

# 程式流程控制結構

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>

## 條件述句

---

1. if
2. if...else
3. if...elseif.....else
4. switch
5. 流程控制的替代語法

### 1. if

語法：

```
if( 條件 )
{
    敘述句;
}
```

範例一：

```
01:  <?php
02:     $a = "123";
03:     if ( $a == 123 )
04:         echo "A = $a ";
05:  ?>
```

範例二：

```
01:  <?php
02:     $a = "123";
03:     if ( $a == 123 )
04:     {
05:         echo "A = $a ";
06:         echo "<br>";
07:         echo "A=".$a;
08:     }
09:  ?>
```

### 2. if .... else

語法：

```

if( 條件 )
{
    敘述句一;
}
else
{
    敘述句二;
}

```

範例：

```

01:  <?php
02:     $a = "123";
03:     if ( $a < 100 )
04:         echo "A 小於 100 ";
05:     else
06:         echo "A 大於 100 ";
07:  ?>

```

### 3. if .... elseif .... else

語法：

```

if( 條件一 )
{
    敘述句一;
}
elseif (條件二)
{
    敘述句二;
}
else
{
    敘述句三;
}

```

範例：

```

01:  <?php
02:     $a = "123";
03:     if ( $a < 100 )
04:         echo "A 小於 100 ";
05:     elseif ( $a >= 100 AND $a < 200)
06:         echo "A 大於 100 ，小於 200";
07:     else
08:         echo "A 大於 200";
09:  ?>

```

#### 4. switch

switch 語法和具有同樣運算式的一系列的 if 語法相似。很多場合下需要把同一個變數（或運算式）與很多不同的值比較，並根據它等於哪個值來執行不同的代碼，這才是 switch 語句的用途。

語法：

```
switch ( switch 條件 )
{
    case '值 1':
        敘述 1;
        break;
    case '值 2':
        敘述 2;
        break;
    .....
    default :
        敘述 n;
        break;
}
```

範例：

用 if 的寫法	用 switch 的寫法
<pre>&lt;?php \$a = \$_REQUEST['no']; if ( \$a == 1 )     echo "冠軍";  if ( \$a == 2 )     echo "亞軍";  if ( \$a == 3 )     echo "季軍"; ?&gt;</pre>	<pre>&lt;?php \$a = \$_REQUEST['no']; switch (\$a) {     case '1':         echo "冠軍";         break;     case '2':         echo "亞軍";         break;     case '3':         echo "季軍";         break; } ?&gt;</pre>

為避免錯誤，理解 switch 是怎樣執行的非常重要。switch 語句一行接一行地執行（實際上是語句接語句）。開始時沒有代碼被執行。僅當一個 case 語句中的值和 switch 運算式的值匹配時 PHP 才開始執行語句，直到 switch 的程式段結束或者遇到第一個 break 語句為止。如果不在 case 的語句段最後寫上 break 的話，PHP 將繼續執行下一個 case 中的語句段。

舉例來說：

```
01: <?php
02:     $a = 2;
03:
04:     switch ($a)
05:     {
06:         case '1':
07:             echo "冠軍";
08:         case '2':
09:             echo "亞軍";
10:         case '3':
11:             echo "季軍";
12:     }
13:
14: ?>
```

執行結果：

亞軍季軍

上述程式的第 04 行，依 \$a 的資料是 2，應該執行第 08 行，但因為沒有 break 指令來中斷，所以會循序執行第 09、10 行，執行結果就會錯誤。

## 迴圈

---

1. while
2. do...while
3. for
4. break
5. continue

### 1. while

while 迴圈是 PHP 中最簡單的迴圈類型。它和 C 語言中的 while 表現得一樣。

語法：

```
while ( 條件 )
{
    敘述;
}
```

只要條件一直成立(真或 true)時，就會一直重覆執行敘述的程式。

範例：

```
01: <?php
02:   $a = 1;
03:   while ( $a<10 )
04:   {
05:       echo "$a ";
06:       $a++;
07:   }
08:   ?>
```

執行結果>>

1 2 3 4 5 6 7 8 9
-------------------

說明：

第 02 行：預設 \$a 變數為 1；

第 03 行：只要 \$a 小於 10 的條件符合，就會執行第 04、05 行；

第 05 行：是讓 \$a 做累加 1 的動作，好讓最後 \$a = 10 的時候，不符合第 03 行的條件，而停止 while 迴圈。

## 2. do .... while

語法：

```
do
{
    敘述;
} while ( 條件 )
```

範例：

```
01: <?php
02:   $a = 1;
03:   do {
04:       echo "$a <BR>";
05:       $a++;
06:   } while ( $a<6 )
07:   ?>
```

執行結果>>

1
2
3
4
5

說明：

第 02 行：預設 \$a 變數為 1；

第 03 行：先執行第 04、05 行；

第 05 行：是讓 \$a 做累加 1 的動作；

第 06 行：判別 \$a 是否符合  $a < 10$  的條件，若符合，則再執行第 04、05 行的程式，否則停止迴圈

while 與 do .... while 的差別是不管條件是否符合「do .... while」至少先執行一次。

### 3. for

for 迴圈是 PHP 中最複雜的迴圈結構。它的行為和 C 語言的相似。

語法：

```
for ( 條件初始值 ; 條件判斷式 ; 條件改變式 )
{
    敘述;
}
```

範例：

```
01:  <?php
02:     for ( $a = 1; $a < 10 ; $a++ )
03:         echo "$a ";
04:  ?>
```

