重覆結構指令

當我們要計算某一事物的數量或某一事件的執行次數時，方法有很多種，

例如：要求某一動作執行５次，計算方法可以如下各行所示

1. 1，2，3，4，5
2. 5，4，3，2，1
3. 1，3，5，7，9

方法隨各人喜好而定(雖然 方法A是大部人的用法)。

不過，不管使用那種方法，我們可以發現，所有的計算方法都存在幾個相同規則：

1. 藉由數字的變化(設定一個 「變數」)來計算
2. 每一種方法都有一個起始數字(稱為：索引初始值)
3. 所有用來計算的數字都符合某項特定條件（稱為：終止條件──不符合條件即終止計算／執行）
4. 每執行一次，用來計算的數字必須做固定的變化（稱為：索引步進值/數字變更方式）

我們將上述規則表列如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方法 | 藉用變數  可自行設定變數名稱 | 索引初始值 | 終止條件 | 索引步進值 |
| 1，2，3，4，5 | $ct | $ct=1 | $ct<=5($ct<6) | $ct++ (每次加1) |
| 5，4，3，2，1 | $x | $x=5 | $x>=1($x>0) | $x-- (每次減1) |
| 1，3，5，7，9 | $y | $y=1 | $y<=9($y<10) | $+=2 (每次加2) |

所以，php程式提供了幾種「重覆結構指令」如下:

1.for指令

指令格式

For(索引初始值；終止條件；索引步進值)

{

重覆執行動作指令；

}

假設我們要求電腦印出20次的「我愛妳」，用不同計算方法設計的程式表列如下：

|  |  |  |
| --- | --- | --- |
| 計算方法  1，2，3，4，5 | 計算方法  5，4，3，2，1 | 計算方法  1，3，5，7，9 |
| For( $ct=1; $ct<=5; $ct++)  {  Echo “我愛妳<br>”;  } | For( $x=5; $x>0; $x--)  {  Echo “我愛妳<br>”;  } | For( $y=1; $y<=10; $y+=2)  {  Echo “我愛妳<br>”;  } |

如果，我們想要確認是否真的有執行２０次，我們也可以將藉用的變數資料值印出來做驗證，如下：

|  |  |  |
| --- | --- | --- |
| 計算方法  1，2，3，4，5 | 計算方法  5，4，3，2，1 | 計算方法  1，3，5，7，9 |
| For( $ct=1; $ct<=5; $ct++)  {  　Echo “第”.$ct.”次：”;  Echo “我愛妳<br>”;  } | For( $x=5; $x>0; $x--)  {  　Echo “第”.$x.”次：”;  Echo “我愛妳<br>”;  } | For( $y=1; $y<=10; $y+=2)  {  　Echo “第”.$y.”次：”;  Echo “我愛妳<br>”;  } |

上表所列，只有第一種方法會依1,2,3,4,5順序列印次數；如果我們也希望另外二種方法也可印出1,2,3,4,5的順序，可將程式修正如下：

|  |  |
| --- | --- |
| 計算方法  5，4，3，2，1 | 計算方法  1，3，5，7，9 |

計算，再將結果印出。

ch10-01.htm

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=big5">

<title>輸入整數</title>

</head>

<body>

<h2>輸入大於1整數：</h2>

<form name="form1" method="post" action="3-2-1.php">

<p>

<input type="text" name="number">

</p>

<p>

<input type="submit" name="Submit" value="送出">

<input name="reset" type="reset" id="reset" value="重設">

</p>

</form>

</body>

</html>

ch10-01.php

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=big5">

<title>for 重覆結構</title>

</head>

<body>

<?php

$in\_Number=$\_REQUEST[number];

$sum=0;

if ($in\_Number>1){

//for 重覆結構

for($a=1;$a<$in\_Number;$a++){

echo $a." + ";

$sum +=$a;

}

echo $in\_Number." = ";

$sum += $in\_Number;

echo $sum;

}

Else

{

echo "<font color='red'>輸入數值<1, 請重新輸入!</font>";

}

?>

</body>

</html>

單元習作1：1.計算　1+3+5+….+99 的和

2.計算 1\*2\*3\*….\*9 的積

**綜合應用練習題組（單元作業１）**

從上面的例子說明中，我們知道，我們可以應用「重覆廻圈」的變數值於工作項目中，如下：

例1：ex10-01-00.php

For($da=1;$da<=31;$da++)

{

Echo $da;

}

程式執行結果如下:

12345678910111213141516171819202122232425262728293131

例2：

For($da=1;$da<=31;$da++)

{

Echo $da.”<br>”;

}

程式執行結果如下:

