

第二章 PHP 語法

本章將依序介紹 PHP 的基本語法、決策敘述、迴圈敘述、函式、陣列、檔案、與輸出入物件。

2_1 基本語法

本節將依序介紹 PHP 的安裝與執行、運算子、變數與資料型別。

2_1_1 PHP 軟體安裝

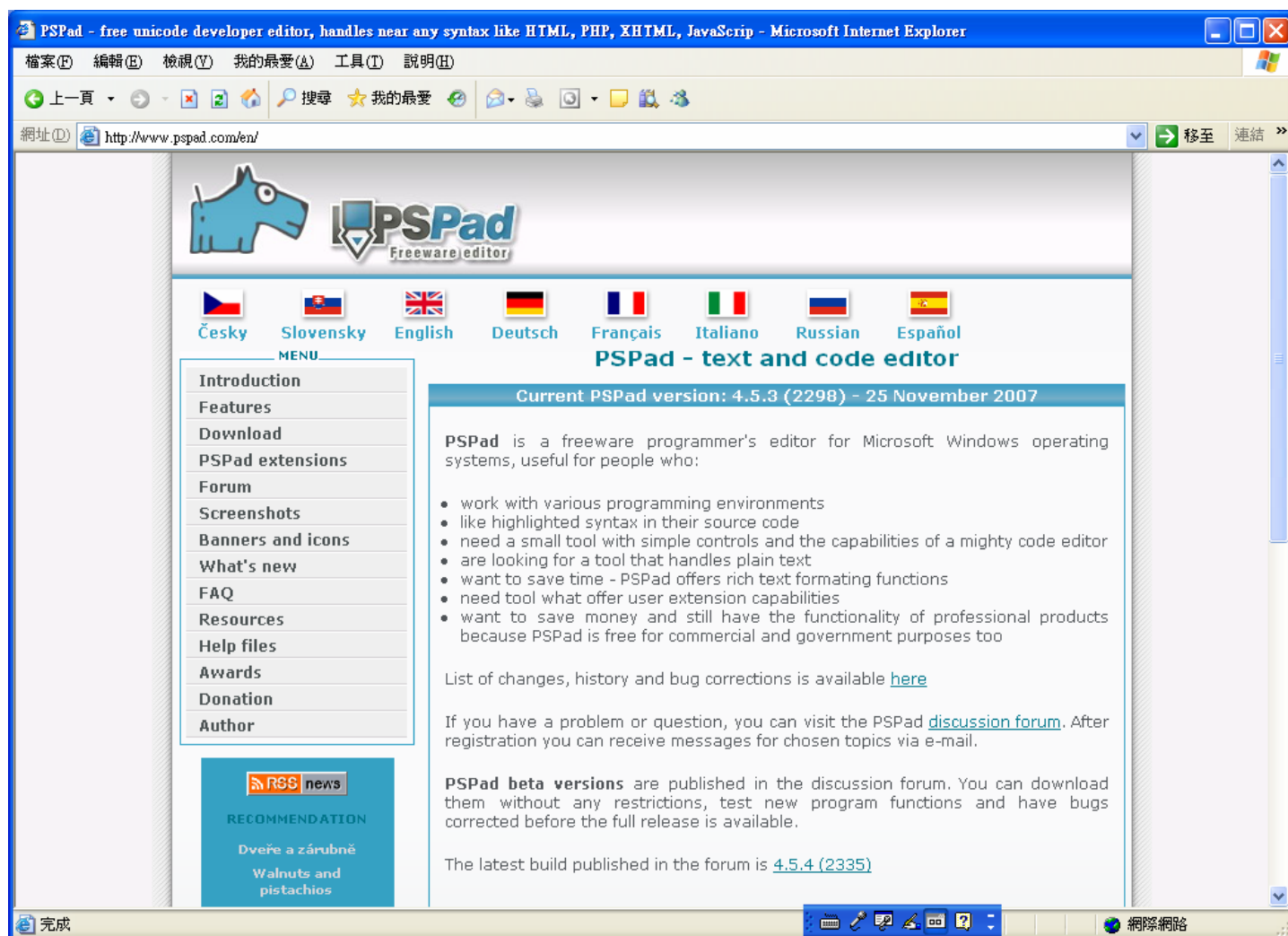
AppServ

PHP檔案不像HTML檔案，只要有瀏覽器就可執行，他必須先安裝伺服器軟體，才能預覽其結果。在Windows系統中，可以執行PHP的伺服器軟體有微軟的IIS與免費公開下載的Apache；在Linux系統中，則有Apache。其次，開發PHP程式要有PHP模組與MySQL資料庫軟體，操作MySQL則需phpMyadmin軟體，也就是共需安裝四次軟體，分別是IIS或Apache、PHP、MySQL與phpMyadmin。還有，此四個軟體依序安裝後，還要設定以上軟體的關連，以上所有操作當然令人昏頭轉向，所幸有一個網站

AppSerNetwork(www.appservnetwork.com)非常用心，已經將以上四個軟體整合在一起，只要下載其整合程式，即可完成以上四種軟體的安裝，所以本書採用並推薦此一軟體，在此由衷感激與致謝。

PSPad

開發 PHP 軟體可用一般的記事本編輯、存檔、上傳至伺服器，再至伺服器執行程式，若有錯誤則要回記事本修改。所幸有一網站 PSPad(www.pspad.com)非常用心，製作一整合編輯環境，在此編輯環境即可編輯、存檔、執行與修改 PHP 程式，所以本書推薦此種編輯方式，在此由衷感激與致謝。



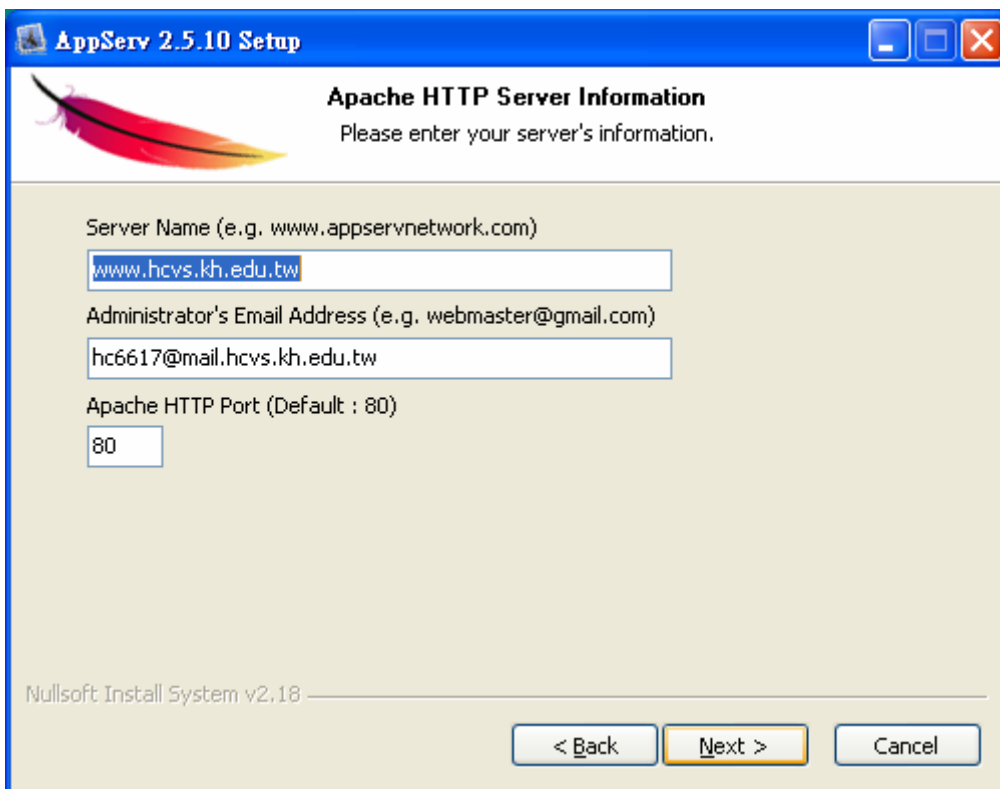
範例 2_1_1a 示範 Appserv 的安裝。

操作步驟

- 1、請自行自 AppServ 網站 <http://www.appservnetwork.com/> 下載整合軟體，或暫行使用本書範例光碟所附的檔案。
- 2、執行安裝程式(本書是 appserv-win32-2.5.10.exe)。
- 3、依照指示，使用預設值並按“下一步”，即可完成安裝。若出現以下畫面，請點選“解除封鎖”。

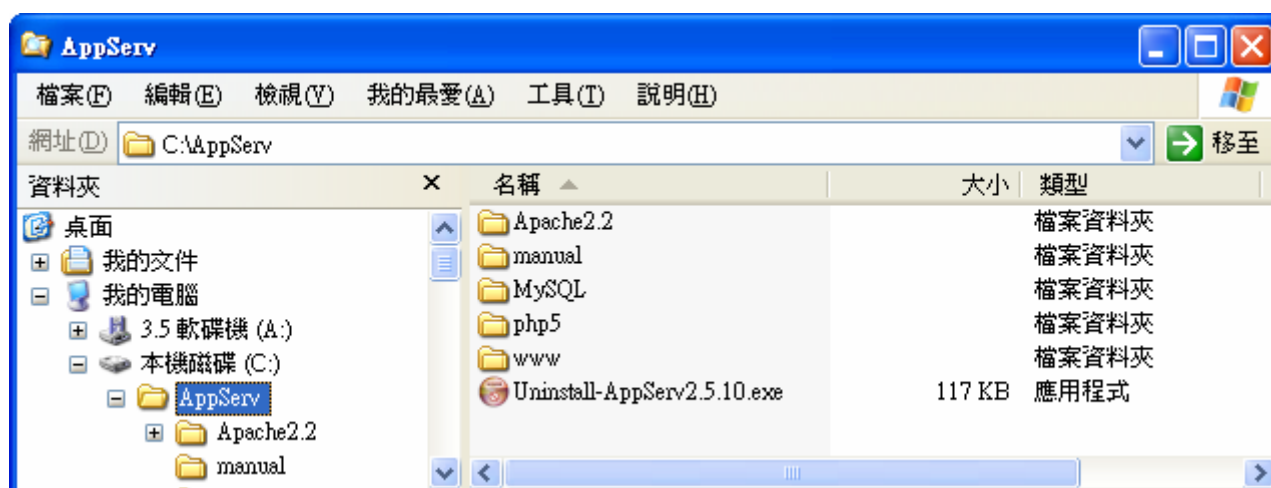


- 4、輸入伺服器名稱與 Email 帳號。此畫面請依照實際情形輸入，若沒有連線伺服器則先行留白即可。(反正練習階段都是在自己的電腦執行程式，所以與伺服器無關)



