



第 4 章

流程控制

4-1 if、if-else判斷式



- 在輸入畫面配置核取方塊

- 核取方塊，使用者自由選擇勾選 / 不勾選的控制元件

- 要做到依勾選結果執行不同的流程就必須使用 if 判斷式

- 首先，在網頁上配置核取方塊，相關程式如下，並將檔案儲存為 chapter4\check-input.php

```
check-input.php
<?php require '../header.php';?>
<form action="check-output.php" method="post">
<p><input type="checkbox" name="mail">訂閱特賣情報電子報</p>
<p><input type="submit" value="確定"></p>
</form>
<?php require '../footer.php';?>
```

核取方塊與確定按鈕

The screenshot shows a web form with a vertical sidebar on the left containing two circular icons with the text 'PHP'. The main content area contains a checkbox labeled '訂閱特賣情報電子報' (Subscribe to special sale information e-newsletter) and a '確定' (Confirm) button below it.

● 配置表單控制元件

–`action` 屬性是用來指定要接收表單資料並進行後續處理的程式
(本例指定為 `check-output.php`)

```
<form action="check-output.php" method="post">
```

▼ 利用 `<form>` 標籤製作表單畫面

```
<form action="check-output.php" method="post">
```

訂閱特賣情報電子報

確定

```
</form>
```

在這裡指定要執行的 PHP 程式

在 `<form>` 與 `</form>` 之間
撰寫建立控制元件的程式

按下按鈕後，執行上面
指定好的 PHP 程式



● 配置核取方塊

–核取方塊是利用 `<input>` 標籤建立，如下所示，將 `type` 屬性指定為 `checkbox`

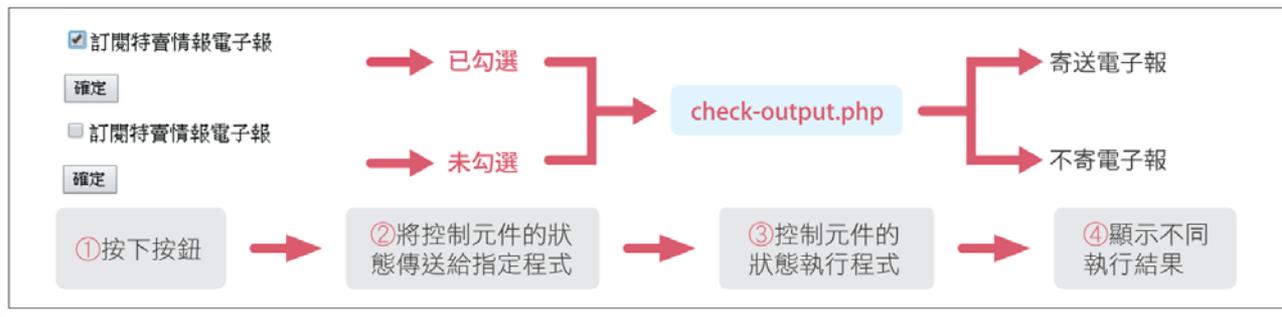
```
<input type="checkbox" name="mail">
```

● 配置按鈕

–同樣利用 `<input>` 標籤建立按鈕。將 `type` 屬性指定為 `submit`，`value` 屬性則用來指定要顯示在按鈕上的文字

```
<input type="submit" value="確定">
```

▼ 依控制元件傳回的狀態執行程式



● 判斷核取方塊是否被勾選

—先判斷表單的核取方塊勾選狀態，再依判斷結果顯示不同訊息

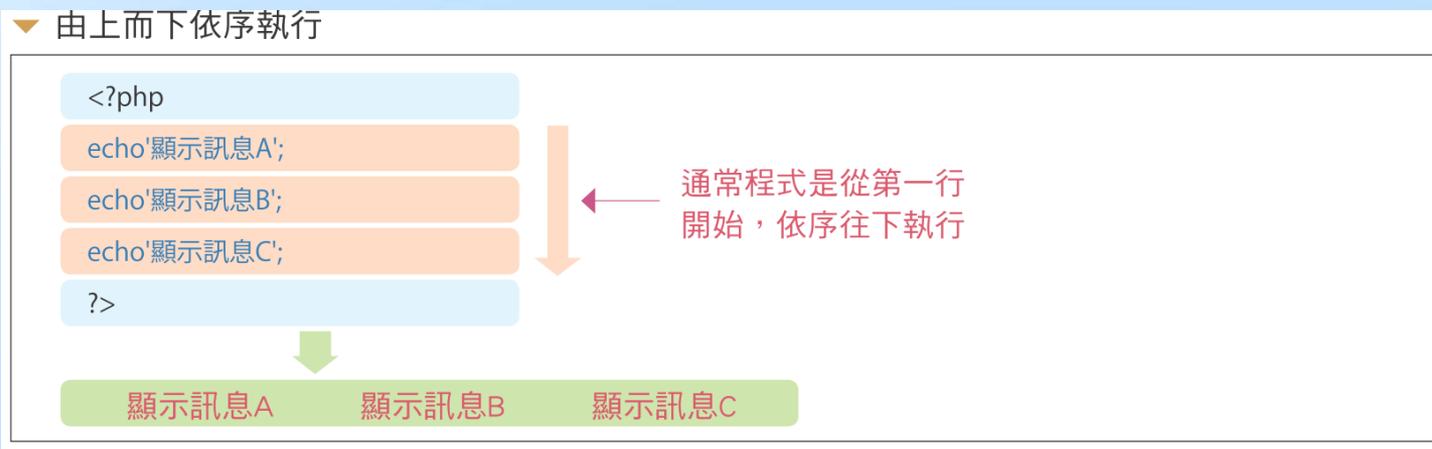
```

List  check-output.php PHP
<?php require '../header.php';?>
<?php
if (isset($_REQUEST['mail'])) {
    echo '已訂閱特賣情報電子報。';
} else {
    echo '未訂閱特賣情報電子報。';
}
?>
<?php require '../footer.php';?>

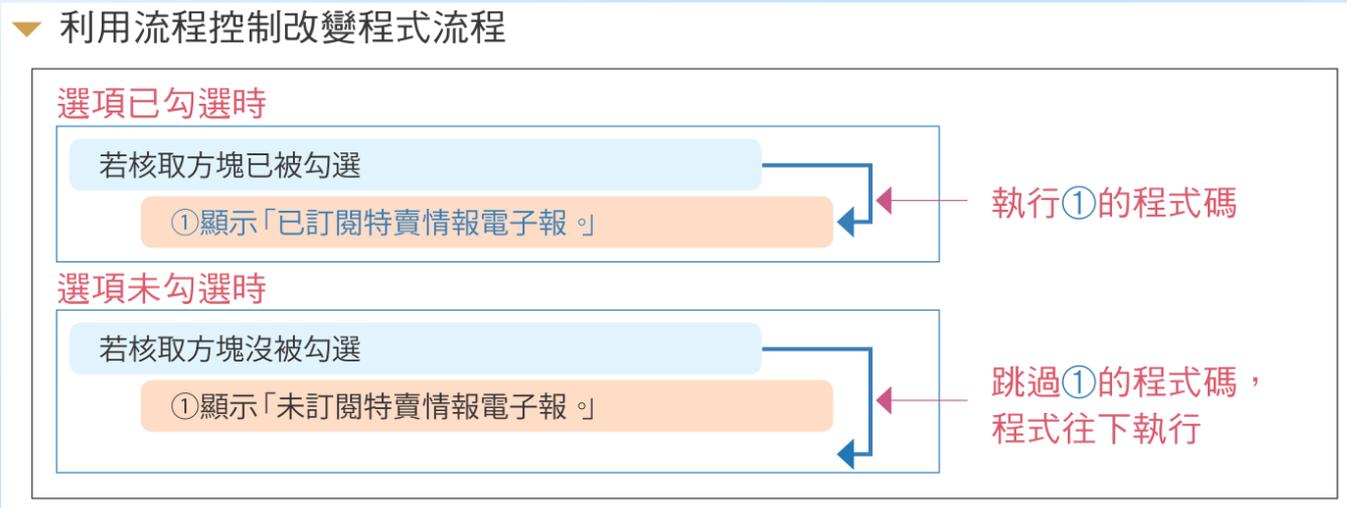
```


