

資料庫專題

資應菁英班四甲

第 3 週 PHP 與表單

許雯綾
chiaohsu@nutc.edu.tw

參考資料：S 小學, [html php 表單傳送與接收資料](#)

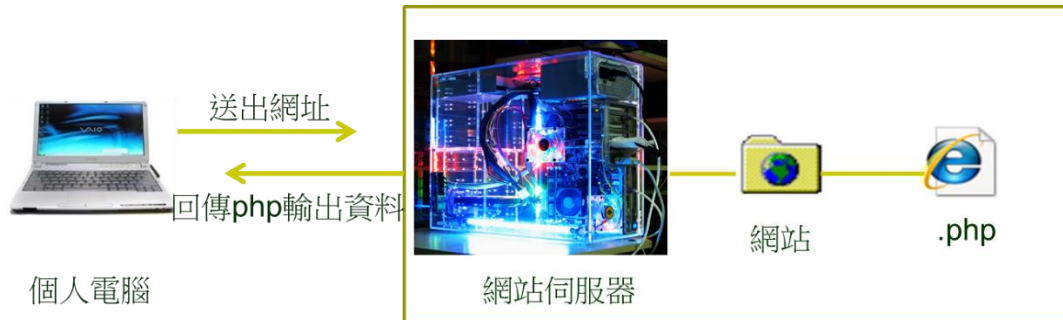
PHP 基礎語法與互動式網頁設計, 林業峻, 簡報

大 綱

- 3-1 PHP 程式簡介
- 3-2 PHP 基本指令
- 3-3 網頁表單 Form
- 3-4 資料傳送方式
- 3-5 表單元素

3-1 PHP 程式簡介

- PHP 是一種內嵌於 HTML 文件裡的程式語言
- 當伺服器送出網頁前，會先執行 PHP 的程式碼，而使用者端所能看到的只有 PHP 的輸出。
 - PHP 藉由輸出產生的 HTML 語法告知使用者的瀏覽器該做什麼
- PHP 程式必須存成 .php 檔

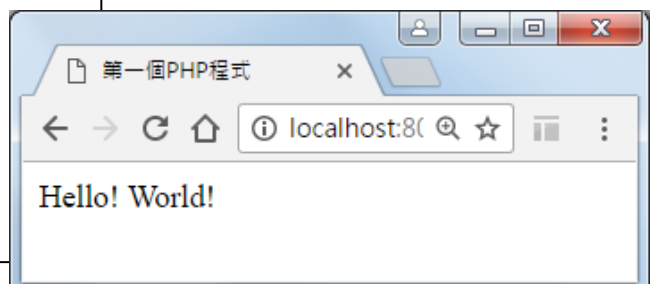


- 用來跟瀏覽網頁的人互動的
 - 輸出 (給瀏覽網頁的人看的)
 - 輸入 (接收瀏覽網頁的人填寫的資料)
- 在主機上執行用來做計算與控制的
 - 資料型態、變數
 - 運算表示法
 - 控制敘述：選擇控制與重覆控制
 - 陣列

3-1-1 PHP 程式寫法

- PHP 是一種內嵌於 HTML 文件裡的程式語言
- 為了能明辨出 PHP 的程式碼與 HTML，因此我們必須加上特製的標籤。

```
<html>
<head><title>第一個 PHP 程式</title></head>
<body>
    <?php
        echo "Hello! World!";
    ?>
</body>
</html>
```



- PHP 的程式碼使用 `<?php` 與 `?>` 將程式碼包起來
- 程式碼由上往下執行
- 每個程式指令敘述以 `;` 做為結束
- 使用 `echo "內容";` 將輸出產生的內容加到 HTML 中

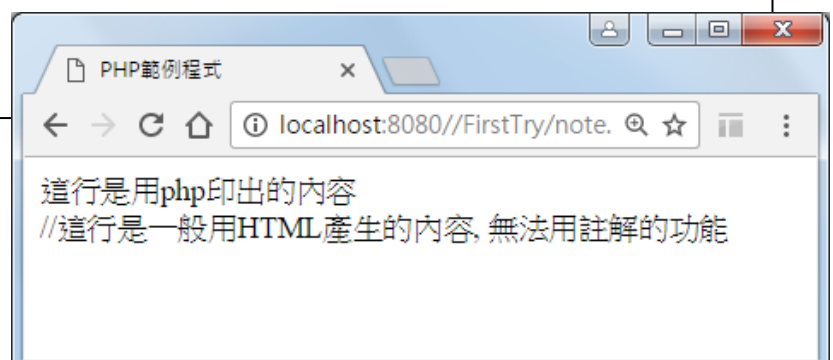
```
<html>
<head><title>PHP 範例程式</title></head>
<body>
    <?php
        echo "這行是用 php 印出的內容<br>";
    ?>
    這行是一般用 HTML 產生的內容<br>
</body>
</html>
```