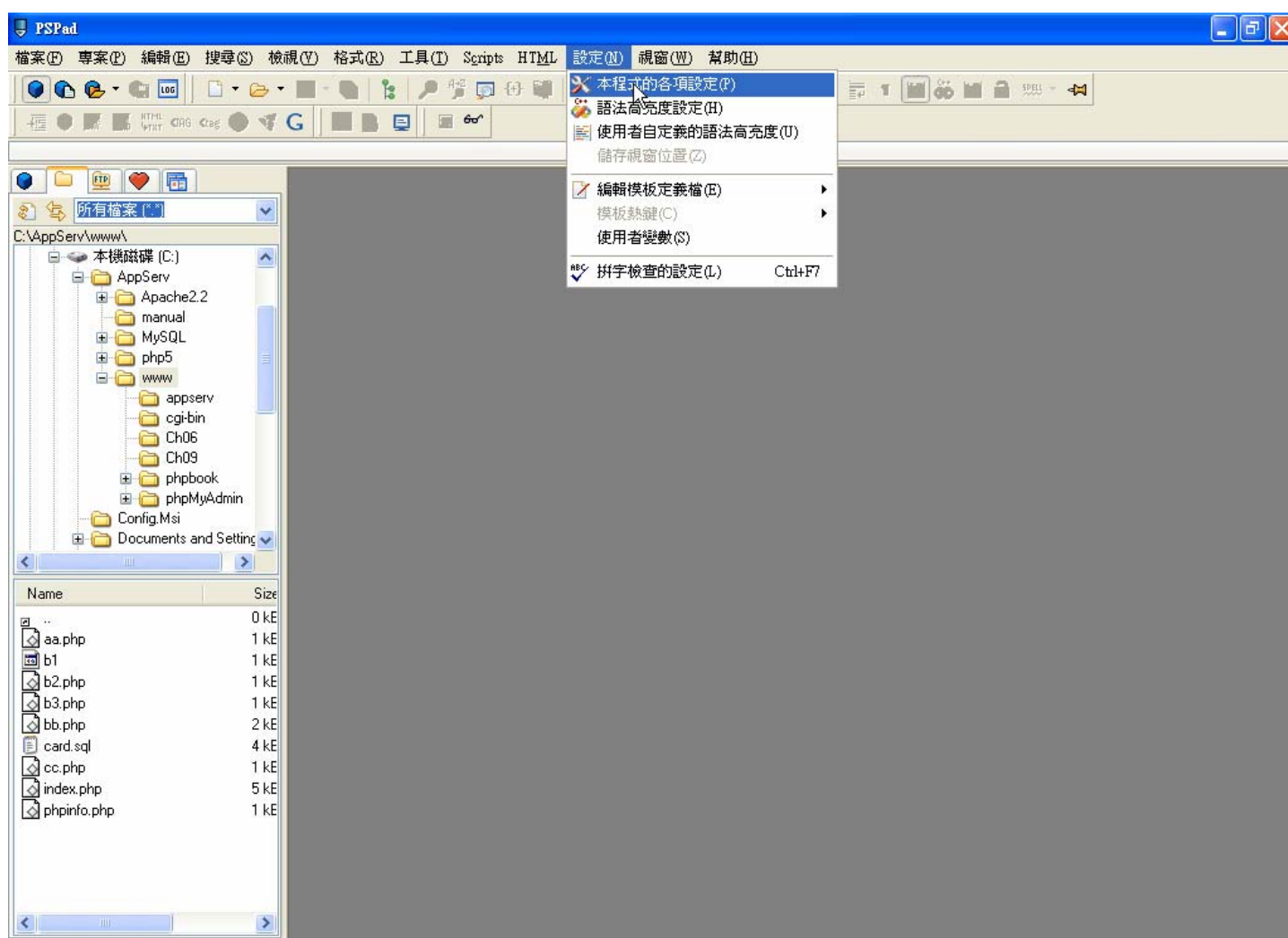
- 5、輸入資料庫密碼。請輸入一個密碼，並記起來以便操作 MySQL 資料庫。

- 6、完成以上操作，電腦會將以上四個軟體安裝在 AppServ 資料夾，如下圖所示：



範例 2_1_1b 示範 PSPad 的安裝。

- 1、請自行自 PSPad 網站 [http:// www.pspad.com](http://www.pspad.com) 下載整合軟體，或暫行使用本書範例光碟所附的檔案。
- 2、執行安裝程式(本書是 pspad453inst_en.exe)，請依照指示，按”Next”即可完成安裝。
- 3、開啓 PSPad 整合編輯視窗。PSPad 整合編輯視窗如下圖：

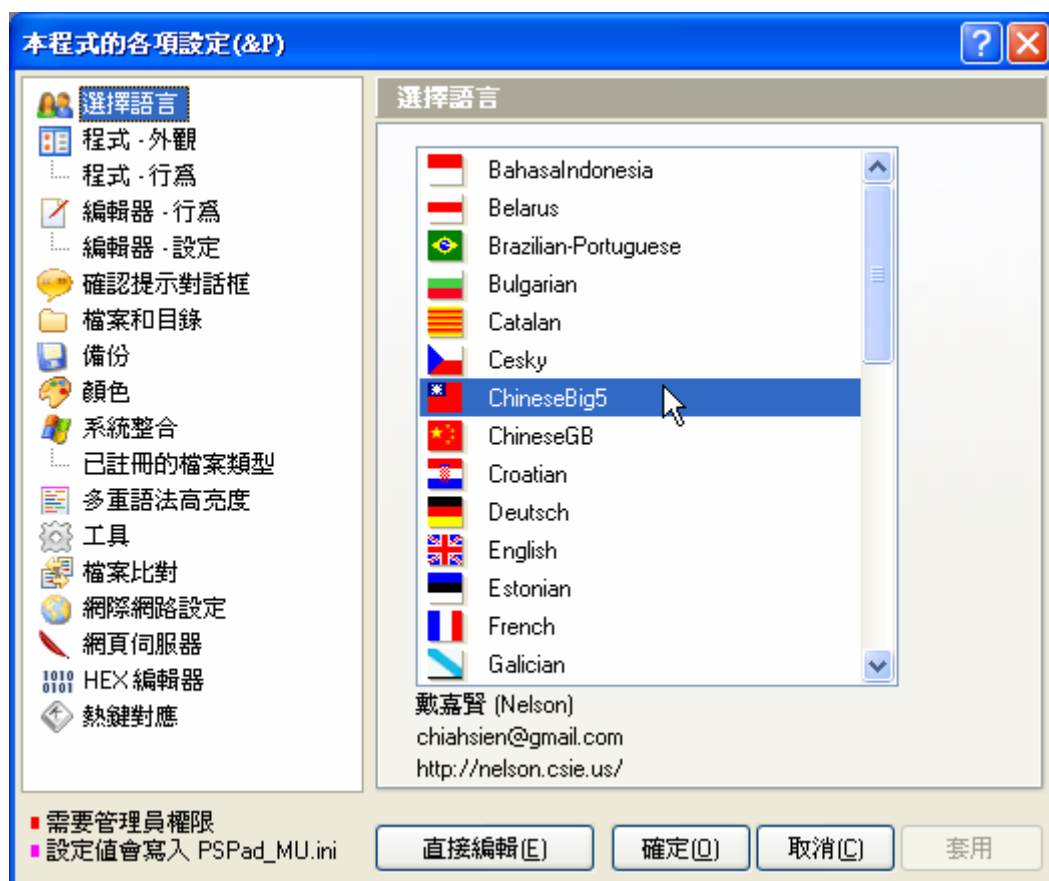


- 4、設定網頁伺服器。

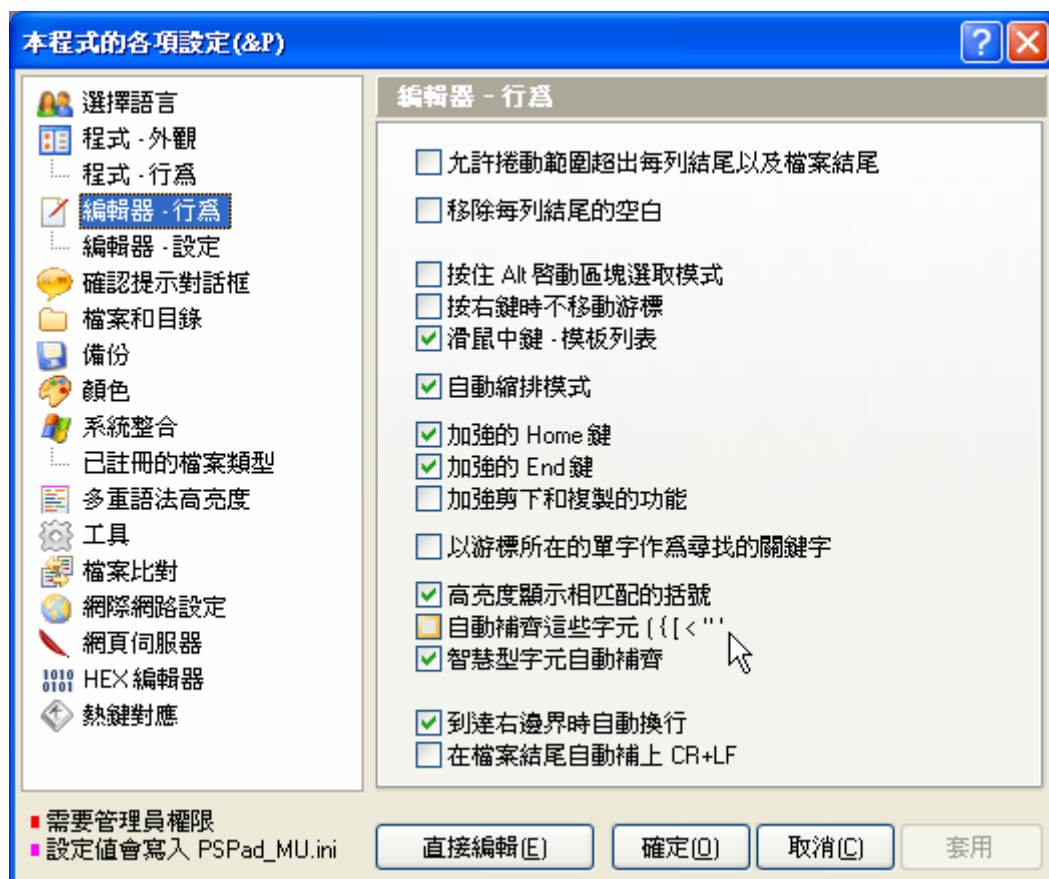
因為 AppServ 預設網頁存放的根目錄為 C:\AppServ\www，本機伺服器名稱為 localhost，所以點選功能表的“設定”/“本程式的各項設定”，並設定如下圖：