1

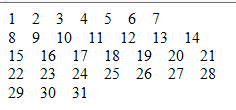
2

3

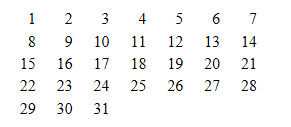
4

31

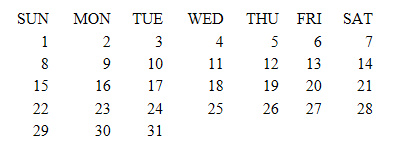
練習1：印出如下結果ex10-01-01.php



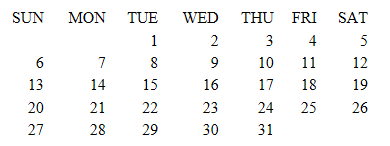
練習2：印出如下結果(列印資料對齊)ex10-01-02.php



練習3：印出如下結果(加入星期標題)ex10-01-03.php

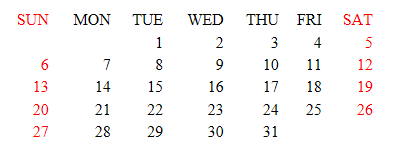


練習4：印出如下結果(1號可以從任一位置開始列印)ex10-01-04.php

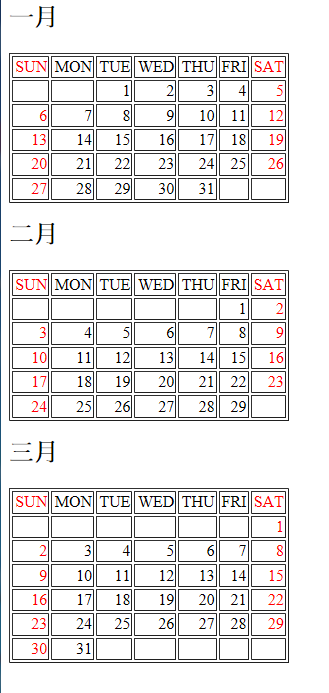


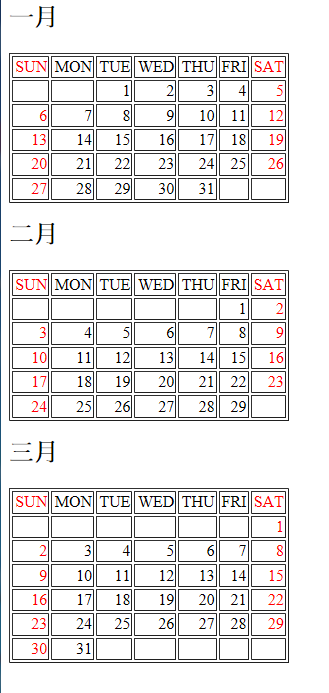
練習4-1：輸入1號位置(W0~W6)，印出上面結果(1號位置會依輸入值變動)

練習5：印出如下結果(星期六、日資料為紅色)ex10-01-05.php

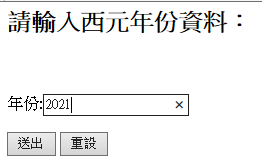
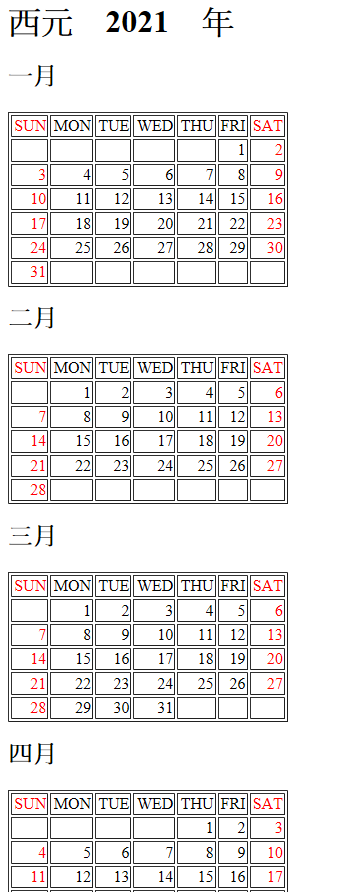