3-1-2 PHP 註解的使用

- 寫給程式設計師看的，不會被執行
 - 單行註解: `//內容`
 - 多行註解: `/* 內容 */`

```
<html>
<head><title>PHP 範例程式</title></head>
<body>
    <?php
        echo "這行是用 php 印出的內容<br>";
        // echo "這行使用單行註解, 不會有任何動作<br>";
        /* echo "這行使用區域註解, 不會有任何動作<br>";
        echo "這行也在註解區域內, 不會有任何動作<br>"; */
    ?>
    //這行是一般用 HTML 產生的內容, 無法用註解的功能<br>
    <!-- 這行才是 HTML 的註解 -->
</body>
</html>
```



3-1-3 換行－HTML vs. PHP

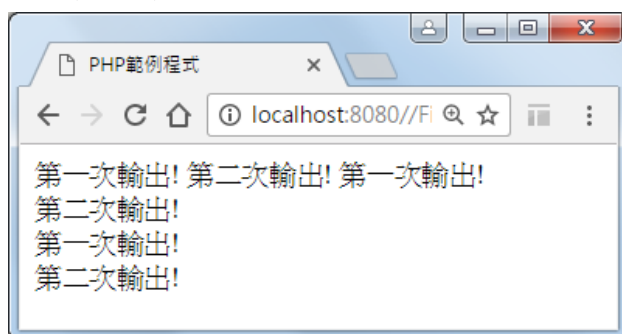
- PHP 的輸出換行：**\n**
 - 用途：在 HTML 原始檔中換一行
- HTML 的換行：**
**
 - 用途：顯示的網頁內容換一行

```
<html>
<head><title>PHP 範例程式</title></head>
<body>
    <?php
        echo "第一次輸出!\n";
        echo "第二次輸出!\n";

        echo "第一次輸出!<br>";
        echo "第二次輸出!<br>";

        echo "第一次輸出!<br>\n";
        echo "第二次輸出!<br>\n";
    ?>
</body>
</html>
```

【瀏覽結果】



【檢視原始碼】

```
1 <html>
2 <head><title>PHP範例程式</title></head>
3 <body>
4 第一次輸出!
5 第二次輸出!
6 第一次輸出!<br>第二次輸出!<br>第一次輸出!<br>
7 第二次輸出!<br>
8 </body>
9 </html>
10
```

3-1-4 echo 中的雙引號

- echo 之雙引號內如有要用到 HTML 中也要雙引號的地方，必須把它改用單引號或乾脆不使用引號
- 如果真的需要在 echo 中輸出雙引號，可以在雙引號前加反斜線: \"

```
<html>
<head><title>PHP 範例程式</title></head>
<body>
    <?php
        echo "<img src='http://otc.nutc.edu.tw/ezfiles/6/1006/img/492/abbb.gif' width='240'>
            <br>";
        echo "\"echo\"<br>";
    ?>
    
</body>
</html>
```

可以用雙引號
或單引號



3-2 PHP 基本指令

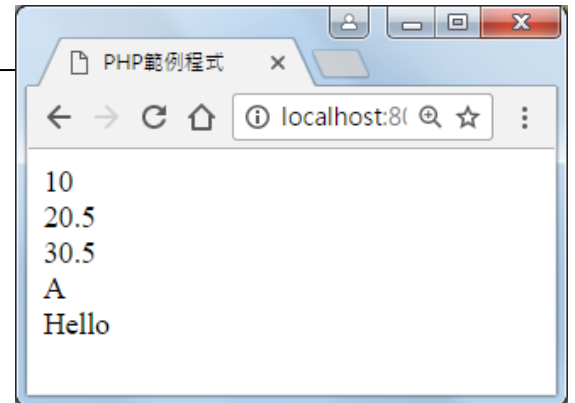
3-2-1 PHP 資料型態

- PHP 的資料型態主要分成：
 - 數字 (整數, 浮點數)
ex: 123、12.45、-100、-13.7
 - 文字 (字串, 字元)
ex: "How are you?"、'Joe Wang'、"\$"、'A'