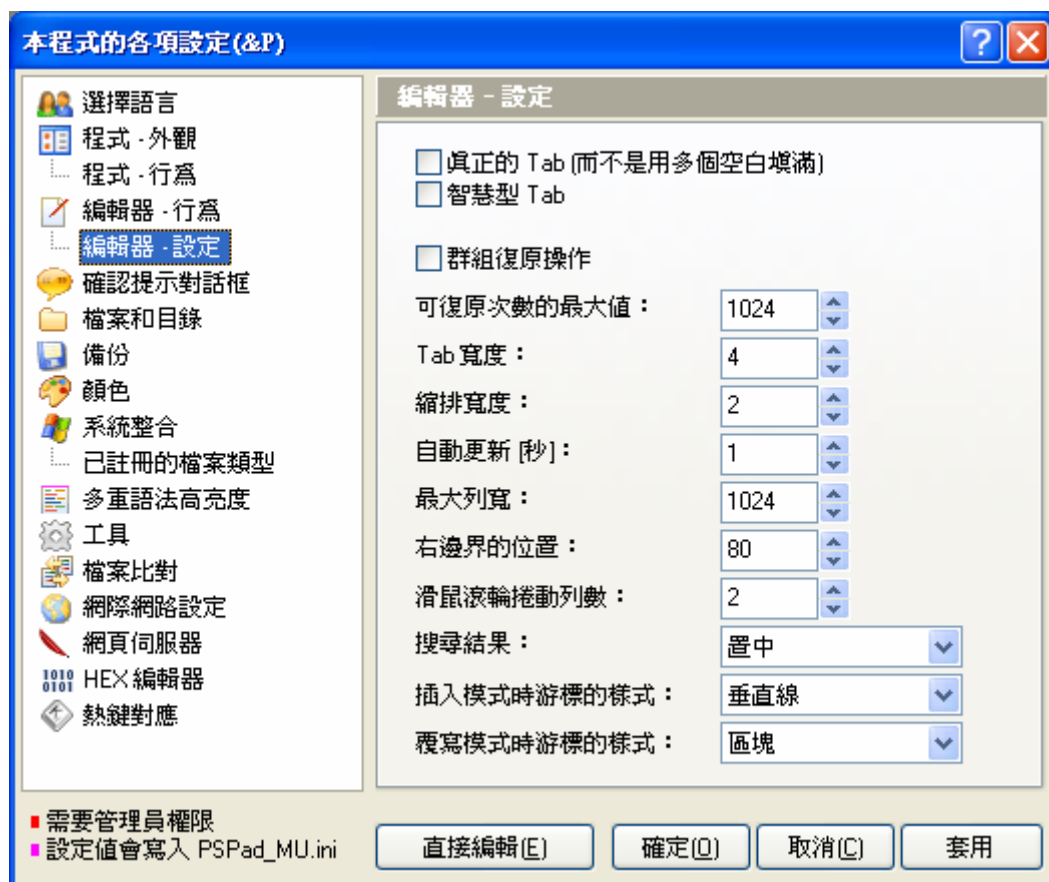
5、語言設定。本例設定如下：



6、編輯器行爲設定。PSPad 預設自動補齊右邊的“}”等符號，筆者覺得不順，所以可修改如下圖：



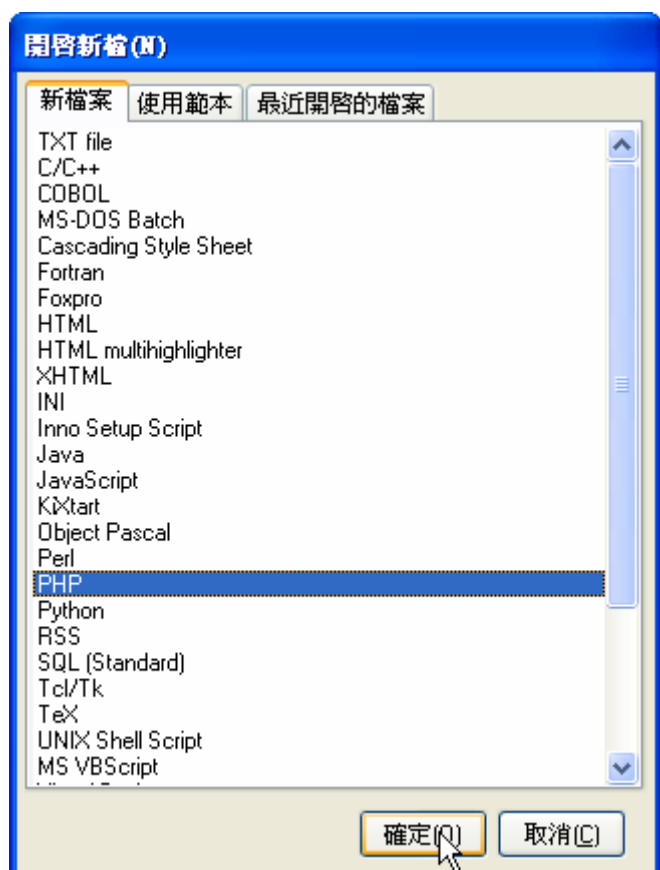
7、設定 Tab 與縮排寬度。本例設定如下圖：



範例 2_1_1c 示範程式的編輯、與執行。

1、開新 PHP 檔案。

點選”檔案”/”開啓新檔”，畫面如下圖，並點選 PHP。

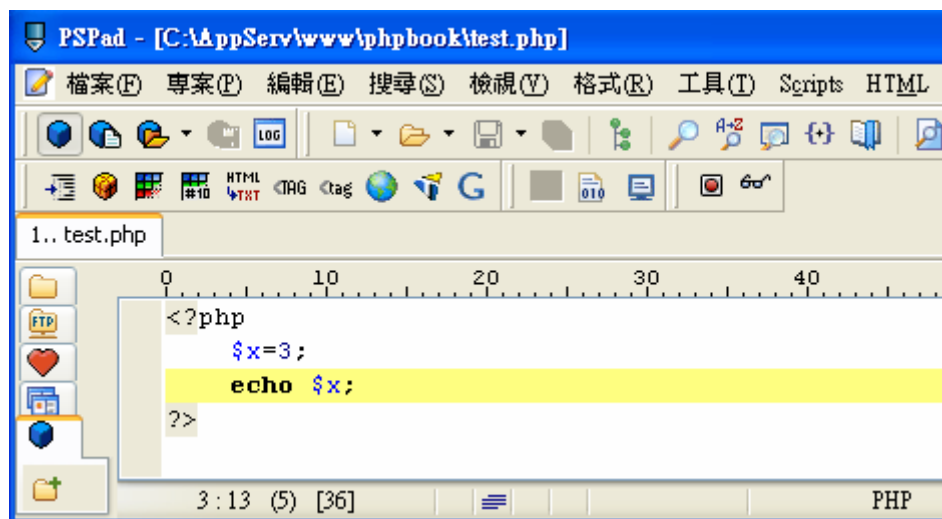


鍵入程式，本例鍵入以下程式：(變數一定要錢字號開頭)

```
0          10          20          30
<?php
    $x=3;
    echo $x;
?>
```

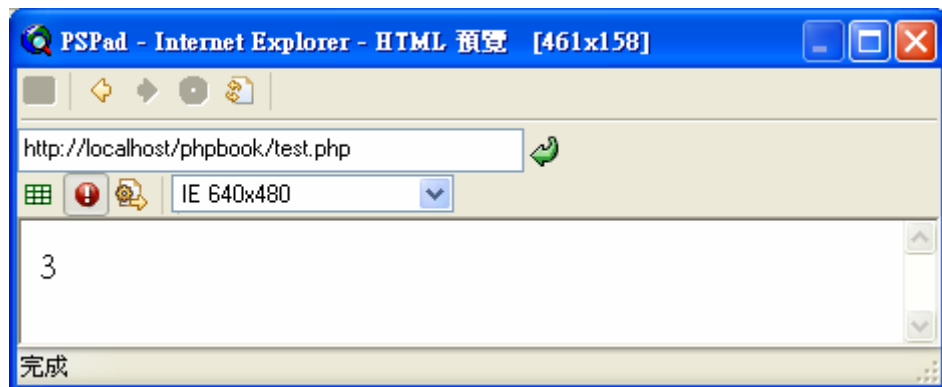
2、存檔。

依照 PHP 的規定，程式一定要存到網頁伺服器的根目錄之下才能執行，而 AppServ 預設網頁根目錄為 C:\AppServ\www，所以一定要將此程式存到此一根目錄之下才能執行。本書的作法是於此根目錄建立一資料夾，名稱爲 phpbook，並將各章程式分類存放，所以將本程式存檔如下：(點選”檔案”/”另存新檔”)



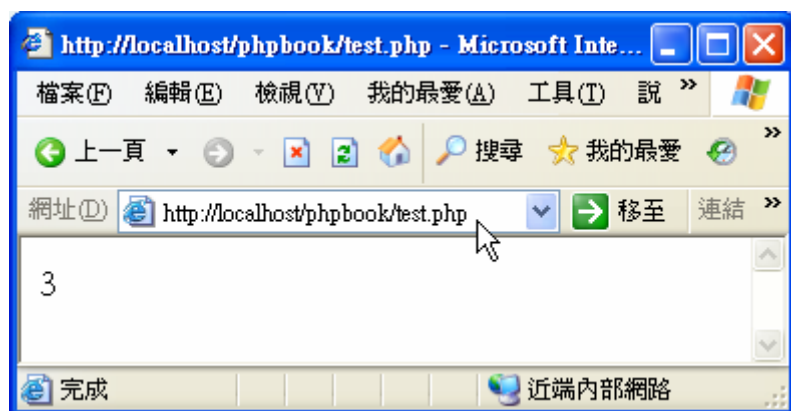
3、執行網頁。

點選功能表的”HTML”/”經由網頁伺服器預覽”，即可得下圖的執行結果。



4、於 IE 執行網頁。

若要在IE執行網頁，一定僅能經由網頁伺服器(localhost)執行網頁，如下圖所示。請於網址列自行鍵入 <http://localhost/phpbook/test.php>。

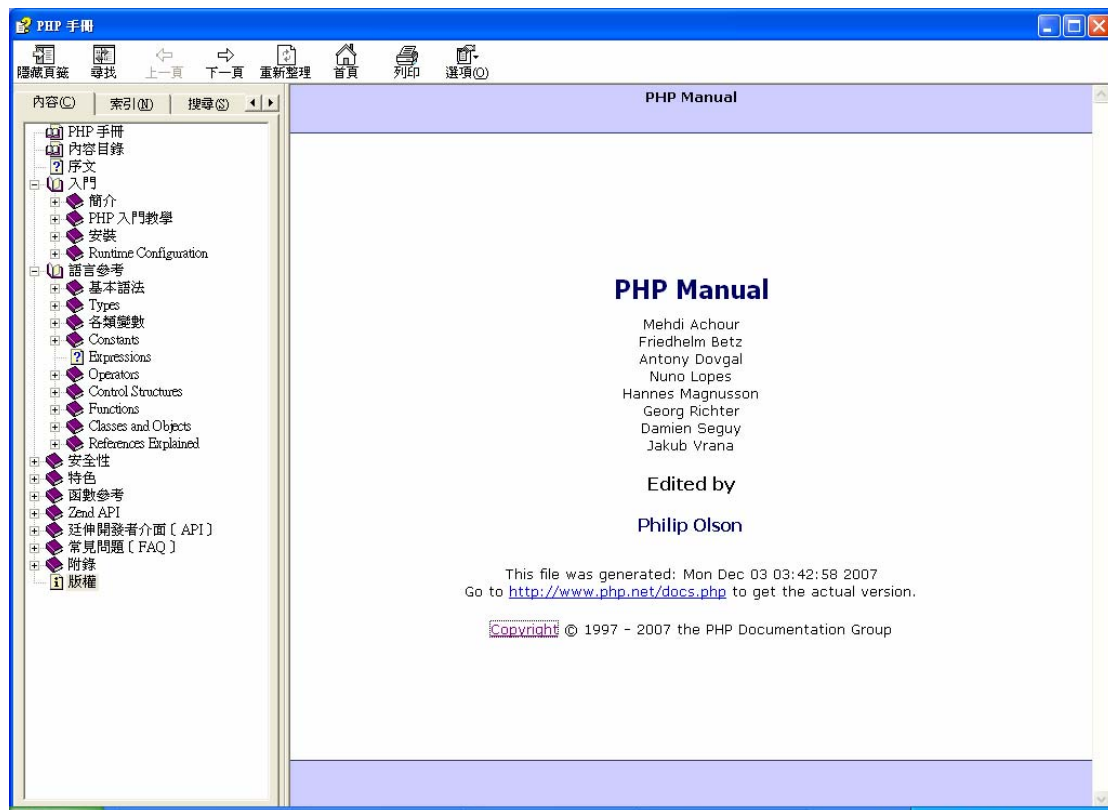


若直接使用”檔案”/”開啓舊檔”，則無法得到正確的結果。

範例 e2_1_1c 示範 PHP 手冊的使用。

操作步驟

- 1、PHP的手冊如下圖所示，請至<http://www.php.net/docs.php>下載此一檔案。下載回來的檔案名稱是 php_manual_tw.chm，使用者只要按兩下此一檔案，即可執行此手冊。



2_1_2 變數與資料型別

本節將先介紹 PHP 的變數命名規則、資料型別、運算子與運算式。

變數的命名

PHP 變數命名規則如下：

- 1、變數必須以錢字號(\$)開頭。
- 2、第二個字元僅能大小寫的英文字母或底線(_)。例如，

```
$a=3;
```

```
$_=3;
```

都是合法的變數設定。但是

```
$7=3;
```

```
$7a=3;
```

都是不合法的變數。

- 3、第三個字元開始可以是大小寫的英文字母或數字。例如，

```
$A3=2;
```

```
$_3=8;
```

```
$if=7;
```

以下則是非法的變數設定

```
a=3;          //未使用$字號開頭，此為初學者最容易產生的錯誤
```

```
$3a=6;        //第二個字元僅能大小寫的英文字母或底線
```

```
$a!s=3;       //不能包含任何非大小寫英文字母與數字當變數名稱
```

```
$ab a=5;      //當然也不能含空白
```

