只不過是給予資料之後，就不可改變資料內容，所以常數是以另一種方式來定義。在 PHP 中，常數是用 define() 函數來建立。例如：

```
01:  <?php
02:    define ("PI", "3.1415926"); // 設定 π 的值
03:    echo PI;
04:  ?>
```

執行結果>>

第 02 行：建立常數「PI」的值是「3.1415926」(圓周率)；

第 03 行：在程式中要使用常數時，直接使用它的名稱就可以，不用「\$」字號。傳統上，常數的名稱大都以「大寫字串」來命名，但不一定強迫限制要大寫；此外，要檢查「某字串」是否被定義為常數，可運用 `defined()` 來識別，例如：

```
01:  <?php
02:      if ( defined ("PI") )
03:          echo "PI 已被設定";
04:      else
05:          echo "未設定"
06:  ?>
```

第 02 行：`defined()` 回應 `true` 或 `false`；如果要判斷常數是否存在，可使用 `isset()` 函數。

資料型態

雖然 PHP 對變數的型態具有彈性(如果給予字串數值，該變數就是字串變數，給予數字，就是數字變數，可做為數字運算用)，但還是具備兩大類型的資料型態供使用變數時加以設定：

標準類型

布林值(boolean)：這是最簡單的類型。`boolean` 表達了事件「真」「假」，可以為 `TRUE` 或 `FALSE`。

整數(integer)：一個 `integer` 是集合 $Z = \{ \dots, -2, -1, 0, 1, 2, \dots \}$ 中的一個數。

浮點數(float)：浮點數（也叫“floats”，“doubles”或“real numbers”）可以用以下任何語法定義：

```
$a = 1.234;
$a = 1.2e3;
$a = 7E-10;
```

浮點數的字長和作業平臺相關，儘管通常最大值是 `1.8e308` 並具有 14 位十進位數字的精度（64 位 IEEE 格式）。

字串(string)：在 PHP 中，字串可以用三種方法來定義：

1. 單引號：

指定一個簡單字串的最簡單的方法是用單引號（字元 `'`）括起來。

如果要表示一個單引號，則需要用另反斜線（`\`）來表示，避免 PHP 在解譯時發生錯誤，和很多其他語言一樣。

如果在單引號之前或字串結尾需要出現一個反斜線，需要用兩個反斜線表示。

例如：

```

01:  <?php
02:    echo 'this is a simple string <BR>';
03:    echo 'You can also have embedded newlines in strings,
04:    like this way. <BR>';
05:
06:    echo 'Arnold once said: "I\'ll be back" <BR>';
07:    // 輸出: ... "I'll be back"
08:
09:    echo 'Are you sure you want to delete C:\\*. *? <BR>';
10:    // 輸出: ... delete C:\\*. *?
11:
12:    echo 'Are you sure you want to delete C:\\*. *? <BR>';
13:    // 輸出: ... delete C:\\*. *?
14:
15:    echo 'I am trying to include at this point: \n a newline <BR>';
16:    // 輸出: ... this point: \n a newline
17:  ?>

```

2. 雙引號：以雙引號圍住的字串，PHP 會對該字串做「變數置入」的動作。

例如：

```

01:  <?php
02:    $name = "大雄";
03:    echo '姓名: $name <BR>';
04:    echo "姓名: $name <BR>";
05:  ?>

```

執行結果>>

```

姓名: $name
姓名: 大雄

```

雙引號字串中有一些符號得用「跳脫字元」才可以正常顯示：

跳脫字元	用途
\n	換行
\r	游標回頭
\t	Tab 鍵
\\	「\」符號
\\$	「\$」符號
\"	「"」符號
\[0-7]{1,3}	八進位數值
\x[0-9A-Fa-f]{1,2}	十六進位數值

複合類型

陣列(array)：將變數做為數組來使用，基本寫法如下，

\$ 變數名稱[索引值]= 值；

例如：

```
01:  <?php
02:  $abc[] = "第一段 <BR>";
03:  $abc[] = "第二種 <BR>";
04:  $abc[] = "第三個 <BR>";
05:
06:  echo $abc[0] ;
07:  echo $abc[1] ;
08:  echo $abc[2] ;
09:  ?>
```