● 用 if 判斷式做判斷

—要做到「依核取方塊的勾選狀態，分別顯示不同的訊息」，由上而下依序執行的機制就無法做到

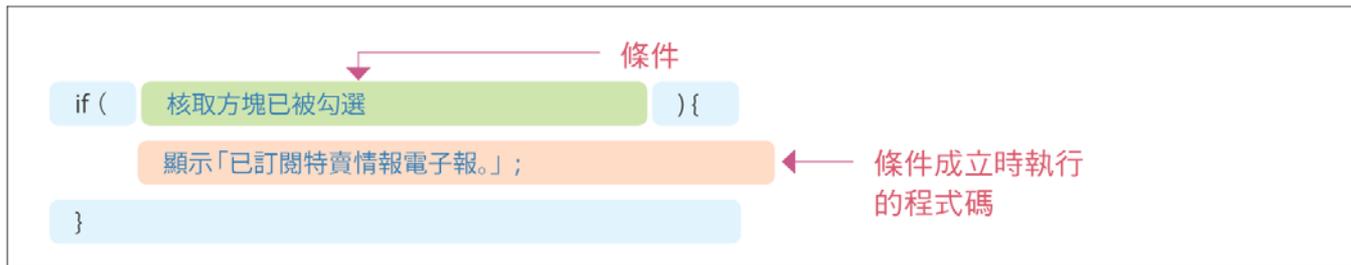


一 要依使用者輸入的內容變更程式的執行流程，必須使用流程控制機制。其運作概念如下圖所示

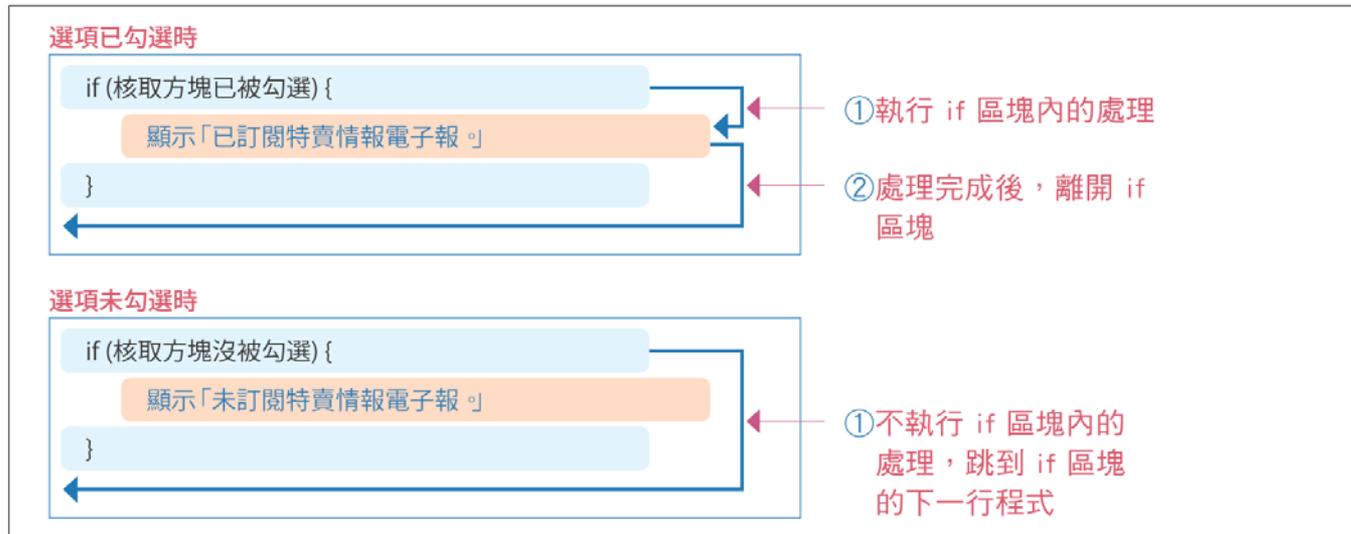


—以核取方塊的例子來說，if 判斷式將依下圖所示的方式撰寫

▼ 利用 if 判斷式切換顯示的訊息



▼ 流程控制的執行順序

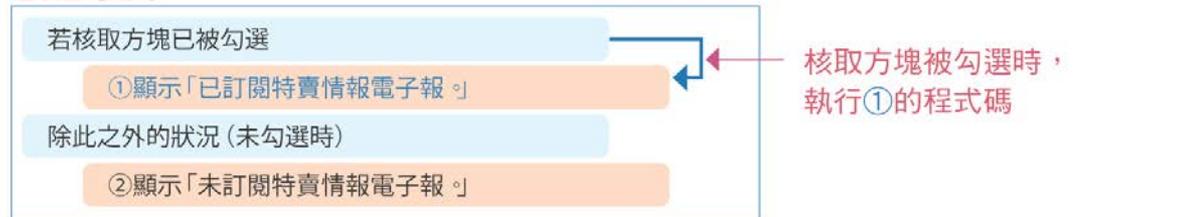


● 利用 if-else 判斷式製作條件判斷

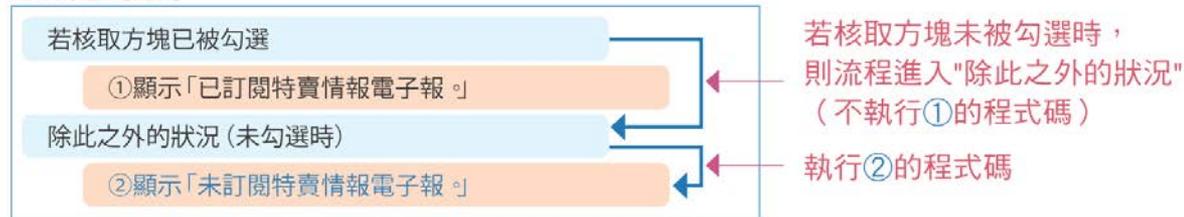
—在條件成立時與條件不成立時，分別執行對應的處理。其運作流程如下所示

▼ 依條件改變執行的流程

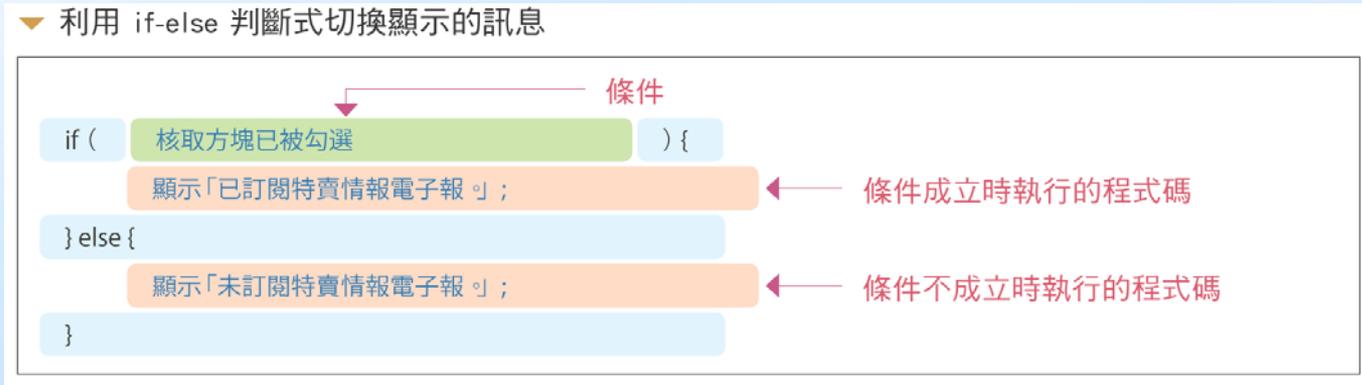
選項已勾選時



選項未被勾選時



—以核取方塊的例子來說，if-else 敘述將依下圖所示的方式撰寫



—實際用來執行的程式碼如下所示

```

if (isset($_REQUEST['mail'])) {
    echo '已訂閱特賣情報電子報。';
} else {
    echo '未訂閱特賣情報電子報。';
}