4、大小寫視為不同變數。例如，

```
$mary=3;
```

```
$Mary=7;
```

```
$MARY=6;
```

都是代表不同的變數。

變數的宣告

PHP 變數使用前並不用事先宣告其資料型別，只要設定其初值即可，編譯器即會依照其初值給予安排適當的型別。例如，

```
$a=5;
```

```
$b="mary";
```

編譯器即會分別安排數值與字串型別所需的記憶體給予\$a 與\$b 變數。

資料型別

PHP 提供以下八種資料型別，分別是整數、浮點數、布林、字串、NULL、陣列、資源(resource)、物件等。陣列請看第 2_5 節。

整數(integer)

整數是最簡單的型別，例如，12、-25 等都是整數型別。

浮點數(double)

帶有小數點的數字，即為浮點數。例如，3.14、-7.5、0.0034 及-0.0034 等都是浮點數。其次，亦可使用科學記號表示浮點數。例如，0.0034 可以寫成

```
$a=3.4e-3
```

其中字母 e 大小寫都可以，也就是 PHP 的保留字大小寫都可以。此外，寫成浮點數的數值，PHP 都視為浮點數。例如，

```
$a=34e5;
```

```
echo (gettype($a);
```

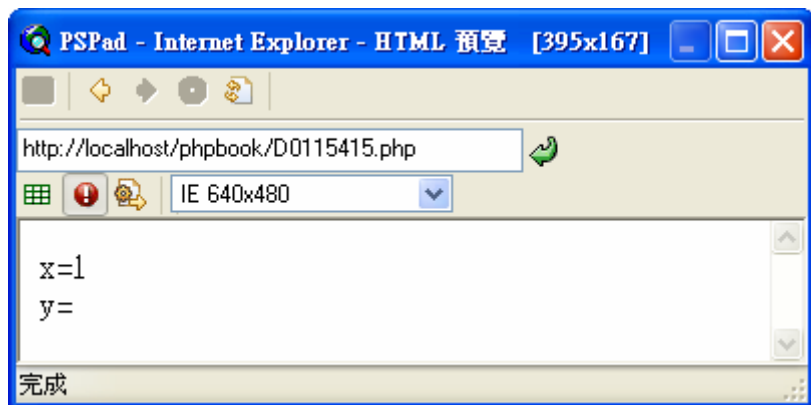
即傳回 double。(附註:gettype 函式可傳回變數的型別)

布林(boolean)

若運算結果僅有 true 與 false，則其型別可規定為布林，如此才可提高運算效率與節省記憶體。其次，true 的輸出為 1，false 的輸出為空白。例如，

```
$x=true;$y=false;  
echo "x=$x <br>";  
echo "y=$y <br>";
```

的輸出如下：



字串

不用進行數值運算的資料，都可視為字串，且應加上單引號(')或雙引號(")。例如，

```
$a= ' Mary' ;  
$b= " Mary" ;
```

等都是字串。若字串中有單引號，則可使用雙引號，例如，

```
I' m a student.
```

則應使用雙引號如下：

```
$a= " I' m a student" ;
```

同樣的道理，若字串中有雙引號，則可使用單引號，例如，

```
It is "apple" .
```

應使用單引號，如以下敘述：

```
$b= ' It is "apple" .' ;
```

其次，單引號與雙引號的差別是雙引號可使用變數剖析。例如，

```
$str= " Mary"  
echo " My name is $str." ;
```

則結果是

```
My name is Mary.
```

但是，

```
$str= " Mary"  
echo ' My name is $str.' ;
```

則結果是

```
My name is $str.
```

NULL

NULL 是空的意思，當任一變數未輸入初值，其結果即為 NULL，而且當然與 0 和空字串(" ")不同。

常數

有些數字在程式中從頭到尾都不會變動的東西，稱為常數。例如，圓周率=3.14，此時即可使用 define 保留字給予定義。例如，

```
define( "PI" ,3.14);  
echo PI;
```

結果是 3.14。請特別留意，常數名稱不要加錢字號(\$)開頭。此外，經常變動的數值，亦應使用常數放在程式的前頭。例如，利息的計算中，利率當然經常變動，此時應在程式的前頭先行定義利率這一常數，且程式所有計算利息的地方都要統一使用此常數，則利率改變時，只要改變程式前頭的利率值即可，不用在程式中到處修改此數字，否則，萬一有些地方沒有修改到，當然會造成執行結果的不一致。

2_1_3 運算子

可以對運算元(Operand)執行特定功能的特殊符號稱為運算子。一般而言，運算子可分為以下幾類：指派(Assignment)運算子、算術(Arithmetic)運算子、關係(Relational)運算子、邏輯(Logical)運算子、字串(String)運算子及複合指派運算子。其次，運算子又可以分為一元(Unary)運算子與二元(Binary)運算子。一元運算子只需要一個運算元就可以操作，而二元運算子則需要兩個運算元才能夠操作。在以下單元中，我們除了檢視各種不同的運算子功能外，還將討論運算子的優先順序(Precedence)。優先順序是用來決定同一式子擁有多個運算子時，每一個運算子進行運算的順序。

I 指派運算子 (Assignment operator)

指派(或稱指定)運算子的符號為=，其作用為將運算符號右邊運算式的值指派給運算符號左邊的運算元。所以，以下敘述\$sum = \$a + \$b，是將\$a + \$b 的值指派給\$sum。

```
$sum = 0 ; $a = 3 ; $b = 5;  
$sum = $a + $b;
```

上式與數學的等號是不同的，所以不要一直懷疑為什麼 0 會等於 8。其次，你不能將常數放在指派運算子的左邊。例如，

```
8 = $x;
```

為一個不合法的敘述，但以下敘述將常數 8 指派給變數\$x 為合法的。

```
$x = 8;
```

I 算術運算子 (Arithmetic operators)

算術運算子用來執行一般的數學運算，包括取正數(+)、取負數(-)、加(+)、減(-)、乘(*)、除(/)、取餘數(%)、遞增(++)、遞減(--)等，如下表所示：

運算子符號	功 能	範 例	結果
+	取正數	+2	
-	取負數	-3	

+	加	6 + 2	8
-	減	6 - 2	4
*	乘	6 * 2	12
/	除	7 / 2	3.5
%	取餘數	7 %2	1
++	加 1	\$a=2;\$a++;echo(\$a);	3
--	減 1	\$a=2;\$a--; echo(\$a);	1
+=	相加且指派	\$a=1;\$a+=5; echo(\$a);	6 (\$a=\$a+5)
-=	相減且指派	\$a=3;\$a-=2; echo(\$a);	1 (\$a=\$a-2)
=	相乘且指派	\$a=2;\$a=4; echo(\$a);	8 (\$a=\$a*4)
/=	相除且指派	\$a=9;\$a/=3; echo(\$a);	3 (\$a=\$a/3)
%=	取餘數且指派	\$a=7;\$a%=3;echo(a);	1 (\$a=\$a%3)

例如，

```
$a=2;
$a++;
echo($a);
```

結果是 3。因為 \$a=\$a+1 這種遞增運算常用，所以有如此簡便語法。又例如

```
$a=1;
$a+=5;
echo$a(a);
```

結果是 6。因為 \$a=\$a+5 這種運算也常用，所以有如此簡便用法。

I 關係運算子 (Relational operators)

關係運算子又稱為比較運算子，用於資料之間的大小比較，比較的結果可得到邏輯的 true 或 false，下表是 PHP 中的關係運算子符號。

運算子符號	功 能	範例	結果
==	等於	2 == 3	false
!=	不等於	2 != 3	true
<>	不等於	2 <>3	true
<	小於	2 < 3	true
>	大於	2 > 3	false
<=	小於或等於	2 <= 3	true
>=	大於或等於	2 >= 3	false

例如，

```
$a=2;$b=3;
echo($a==$b);
```

結果是 false(輸出空白)。

I 邏輯運算子 (Logical operators)

當同一個運算式同時具有兩個以上的關係運算子時，則每兩個關係運算子之間必須使用邏輯運算子連結。PHP 的邏輯運算子如下表所示：

運算子符號	功 能	範例	結果	備註
and	And	(3 > 2) and (3 == 1)	false	兩者都為真，結果才為真
&&	And	(3 > 2) && (3 == 1)	false	兩者都為真，結果才為真
or	Or	(3 > 2) or (3==1)	true	任一為真，結果即為真
	Or	(3 > 2) (3==1)	true	任一為真，結果即為真
!	Not	!(3>2)	false	反相
		! false	true	
xor	XOR	true xor true	false	一真一假時為真；兩個都真或兩個都假則為假。
		true xor false	true	
		false xor true	true	
		false xor false	false	

I 字串運算子(String operator)

前面關係運算子的 ==、!=、<>、>、<、>= 及 <= 亦均適用於字串的大小比較。例如，

```
$a="a";$b="b";
```

```
echo ($a>$b);
```

其結果是 false。因為，字串的大小是以 ASCII 值的大小為依據，a 的 ASCII 值為 97，b 的 ASCII 值為 98，所以字串 b 較大。而下表的字串連結運算子(.)，則可連結兩個字串。

運算子符號	功 能	範例	結果
.	字串連結	"abc" . "xyz"	abcxyz

I 雜項運算子

以下是一些無法歸類的運算子，列表如下。這些運算子將會在往後各章節陸續介紹。

運算子符號	使用範例	說明
[]	\$a[2]	一維陣列
[][]	\$b[2][3]	二維陣列
()	f(3)	函式 f 的叫用(特別留意，函式名稱前不用錢字號)

I 運算子的優先順序 (Precedence)

於較複雜的運算式中，通常同時存在許多運算子，此時就需要定義其優先順序，下表即是 PHP 關於運算子的優先順序表，優先權高的在上面。

結合方式	運算子	備註
non-associative	new	
right	[	
right	! ~ ++ -- (int) (float) (string) (array) (object) @	型別轉換
left	* / %	
left	+ - .	
left	<< >>	位元運算
non-associative	< <= > >=	關係運算
non-associative	== != === !==	陣列運算
left	&	
left	&&	
left		
left	? :	
right	= += -= *= /= .= %= &= = ^= <<= >>=	
right	print	
left	and	
left	xor	
left	or	
left	,	

例如，

```
$a=2;$b=3;
```

```
$c=++$a+$b;
```

則等效於以下敘述：

```
$c =((++$a)+$b) ;
```

因爲一元運算子的優先順序較高，所以結果是 6。又例如，

```
$a=2;$b=3;  
$c=$a+$b*2;
```

則等效於以下敘述:

```
$c=($a+($b*2));
```

又例如，

```
$x=3;$y=4;  
$z= $x > 2 && $y > 3;
```

則等效於以下敘述。

```
$z = (($x > 2)&& ($y > 3));
```

運算子結合方式

由上表亦可知運算子結合方式有左結合與右結合。例如，

```
$x=8;$y=2;  
echo $x/$y/$y;    //2
```

的結果是 2，因為除法運算是左結合。又例如，

```
$x=$y=3;  
echo "x=$x <br>"; //3  
echo "y=$y <br>"; //3
```

則可同時設定\$x,\$y 的值均為 3，因為指派運算為右結合。

2_1_4 敘述與運算式

I 敘述 (Statement)

凡是控制執行的順序、對運算式取值或不作任何事，均可稱為敘述或陳述式。其次，每一敘述均應以分號(;)結束，以下式子即是一個敘述。

```
$sum = $sum + 1;
```