執行結果>>

如果沒有指定索引值，是以 0 開始紀錄

物件(object)：要初始化一個物件，用 new 語句將物件實例傳到一個變數中。

由於物件的學習屬 PHP 進階內容，僅舉例如下：

```
01:  <?php
02:  class foo
03:  {
04:      function do_foo()
05:      {
06:          echo "使用物件！";
07:      }
08:  }
09:  $bar = new foo;
10:  $bar->do_foo();
11:  ?>
```

執行 結果>>

第 02~08 行：定義個 class 類別；

第 09 行：產生個新 class 物件變數；

第 10 行：輸出變數內容。

特殊類型

空值(null)：特殊的 NULL 值表示一個變數沒有值。NULL 類型唯一可能的值就是 NULL。

在下列情況下，變數的預設值皆為 NULL；變數也可以利用 unset() 函數把變數內容「清空」；當然下列方法也可以把變數內容「清空」。

```

01:  <?php
02:      $var = NULL;
03:  ?>

```

備註：

1. 變數的類型通常不是由程式師設定的，確切地說，是由 PHP 根據該變數使用的上下文在運行時決定的。
2. 如果想查看某個運算式的值和類型，用 `var_dump()` 函數。
3. 要查看某個類型，不要用 `gettype()`，而用 `is_type` 函數。
4. 如果要將一個變數強制轉換為某類型，可以對其使用強制轉換或者 `settype()` 函數。
5. 注意變數根據其當時的類型在特定場合下會表現出不同的風格。

運算子 (Operator)

什麼是運算子(有人翻譯為「運算元」或「運算符號」)，以下例來解釋：

```
$total = $a + $b;
```

`$a + $b` 稱為「表示式」(expressions)

`+` 即稱為運算子(operator)

在 PHP 的運算子裡，大致歸類下列十種：

1. 算術運算子
2. 指定運算子
3. 位元運算子
4. 比較運算子
5. 錯誤控制運算子
6. 執行運算子
7. 加一／減一運算子
8. 邏輯運算子
9. 字串運算子
10. 陣列運算子

1. 算術運算子

其實與一般的數學沒甚麼兩樣：

運算子	範例	用途
+	<code>\$a + \$b</code>	<code>\$a</code> 和 <code>\$b</code> 的和

-	$\$a - \b	$\$a$ 和 $\$b$ 的差
*	$\$a * \b	$\$a$ 和 $\$b$ 的乘積
/	$\$a / \b	$\$a$ 除以 $\$b$ 的商
%	$\$a \% \b	$\$a$ 除以 $\$b$ 的餘數

2. 指定運算子

基本的指定運算符就是“=”。並不稱做“等於”，實際上意味著把右邊運算式的值指定給左運算數。例如：

```
 $\$a = \$a + 2;$ 
```

正確的解釋是將「右邊的 $\$a + 2$ 的值」指定給「左邊的 $\$a$ 」。所以在程式技巧中，可以把表示式簡寫為下列方式：

標準式	簡單式
$\$a = \$a + \$b$	$\$a += \b
$\$a = \$a - \$b$	$\$a -= \b
$\$a = \$a * \$b$	$\$a *= \b
$\$a = \$a / \$b$	$\$a /= \b
$\$a = \$a \% \$b$	$\$a \% = \b

其他的運算子也可以以類似的方式簡單表達，而達成運算目的。

3. 位元運算子

將整數視為二進位的字串來加以運算；因為有點難，大致了解即可：

範例	名稱	結果
$\$a \& \b	And (且)	將在 $\$a$ 和 $\$b$ 中都為 1 的位元設為 1
$\$a \b	Or (或)	將在 $\$a$ 或者 $\$b$ 中為 1 的位元設為 1
$\$a \wedge \b	Xor (互斥)	將在 $\$a$ 和 $\$b$ 中不同的位元設為 1
$\sim \$a$	Not (補數)	將 $\$a$ 中為 0 的位元設為 1，反之亦然
$\$a \ll \b	Shift left (左移)	將 $\$a$ 中的位元向左移動 $\$b$ 次
$\$a \gg \b	Shift right (右移)	將 $\$a$ 中的位元向右移動 $\$b$ 次

