

Chapter

4

Dreamweaver設計萬年曆網頁

技能指標

1. 了解流程控制的迴圈目的
2. 了解「for」計數迴圈
3. 了解「while」、「do … while」條件迴圈
4. 了解巢狀迴圈
5. 了解「break, continue」跳躍敘述
6. 建立萬年曆網頁

學習總結

完成本章後、您將了解程式語當中的流程控制「迴圈」、它所扮演的重要角色，以及在程式執行的過程中它所擔任的重要任務，並且清楚知道流程控制「迴圈」，它在實際的應用上有多重要及使用的頻繁度。

了解流程控制的迴圈目的

迴圈是用來讓程式在執行時，可以針對某些動作一再的重複執行，在很多的情況下我們需要這樣情況來達到這樣的需求。

底下的程式片段中，一共4個動作要執行，在不受到迴圈的控制下，每一個動作僅會進行單一次的執行，如圖4-2結果一樣。

```

7  <?php
8  echo "1. 打開冰箱<br/>";
9  echo "2. 拿出一瓶可樂<br/>";
10 echo "3. 倒一杯可樂<br/>";
11 echo "4. 喝一口可樂<br/>";
12 ?>

```

圖4-1 標準敘述



圖4-2 執行結果

那麼底下的程式碼片段我們來看一下，前的4個動作，若是搭配了迴圈後，那麼在執行的結果上又會有什麼樣的變化呢？從圖4-4的結果中我們發現，在動作「4. 喝一口可樂」，這個動作一共會執行5次；這就是迴圈所帶來的好處。

```

7  <?php
8  echo "1. 打開冰箱<br/>";
9  echo "2. 拿出一瓶可樂<br/>";
10 echo "3. 倒一杯可樂<br/>";
11 for($i=1; $i<=5; $i++){
12     echo "4. 喝一口可樂<br/>";
13 }
14 ?>

```

圖4-3 for迴圈介紹

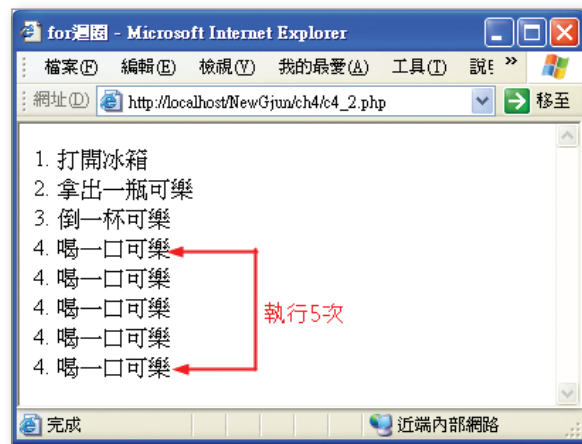


圖4-4 執行結果

CHAPTER
01

CHAPTER
02

CHAPTER
03

CHAPTER
04

CHAPTER
05

CHAPTER
06

CHAPTER
07

CHAPTER
08

CHAPTER
09

CHAPTER
10

CHAPTER
11

CHAPTER
12

CHAPTER
13

CHAPTER
14

CHAPTER
15

了解「for」計數迴圈

for迴圈是一種用來計算次數的迴圈，藉由計算次數的方式，重複的執行某一個動作。接下來讓我們搭配底下的程式碼片段及說明的，讓您可以更清楚的了解for迴圈的運作方式。

● 劇情：

小明有天早上起床後，想要來個運動，他選擇到操場跑個6圈，那麼這樣的迴圈敘述該如何撰寫呢？

語法：	舉例說明：
for([初始值運算]; [條件]; [增/減值]){ 連續動作; }	for(圈數=1; 圈數<=5; 圈數++){ 跑操場; }

```

9  <?php
10 for($ring=1; $ring<=6; $ring++){
11     echo "小明現在跑第 $ring 圈<br/>";
12 }
13 ?>