敘述區塊 (Block Statement) 或複合敘述 (Compound Statement)

在任何可以放單一敘述的地方，也就能放敘述區塊，敘述區塊亦稱為複合敘述。一個複合敘述是由一對大括號組合而成，如下所示：

```
{  
t = a;  
a = b;  
b = t;  
}
```

I 運算式 (Expression)

任何可求得值的式子，均稱為運算式。例如，5+3 會傳回一個數值，所以 5+3 是一個運算式。一般而言，可以放在等號右邊與括號內的東西，都可以稱為運算式。例如，以下等號右邊及括號內的東西均稱為運

算式。

```
$a = 3 ; $b = 5;  
$c=$a+$b;  
if ($a>0)           /*true  
while($a==$b)       //false
```

I 註解 (Comments)

適當的程式註解能增加程式的可讀性。其次，註解是給人看的，編譯器均不予理會註解的內容。PHP 的註解有兩種表示方式，分別是單列註解及多列註解。

II 單列註解

凡是雙斜線(//)或井號(#)以後的敘述，PHP 均視為註解。例如，以下敘述的“將 y 值累加至 sum”為註解，不會被執行。

```
sum=sum+y;  
//將 y 值累加至 sum  
#將 y 值累加至 sum
```

II 多列註解

凡是/*以後的敘述，PHP 均視為註解，直到*/為止。例如，以下敘述全為註解，均不會被執行。

```
/* 這是多列註解  
sum=sum+y;  
將 y 值累加至 sum */
```

範例 2_1_3a

請將以下數學敘述以 PHP 敘述表示。

1. $x = \frac{(a+b+c)}{3}$

2. $y = \frac{a+b}{a-b}$

3. $y = a^2$

4. $y = \sqrt{a}$

5. $y = b^{10}$

6. $y = \sqrt[3]{1000}$

執行結果



程式列印

```
<?php
    $a=3;$b=2;$c=1;
    $x=($a+$b+$c)/3;
    echo($a)."<BR>";
    $y=($a+$b)/($a-$b);
    echo($y)."<BR>";
    $y=pow($a,2);
    echo($y)."<BR>";
    $y=pow($a,0.5);
    echo($y)."<BR>";
    $y=pow($b,10);
    echo($y)."<BR>";
    $y=pow(1000,1/3);
    echo($y)."<BR>";
?>
```

【習題】

- 1、令 $a=1, b=-1, c=-6$ ， $x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ ， $x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$ 請於 PHP 求出 x_1 與 x_2 的結果。
- 2、令三角形三邊長 $a=3$ 、 $b=4$ 、 $c=5$ ，求其面積。(面積 $= \sqrt{d(d-a)(d-b)(d-c)}$ ，其中 $d = \frac{(a+b+c)}{2}$)
- 3、若 a 點座標是 $(3,0)$ ， b 點座標是 $(0,4)$ ，請計算此二點的距離。

2-2 決策敘述

PHP 的決策敘述有 if 、if~else、if~elseif、及 switch 等，分別說明如下：

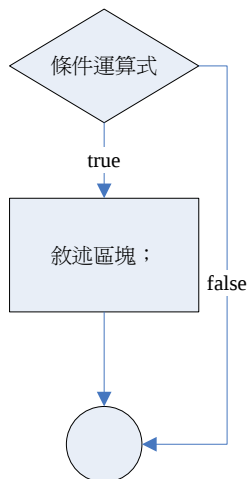
if

if 通常用於單一分歧的決策，它的使用時機為“假如～則～”，也就是條件成立時，則執行某項工作，但若條件不成立時，則不予理會。其語法如下：

```
if (運算式) {  
    敘述 1;  
    敘述 2;  
    ...  
}
```

以上語法說明如下：

1、若運算式的值為 true 則執行敘述區塊，運算式的值若為 false，則不會被執行，其流程如下：



2、例如，以下敘述可判斷變數 \$a 是否及格。(t2_2a)

```
<?php  
    $a=77;  
    $b="Done";  
    if ($a>=60){  
        $b="Pass";  
    }  
    echo $b;  
?>
```

3、當運算式成立時，僅執行一個敘述，則大括號可省略。例如以上敘述亦可寫成如下：

```
<?php
```

```

$a=77;
$b="Done";
if ($a>=60)
    $b="Pass";
echo $b;
?>

```

if~else

if~else 適用假如～否則～。其語法如下：

```

if (運算式){
    敘述 1；
}
else {
    敘述 2；
}

```

也就是當運算式成立時，執行敘述 1，不成立時則執行敘述 2。例如，以上範例以 if ~else 改寫，則其程式如下：(t2_2b)

```

<?php
    $a=77;
    if ($a>=60){
        $b="Pass";
    }
    else {
        $b="Done";
    }
    echo $b;
?>

```

三元運算子(? :)

以上 if ~else 亦可使用三元運算子(? :)， 以上程式使用三元運算子可改寫如下，也就是條件成立時，**\$b="Pass"**否則**\$b="Done"**。

```

<?php
    $a=77;
    $b=($a>60)? "Pass":"Done";
    echo $b;
?>

```

記得問號(?)後面要加一個空白。

巢狀 if

if 內有 if 或 else 內有 if ，稱為巢狀 if 。例如，以下程式(t2_2c)可判斷\$a 是正數、0、或負數。

```
<?php
    $a=0;
    if ($a>0)
        $b="Plus";
    else
        if ($a==0)
            $b="Zero";
        else
            $b="Minus" ;
    echo $b;
?>
```

if~elseif~

於多選一的情況，除了使用巢狀 if 外，還可使用 if~elseif，if ~elseif 的語法如下：

```
if (運算式 1){
    敘述 1；
}
elseif (運算式 2){
    敘述 2;
}
else {
    敘述 3;
}
```

例如，以上判斷正數、零、或負數的程式亦可改寫如下：(t2_2d)

```
<?php
    $a=-7;
    if ($a>0)
        $b="Plus";
    elseif ($a==0)
        $b="Zero";
    else
        $b="Minus" ;
    echo " $a  is $b";
?>
```

switch

switch 適用於多選一的決策敘述。例如，以下敘述(t2_2e)可將 1、2、3、4 轉為“春”、“夏”、“秋”、“冬”。

```
<?php
    $a=1;
    switch ($a){
        case 1:
            $b="春";
            break;
        case 2:
            $b="夏";
            break;
        case 3:
            $b="秋";
            break;
        case 4:
            $b="冬";
            break;
    }
    echo $b."<br>"
?>
```

請留意 break 一定不可以省略，否則會往下一直執行，造成無法預期的結果。例如，以下程式(t2_2f)，不論變數\$a 的值是 A+、A、或 A-都可以得到“甲”。

```
<?php
    $a="A-";
    switch ($a){
        case "A+":
        case "A":
        case "A-":
            $b="甲";
            break;
        case "B":
            $b="乙";
            break;
        case "C":
            $b="丙";
            break;
        case "D":
            $b="丁";
    }
```

```
        break;
    }
    echo $b."<br>"
?>
```

習題

- 1、若有三個數，分別是 3,7,2，請以變數儲存，並輸出最大數，也就是求此三數極大值。
- 2、若有三個數，分別是 3,7,2，請以變數儲存，並由小而大排序輸出。
- 3、自來水費計算。假設自來水費率如下：
 - A、100 度以下，每度 3 元。例如，用水 50 度，則水費=50*3=150 元。
 - B、100~300 度，超過 100 度的部分，每度 5 元。例如，用水 200 度，則水費=100*3+100*5=800 元。
 - C、300 度以上，超過 300 度的部分，每度 6 元。例如，用水 400 度，則水費= 100*3+200*5+100*6=1900 元。請設計一網頁，可以計算自來水費，度數則由變數設定。

- 4、PHP 產生 0 到 2 的亂數函式為 rand(0,2)，請寫一程式，可以產生兩個亂數，0 代表剪刀，1 代表石頭，2 代表布，並評定勝負。

2-3 迴圈敘述

PHP 的迴圈敘述有 for、while、do~while 分別說明如下：

for

for 用於程式設計階段即已知迴圈的起始值與終值。例如，以下程式(t2_3a)可輸出整數 1 到 10。

```
<?php
    for ($i=1;$i<=10;$i++){
        echo $i."<br>";
    }
?>
```

以下程式(t2_3b)則可以輸出 9 7 5 3 1。

```
<?php
    for ($i=9;$i>=1;$i=$i-2)
        echo $i." ";
?>
```

continue

continue 通常用於迴圈敘述內，若程式遇到 continue 則提早離開迴圈，且從迴圈的開頭繼續進行迴圈。例如，以下程式(t2_3c)，當 \$i==3 時，提早離開迴圈(未輸出 3)且繼續從迴圈的開頭繼續執行，此時 \$i 已經等於 4。

break

break 通常用於迴圈敘述內，且無條件提早離開迴圈。例如，以下程式(t2_3c)，執行到 6 時，提早離開迴圈，所以 6 以後都未輸出。

```
<?php
    for ($i=1;$i<=10;$i++){
        if ($i==3)
            continue;
        if ($i==6)
            break;
        echo $i."<br>";
    }
?>
```

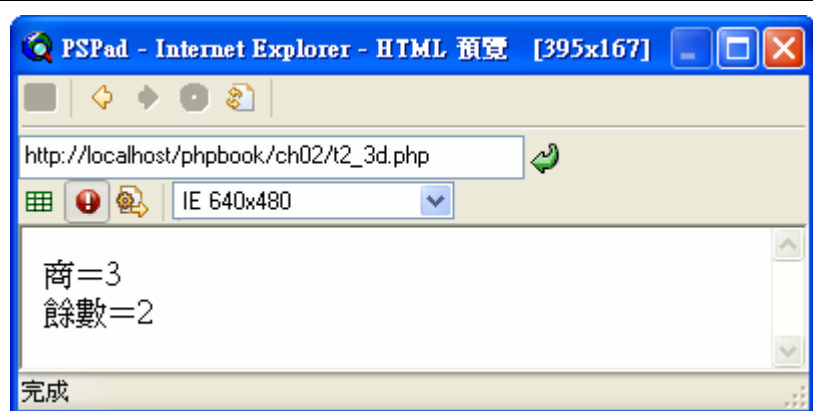


1
2
4
5
完成

while

while 是一種未知計數終止值的前測試迴圈。例如，於以下程式可進行除法運算，被除數可減去除數的個數即為商。被除數(11)到底可以減去多少個除數(3)，此種運算於程式設計階段並不知計數的終值，所以應該使用此前測試的 while。本例程式(t2_3d)如下：

```
<?php
    $a=11;
    $b=3;
    $q=0;
    while ($a> $b){
        $a=$a-$b;
        $q++;
    }
    echo "商=".$q."<br>";
    echo "餘數=".$a."<br>";
?>
```



商=3
餘數=2
完成

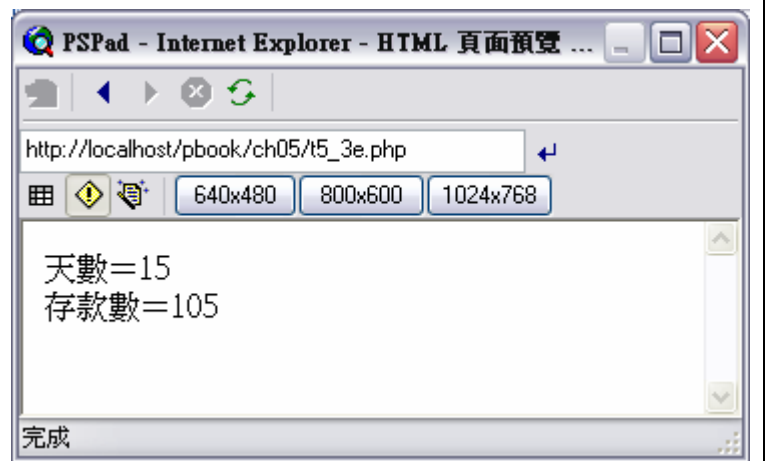
do~while

do~while 也是一種未知計數終止值的後測試迴圈。例如，每天存 7 元，到底幾天可以存款超過 100 元，此種運算於程式設計階段並不知計數的終值，所以應該使用此後測試的 do~while。本例程式(t2_3e)如下，請特別留意後測試迴圈的 while()之後要加分號(;)。

```

<?php
    $sum=0;
    $b=7;
    $i=0;
    do {
        $i++;
        $sum=$sum+$b;
    }while ($sum <100);
    echo "天數=".$i."<br>";
    echo "存款數=".$sum."<br>";
?>