```

● 真偽值

—是用來表示條件是否成立的值，其值分為 TRUE 與 FALSE

語法 以真偽值代入 if 判斷式

```
if (條件) {
    條件為 TRUE 時執行的處理;
}
```

語法 以真偽值代入 if-else 判斷式

```
if (條件) {
    條件為 TRUE 時執行的處理;
} else {
    條件為 FALSE 時執行的處理;
}
```



● if...elseif....else 語句

```
if (條件1) {  
    條件為 true 時執行的代碼;  
} elseif (條件2) {  
    條件為 true 時執行的代碼;  
} else {  
    條件1和條件2都是 false 時執行的代碼;  
}
```



● 依核取方塊的狀態執行對應處理

–藉由 PHP 預設的 `$_REQUEST` 陣列變數，依下列語法取得 REQUEST 參數

語法 取得 REQUEST 參數

```
$_REQUEST['REQUEST 參數名稱']
```

–而變數是否已宣告，可利用 `isset` 函式檢查

語法 `isset`

```
isset(變數)
```

–利用 REQUEST 參數撰寫出來的程式如下

```
isset($_REQUEST['mail'])
```



← 將 if 判斷式、isset 函式以及 REQUEST 參數組合在一起，依以下語法撰寫程式

語法 在 if 判斷式中使用 REQUEST 參數

```
if (isset(REQUEST 參數中的變數) ) {  
    變數已宣告時要執行的處理；  
}
```

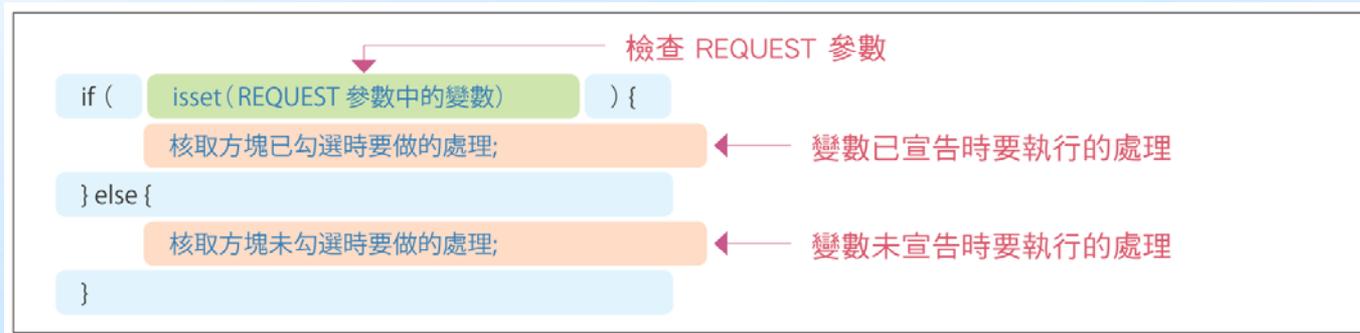
← 若是使用 if-else 判斷式與 isset 函式、REQUEST 參數撰寫，則語法如下

語法 在 if-else 的條件式中使用 REQUEST 參數

```
if (isset(REQUEST 參數中的變數) ) {  
    變數已宣告時要執行的處理；  
} else {  
    變數未宣告時要執行的處理；  
}
```



—依REQUEST參數的宣告狀態進流程控制



—實際執行的程式碼如下

```
if (isset($_REQUEST['mail'])) {
    echo '已訂閱特賣情報電子報。';
} else {
    echo '未訂閱特賣情報電子報。';
}
```

PHP isset() 函數



● 條件運算式

-運算式是由常數、變數、算符、函式所組成

-在 if 判斷式和 if-else 判斷式的條件中，也可使用運算式

語法 在 if 判斷式的條件中使用運算式

```
if (運算式) {  
    當運算結果為 TRUE 時執行的處理;  
}
```

運算式：**(3+5) > 7**



-在 if-else 判斷式的條件中使用運算式

語法 在 if-else 判斷式的條件中使用運算式

```
if (運算式) {  
    當運算結果為 TRUE 時執行的處理;  
} else {  
    當運算結果為 FALSE 時執行的處理;  
}
```

-在 if-else 判斷式中所使用的條件式，也是運算式的一種，如下

```
isset($_REQUEST['mail'])
```



● 省略大括號 {}

–當 if 判斷式的 **{}** 中只有 **1** 行程式時，可省略 {}。但若 {} 內有多行程式，則不可省略

語法 省略 {} 的 if 判斷式

```
if (運算式) 當運算結果為 TRUE 時執行的處理;
```

–if-else 敘述也一樣

語法 省略 {} 的 if-else 判斷式

```
if (運算式) 運算結果為 TRUE 時執行的處理; else 運算結果為 FALSE 時執行的處理;
```

PHP If...Else 語句

變數名稱的命名方式



- 變數名稱取得好，可以讓程式變得更易讀易懂，提高開發效率
- 變數名稱建議使用簡單的英文命名

4-2 switch 判斷式 (一)



● 輸入畫面上配置單選鈕

—請參照下列程式，並將檔案儲存為 chapter4\radio-input.php

```
chapter4\radio-input.php PHP
<?php require '../header.php';?>
請選擇餐點：
<form action="radio-output.php" method="post">
<p><input type="radio" name="meal" value="日式套餐" checked>日式套餐</p>
<p><input type="radio" name="meal" value="西式套餐">西式套餐</p>
<p><input type="radio" name="meal" value="中式套餐">中式套餐</p>
<p><input type="submit" value="確定"></p>
</form>
<?php require '../footer.php';?>
```

▼ 單選鈕與〔確定〕按鈕

請選擇餐點：

- 日式套餐
- 西式套餐
- 中式套餐

確定

顯示出〔日式套餐〕〔西式套餐〕〔中式套餐〕3個單選鈕以及〔確定〕按鈕。(多選一，name="meal")

● 建立單選鈕

—單選鈕 (Radio) 是利用 `<input>` 標籤建立，`type` 屬性值應指定為「radio」

```
<input type="radio" name="meal" value="日式套餐">
```

—要利用 **name** 屬性為單選鈕命名

```
<input type="radio" name="meal" value="西式套餐">  
<input type="radio" name="meal" value="中式套餐">
```

-name 屬性的值會成為 REQUEST 參數名

```
<input type="radio" name="meal" value="日式套餐">
<input type="radio" name="meal" value="西式套餐">
<input type="radio" name="meal" value="中式套餐">
```



REQUEST 參數
meal

-在 `<input>` 標籤中加上「**checked**」，就可設定該選項為預設選取的選項

```
<input type="radio" name="meal" value="日式套餐" checked>
```