```

圖4-5 for迴圈介紹



圖4-6 執行結果

在撰寫迴圈敘述時，只需要決定哪些動作是需要重複執行，並且將它放到for迴圈的敘述區塊「{ … }」即可達成目的。

接下來我們再來看另外一個例子，這個例子要比前一個例子要稍為困難一些，當然也不困難，仔細想想看清楚，一定能夠了解的。

● 劇情：

假設阿財想知道 1 ~ 100 之間的奇數累加起來的值是多少？那麼程式該如何撰寫。

語法：	舉例說明：
<pre>for([初始值運算]; [條件]; [增/減值]){ 連續動作; }</pre>	<pre>for(計數數字=1; 計數數字<=5; 計數數字 ++){ 數字累加; }</pre>

```
9  <?php  
10 $sum = 0;  
11 for($count=1; $count<=100; $count++){  
12     if($count % 2 == 0){  
13         $sum += $count;  
14     }  
15 }  
16 echo "1 ~ 100 累加和: $sum<br/>";  
17 ?>
```

圖4-7 累加和計算



圖4-8 執行結果

了解「while」、「do … while」條件迴圈

除了前面所介紹的計數迴圈外，在迴圈的敘述還有另一外種也非常好用的迴圈敘述，條件迴圈、它與for迴圈不同的地方，因為for迴圈是一種單純的計算次數；但是while 或 do … while 是以條件的方式在執行，因此將會有些許不同的使用時機。

好了，底下我們一樣透過範例程式，還有程式說明的方式，讓您可以更清楚的了解條件迴圈的運作重點在那。

● 劇情：

小明有天早上起床後，想要來個運動，他選擇到操場跑個6圈，那麼這樣的迴圈敘述該如何撰寫呢？

語法：	舉例說明：
<pre>while(條件) { 連續動作; }</pre>	<pre>while(圈數<=6) { 跑操場; }</pre>

```

9  <?php
10 $ring=1; //代表圈數
11 while($ring<=6){
12     echo "小明正在跑第 $ring 圈(while)<br/>";
13     $ring++;
14 }
15 ?>
```

圖4-9 while條件迴圈介紹

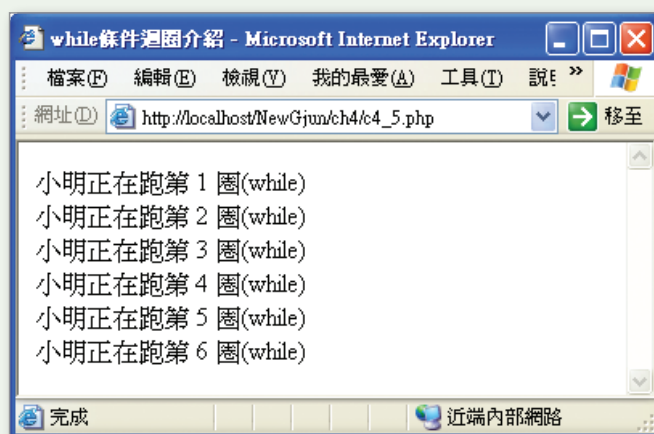


圖4-10 執行結果

那麼緊接著讓我們來看看do … while若是在相同的劇情下，它的寫法又會是如何撰寫的呢？

語法：	舉例說明：
do{ 連續動作; } while(條件);	do{ 跑操場; } while(圈數<=6);

```

8  <?php
9  $ring=1;
10 do{
11     echo "小明正在跑第 $ring 圈(do ... while)<br/>";
12     $ring++;
13 }while($ring<=6);
14 ?>

```

圖4-11 do…while條件迴圈介紹

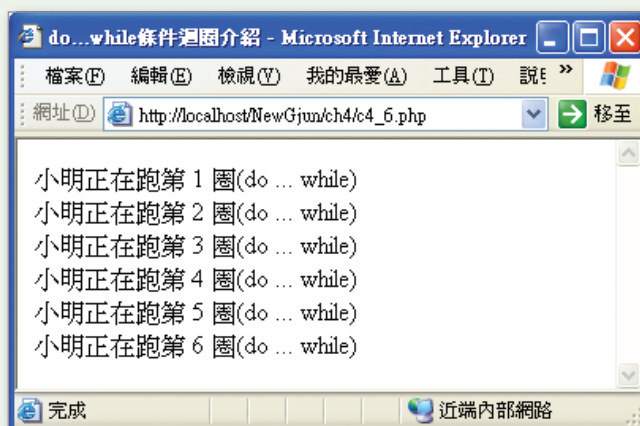


圖4-12 執行結果

那麼while與do … while藉由前一範例程式，實在沒有辦法找出有任何差異的部份，底下我們再就再來看看另一個範例程式，相信一定能夠讓您了解這兩者之間除了語法之外還有什麼不同之處。