```



前測試與後測試迴圈的差別在於，前測試迴圈先行測試某一條件，再決定是否執行迴圈敘述，但是後測試迴圈則是先行執行一次迴圈內的敘述，再測試是否繼續重複迴圈，所以兩者的差別是前測試迴圈可能一次都未執行迴圈內敘述，後測試迴圈則是至少執行一次迴圈。

習題

- 1、若 $y=f(x)=x^2+x-6$ ，其解在-10 到 10 之間的整數，請設計一網頁，使用暴力代入法求其解。
- 2、請任意產生一個 2 到 100 的亂數，並判斷此數是否為質數。(附註：PHP 產生 2 到 100 的亂數是 rand(2,100))
- 3、請列出 2 到 100 的所有質數。
- 4、試設計一網頁，找出三位數“阿姆斯壯數”。所謂阿姆斯壯數是指一數等於各個位數的立方和，即 $abc=a^3+b^3+c^3$ ，例如 $153=1^3+5^3+3^3$ 。

- 5、請設計一網頁，滿足以下條件：

- (1) 可以產生 0 至 6 的亂數。(隨機產生 0 至 1 的亂數函式為 rand(0,6)，關於亂數的使用，請先看第 2_4_2 節)
- (2) 累加以上亂數。
- (3) 輸出此亂數與其和。
- (4) 若亂數不為 0，則重複步驟(1)~(3)，直到亂數為 0，則程式結束。

- 6、請設計一網頁，可以連續產生 3 個 1~6 的亂數，並輸出此 3 個亂數，直到有其中兩個亂數相等為止，並輸出另一亂數之值。

- 7、請設計一網頁，可以連續產生 4 個 1~6 的亂數，並輸出此 4 個亂數，直到有其中兩個亂數相等為止，並輸出另兩亂數之和。例如，亂數為 1,5,2,6 則繼續產生亂數，亂數若為 2,5,2,6 則輸出其和為 11。

2-4 函式

若某些程式片段常常出現，此時即可將此程式片段，寫成函式，當要用這些程式片段時，即可使用函式名稱呼叫，此稱為函式。本節將分別介紹如何自訂函式，及 PHP 預設的一些函式，例如，數值函式、字串函式、時間函式、及一些無法歸類的雜項函式。

2-4-1 自訂函式

PHP 自訂函式的簡要語法如下：

```
function 函式名稱(參數列)
{
    敘述；
    [return;return 值；]
    [敘述；]
}
```

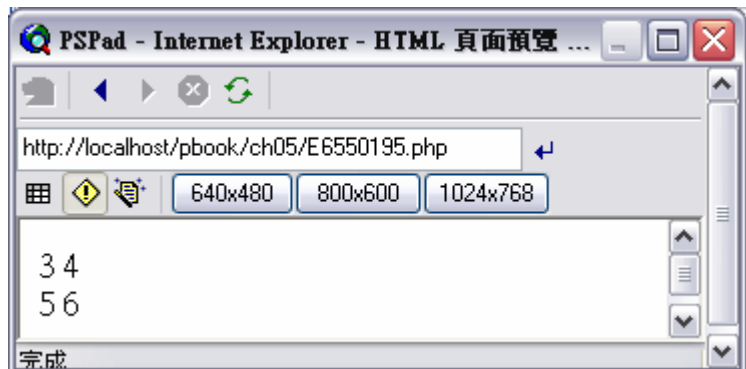
例如，以下程式(t2_4_1a)，可得 $y=f(x)=3x+1$ 之值，function 放在主程式的前面或後面均可。其次，函式名稱前不用加錢字號(\$)。

```
<?php
function f($x){
    $y=3*$x+1;
    return ($y);
}
$y=f(2);
echo $y."<br>";
?>
```

傳址呼叫與傳值呼叫

傳址呼叫時，函式的參數若有變動，將會回傳主程式所對應的變數；其次，傳值呼叫時，函式的參數若有變動，並不會回傳主程式所對應的變數。PHP 內定傳值呼叫，也就是函式的變數值並不回傳。若要特別強調傳址呼叫，則應於參數列加上&。例如，以下程式(t2_4_1b)f1()函式是傳值呼叫，f2()函式則是傳址呼叫，所以 f2()函式內所變動的參數將會回傳，f1()則不回傳。

```
<?php
function f1($m,$n){
    $m=5;$n=6;
}
function f2(&$m,&$n){
    $m=5;$n=6;
}
$x=3;$y=4;
f1($x,$y);
echo $x." ".$y."<br>"; //3 4
```



```
$x=3;$y=4;  
f2($x,$y);  
echo $x." ".$y."<br>"; //5 6
```

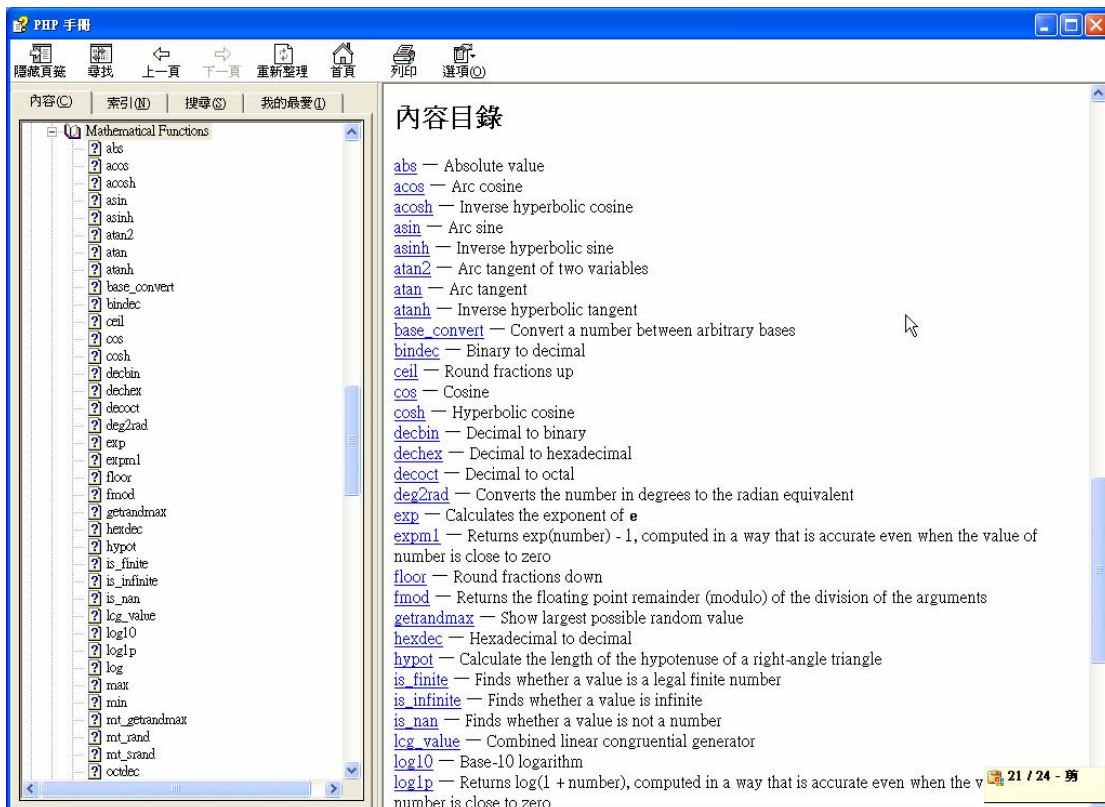
?>

習題

- 1、計算 $C_n^m = \frac{m!}{n!*(m-n)!}$ 的組合運算，但使用 function。
- 2、 $f(x,y)=3x^2+xy-6y^2$ ，請寫一程式，可以將 $f(x,y)$ 寫成 function，且可使用 $f(x,y)$ 呼叫函式。例如， $f(0,1)=-4$ 、 $f(1,1)=-2$ 。
- 3、請寫一個程式，使用 function，計算 $1!+2!+3!+4!+5!$ 之值。

2_4_2 數值函式

PHP 的數值函式如下圖：



常用的數值常數與函式說明如下：

`pi()`

傳回圓周率 $\pi = 3.14159$ 之精確值。例如，以下敘述可傳回圓周率 π 之值。

```
echo pi();
```

abs()

絕對值函式。例如，以下敘述得到 3。

```
echo abs(-3);
```

pow()

次方函式。例如，以下敘述傳回 2 的 3 次方。

```
echo pow(2,3);
```

sqrt()

開根號函式。例如，以下敘述可傳回 9 的開根號，其值是 3。

```
echo sqrt(9);
```

round()

將數值四捨五入，其語法如下：

```
float round ( float $val [, int $precision] )
```

例如，以下敘述可將小數點後面取到第 2 位，第 3 位四捨五入。

```
echo round(2.555,2); //2.56
```

rand()

亂數函式，其語法如下：

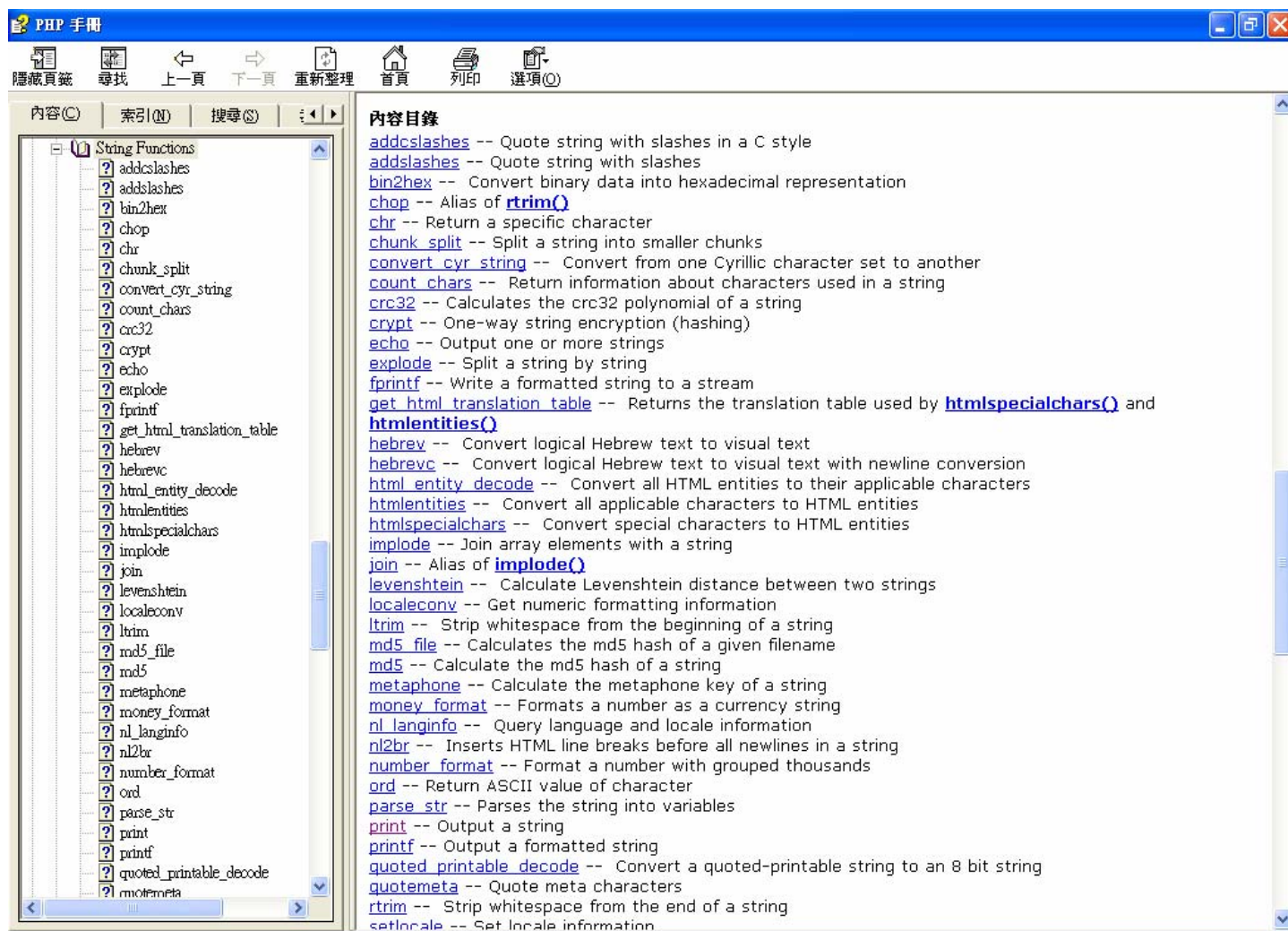
```
int rand ( [int $min, int $max] )
```