● 依選取項目顯示不同訊息

—參照以下所示，撰寫程式判斷單選鈕的哪一個選項被選取

 radio-output.php.php
PHP

```

<?php require '../header.php';?>
<?php
switch ($_REQUEST['meal']) {
case '日式套餐':
    echo '烤魚、燉菜、味噌湯、白飯、水果';
    break;
case '西式套餐':
    echo '果汁、水波蛋、薯餅、麵包、咖啡';
    break;
case '中式套餐':
    echo '春捲、煎餃、蛋花湯、炒飯、杏仁豆腐';
    break;
}
echo '將稍候送達';
?>
<?php require '../footer.php';?>

```

▼ 選擇日式套餐時的執行結果


烤魚、燉菜、味噌湯、白飯、水果將稍候送達

▼ 選擇西式套餐時的執行結果


果汁、水波蛋、薯餅、麵包、咖啡將稍候送達

▼ 選擇中式套餐時的執行結果


春捲、煎餃、蛋花湯、炒飯、杏仁豆腐將稍候送達

● 利用 switch 判斷式控制流程

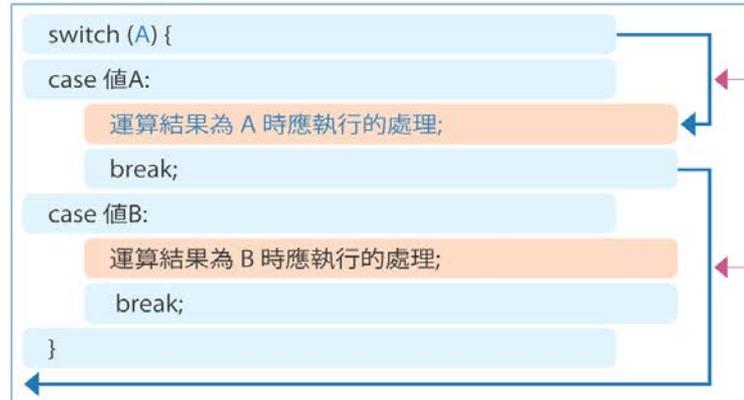
- 依照單選鈕被選取的項目顯示出不同訊息的程式，是常見將程式流程分成多段，再依使用者所選內容執行對應的流程，可用 if ... elseif ... 處理。
- switch 判斷式，讓程式更簡潔。switch 判斷式的語法如下

語法 switch

```
switch (運算式) {  
    case值 A:  
        運算結果為 A 時應執行的處理;  
        break;  
    case值 B:  
        運算結果為 B 時應執行的處理;  
        break;  
    case值 C:  
        運算結果為 C 時應執行的處理;  
        break;  
    .....  
}
```

—依switch的運算結果執行對應的case區塊

當運算結果為 A 時



①執行「case 值 A」
區塊內的程式

②執行完「case 值 A」區塊內
的程式後，跳出 switch 區塊

當運算結果為 B 時



①執行「case 值 B」
區塊內的程式

②執行完「case 值 B」區塊內
的程式後，跳出 switch 區塊

—以單選鈕選擇餐點，使用 switch 判斷式的程式撰寫概念如下

```

switch ( 選取的單選鈕 ) {
case 日式套餐:
    顯示烤魚、燉菜等;
    break;
case 西式套餐:
    顯示果汁、水波蛋等;
    break;
case 中式套餐:
    顯示春捲、煎餃等;
    break;
}

```

條件運算式

← 選取〔日式套餐〕時執行的程式

← 選取〔西式套餐〕時執行的程式

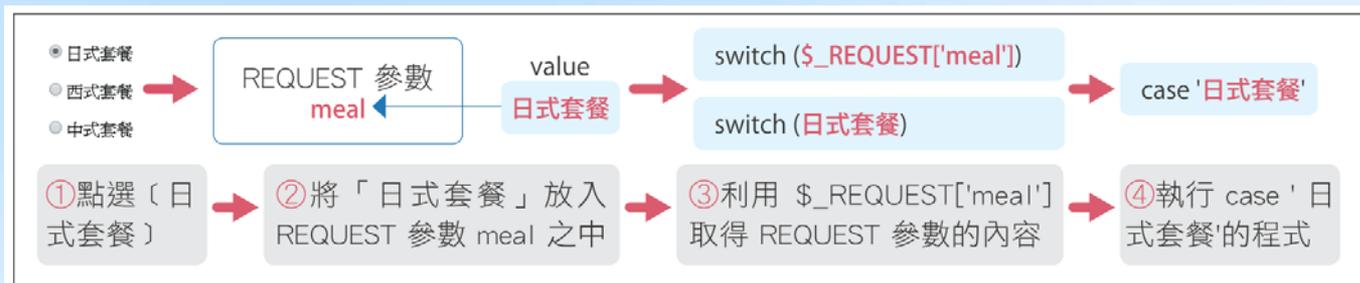
← 選取〔中式套餐〕時執行的程式

● 取得選取的單選鈕

– 這個 REQUEST 參數裡所存放的內容，即是被選取項目的 value 屬性值



– 在條件運算式中使用 REQUEST 參數



4-3 switch 判斷式 (二)



- 在輸入畫面上配置下拉式選單

—本例配置 1 個**下拉式選單**與 1 個確定鈕。請參照下列程式，並將檔案儲存為 chapter4\select-input.php

```
List select-input.php PHP
<?php require '../header.php';?>
<p>請選擇座位類型：</p>
<form action="select-output.php" method="post">
<select name="seat">
<option value="自由席">自由席</option>
<option value="指定席">指定席</option>
<option value="商務車廂">商務車廂</option>
</select>
<p><input type="submit" value="確定"></p>
</form>
<?php require '../footer.php';?>
```

▼ 下拉式選單與確定按鈕

請選擇座位類型：

自由席 ▼

確定

- 建立下拉式選單

- 利用 HTML 的 `<select>` 標籤

```
<select name="seat">  
...  
</select>
```

- 接著在 `<select>` 與 `</select>` 之間，利用 `<option>` 標籤加上選項

```
<option value="自由席">自由席</option>
```

● 依所選項目顯示不同訊息

— 撰寫「可依下拉式選單被選擇的項目顯示對應訊息」的程式

List
select-output.php
PHP

```

<?php require '../header.php';?>
<?php
switch ($_REQUEST['seat']) {
case '指定席':
    echo '需加付120元補票。';
    break;
case '商務車廂':
    echo '需加付250元補票。';
    break;
default:
    echo '不需補票。';
    break;
}
?>
<?php require '../footer.php';?>

```

▼ 選擇「自由席」時的執行結果

不需補票。

▼ 選擇「指定席」時的執行結果

需加付120元補票。

▼ 選擇「商務車廂」時的執行結果

需加付250元補票。

● switch 判斷式與 default

— 要依選單中被選取的項目分別顯示不同訊息

- 選擇〔自由席〕時 → 顯示不需補票的訊息
- 選擇〔指定席〕時 → 顯示需付 120 元補票的訊息
- 選擇〔商務車廂〕時 → 顯示需付 250 元補票的訊息



-default 敘述

- 選擇〔指定席〕時 → 顯示需付 120 元補票的訊息
- 選擇〔商務車廂〕時 → 顯示需付 250 元補票的訊息
- 以上皆非 → 顯示不需補票的訊息。



←default 敘述一般會寫在 switch 判斷式 {} 內的最後。與 case 敘述相同，它的最末行最好都加上 break。

語法 使用 default 的 switch 判斷式

```
switch (條件運算式) {  
  case 值 A:  
    運算結果為 A 時應執行的處理;  
    break;  
  case 值 B :  
    運算結果為 B 時應執行的處理;  
    break;  
  default:  
    運算結果與所有 case 的值都不符合時應執行的處理;  
    break;  
}
```