while語法：	範例說明：
<pre> 9 <?php 10 \$money = 19; //代表目前的金額 11 echo "1. 打開冰箱
"; 12 echo "2. 拿出一瓶可樂
"; 13 echo "3. 倒一杯可樂
"; 14 while(\$money >= 20){ 15 echo "4. 喝一杯可樂
"; 16 } 17 ?> </pre>	<p>假設迴圈重複執行的條件是「\$money>=20」，因此迴圈在執行動作之前，必須先確定條件是符合。</p> <p>左邊範例程式碼中，因為條件並未符合，因此迴圈會立即結束，而不會執行任何動作。</p>

圖4-13 while迴圈範例

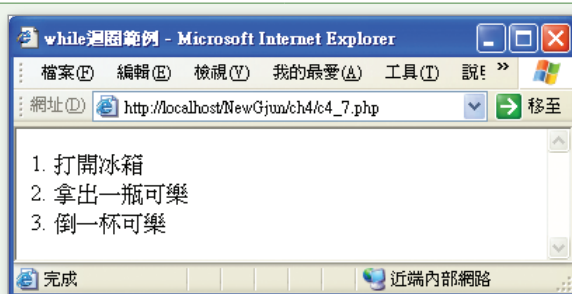


圖4-14 執行結果

do ... while語法：

```

9  <?php
10 $money = 19; //代表目前的金額
11 echo "1. 打開冰箱<br/>";
12 echo "2. 拿出一瓶可樂<br/>";
13 echo "3. 倒一杯可樂<br/>";
14 do{
15     echo "4. 喝一杯可樂<br/>";
16 }while($money >= 20);
17 ?>

```

圖4-15 do...while迴圈範例

範例說明：

程式執行至do的時候，將會先執行「{ ... }」區塊中的敘述，接著再依照條件是否符合，決定是否繼續執行下次。左邊範例程式碼中，雖然條件並未符合，但是當程式遇到do時已經執行了第1次，因此就算迴圈條件未符合，迴圈至少執行了單一次動作。

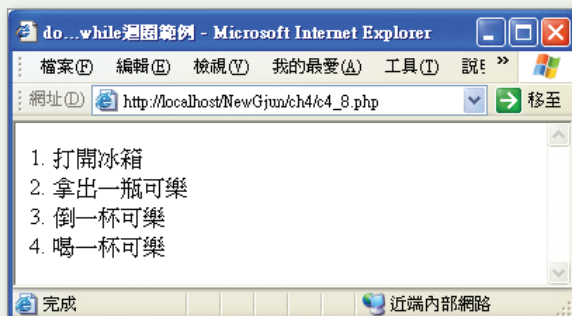


圖4-16 執行結果

了解巢狀迴圈

所謂巢狀迴圈就是在迴圈當中，在包含著另外一個迴圈，也因為如此的關係，我們通常被包含的為「內迴圈」、而包含的為「外迴圈」，透過圖4-9您可以更清楚的了解。

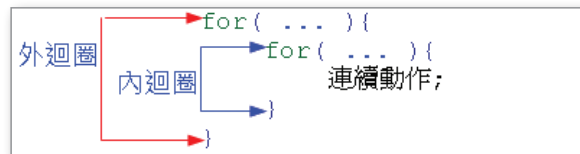


圖4-17 巢狀迴圈語法說明

巢狀迴圈並沒有限制迴圈使用的語法，只要符合巢狀迴圈的語法結構就可以，因此我們可以將for, while, do ... while這三個迴圈，任意的搭配應用。

接一來所要介紹的是，沒有巢迴圈以及使用巢狀迴圈，對程式執行上究竟有什麼樣的不同，它有什麼過人特別地方？那麼請看底下的範例，您就能夠清楚的了解。

範例一：	範例說明：
<pre> <?php for(\$row=1; \$row<=3; \$row++){ echo "row執行第 \$row 次
"; } for(\$col=1; \$col<=5; \$col++){ echo "col執行第 \$col 次
"; } ?> </pre>	<p>左邊的程式碼中，使用了兩個for迴圈，但是要注意喔！這兩個迴圈之間，其實並沒有任何的關係，它們是各自獨立的兩個迴圈。</p>
<pre> 9 <?php 10 for(\$row=1; \$row<=3; \$row++){ 11 echo "row執行第 \$row 次
"; 12 } 13 for(\$col=1; \$col<=5; \$col++){ 14 echo "col執行第 \$col 次
"; 15 } 16 ?> </pre>	