練習6：印出如下結果(設元旦為任意位置，印出1~12月份的月曆) ex10-01-  
06.php

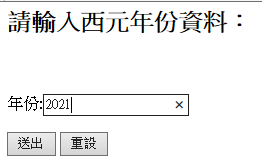




練習7：輸入任一西元年份，印出該年年曆，如下：ex10-01-07-1.php

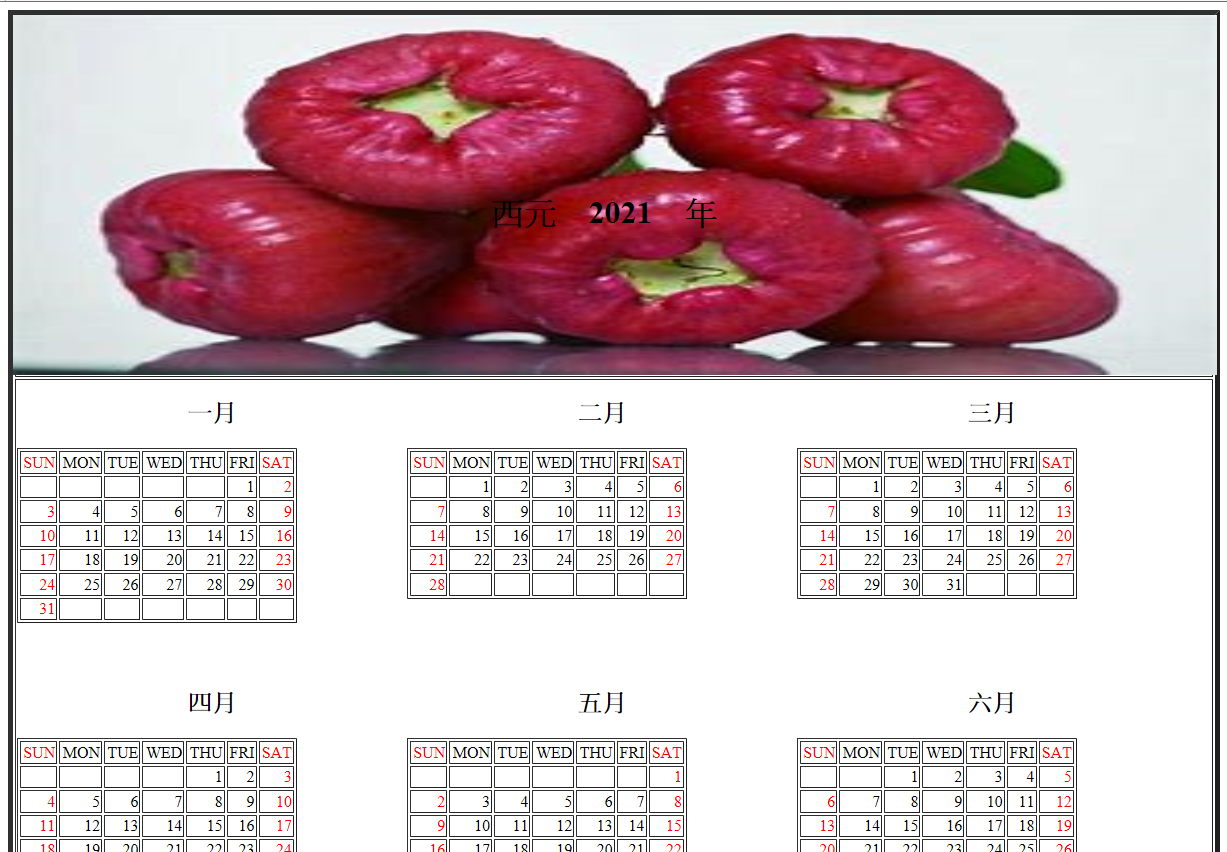


練習8：輸入任一西元年份，印出該年年曆(月份排列可隨視窗自動調整)，如下：ex10-01-08.php

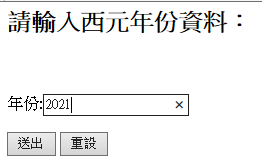




練習9：承上題，印出的年曆分上下二列，第一列為圖案及年份，第二列為月曆排列，如下：ex10-01-09.php

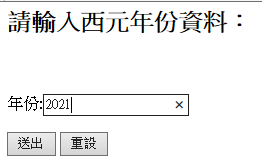


練習10：承上題，除輸入年份外,另增加月曆種類選擇如下:



請選擇月曆類型

水果 空姐 消防 風景



2.while重覆結構　與　Do…while重覆結構

For(索引初始值；終止條件；索引步進值)

{

重覆執行動作指令；

}

除了for重覆結構外，大部的程式語言，也另外提供了「while」跟「Do…while」兩種結構，不過，無論是那種結構，都必須應用「控制廻圈三要素」來完成重覆次數的控制。

While 指令格式：

　索引初始值設定

While(終止條件)

{

重覆執行動作指令；

　　索引步進值

}

Do…while重覆結構

索引初始值設定

Do

{

重覆執行動作指令；

索引步進值

} While(終止條件)