3-2-1 PHP 變數

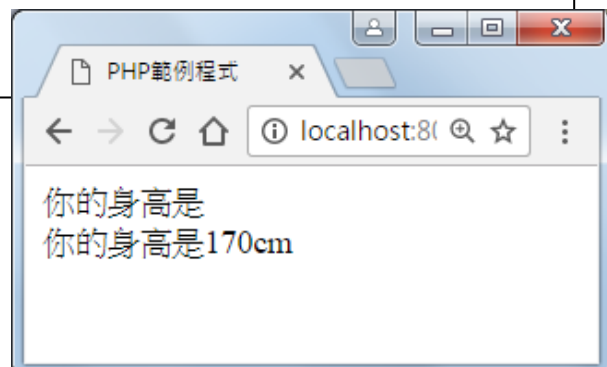
- 變數不分型態，變數名稱前要加 **\$** 號
- 變數名稱開頭必須是英文字母或底線
- 字元的大寫小寫所代表的意義不同
- 使用等號 **=** 做儲存(或稱指定)的動作

```
<html>
<head><title>PHP 範例程式</title></head>
<body>
    <?php
        $num1 = 10;
        $num2 = 20.5;
        $num3 = $num1 + $num2;
        $sch = 'A';
        $str = "Hello";
        echo "$num1<br>$num2<br>$num3<br>$sch<br>$str";
    ?>
</body>
</html>
```



- 當變數名稱跟你要輸出的內容黏在一起時，可以使用 **{ \$變數名稱 }** 把變數包起來

```
<html>
<head><title>PHP 範例程式</title></head>
<body>
    <?php
        $n = 170;
        echo "你的身高是$ncm<br>"; //沒有$ncm 這個變數
        echo "你的身高是{$n}cm<br>";
    ?>
</body>
</html>
```

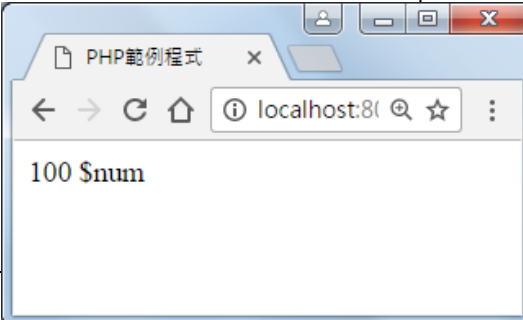


- 觀察下面 a 變數與 b 變數輸出結果 (變數時使用單/雙引號的差別)

```

<html>
<head><title>PHP 範例程式</title></head>
<body>
    <?php
        $num = 100;
        $a = "$num";
        $b = '$num';
        echo "$a $b";
    ?>
</body>
</html>

```



3-2-2 資料的運算

- 指定 (儲存): =
- 算數

負號 (-)
乘 (*)、除 (/)、取餘數 (%)
加 (+)、減 (-)
遞增(++), 遞減(--)
累加(+=), 累積(*=)

← 高優先順序

← 中優先順序

← 低優先順序

- 邏輯

運算子	功能敘述
&&	AND(及)
	OR(或)
!	NOT(反向)
xor	XOR(互斥)

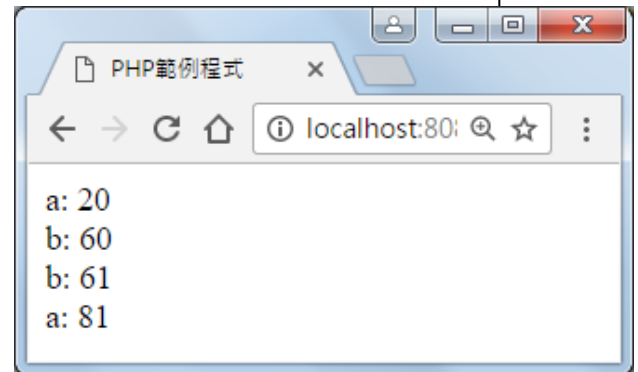
- 比較

運算子	功能敘述
<	小於
<=	小於等於
>	大於
>=	大於等於
==	等於
!=	不等於

```

<html>
<head><title>PHP 範例程式</title></head>
<body>
    <?php
        $a = 10+2*5;
        echo "a: $a<br>";
        $b = (10+2)*5;
        echo "b: $b<br>";
        $b++;
        echo "b: $b<br>";
        $a+=$b;
        echo "a: $a<br>";
    ?>
</body>
</html>