圖4-18 範例二

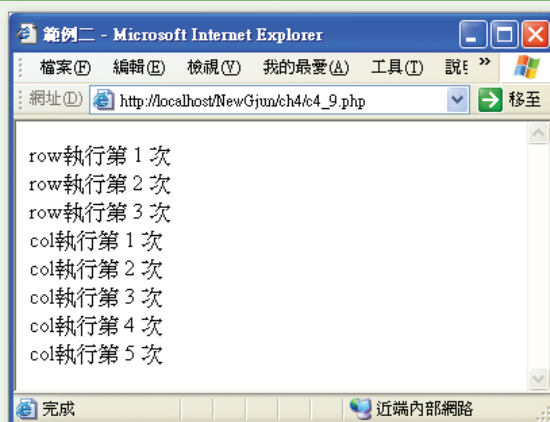


圖4-19 兩個單一for迴圈執行結果

範例二：

```

9  <?php
10 for($row=1; $row<=3; $row++){
11     echo "row執行第 $row 次<br/>";
12     for($col=1; $col<=5; $col++){
13         echo "col執行第 $col 次, ";
14     }
15     echo "<br/>";
16 }
17 ?>

```

圖4-20 for巢狀迴圈

範例說明：

左邊程式碼中，注意看清楚了，此時的for迴圈已經是巢狀迴圈之間的關係了。

此時，請將\$row當成是「列」，接著將\$col當成是「欄」，這樣一來就可以形成一個表格了。

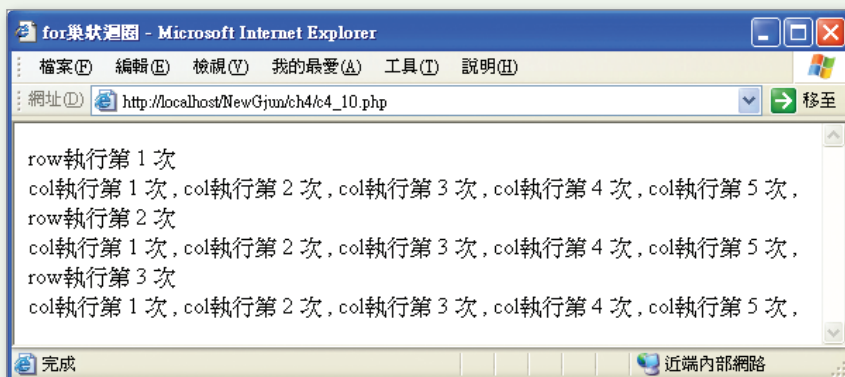


圖4-21 巢狀迴圈執行結果

藉由前面的介紹，您是不是更了解巢狀迴圈了呢？接下來、巢狀迴圈還有一般典型常見的範例，一樣請您一起跟著我的說明，相信一定更可以了解巢狀迴圈的應用。

九九乘法表範例搭配表格：

```

9  <?php
10 echo "<table border=\"1\">"; //印出<table border="1">
11 for($row=2; $row<=9; $row++){ //代表「列」
12     echo "<tr>"; //印出表格「列」<tr>
13     for($col=1; $col<=9; $col++){ //代表「欄」
14         //印出列的「欄<td>」 內容 結束「欄</td>」
15         echo "<td>$row*$col= " . ($row*$col) . "</td>";
16     }
17     echo "<tr/>"; //結束「列</tr>」
18 }
19 echo "</table>"; //結束表格</table>
20 ?>

```

圖4-22 九九乘法表搭配表格

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

圖4-23 九九乘法表執行結果

CHAPTER
01

CHAPTER
02

CHAPTER
03

CHAPTER
04

CHAPTER
05

CHAPTER
06

CHAPTER
07

CHAPTER
08

CHAPTER
09

CHAPTER
10

CHAPTER
11

CHAPTER
12

CHAPTER
13

CHAPTER
14

CHAPTER
15

了解「break, continue」跳躍敘述

迴圈執行的過程中，程式不一定每一次都必須要執行完畢，某些時候或許您需要，在某一個特定的情況發生時，立即將程式結束停止執行；在PHP當中提供兩個關鍵字，一個「break」、另一個「continue」，接下來一起來觀察有什麼不一樣的地方。