我們用下三個程式來瞭解比較三種廻圈的架構

重覆５次

|  |  |  |
| --- | --- | --- |
| for | while | Do…while |
| For( $ct=1; $ct<=5; $ct++)  {  Echo "我愛妳<br>";  } | $ct=1;  while( $ct<=5)  {  Echo "我愛妳<br>"  $ct++;  } | $ct=1;  Do  {  Echo"我愛妳<br>";  $ct++;  } while( $ct<=5) |

同學可能會覺得while 跟do…while是一樣的架構，其實，他們的差異點是，

「For廻圈」 與「while廻圈」的重覆動作有可能不會被執行到，

但「do…while廻圈」的動作則至少會被執行一次。

我們把上表程式中的條件改為 $ct<=0，再分別看執行結果，就可瞭解了。

|  |  |  |
| --- | --- | --- |
| for | while | Do…while |
| For( $ct=1; $ct<=0; $ct++)  {  Echo"我愛妳<br>";  } | $ct=1;  while( $ct<=0)  {  Echo "我愛妳<br>";  $ct++;  } | $ct=1;  Do  {  Echo "我愛妳<br>";  $ct++;  } while( $ct<=0); |

例：應用while 指令做出同ch10-01.php的結果 ，輸入一個大於1的整數n ，再由後端程式進行 1+2+….+n的計算，再將結果印出。

ch10-02.htm （同 ch10-01.htm）

ch10-02.php

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=big5">

<title>while 重覆結構</title>

</head>

<body>

<?php

$in\_Number=$\_REQUEST[number];

$sum=0;

$a=1; //初值設定

if ($in\_Number>1){

//while 重覆結構

while($a<$in\_Number){

echo $a." + ";

$sum +=$a;

$a ++; //變數$a遞增

}

echo $in\_Number." = ";

$sum += $in\_Number;

echo $sum;

}else{

echo "<font color='red'>輸入數值< 1, 請重新輸入!</font>";

}

?>

</body>

</html>

例：輸入一個大於3的奇數整數n ，再由後端程式進行 1+3+5+….+n的計算，再將結果印出。

ch10-03.htm

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=big5">

<title>輸入奇數</title>

</head>

<body>

<h2>輸入大於1的奇數：</h2>

<form name="form1" method="post" action="ch10-03.php">

<p>

<input type="text" name="number">

</p>

<p>

<input type="submit" name="Submit" value="送出">

<input name="reset" type="reset" id="reset" value="重設">

</p>

</form>

</body>

</html>

ch10-03.php

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=big5">

<title>do...while 重覆結構</title>

</head>

<body>

<?php

$in\_Number=$\_REQUEST[number];

$sum=0;

$a=1;

if(($in\_Number>1) && ($in\_Number%2!=0)){

//do...while重覆結構

do{

echo $a." + ";

$sum +=$a;

$a +=2; //變數$a遞增2

}while($a<$in\_Number);

echo $in\_Number." = ";

$sum += $in\_Number;

echo $sum;

}else{

echo "<font color='red'>所輸入的數值小於3,或不是奇數,請重新輸入!</font>";

}

?>

</body>

</html>

單元作業２：用while結構設計萬年曆(練習10)程式

中斷廻圈指令

1.break；指令(中斷整個廻圈的執行)

例：設計一個顯示九九乘法表格程式，但程式只執行到6\*9即停止

ch10-04.php

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=big5">

<title>break 指令</title>

<style>

h2 {color:#CC00FF}

</style>

</head>

<body>

<h2>九九乘法表</h2>

<?php

echo "<table bgcolor='#FFFFCC' border=2>";

for($a=1;$a<10;$a++){

//加入if條件式

if($a>6){

break; //跳出迴圈

}

echo "<tr>";

for($b=1;$b<10;$b++){

echo "<td>".$a."\*";

echo $b." = ".$a\*$b."</td>";

}

echo "</tr>";

}

echo "</table>";

?>

</body>

</html>

2.continue；(中斷本次廻圈的執行，直接跳到廻圈的起始處繼續執行後續動作)

例：設計一個顥示九九乘法表格程式，但程式只執行到\*7的資料

ch10-05.php

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=big5">

<title>break 指令</title>

<style>

h2 {color:#CC00FF}

</style>

</head>

<body>

<h2>九九乘法表(二)</h2>

<?php

echo "<table bgcolor='#FFFFCC' border=2>";

for($a=1;$a<10;$a++){

echo "<tr>";

for($b=1;$b<10;$b++){

//加入if條件式

if($b>7){

//跳出此次迴圈,回到迴圈的起始

continue;

}

echo "<td>".$a."\*";

echo $b." = ".$a\*$b."</td>";

}

echo "<tr>";

}

echo "</table>";

?>

</body>

</html>