一用下拉式選單選擇座位類型的範例

```

switch ( 用下拉式選單選擇的項目 ) {
case 指定席:
    顯示需付 120 元補票的訊息;
    break;
case 商務車廂:
    顯示需付 250 元補票的訊息;
    break;
default:
    顯示不需補票的訊息;
    break;
}

```

條件運算式

選取 [指定席] 時應執行的程式

選取 [商務車廂] 時應執行的程式

以上皆非時應執行的程式



● 取得選單中所選的選項

–利用 **switch** 判斷式，依選項分別顯示不同訊息的程式如下

```
switch ($_REQUEST['seat']) {  
  case '指定席':  
    echo '需加付120元補票。';  
    break;  
  case '商務車廂':  
    echo '需加付250元補票。';  
    break;  
  default:  
    echo '不需補票。';  
    break;  
}
```

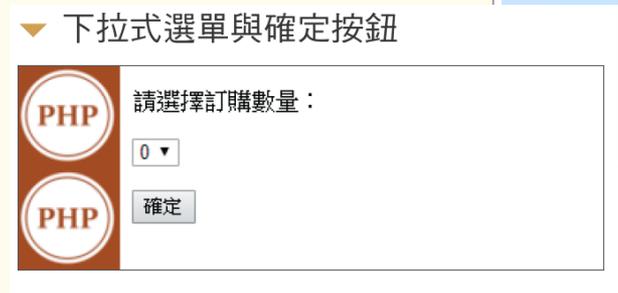
4-4 for迴圈、while迴圈



● 逐筆撰寫選項

一試著以手動標記下拉式選單和選項。程式內容如下

```
List select-for-input.php PHP
<?php require '../header.php';?>
<p>請選擇訂購數量：</p>
<form action="select-for-output.php" method="post">
<select name="count">
<option value="0">0</option>
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
<option value="5">5</option>
<option value="6">6</option>
<option value="7">7</option>
<option value="8">8</option>
<option value="9">9</option>
</select>
<p><input type="submit" value="確定"></p>
</form>
<?php require '../footer.php';?>
```



● 利用程式產生選項

—請參照下列程式，並將檔案儲存為 `chapter4\select-for-input2.php`

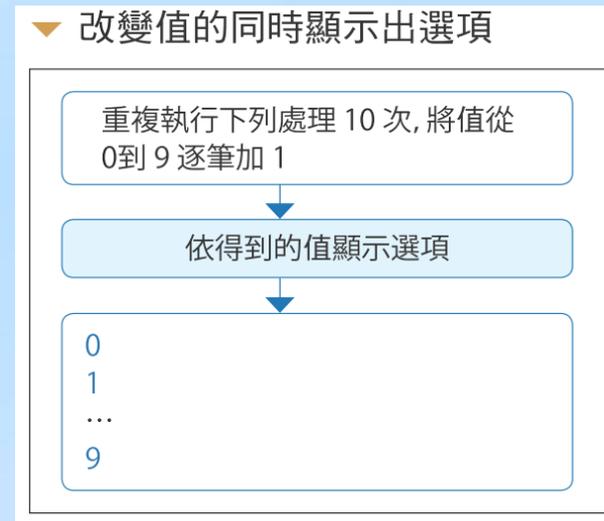
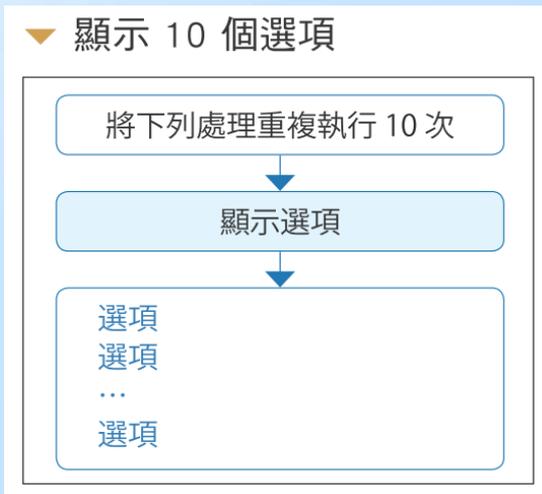
```

List  select-for-input2.php PHP
<?php require '../header.php';?>
<p>請選擇訂購數量：</p>
<form action="select-for-output.php" method="post">
<select name="count">
<?php
for ($i=0; $i<10; $i++) {
    echo '<option value="' . $i . '>' . $i . '</option>';
}
?>
</select>
<p><input type="submit" value="確定"></p>
</form>
<?php require '../footer.php';?>

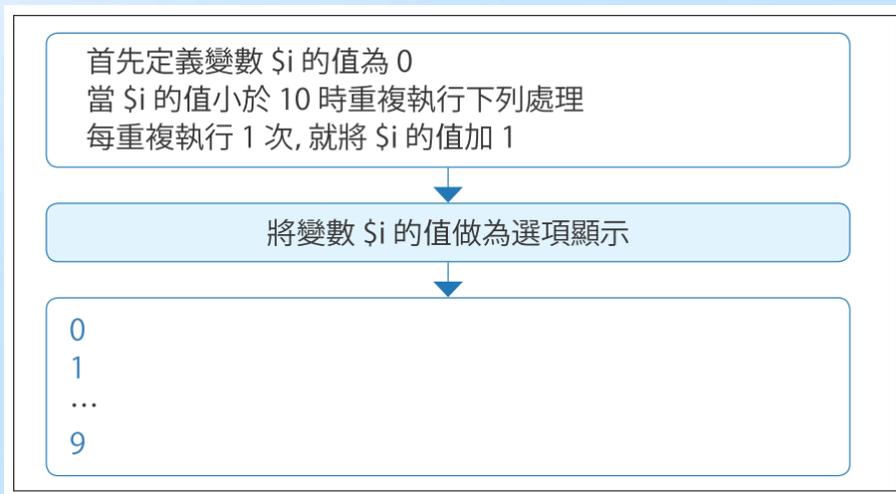
```

● for 迴圈

—在程式中，有時會需要將指定的處理重複執行指定次數，這種重複執行的機制就稱為迴圈



- 要利用程式重複執行這些處理，首先需要用到變數，在本例中使用的是變數 $\$i$
- 將變數代入後，處理流程如下



一變數 \$i 的變化與對應的處理





-for 迴圈是 PHP 中用來進行重複處理的語法之一，寫法如下

語法 for

```
for (開始處理; 條件式; 更新處理) {  
    重複執行的處理;  
}
```

—開始處理

- 此設定用來控制迴圈重複次數的變數起始值

—條件式

- 用來判斷是否重複執行的條件式

—更新處理

- 通常用來加減控制迴圈執行的變數值
- for 迴圈產生選項〔0〕到〔9〕時的程式流程：





● 比較算符

演算子	讀法	判斷結果為 TRUE 的情況
<	小於	左邊值小於右邊
>	大於	左邊值大於右邊
<=	小於等於	左邊值小於等於右邊（左邊值在右邊以下）
>=	大於等於	左邊值大於等於右邊（左邊值在右邊的以上）
==	相等	左右相等
!=	不相等（!為否定之意）	左右不相等

顯示選擇的結果



- 請參照下列程式，顯示出在下拉式選單選擇的結果，檔案儲存為 `chapter4\select-for-output.php`

```
select-for-output.php  
<?php require '../header.php';?>  
<?php  
echo $_REQUEST['count'], '個商品放入購物車。';  
?>  
<?php require '../footer.php';?>
```