未搭配break介紹：

```

9  <?php
10 $sum = 0;
11 for($count=1; $count<=10; $count++){
12     $sum += $count;
13     echo "count= $count , sum= $sum<br/>";
14 }
15 echo "-----<br/>";
16 echo "count= $count &nbsp; sum = $sum<br/>";
17 ?>

```

圖4-24 未搭配break介紹

範例說明：

左邊的程式碼，在尚未搭配break之前，for迴圈能夠更正確的執行10次，一直到\$count變數在未小於10，才結束迴圈執行。

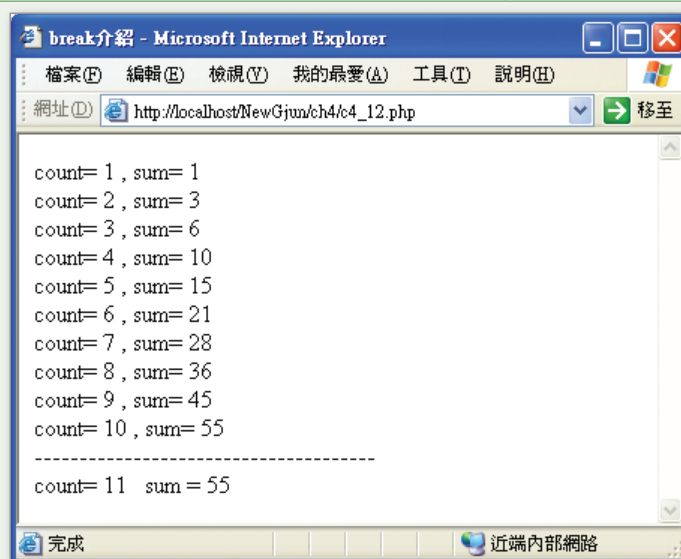


圖4-25 執行結果

搭配break介紹：

```

9  <?php
10 $sum = 0;
11 for($count=1; $count<=10; $count++){
12     $sum += $count;
13     if( $sum >= 25 ){
14         break; //立即結束迴圈執行
15     }
16     echo "count= $count , sum= $sum<br/>";
17 }
18 echo "-----<br/>";
19 echo "count= $count &nbsp; sum = $sum<br/>";
20 ?>

```

圖4-26 搭配break介紹

範例說明：

左列程式碼中，for迴圈搭配了break，則使得當for迴圈中\$sum的變數在大於或等25時，立即將迴圈停止。也因此，for迴圈在這個範例中，也僅執行了6次，藉由圖4-27就可以了解。

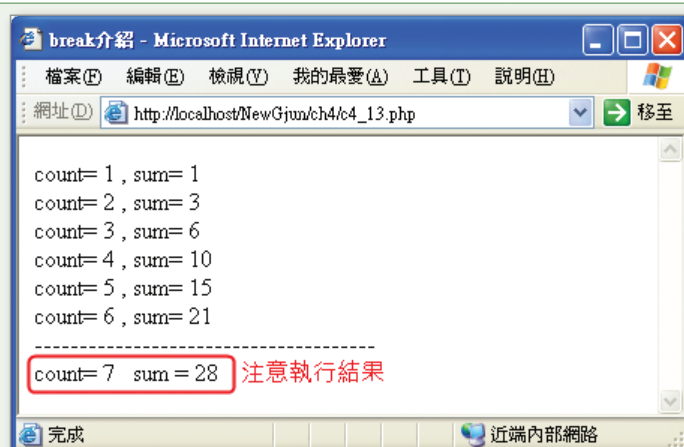


圖4-27 搭配break執行結果

除了break外，在PHP中還提供了另外一個跳躍敘述，那就是continue，不過continue與break卻有非常不一樣的執行情況，與需求的時機，底下我們一樣藉由程式碼範例來進行說明。