可傳回指定範圍的整數亂數。例如，以下敘述可傳回 1 到 6 的整數亂數。

```
echo rand(1,6);
```

2_4_3 字串函式

PHP 的字串函式如下表所示：



其中常用的字串函式如下：

`strlen()`

傳回字串長度。其語法如下：

```
int strlen(string str)
```

例如，

```
echo str( "Hello" );
```

可傳回 5。

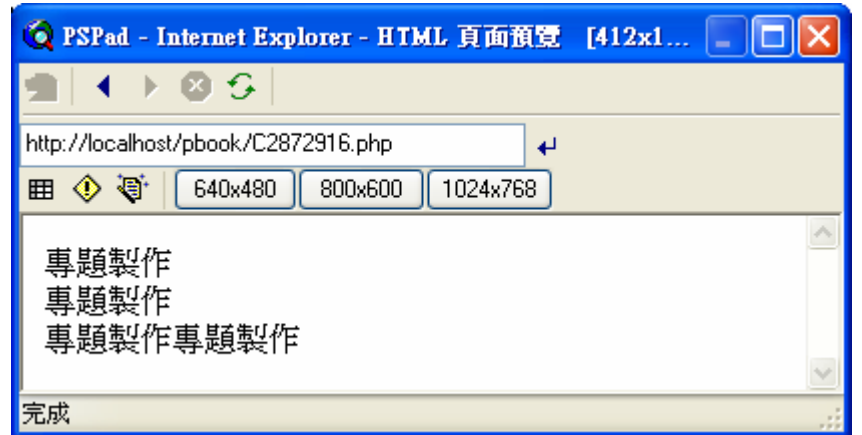
`nl2br()`

此函式語法如下：

```
string nl2br (string str)
```

將字串 `str` 中所包含的換行符號 `\n`、`\r`、`\r\n` 轉為 HTML 換行標籤 `<BR>`，然後傳回經過轉換的字串。例如以下程式(t2_4_3a) 的執行結果如右：

```
$string="專題製作\r\n";  
echo nl2br($string);  
echo nl2br($string);  
echo ($string);  
echo ($string);
```



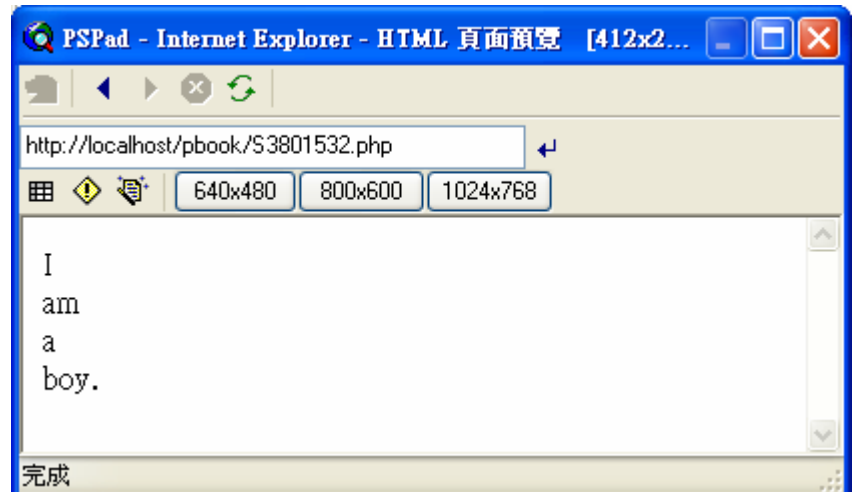
explode()

字串分解函式。其語法如下：

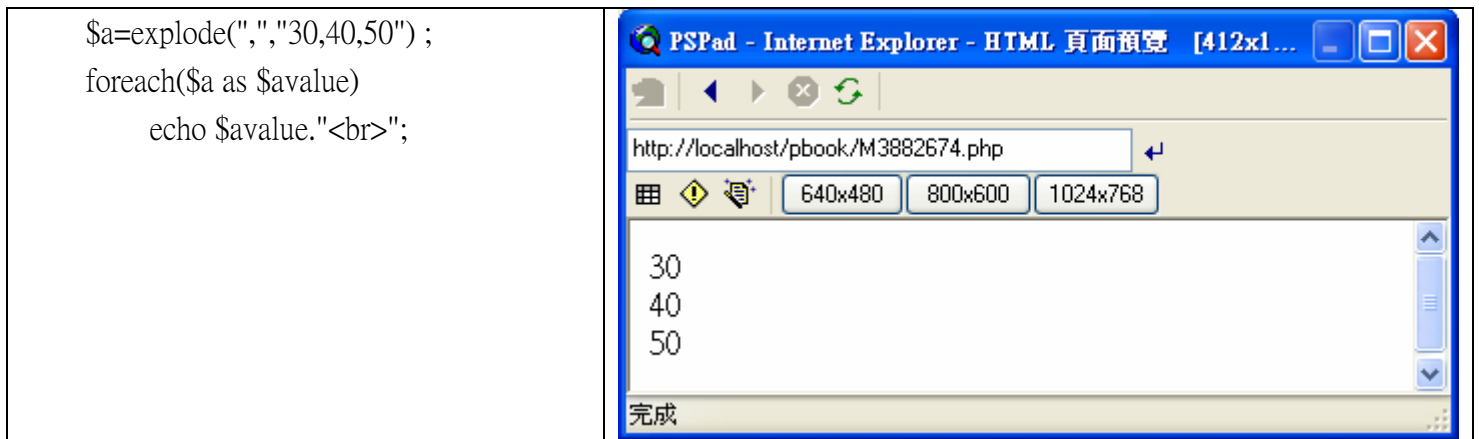
array **explode** (string separator, string string)

separator 是分解的字串，string 是待分解字串，傳回值則為陣列。例如，

```
$a=explode(" ","I am a boy.");  
foreach($a as $avalue)  
    echo $avalue."<br>";
```



又例如，



習題

1、請寫一程式，可以執行統一發票的對獎，本例假設統一發票只有 6 碼，且只有一組中獎號碼。例如，

- (1).後面 6 個號碼相同，中頭獎。
- (2).後面 5 個號碼相同，中二獎。
- (3).後面 4 個號碼相同，中三獎。
- (4).後面 3 個號碼相同，中四獎。

2、請寫一程式，可以核對兩個字串相同的程度，數字與位置相同定義為 A，數字相同，位置不同定義為 B。例如，

字串 1	字串 2	比對結果
5218	5218	4A0B
5218	5281	2A2B
5218	1852	0A4B

2_4_4 時間函式

PHP 的時間函式如下圖所示：



以下將介紹一些常用的時間函式。

`int time(void)`

傳回電腦系統的目前時間刻記(timestamp)，此刻記是指 1970/1/1 到目前時間所經過的秒數。例如，

```
$t1=time();
```

將可得到電腦系統的時間刻記。

`date()`

將指定的時間刻記，轉為指定的字串型態。其語法如下：

`string date(string format [,int timestamp])`

format 的可用字元如下表，timestamp 若省略，則表示取系統時間。

<i>format character</i>	Description	Example returned values
a	Lowercase Ante meridiem and Post meridiem	am Or pm
A	Uppercase Ante meridiem and Post meridiem	AM or PM
B	Swatch Internet time	000 through 999
c	ISO 8601 date (added in PHP 5)	2004-02-12T15:19:21+00:00
d	Day of the month, 2 digits with leading zeros	01 to 31
D	A textual representation of a day, three letters	Mon through Sun
F	A full textual representation of a month, such as January or March	January through December
g	12-hour format of an hour without leading zeros	1 through 12
G	24-hour format of an hour without leading zeros	0 through 23
h	12-hour format of an hour with leading zeros	01 through 12
H	24-hour format of an hour with leading zeros	00 through 23
i	Minutes with leading zeros	00 to 59
I (capital i)	Whether or not the date is in daylight savings time	1 if Daylight Savings Time, 0 otherwise.
j	Day of the month without leading zeros	1 to 31
l (lowercase 'L')	A full textual representation of the day of the week	Sunday through Saturday
L	Whether it's a leap year	1 if it is a leap year, 0 otherwise.
m	Numeric representation of a month, with leading zeros	01 through 12
M	A short textual representation of a month, three letters	Jan through Dec
n	Numeric representation of a month, without leading zeros	1 through 12
O	Difference to Greenwich time (GMT) in hours	Example: +0200
r	RFC 2822 formatted date	Example: Thu, 21 Dec 2000 16:01:07 +0200
s	Seconds, with leading zeros	00 through 59

例如，

```
$today = date("Y,m,d");
```

```
echo $today ;
```

將可得到系統的年,月,日。又例如，

```
$today = date("Y,m,d H-i-s");
```

```
echo $today ;
```

將可得到系統的年,月,日 時-分-秒。

getdate

將時間刻記以陣列型態傳回，其語法如下：

```
array getdate([int timestamp])
```

timestamp 稱為時間刻記，在 Windows 作業系統是指 1970/1/1 到指定的日期或時間所經過的秒數。例如，

```
$t=getdate(0);
```

將會得到 1970/1/1，而

```
$t=getdate(60*60*24);
```

將會得到 1970/1/2。若省略參數，則表示傳回系統的時間。例如，

```
$t=getdate();
```

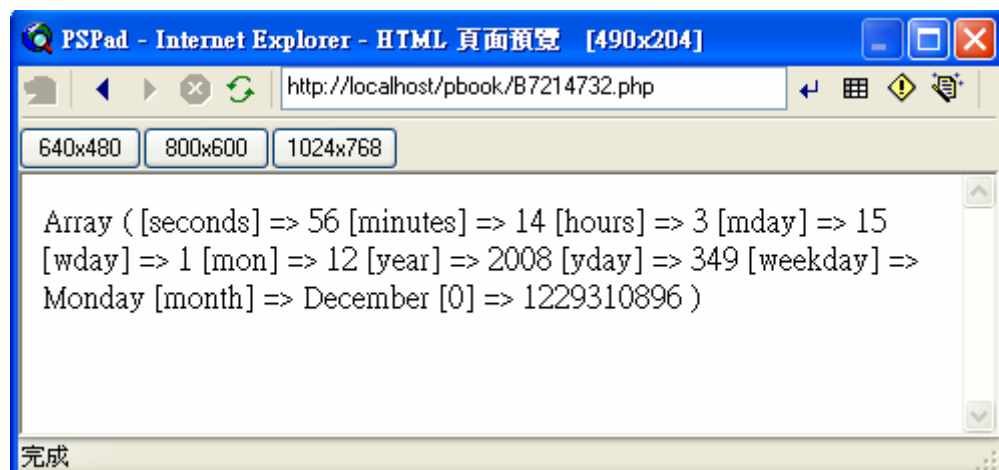
將會得到目前系統的時間。其次，傳回值是一個陣列型態，其元素如下表：

Key	Description	Example returned values
"seconds"	Numeric representation of seconds	0 to 59
"minutes"	Numeric representation of minutes	0 to 59
"hours"	Numeric representation of hours	0 to 23
"mday"	Numeric representation of the day of the month	1 to 31
"wday"	Numeric representation of the day of the week	0 (for Sunday) through 6 (for Saturday)
"mon"	Numeric representation of a month	1 through 12
"year"	A full numeric representation of a year, 4 digits	Examples: 1999 or 2003
"yday"	Numeric representation of the day of the year	0 through 366
"weekday"	A full textual representation of the day of the week	Sunday through Saturday
"month"	A full textual representation of a month, such as January or March	January through December
0	Seconds since the Unix Epoch, similar to the values returned by time() and used by date() .	System Dependent, typically - 2147483648 through 2147483647.