▼ 顯示執行結果



5個商品放入購物車。

● 顯示 REQUEST 參數值

—本例中所取得的是 `<option>` 標籤的 `value` 值，也就是被選取的 `<option>` 標籤的值，將會被放入 `REQUEST` 參數中

```
select-for-input2.php PHP
<?php require '../header.php';?>
<p>請選擇訂購數量：</p>
<form action="select-for-output.php" method="post">
<select name="count">
<?php
for ($i=0; $i<10; $i++) {
    echo '<option value="' . $i . '>' . $i . '</option>';
}
?>
</select>
<p><input type="submit" value="確定"></p>
</form>
<?php require '../footer.php';?>
```

[PHP 迴圈 - For 迴圈](#)

● 利用 while 迴圈製作選項

–while 迴圈與 for 迴圈一樣是 PHP 中用來重複執行處理的語法

```

List  select-for-input3.php PHP
<?php require '../header.php';?>
<p>請選擇訂購數量：</p>
<form action="select-for-output.php" method="post">
<select name="count">
<?php
$i=0;
while ($i<10) {
    echo '<option value="' . $i . '>' . $i . '</option>';
    $i++;
}
?>
</select>
<p><input type="submit" value="確定"></p>
</form>
<?php require '../footer.php';?>
    
```

● while 迴圈

–while 迴圈與 for 迴圈都是 PHP 中用來重複執行處理的語法，其寫法如下

語法 while

```
while (條件式) {  
    重複執行的處理;  
}
```

語法 for

```
for (開始處理; 條件式; 更新處理) {  
    重複執行的處理;  
}
```


4-5 foreach迴圈 (一)



● 利用程式產生選項

- 首先建立下拉式選單及其選項

```
select-foreach-input.php PHP
<?php require '../header.php';?>
<p>請選擇密碼提示問題：</p>
<form action="select-foreach-output.php" method="post">
<select name="question">
<?php
$question=[
    '第一部看的電影',
    '第一隻寵物的名字',
    '第一部車的車款',
    '畢業的小學名稱',
    '小學時的導師姓名',
    '出生地的地名'
];
foreach ($question as $item) {
    echo '<option value="", $item, '>', $item, '</option>';
}
?>
</select>
<p>密碼提示問題的答案</p>
<p><input type="text" name="answer"></p>
<p><input type="submit" value="確定"></p>
</form>
<?php require '../footer.php';?>
```

問題與答案輸入畫面

請選擇密碼提示問題：

第一部看的電影 ▼

密碼提示問題的答案

確定

● 利用陣列產生選項

-以字串為值的問題選單選項，可以像下列這行敘述一樣，利用 `<option>` 標籤直接在程式裡寫明

```
<option value="第一部看的電影">第一部看的電影</option>
```

-在有多個選項時，陣列 (Array) 就能輕鬆地管理所有選項

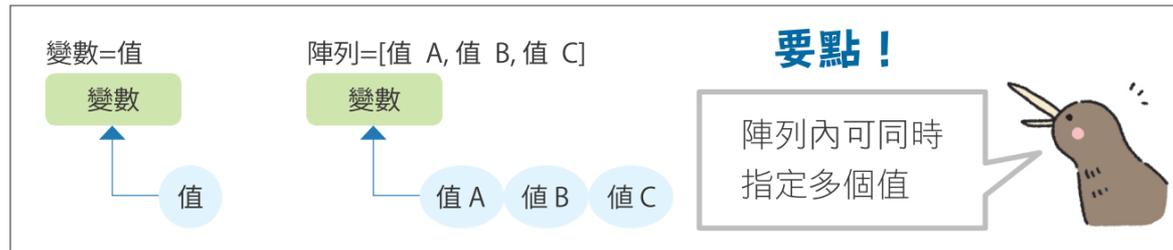
```
<option value="第一隻寵物的名字">第一隻寵物的名字</option>  
<option value="第一部車的車款">第一部車的車款</option>  
...
```

語法 指定陣列的值

```
陣列=[值 A,值 B,值 C, ... ];
```

—廣義來說陣列是變數的一種，與其它變數一樣，可以將值放入變數內

▼ 將值代入陣列中

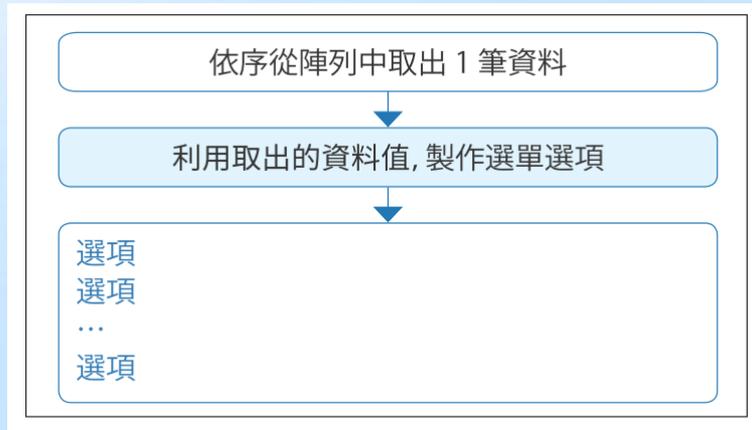


—程式中，如下所示

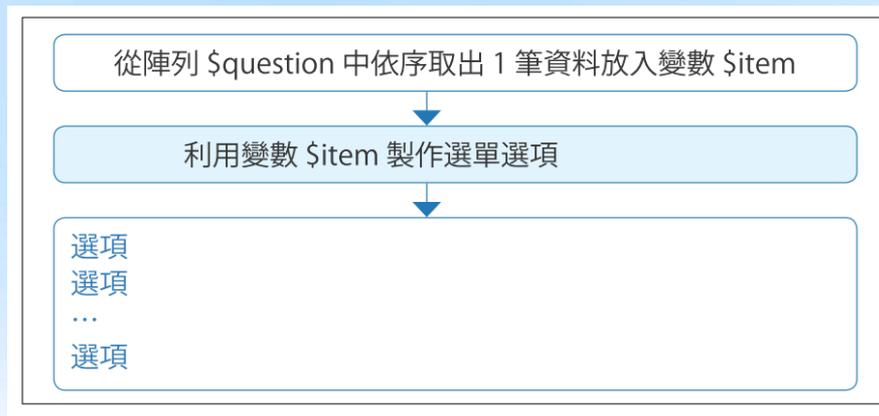
```
$question=[
    '第一部看的電影',
    '第一隻寵物的名字',
    '第一部車的車款',
    '畢業的小學名稱',
    '小學時的導師姓名',
    '出生地的地名'
];
```