```



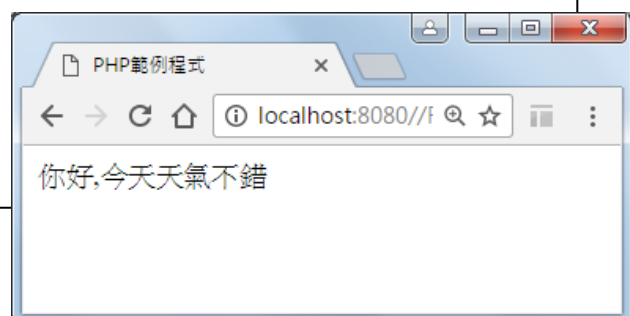
3-2-3 文字的連結

- 用來連結兩段文字內容: “.”

```

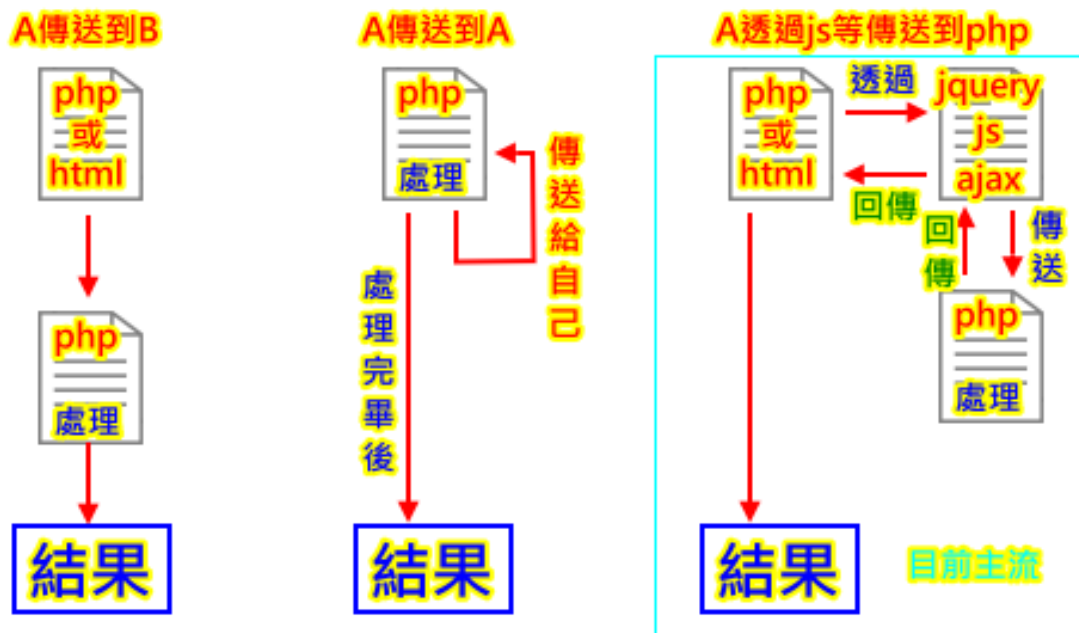
<html>
<head><title>PHP 範例程式</title></head>
<body>
    <?php
        $s1 = "你好, ";
        $s2 = "今天天氣不錯";
        $s3 = $s1 . $s2;
        echo "$s3";
    ?>
</body>
</html>

```



3-3 網頁表單 Form

- 表單 (Forms) 是 HTML 的一個重要標籤，可以提供輸入介面，讓使用者輸入資料，然後將資料傳回 Web 伺服器，以便做進一步的處理。
- 其應用如：Web 搜尋、網路票選、線上問卷、會員登入、線上訂購...等。
- 表單除了有呈現資訊的功能，更重要的是，它包含了數種表單元素 (Form Elements)，可以讓使用者進行資料填寫或選取的功能，並將使用者所填寫的資料送到伺服器端，進行必要的處理。因此我們可以說，表單就是客戶端和伺服器進行資訊溝通的第一個門面。
- 網頁 html 及 php 表單傳送與接收的方法，基本上有三種傳遞接收表單的模式：



- 標準的 HTML 表單是由 <form> 標籤開始，結尾使用 </form> 標籤，兩個標籤之間就是放置表單元素的地方，簡單的表單結構如下：

```
<form action="送出目的地" method="資料傳送方式" name="表單標記名稱" id="表單標記名稱">
... 表單內容...
</form>
```