4. 比較運算子

範例	名稱	解釋
<code>\$a == \$b</code>	判斷等於	如果 <code>\$a</code> 等於 <code>\$b</code> ，則結果為真
<code>\$a === \$b</code>	全等	如果 <code>\$a</code> 等於 <code>\$b</code> ，並且它們的類型也相同，則結果為真
<code>\$a != \$b</code>	不等	如果 <code>\$a</code> 不等於 <code>\$b</code> ，則結果為真
<code>\$a <> \$b</code>	不等	如果 <code>\$a</code> 不等於 <code>\$b</code> ，則結果為真
<code>\$a !== \$b</code>	非全等	如果 <code>\$a</code> 不等於 <code>\$b</code> ，或者它們的類型不同，則結果為真
<code>\$a < \$b</code>	小於	如果 <code>\$a</code> 小於 <code>\$b</code> ，則結果為真
<code>\$a > \$b</code>	大於	如果 <code>\$a</code> 大於 <code>\$b</code> ，則結果為真
<code>\$a <= \$b</code>	小於等於	如果 <code>\$a</code> 小於或者等於 <code>\$b</code> ，則結果為真
<code>\$a >= \$b</code>	大於等於	如果 <code>\$a</code> 大於或者等於，則結果為真

另外一個條件運算符是“?:”（或三元）運算符，其語法如下：

```
(expr1) ? (expr2) : (expr3);
```

如果 `expr1` 的值為 `TRUE`，則此運算式的值為 `expr2`，如果 `expr1` 的值為 `FALSE`，則此運算式的值為 `expr3`。

5. 錯誤控制運算子

PHP 支援一個錯誤控制運算符「@」。當將其放置在一個 PHP 運算式之前，該運算式可能產生的任何錯誤資訊都被忽略掉。這功能有幾個目的：

安全性：避免因程式上的某些錯誤訊息將一些訊息告知了外界，而暴露系統上可能的安全漏洞。

美觀：因為錯誤訊息會造成顯示畫面的混亂。

基本上，這個功能得在安裝 PHP 時啟動 `track_errors` 功能(也可事後在 `php.ini` 中修改)，運算式所產生的任何錯誤資訊才會被存放在總體變數 `$php_errormsg` 裡，並顯示出來。

```
01: <?php
02:     $abc = @mysql_connect ('root', 'test', 'database');
03: ?>
```

6. 執行運算子

PHP 支持一個執行運算符：反引號「`...`」。注意這不是單引號！PHP 將嘗試將反引號中的內容作為 shell 命令來執行，並將其輸出資訊返回（例如，可以賦給一個變數而不是簡單地丟棄到標準輸出）。

```
01: <?php
02:     $output = `ls -al`;
03:     echo "<pre> $output </pre>";
04: ?>
```

這只能在 command 文字命令模式下執行，不可透過 Web Server 執行。

注：反引號運算子在啟動了 safe mode 或者關閉了 shell_exec() 時是無效的。

7. 加一／減一運算子

PHP 支持 C 風格的前／後加一與減一運算子。

範例	名稱	解釋
++\$a	前加	\$a 的值加一，然後傳回\$a 的值
\$a++	後加	傳回\$a 的值，然後將\$a 的值加一
--\$a	前減	\$a 的值減一，然後傳回\$a 的值
\$a--	後減	傳回\$a 的值，然後將\$a 的值減一

例子：

```
01: <?php
02:     echo "<h3>a++試驗</h3>";
03:     $a = 5;
04:     echo "應該是 5: ".$a++."<br />\n";
05:     echo "應該是 6: ".$a."<br />\n";
06:
07:     echo "<h3>++a 試驗</h3>";
08:     $a = 5;
09:     echo "應該是 6: ".$a."<br />\n";
10:     echo "應該是 6: ".$a."<br />\n";
11:
12:     echo "<h3>a--試驗</h3>";
13:     $a = 5;
14:     echo "應該是 5: ".$a--."<br />\n";
15:     echo "應該是 4: ".$a."<br />\n";
16:
17:     echo "<h3>--a 試驗</h3>";
18:     $a = 5;
19:     echo "應該是 4: ".$a."<br />\n";
20:     echo "應該是 4: ".$a."<br />\n";
21:
22: ?>
```