搭配continue介紹：

```

9  <?php
10 $sum = 0;
11 for($count=1; $count<=10; $count++){
12     $sum += $count;
13     if( $sum >= 25 ){
14         continue; //結束迴圈單次執行
15     }
16     echo "count= $count , sum= $sum<br/>";
17 }
18 echo "-----<br/>";
19 echo "count= $count &nbsp; sum = $sum<br/>";
20 ?>

```

圖4-28 continue介紹

範例說明：

左列程式碼中，for迴圈搭配了continue，則使得當for迴圈中\$sum的變數在大於或等25時，將不再進行列印、但卻會繼續累加直到迴圈結束。也因此，for迴圈在這個範例中，最結果累加和依然為55。

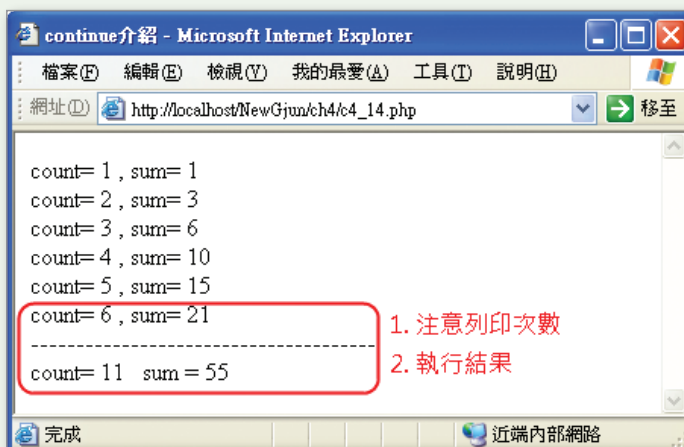


圖4-29 搭配continue執行結果

萬年曆網頁設計

最後一個部份讓我們綜合一下前面所學習到的觀念，我們來設計一個萬年曆的網頁，以提供在有需要輸入日期的時候可以搭配應用。

在這萬曆的設計部份需要使用到二個額外的函式，在此我們進行說明。

函數1： mktime([[int \$hour [, int \$minute [, int \$second [, int \$month [, int \$day [, int \$year]]]]]]))

用途： 取得UNIX時間戳記

說明： \$hour : 時

\$minute : 分

\$second : 秒

\$month : 月

\$day : 日

\$year : 年

函數2： date(string \$format,int \$timestamp)

用途： 格式化時間或時間

說明： \$format : 格式化字串符號

\$timestamp : 傳入的時間戳記

使用參數：

w : 取得星期幾代號（日～六、0～6）

t : 取得該月份天為（28～31）

接下來讓我們來完成萬年曆的網頁。

01 STEP 請在Dreamweaver檔案頁籤空白處建立一個新的頁面，並且將其命名為Calendar.php。

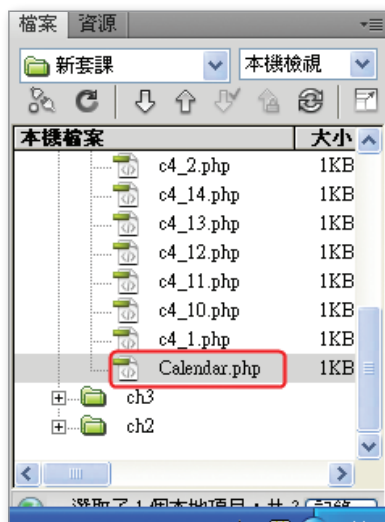


圖4-30 檔案頁籤

02 STEP 請開啓Calendar.php後切換到「設計」，接著點選「插入(I)」功能表→「表單(F)」→「表單(F)」，接著點選「插入(I)」→「表格」，設定為「3列2欄」。接著參考圖4-31完成其它設定。



圖4-31 萬年曆表格

03 STEP 接著將二個清單項目分為修改名稱為year以及month。



圖4-32 清單屬性

04 STEP 請點選「<form#form>」後在屬性視窗中輸入動作「Calendar.php」為接收處理的對象。即完成輸入表單的建置。

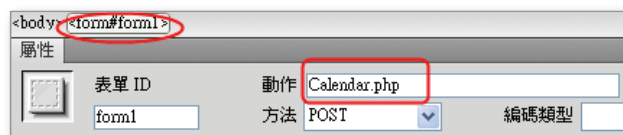


圖4-33 表單屬性

05 STEP 請切換到 **程式碼**，請將鍵盤的游標停駐在<body>標記後，接著請點選「插入(I)」功能表→「PHP物件(P)」→「程式碼區塊(O)」。接著請參考圖加入底下接收的程式碼。

```

16 <body>
17 <?php
18 $year      = $_POST['year']; //讀取清單選取年度
19 $month     = $_POST['month']; //讀取清單選取月份
20 //依選取年度及月份建立一時間戳記
21 $newtime   = mktime(0,0,0,$month,1,$year);
22 $nowyear   = date("Y"); //取得系統年度
23 $dayweek   = date("w",$newtime); //取得星期幾代號
24 $month_day = date("t",$newtime); //取得月份天數
25 ?>