- **action**：form 會將使用者填完的資料，傳送至 action 所設定的資料傳送目的地，
例如：action=" <http://127.0.0.1/index.asp>"
- **name** 與 **id** 基本意義都是為了標記元素的名稱。
 - **id** 常用於 Client 端 HTML 元素的 Identity(識別值)。
 - 而表單元素(form input textarea select)與框架元素(iframe frame) 則用 **name** 來識別。
- 這些元素都與表單提交有關，在表單的接收頁面只接收有 name 的元素，id 的元素通過表單是接收不到值的。
- 另外像 HTML 元素 **input type='radio'** 分組，當 radio button 控制元件在同一個分組時，同一時間只能選中一個 radio，這個分組就是根據相同的 Name 屬性來實現的。

3-4 資料傳送方式

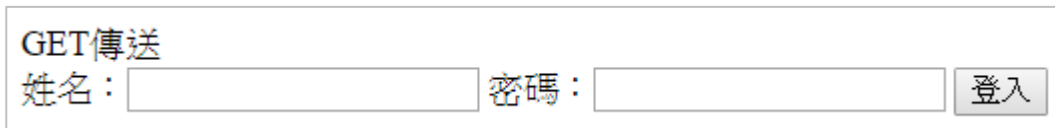
- PHP 接收 form 表單變數傳遞有兩種方式，分別為 POST 與 GET 這兩種方法，兩者的主要差異在傳遞資料的過程，PHP 是否會將 HTML 表單欄位的參數及參數值顯示於網址(url)上，POST method 不會顯示參數，而 GET method 則會顯示。
- HTML 表單的 method 參數與 PHP 的接收方式必須是對應的，例如表單 method 設定用 post 傳遞資料，PHP 必須使用 **\$_POST** 來接收，如果表單的 method 設定用 get 方式傳遞資料，則 PHP 就必須使用 **\$_GET** 來接收

【範例一】透過 GET 接收 form 表單變數資料

首先設計一個 Form 表單 (formA.php)

```
<html>
<head> <meta charset="utf-8"> <title>登入</title>
</head>
<body>
<form action="getA.php" method="get">
  GET 傳送<br>
  姓名：<input name="user_name" type="text">
  密碼：<input name="user_pass" type="password">
  <input name="submit" type="submit" value="登入">
</form>
</body>
</html>
```

【呈現結果】



表單內使用了 method="get" 這樣的宣告，代表當使用者按下送出表單之後，整個表單的欄位資料都會透過 get 的方式傳遞出去給 getA.php 這支程式，而 getA.php 將會使用 **\$_get** 來接收資料

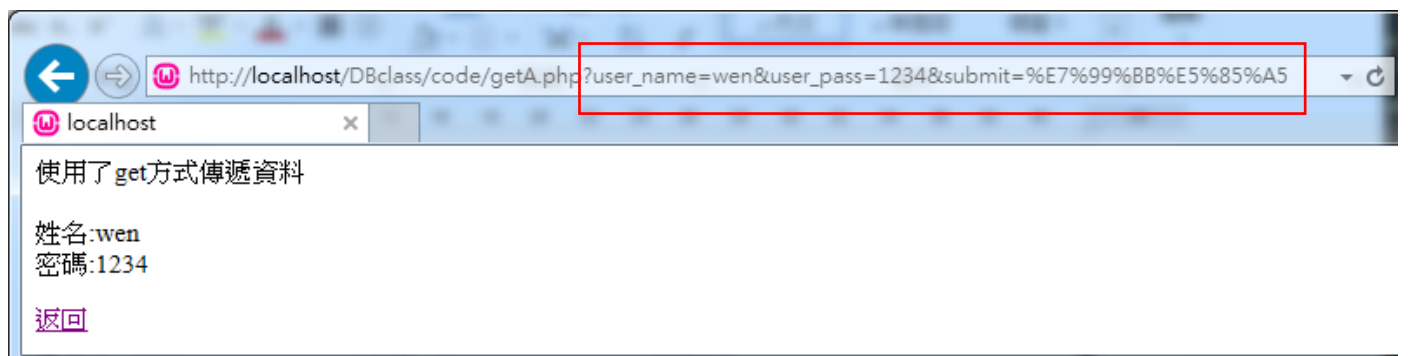
接著設計一個接收 formA.php 表單資訊的網頁 (getA.php)

```
<?php
    echo '使用了 get 方式傳遞資料<br /><br />';
    if ( $_GET['user_name'] != '' )
        { echo '姓名:' . $_GET['user_name'] . '<br />'; }
    else
        { echo '無名氏<br />'; }

    if( $_GET['user_pass'] != '' )
        { echo '密碼:' . $_GET['user_pass'] . '<br />'; }
    else
        { echo '無密碼<br />'; }

    echo '<p><a href="formA.php">返回</a></p>';
?>
```