執行結果>>

1 2 3 4 5 6 7 8 9
-------------------

### 4. break

break 結束當前 for，foreach，while，do..while 或者 switch 結構的執行。

break 可以接受一個可選的數字參數來決定跳出幾重迴圈。

### 5. continue

continue 在迴圈結構用來跳過本次迴圈中剩餘的代碼並開始執行下一次迴圈。

continue 接受一個可選的數字參數來決定跳過幾重迴圈到迴圈結尾。

## 陣列

陣列是一個結構性的資料儲存空間，在程式設計上是不可或缺的工具。陣列由元素組成，每一個元素都持有一個數值。每個元素都是以它的索引(或鍵值)來參照的。

索引可以是字串或數位。如果省略了索引，會自動產生從 0 開始的整數索引。如果索引是整數，則下一個產生的索引將是目前最大的整數索引 + 1。注意如果定義了兩個完全一樣的索引，則後面一個會覆蓋前一個。

## 陣列初始化

在陣列初始化中，我們會用到下列的方法：

- 直接將值設定給陣列
- 由 `list()` 或 `array()` 函數來宣告

「直接將值設定給陣列」是最簡單的方法，例如：

```
01: <?php
02:     $a[] = "台北";
03:     $a[] = "台中";
04:     $a[] = "高雄";
05:
06:     for ( $i=0 ; $i<3 ; $i++)
07:         echo "$i -> $a[$i] <BR>";
08: ?>
```

執行結果>>

```
0 -> 台北
1 -> 台中
2 -> 高雄
```

第 02~04 行：指定值給 `$a[]` 陣列；

第 06~08 行：將 `$a[]` 陣列的內容顯示出來。

以上面的示範為例，如果沒有指定索引值給 `$a[]`，那麼，PHP 會預設索引值由 0 開始，依序給 1、2，

```
$a[0] = "台北";
$a[1] = "台中";
$a[2] = "高雄";
```

如果要用文字的索引值的話，要如何設定及使用呢？舉例如下：

```
01: <?php
02:     $c["Jan"] = "一月";
03:     $c["Feb"] = "二月";
04:     $c["Mar"] = "三月";
05:
06:     echo $c["Feb"];
07: ?>
```

執行結果>>

```
二月
```

第 02~04 行：指定陣列「文字」的索引值及內容；

第 06 行：顯示陣列元素的內容

以經驗來說，文字的索引值可用雙引號，也可以用單引號來表達。

「由 list() 或 array() 函數來宣告」也是在 PHP 程式設計上蠻常用的方法，舉例如下：

```
01:  <?php
02:      $a = array ( "台北","台中","高雄");
03:
04:      for ( $i=0 ; $i<3 ; $i++)
05:          echo "$i -> $a[$i] <BR>";
06:  ?>
```

執行結果>>

```
0 -> 台北
1 -> 台中
2 -> 高雄
```

第 02 行：指定值給 \$a[] 陣列；

第 04~06 行：將 \$a[] 陣列的內容顯示出來。

上述範例，也沒有指定索引值，只是將「值」指定給陣列，所以陣列也會自動給 0、1、2 的索引值。

如果指定索引值的話：

```
01:  <?php
02:      $b = array ( 100 => "校長室",
03:                  200 => "教務處",
04:                  300 => "訓導處",
05:                  400 => "總務處" );
06:
07:      echo $b[200];
08:  ?>
```

執行結果>>

```
教務處
```

第 02~04 行：指定陣列索引值及內容；

第 05 行：不指定索引值；其實 02~05 是同一段，這種寫法只是為程式的可讀性及美觀。

第 07 行：顯示陣列元素的內容

如果索引值是用「文字索引值」，則寫法如下：



```

01: <?php
02:     $c = array ("Jan" => "一月",
03:                 "Feb" => "二月",
04:                 "Mar" => "三月" );
05:
06:     echo $c["Feb"];
07: ?>

```

執行結果>>

二月

第 02~04 行：指定陣列「文字」的索引值及內容；

第 06 行：顯示陣列元素的內容

## 二維陣列

一般來說，一般(一維)的陣列已經很好用了，但有時會應用到多維的陣列，例如：

Brand	Money	Color	Type
BMW	200	Black	328i
MAZDA	70	Blue	323
HONDA	40	White	City
VOLVO	100	Pink	s40

PHP 程式設計的範例如下：

```

01: <?php
02: $car = array (
03:     "BMW" => array ("Money" => 200, "Color" => "Black", "Type" => "328i"),
04:     "MAZDA" => array ("Money" => 70, "Color" => "Blue", "Type" => "323"),
05:     "HONDA" => array ("Money" => 40, "Color" => "White", "Type" => "City"),
06:     "VOLVO" => array ("Money" => 100, "Color" => "Pink", "Type" => "s40"));
07:
08:     echo $car["BMW"]["Type"];
09: ?>

```

執行結果>>

328i

第 02 行：設定 \$car 是一個陣列，它的索引值是 BMW、MAZDA、HONDA、VOLVO。

第 03 行：設定 BMW 這個索引值下面又有「第二層」索引值，其他以此類推。

第 09 行：二維陣列值(存)取的方法：\$變數名稱[第一層索引件][第二層索引件]