● foreach 迴圈

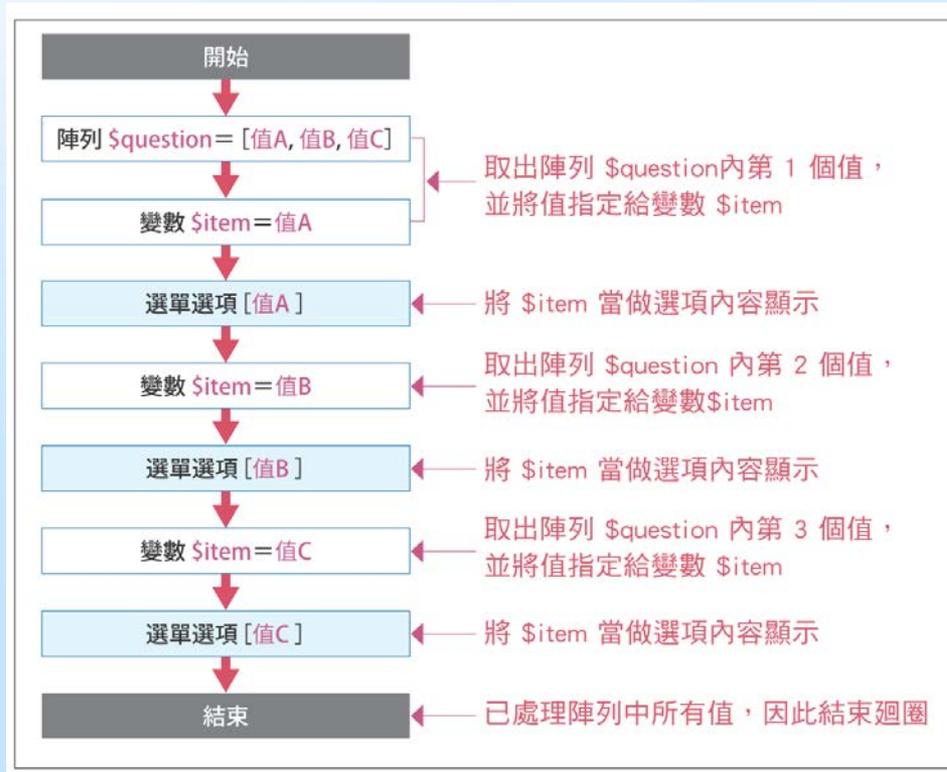
—利用陣列製作選單選項



—將陣列取出的值改放入變數中的流程如下



–foreach 迴圈可將陣列內的資料一筆一筆循序取出

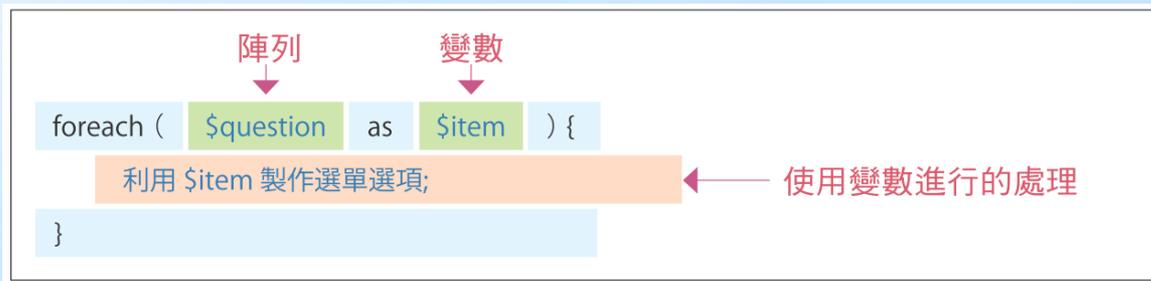


–foreach 迴圈的語法如下

```

    語法  foreach
    foreach(陣列 as 變數)
        使用變數進行的處理;
    }
  
```

–利用 `foreach` 迴圈將陣列 `$question` 內的資料值指定給變數 `$item`，再以此建立選單選項的程式示意圖如下



```

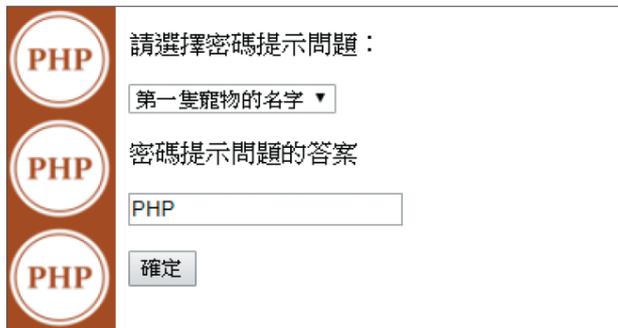
foreach ($question as $item) {
    echo '<option value="", $item, '>', $item, '</option>';
}
  
```

● 顯示問題與回答

— 參照下列程式，將使用者所選擇的問題及填寫的答案顯示在畫面

```
List select-foreach-output.php PHP
<?php require '../header.php';?>
<?php
echo '<p>您選擇的問題是：「', $_REQUEST['question'], '」</p>';
echo '<p>你輸入的答案是：「', $_REQUEST['answer'], '」</p>';
?>
<?php require '../footer.php';?>
```

▼ 選擇問題並輸入答案



請選擇密碼提示問題：

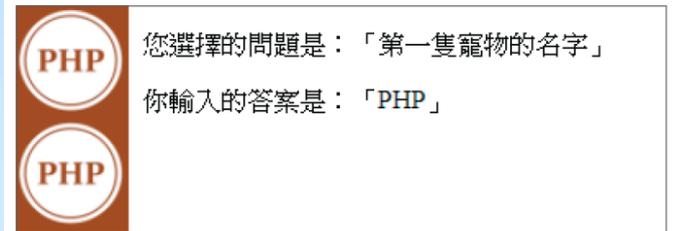
第一隻寵物的名字 ▼

密碼提示問題的答案

PHP

確定

▼ 顯示問題與答案



您選擇的問題是：「第一隻寵物的名字」

你輸入的答案是：「PHP」

4-6 foreach迴圈 (二)



● 手動製作選單選項

–以手動逐項輸入下拉式選單的每個選項，程式內容如下，檔案儲存為chapter4\store-input.php

```
store-input.php PHP
<?php require '../header.php';?>
<p>請選擇分店：</p>
<form action="store-output.php" method="post">
<select name="code">
<option value="100">新宿</option>
<option value="101">秋葉原</option>
<option value="102">上野</option>
<option value="200">橫濱</option>
<option value="201">川崎</option>
<option value="300">札幌</option>
<option value="400">仙台</option>
<option value="500">名古屋</option>
<option value="600">京都</option>
<option value="700">博多</option>
</select>
<p><input type="submit" value="確定"></p>
</form>
<?php require '../footer.php';?>
```

▼ 下拉式選單與確定按鈕

	請選擇分店：
	<input type="text" value="新宿"/>
	<input type="button" value="確定"/>

- 製作選單選項

一如前所述，下拉式選單的選項是以 `<option>` 標籤製作

```
<option value="100">新宿</option>
```

● 利用程式製作選單選項

— 參照下列程式，改以程式產生選單選項

```

List  store-input2.php PHP

<?php require '../header.php';?>
<p>請選擇分店：</p>
<form action="store-output.php" method="post">
<select name="code">
<?php
$store=[
    '新宿'=>100, '秋葉原'=>101, '上野'=>102, '橫濱'=>200, '川崎'=>201,
    '札幌'=>300, '仙台'=>400, '名古屋'=>500, '京都'=>600, '博多'=>700
];
foreach ($store as $key=>$value) {
    echo '<option value="", $value, "">', $key, '</option>';
}
?>
</select>
<p><input type="submit" value="確定"></p>
</form>
<?php require '../footer.php';?>

```

● 陣列的索引鍵與值

- 本例利用程式將下列「分店名稱」與「分店編號」，一組一組放入陣列
- 分店名稱（新宿）是索引鍵，分店編號（100）是值

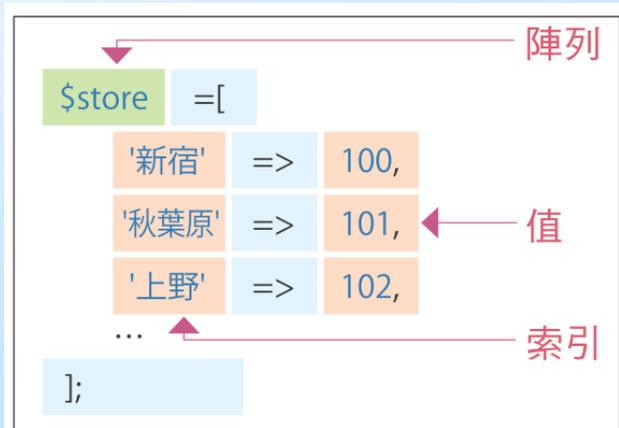
語法 將索引與值放入陣列

```
陣列=[  
    索引A => 值A,  
    索引B => 值B,  
    索引C => 值C,  
    ...  
];
```