【呈現結果】

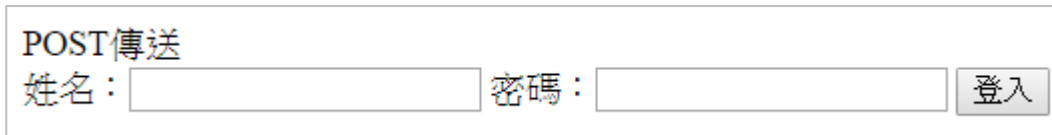


【範例二】透過 POST 接收 form 表單變數資料

首先設計一個 Form 表單 (formB.php)

```
<html>
<head> <meta charset="utf-8"> <title>登入</title>
</head>
<body>
<form action="postB.php" method="post">
  POST 傳送<br>
  姓名：<input name="user_name" type="text">
  密碼：<input name="user_pass" type="password">
  <input name="submit" type="submit" value="登入">
</form>
</body>
</html>
```

【呈現結果】



表單內使用了 method="post" 這樣的宣告，代表當使用者按下送出表單之後，整個表單的欄位資料都會透過 POST 的方式傳遞出去給 postB.php 這支程式，而 postB.php 將會使用 \$_POST 來接收資料

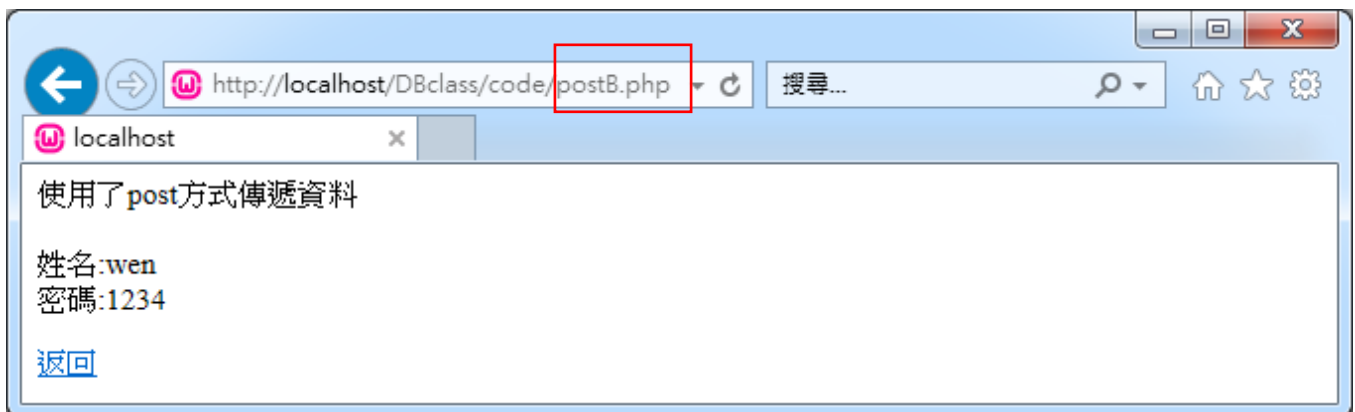
接著設計一個接收 formB.php 表單資訊的網頁 (postB.php)

```
<?php
echo '使用了 post 方式傳遞資料<br /><br />';
if( !empty( $_POST['user_name'] ) )
    { echo '姓名:'. $_POST['user_name'] . '<br />'; }
else
    { echo '無名氏<br />'; }

if( !empty($_POST['user_pass']) )
    { echo '密碼:'. $_POST['user_pass'] . '<br />'; }
else
    { echo '無密碼<br />'; }

echo '<p><a href="formB.html">返回</a></p>';
?>
```

【呈現結果】



3-5 表單元素

■ 表單內的元素是用來設計使用者填寫資料的方式，常見的 HTML form 表單元素說明如下：

(1) 表單按鈕 HTML button

表單按鈕，用來送出或重設所填寫的表單內容。

```
<form>  
<input type="submit" name="button" id="btnSubmit" value="送出資料">  
<input type="reset" name="button2" id="btnReset" value="重新填寫">  
</form>
```