```

圖4-34 程式區塊

06 STEP 接著分別加入年度產生以及月份產生的紅色程式碼區塊，年度是系統年度加減5年，而月份則是單純1~12。

```

30 <td>西元年：
31 <select name="year" id="year">
32 <?php
33 for($y=$nowyear-5; $y<=$nowyear+5; $y++){
34     if($y!=$year){
35         echo "<option value=\"$y\">$y</optgroup>";
36     }else{
37         echo "<option value=\"$y\" selected>$y</optgroup>";
38     }
39 }
40 ?>
41 </select></td>

```

圖4-35 年度產生

```

42 <td>月：|
43 <select name="select2" id="select2">
44 <?php
45 for($m=1; $m<=12; $m++){
46     if($m==$month){
47         echo "<option value=\"$m\">$m</optgroup>";
48     }else{
49         echo "<option value=\"$m\" selected>$m</optgroup>";
50     }
51 }
52 ?>
53 </select></td>

```

圖4-36 月份產生

07
STEP

接著將月曆產生出來。在這個例子當中使用了巢狀迴圈，外迴圈代表了週，而內迴圈則代表了天數。

```

60 <table border="1"><!--顯示表格-->
61 <tr>
62 <!-- 顯示月曆的星期幾 -->
63 <td>日</td><td>一</td><td>二</td><td>三</td><td>四</td>
64 <td>五</td><td>六</td>
65 </tr>
66 <?php
67 for($r=1; $r<=5; $r++){ //外迴圈用來表示一個月有「5週」
68     echo "<tr>"; //表格「列」
69     for($c=0; $c<=6; $c++){ //內迴圈用來表示一週有「7天」
70         if($r==1 && $c<$dayweek){ //第1週才會有要處理1號是星期幾
71             echo "<td>&nbsp;</td>"; //該「欄」顯示一個空白
72         }elseif($r==5 && $day>=$month_day){ //第5週要處理當月共幾天
73             echo "<td>&nbsp;</td>"; //該「欄」顯示一個空
74         }else{
75             echo "<td>".$day."</td>"; //該「欄」顯示正常的天數代號
76         }
77     }
78     echo "</tr>";
79 }
80 echo "</table>";
81 ?>
82

```

圖4-37 月曆產生

08
STEP

完成後存檔，請切換檔案到Calendar.php，接著按下「F12」執行，讓我們來測試結果是否正確。



圖4-38 執行結果

CHAPTER
01

CHAPTER
02

CHAPTER
03

CHAPTER
04

CHAPTER
05

CHAPTER
06

CHAPTER
07

CHAPTER
08

CHAPTER
09

CHAPTER
10

CHAPTER
11

CHAPTER
12

CHAPTER
13

CHAPTER
14

CHAPTER
15

自我測驗 & 能力檢核

完成每一項測驗後，請在□中打勾、代表完成您已了解。

☆流程控制—迴圈綜合練習、請試著寫出底下每道練習結果 ☐我已了解

1. 請撰寫一程式，接受輸入一個整數，並且列出 1 ~ 100 當中，為該整數的被數分別有哪些？

建議事項：請分別使用for、do ... while迴圈完成。

2. 請撰寫一程式，可以輸出費氏數列：假設數列的第1與第2數皆為1；而第3數則為前兩數相加的合，如此並依序類推。

建議事項：請使用 while迴圈完成。

輸出參考：1, 1, 2, 3, 5, 8, 13, 21, 34, ...

3. 請使用巢狀迴圈，輸出如下圖之九九乘法表。

建議事項：請搭配HTML語法<TABLE>, <TR>, <TD>來完成

輸出參考：



九九乘法表

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