語法

```
陣列=[索引A => 值A,索引B => 值B,索引C => 值C,...];
```

—在的程式中，將分店名稱與分店編號放入 \$store 陣列時，以分店名稱為索引，分店編號為值



—實際的程式如下。這裡為了縮減行數，將部份敘述合併在一行

```
$store=[
    '新宿'=>100, '秋葉原'=>101, '上野'=>102, '橫濱'=>200, '川崎'=>201,
    '札幌'=>300, '仙台'=>400, '名古屋'=>500, '京都'=>600, '博多'=>700
];
```


—這裡與前面介紹過的 `foreach` 迴圈語法的差異，在於 `as` 後面不是直接用變數，而是改為「索引鍵的變數 => 值的變數」

語法 利用 `foreach` 迴圈取出索引鍵與值

```
foreach (陣列 as 索引鍵的變數 => 值的變數) {
    以索引鍵與值進行的處理；
}
```

▼ 將成組的索引鍵與值放入變數





—將陣列改為 `$store`，索引鍵的變數改為 `$key`，值的變數改為 `$value` 後，實際的程式如下

```
foreach ($store as $key=>$value) {  
    echo '<option value="' . $value . '>' . $key . '</option>';  
}
```

—當程式從陣列中取出「新宿」和「100」的組合時

```
<option value="100">新宿</option>
```

● 顯示所選分店的分店編號

— 參照下列程式，撰寫程式顯示目前所選分店的對應編號，並將檔案儲存為chapter4\store-output.php

```
store-output.php
<?php require '../header.php';?>
<?php
echo '分店編號為', $_REQUEST['code'], '號';
?>
<?php require '../footer.php';?>
```

▼ 顯示分店編號



分店編號為100號

4-7 foreach迴圈 (三)



● 產生核取方塊

—參照下列程式，撰寫可一次產生所有核取方塊選項的程式，並將檔案儲存為 `chapter4\checks-input.php`

```
checks-input.php PHP
<?php require '../header.php';?>
<p>請選擇您有興趣的商品種類：</p>
<form action="checks-output.php" method="post">
<?php
$genre=['攝錄影機', '個人電腦', '鐘錶', '家電', '書籍', '文具用品', '食品'];
foreach ($genre as $item) {
    echo '<p>';
    echo '<input type="checkbox" name="genre[]" value="", $item, "">';
    echo $item;
    echo '</p>';
}
?>
<p><input type="submit" value="確定"></p>
</form>
<?php require '../footer.php';?>
```

核取方塊與（確定）按鈕

PHP 請選擇您有興趣的商品種類：

- 攝錄影機
- 個人電腦
- 鐘錶
- 家電
- 書籍
- 文具用品
- 食品

PHP 確定

● 用程式產生所有核取方塊

–將各核取方塊旁要顯示的文字全部放入陣列，本例要顯示的是商品種類，因此將陣列命名為 `$genre`

```
$genre=['攝錄影機', '個人電腦', '鐘錶', '家電', '書籍', '文具用品', '食品'];
```

–再從陣列 `$genre` 中逐筆取出值來製作核取方塊，此時取出的值要放到變數 `$item` 中

```
foreach ($genre as $item) {  
    ...  
}
```



● 核取方塊的屬性(可多選)

```
echo '<input type="checkbox" name="genre[]" value="", $item, ">';
```

—核取方塊敘述內容的例子如下

```
<input type="checkbox" name="genre[]" value="攝錄影機">
```

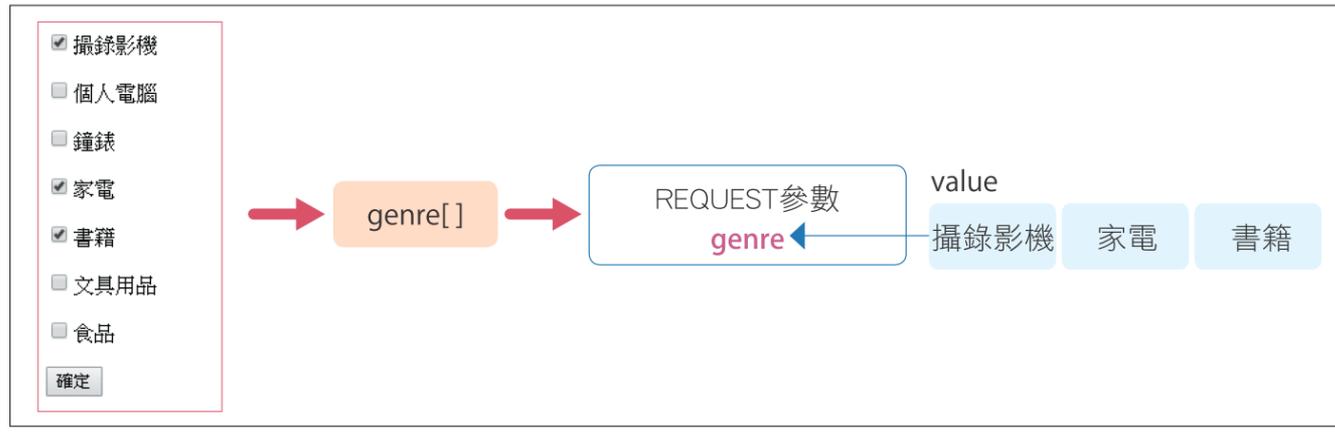
```
<input type="checkbox" name="genre[]" value="個人電腦">
```

```
<input type="checkbox" name="genre[]" value="鐘錶">
```

```
...
```

核取方塊是以 name 屬性值做為 REQUEST 參數名

將 name 屬性指定為陣列



● 取得所有勾選項目

— 參照下列程式，撰寫顯示所有目前勾選項目一覽表的程式，並將檔案儲存為chapter4/checks-output.php

```

List checks-output.php PHP
<?php require '../header.php';?>
<?php
foreach ($_REQUEST['genre'] as $item) {
    echo '<p>', $item, '</p>';
}
echo '以上商品的特惠情報電子報訂閱成功。';
?>
<?php require '../footer.php';?>

```

▼ 顯示所有勾選的項目

	個人電腦
	書籍
以上商品的特惠情報電子報訂閱成功。	



● 取得所有勾選的項目

- 被勾選的項目內容，會存放在核取方塊的 REQUEST 參數中
- 下列程式是將取出的項目放入變數 \$item 中

```
foreach ($_REQUEST['genre'] as $item) {  
    ...  
}
```

genre 這個參數是一個陣列，可以存放多個資料



● 未勾選任何項目時的處理方式

–利用 4-1 節所介紹的 if 條件與 isset 函數將程式修改如下，就可避免錯誤的發生

```
if (isset($_REQUEST['genre'])) {  
    foreach ($_REQUEST['genre'] as $item) {  
        ...  
    }  
}
```

php foreach 的普通用法和高級用法

do..while 迴圈



- do...while 迴圈首先會執行一次代碼塊，然後檢查條件，如果指定條件為真，則重複迴圈。

```
do {  
    要執行的代碼;  
} while (條件為真);
```

```
<?php  
$x=1;  
  
do {  
    echo "這個數字是 : $x <br>";  
    $x++;  
} while ($x<=5);  
?>
```

PHP do while Loop

Chapter 4 小結



- 本章介紹了 PHP 中判斷句（if、switch）與迴圈（for、while、foreach）的語法。在介紹迴圈的同時，也一併說明了配合陣列時的使用方法。此外，也介紹了核取方塊、單選鈕、下拉式選單等控制元件的操作方式。
- 下一章將針對 PHP 所提供的各種函式進行說明。