【呈現結果】

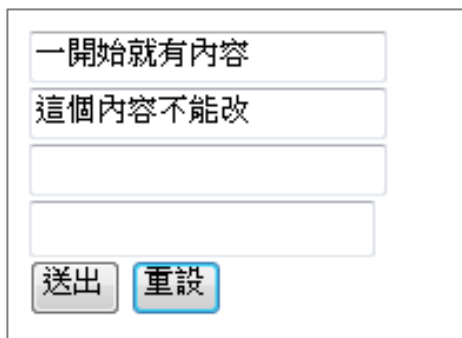


(2) 文字輸入欄位 <input type="text"/>/密碼輸入欄位 <input type="password">

input 標籤會在頁面上顯示一個輸入框，用來填入資料，例如會員登入的帳號密碼或是線上報名表單的姓名、電話、地址之類的資訊。

```
<html>
<head> <title> 文字輸入欄位 </title> </head>
<body>
<form method="POST" action="action1.php">
  <input type="text" name="T1" size="20" value="一開始就有內容"> <br>
  <input type="text" name="T2" size="20" value="這個內容不能改" readonly> <br>
  <input type="text" name="T3" size="20"> <br>
  <input type="password" name="T4" size="20" maxlength="4"> <br>
  <input type="submit" value="送出" name="B1">
  <input type="reset" value="重設" name="B2">
</form>
</body>
</html>
```

【呈現結果】



The screenshot shows a web form with four text input fields stacked vertically. The first field contains the text "一開始就有內容". The second field contains the text "這個內容不能改". The third and fourth fields are empty. Below the input fields are two buttons: "送出" (Submit) and "重設" (Reset).

【action1.php】

```
<html>
<head> <title> PHP 範例程式 </title> </head>
<body>
<?php
$text1 = $_POST['T1'];
$text2 = $_POST['T2'];
$text3 = $_POST['T3'];
$text4 = $_POST['T4'];
echo "文字方塊 1 : $text1<br>";
echo "文字方塊 2 : $text2<br>";
echo "文字方塊 3 : $text3<br>";
echo "文字方塊 4 : $text4<br>";
?>
</body>
</html>
```

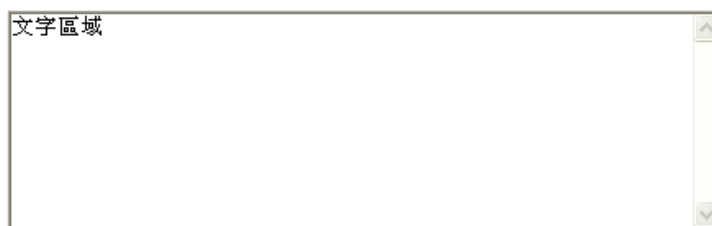
(3) 長文字輸入欄位 <textarea></textarea>

文字輸入框 <textarea></textarea> 與 <input type="text"> 不同的是可以多行輸入，通常用來填入比較多的資訊，當輸入的文字長度較長的時候，<textarea></textarea> 可以自動產生 scrollbar。

```
<html>
<head> <title>PHP 範例程式</title> </head>
<body>
<form method="POST" action="action2.php">
<textarea rows="8" name="S1" cols="59">
文字區域
</textarea>
<br>
<input type="submit" value="送出" name="B1">
</form>
</body>
</html>
```

- ▶ rows: 行數
- ▶ cols: 字元寬度

【呈現結果】



【action2.php】

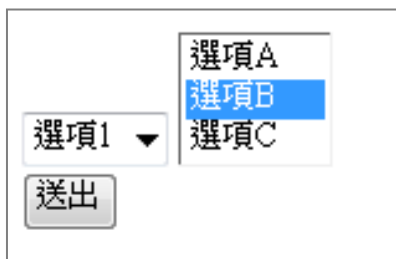
```
<html>
<head> <title>PHP 範例程式</title> </head>
<body>
<?php
$textarea1 = nl2br($_POST['S1']);
echo "文字區域：<br>$textarea1";
?>
</body>
</html>
```