8. 邏輯運算子

邏輯運算子主要用來測試條件句的真(true)或假(false)，以判斷程式的某個段落是否繼續執行或跳脫。

範例	名稱	解釋
<code>\$a and \$b</code>	And (邏輯與)	TRUE, 如果 \$a 與 \$b 都為 TRUE
<code>\$a or \$b</code>	Or (邏輯或)	TRUE, 如果 \$a 或 \$b 任一為 TRUE
<code>\$a xor \$b</code>	Xor (邏輯互斥)	TRUE, 如果 \$a 或 \$b 任一為 TRUE, 但不同時是
<code>! \$a</code>	Not (邏輯非)	TRUE, 如果 \$a 不為 TRUE
<code>\$a && \$b</code>	And (邏輯與)	TRUE, 如果 \$a 與 \$b 都為 TRUE
<code>\$a \$b</code>	Or (邏輯或)	TRUE, 如果 \$a 或 \$b 任一為 TRUE

9. 字串運算子

有兩個字串運算符。第一個是連接運算符 (“.”), 它傳回其左右參數連接後的字串。第二個是連接指定運算子 (“.=”), 它將右邊參數附加到左邊的參數後。

```
01: <?php
02:     $a = "花好月圓";
03:     $b = $a . "萬事如意";
04:     echo $b . "<BR>";
05:
06:     $c = "春風得意";
07:     $c .= "國泰民安";
08:     echo $c;
09: ?>
```

執行結果 >>

```
花好月圓萬事如意
春風得意國泰民安
```

10. 陣列運算子

PHP 僅有的一個陣列運算子是 「 + 」 運算子。它把右邊的陣列附加到左邊的陣列後，但是重複的索引值不會被覆蓋。

```
01: <?php
02:     $a = array("1" => "一月", "2" => "二月");
03:     $b = array("1" => "星期一", "2" => "星期二", "3" => "星期三");
04:     $c = $a + $b;
05:
06:     var_dump($c); // var_dump( ) 是顯示變數內容的函數
07: ?>
```

執行結果 >>

第 06 行的結果將會是：

```
array(3) { [1]=> string(4) "一月" [2]=> string(4) "二月" [3]=> string(6) "星期三" }
```

上式表示 \$c 變數是 array (陣列)，裡面的元素中，索引值 1 是字串，字串長度是 4，內容是"一月"，以下以此類推。

運算子優先順序

運算子優先順序指定了兩個運算式綁定得有多“緊密”。例如，運算式 $1 + 5 * 3$ 的結果是 16 而不是 18，原因是乘號（“*”）的優先順序比加號（“+”）高。若要強制改變優先順序，可以使用括弧來完成。例如： $(1 + 5) * 3$ 的值為 18。

函數(或稱函式)

在 PHP 中，允許程式設計者將常用的流程或者變數等元件，組織成一個固定的格式。也就是說使用者可以自行組合函式或者是物件。

一個函數可由以下的語法來定義：

```
function foo ($arg_1, $arg_2, ..., $arg_n)
{
    echo "Example function.\n";    // 程式動作
    .
    .
    .
    return $retval;                // 傳回結果
}
```

\$arg_1, \$arg_2, ..., \$arg_n 為函數所使用的參數，參數間用「，」分隔。舉例來說：

```
01:  <?php
02:    // 圓周長=直徑 x 3.14
03:    function circle_len($r)
04:    {
05:        $result = $r * 2 * 3.14;
06:        return ($result);
07:    }
08:
09:    $r = 10; // 半徑
10:    echo "圓周長=".circle_len($r);
11:  ?>
```

執行結果 >>

圓周長=62.8

第 03~07 行：自訂了一個函數，名稱叫做「circle_len」

第 10 行：使用自訂的函數傳回結果。