(4) 下拉選單/清單方塊 <select><option></option></select>

<select></select> 呈現的是 HTML form 表單的下拉式選單，例如日期選擇或是購物的商品規格都是常見的應用，其中 <option> 是選項，可以設定多個下拉選單的選項，每個選項給不同的值，表單送出後讓程式取得的選擇的項目。

```
<html>
<head> <title>PHP 範例程式</title></head>
<body>
<form method="POST" action="action3.php">
<select size="1" name="D1">
    <option selected>選項 1</option>
    <option>選項 2</option>
    <option>選項 3</option>
</select>
<select size="3" name="D2">
    <option>選項 A</option>
    <option selected>選項 B</option>
    <option>選項 C</option>
</select>
<br>
<input type="submit" value="送出" name="B1">
</form>
</body>
</html>
```

【呈現結果】



【action3.php】

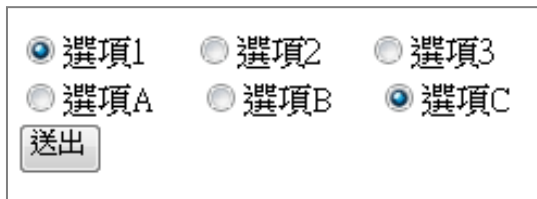
```
<html>
<head> <title>PHP 範例程式</title></head>
<body>
<?php
$select1 = $_POST['D1'];
$select2 = $_POST['D2'];
echo "第一組選項: $select1<br>";
echo "第二組選項: $select2<br>";
?>
</body>
</html>
```


(5) 選項按鈕 Radio Buttons <input type="radio">

Radio Buttons <input type="radio"> 通常會做成一組的選項，讓使用者僅能選擇其中一項，做成一組的技巧是設定整組 <input type="radio"> 的 **name** 都是一樣的，這樣 form 就會自動判斷為一整組。

```
<html>
<head> <title>PHP 範例程式</title> </head>
<body>
<form method="POST" action="action4.php">
<input type="radio" value="V1" name="R1" checked> 選項 1
<input type="radio" value="V2" name="R1"> 選項 2
<input type="radio" value="V3" name="R1"> 選項 3
<br>
<input type="radio" value="A" name="R2"> 選項 A
<input type="radio" value="B" name="R2"> 選項 B
<input type="radio" value="C" name="R2" checked> 選項 C
<br>
<input type="submit" value="送出" name="B1">
</form>
</body>
</html>
```

【呈現結果】



【action4.php】

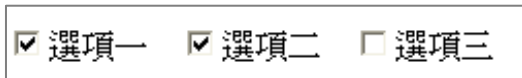
```
<html>
<head> <title>PHP 範例程式</title> </head>
<body>
<?php
$radio1 = $_POST['R1'];
$radio2 = $_POST['R2'];
echo "第一組選項: $radio1<br>";
echo "第二組選項: $radio2<br>";
?>
</body>
</html>
```

(6) 核取方塊 Checkboxes <input type="checkbox">

Checkboxes 通常用來製作多選項的表單，例如使用者在填寫個人興趣的時候，可以勾選許多項目，就是用 Checkboxes 做到的。

```
<html>
<head> <title>PHP 範例程式</title> </head>
<body>
<form method="POST" action="action5.php">
<input type="checkbox" name="C1" value="C1" checked> 選項一
<input type="checkbox" name="C2" value="C2" checked> 選項二
<input type="checkbox" name="C3" value="C3"> 選項三<br>
<input type="submit" value="送出" name="B1">
</form>
</body>
</html>
```

【呈現結果】



【action5.php】

```
<html>
<head> <title>PHP 範例程式</title> </head>
<body>
<?php
if(!empty($_POST['C1']))
    echo "選項一: ".$_POST['C1']."<br>";
if(!empty($_POST['C2']))
    echo "選項二: ".$_POST['C2']."<br>";
if(!empty($_POST['C3']))
    echo "選項三: ".$_POST['C3']."<br>";
</body>
</html>
```

(7) 隱藏欄位 input hidden

隱藏欄位用來傳遞一些表單參數，並不會顯示在網頁上。

```
<html>
<head> <title>PHP 範例程式</title> </head>
<body>
```

```
<form method="POST" action="action6.php">
<input type="hidden" name="H1" value="Hello">
<input type="submit" value="送出" name="B1">
</form>
</body>
</html>
```

【呈現結果】



【action6.php】

```
<html>
<head> <title>PHP 範例程式</title> </head>
<body>
<?php
$hidden1 = $_POST['H1'];
echo "隱藏元件：$hidden1";
</body>
</html>
```