例如，

```
$today = getdate();  
print_r($today);
```

將會得到輸出結果如下



又例如，

```
print($today["year"]);
```

將會得到 2008。

`mktime`

根據指定的時間參數，建立一個時間刻記。其語法如下：

```
int mktime ( [int hour [, int minute [, int second [, int month [, int day [, int year]]]]]])
```

例如，

```
$t=mktime(14,0,0,10,2,2008);
```

可以建立一個 2008/10/2 14：0：0 的時間刻記。

習題

1.停車場計費。假設某一停車場第一小時 60 元，不足一小時以一小時計，往後每半時為 20 元，且不足半小時以半小時計，請寫一網頁可以協助管理此停車場的計費網頁。

2 計算兩個日期所經過的天數。

2_4_5 雜項函式

前面已經介紹一些關於數值、字串與時間的常用函式，但是還是遺漏一些常用函式，現在說明如下：

die() vs exit()

此兩函式的功能都相同，都是讓程式停止執行，其語法如下：

```
void exit ( [string $status] )
```

\$status 是要顯示的錯誤訊息。例如，當使用者所開啓的資料庫錯誤，或是密碼輸入錯誤等，都可使用此函式結束程式的執行。

header()

header()函式可讓網頁轉向，或是設定網頁的編碼。其簡易語法如下：

```
void header ( string $string)
```

例如，以下敘述可讓網頁導向與本網頁同資料夾的 aa.php。

```
header ( "Location:aa.php" );
```

以下敘述可讓網頁導向www.goodbooks.idv.tw。

```
header( "Location:http://www.goodbooks.idv.tw" );
```

以下敘述可設定網頁使用 UTF8 編碼。

```
header( "Content-Type:text/html;charset=utf-8" );
```

其次，使用 header()前，應先確認沒有任何的 HTML 或 echo 輸出，否則將會造成執行錯誤。例如，以下敘述已經輸出 HTML，當然無法使用 header 轉址。

```
<html>
```

```
<head>
```

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

```
<title>無標題文件</title>
```

```
</head>
```

```
<?
```

```
echo "<script>location.replace('index.php')</script>";
```

```
//本例無法使用以下方式轉址，因為已經輸出 Html
```

```
//header("Location:index.php");//網頁轉址到 index.php
?>
<body>
</body>
</html>
```

isset()

檢驗某一變數是否存在。其簡易語法如下：

```
bool isset ( mixed $var)
```

empty()

檢驗某一變數是否為 0，或空字串。其語法如下：

```
bool empty ( mixed $var )
```

範例 2_4_5a 請寫一程式，可以依照星期一～日，開啓不同的網頁。

操作步驟

1、先行準備 7 個網頁，p0.html~p6.html 如下圖，放在 e2_4_5a 資料夾。

網址(D) C:\AppServ\www\pbook\ch04\e4_4_5a			
資料夾	名稱	大小	類型
[-] pbook	p0	1 KB	HTML Document
[+] ch02	p1	1 KB	HTML Document
[+] ch03	p2	1 KB	HTML Document
[-] ch04	p3	1 KB	HTML Document
e4_4_5a	p4	1 KB	HTML Document
ch05	p5	1 KB	HTML Document
game	p6	1 KB	HTML Document

程式撰寫如下，請留意網址為 /，若寫成 \ 則有些作業系統無法順利執行。

2、

```
<?php
    $today=getdate(); //取得日期
    $wday=$today["wday"]; //取星期幾
    $page="p".$wday.".html";
    //echo $page;
    header("Location:e2_4_5a\$page");
?>
```

2_5 陣列

陣列的宣告

大部分的程式語言都要宣告陣列的大小與型態，但是 PHP 並不用事先宣告。

I 陣列的初值設定

陣列的初值設定可配合 `Array` 函式設定，例如，以下敘述設定 `a` 陣列的內容，分別是 `$a[0]=0`、`$a[1]=1`、`$a[2]=2`、`$a[3]=3`、`$a[4]=4`。

```
$a= Array(0,1,2,3,4);
```

```
echo $a[0];
```

則其值為 0。以下敘述則設定字串陣列的初值

```
$b=array("aa","bb","cc");
```

完成以上設定，則 `$b[1]="bb"`。其次，亦可數值與字串混和儲存，例如，

```
$c=array("aa",1,"cc");
```

則 `$c[2]="cc"`。更神奇的是陣列亦可繼續增加其容量，例如，繼續設定

```
$c[3]=3;
```

或

```
$c[]=3; //效果同$c[3]=3;
```

則其容量即為 4，所以此時輸出其長度則為 4，如以下敘述。

```
echo(count($c));
```

陣列的索引

大部分的程式語言都是使用整數索引陣列，但是 PHP 卻可使用字串索引。例如，

```
$d["name"]="aa";$d["chi"]=88;$d["eng"]=99;$d["math"]=77;
```

```
echo $d["chi"];
```

則其結果是 88。

陣列的輸出

陣列的輸出可使用兩種方式，分別是 `for` 與 `foreach`

`for`

使用 `for` 輸出 `$a` 陣列的方法如下：

```
for ($i=0 ;$i<=count($a)-1;$i++)
```

```
echo $a[$i]."<br>";
```

`foreach`

使用 `foreach` 輸出 `$a` 陣列的方法如下：

```
foreach($a as $avalue)
```

```
echo $avalue."<br>";
```

以下敘述可輸出索引與值。

```
foreach($a as $key=>$avalue)
```

```
echo "a[\".$key.\"]=\".$value.\"<br>";
```

其結果如下：

```
a[0]=0  
a[1]=1  
a[2]=2  
a[3]=3  
a[4]=4
```

範例 2_5a

假如有資料如下

55,66,77,88,99

(1)請用陣列儲存以上資料。

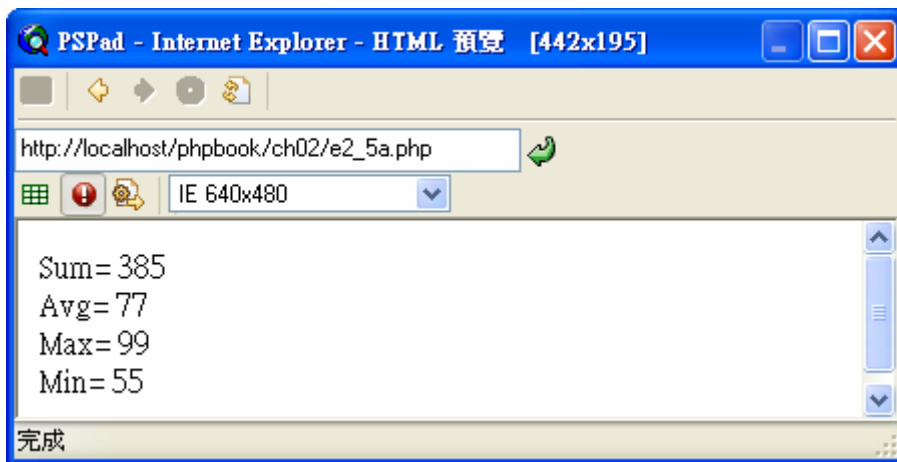
(2)請統計以上資料的和，並輸出。

(3) 請統計以上資料的平均，並輸出。

(4) 請統計以上資料的極大值，並輸出。

(5) 請統計以上資料的極小值，並輸出。

輸出結果



程式列印

<?php

```
//設定陣列初值
```

```
$a=Array(55,66,77,88,99);
```

```
$max=$a[0];
```

```
$min=$a[0];
```

```
$sum=$a[0];
```

```
$n=count($a);//陣列個數
```

```
for ($i=1 ;$i<=$n-1;$i++) {
```

```
    if ($a[$i]> $max)
```

```

        $max=$a[$i];
    if ($a[$i] <$min)
        $min=$a[$i];
    $sum+=$a[$i];
}
$avg=$sum/$n;
echo "Sum= $sum "."<br>";
echo "Avg= $avg "."<br>";
echo "Max= $max "."<br>";
echo "Min= $min "."<br>";
?>

```

自我練習

1、設有資料如下：

-1, 0, 3, 4, -5, 8

(1)請以陣列儲存以上資料。

(2)統計正數、0、負數的個數。

2、設有\$a 陣列資料如下：

3,5,6,7,8

\$b 陣列資料如下：

1,2,5,6,8,9

請寫程式找出重複的資料。

3、設有\$a 陣列資料如下：

3,5,6,7,8,3,9,1,6,3,7

請將重複的資料刪除。

I 二維陣列的初值設定

二維陣列的初值設定可使用巢狀 **array** 函式。例如，假設有表格資料如下：

| | | | | |
|---|----|----|----|----|
| 0 | 91 | 92 | 93 | 94 |
| 1 | 81 | 82 | 83 | 84 |
| 2 | 71 | 72 | 73 | 74 |

則其 PHP 設定初值程式如下：

```

$a=array (array(0,91,92,93,94),
          array(1,81,82,83,84),
          array(2,71,72,73,74));

```

且其索引亦預設從 0 開始，所以

```
echo $a[0][0];
```

的值為 0。以下敘述可用 for 輸出\$a 陣列。

```

for ($i=0;$i<=2;$i++)
{
    for ($j=0;$j<=4;$j++)
        echo $a[$i][$j];
    echo "<br>";
}

```

其結果如下：

```

091929394
181828384
271727374

```

以下敘述可用 foreach 輸出\$a 陣列，結果同上。

```

foreach($a as $avalue ){
    foreach ($avalue as $aavalue)
        echo $aavalue;
    echo "<br>";
}

```

以下敘述可用表格輸出以上\$a 陣列。

```

<table border=1>
<?php
    for ($i=0;$i<=2;$i++)
    {
        for ($j=0;$j<=4;$j++)
            echo "<td>".$a[$i][$j];
        echo "<tr>";
    }
?>
</table>

```

其結果如下：

| | | | | |
|---|----|----|----|----|
| 0 | 91 | 92 | 93 | 94 |
| 1 | 81 | 82 | 83 | 84 |
| 2 | 71 | 72 | 73 | 74 |

以下敘述可用 foreach 配合表格輸出二維陣列，其結果同上。

```

<table border=1>
<?php
    foreach($a as $avalue ){

```

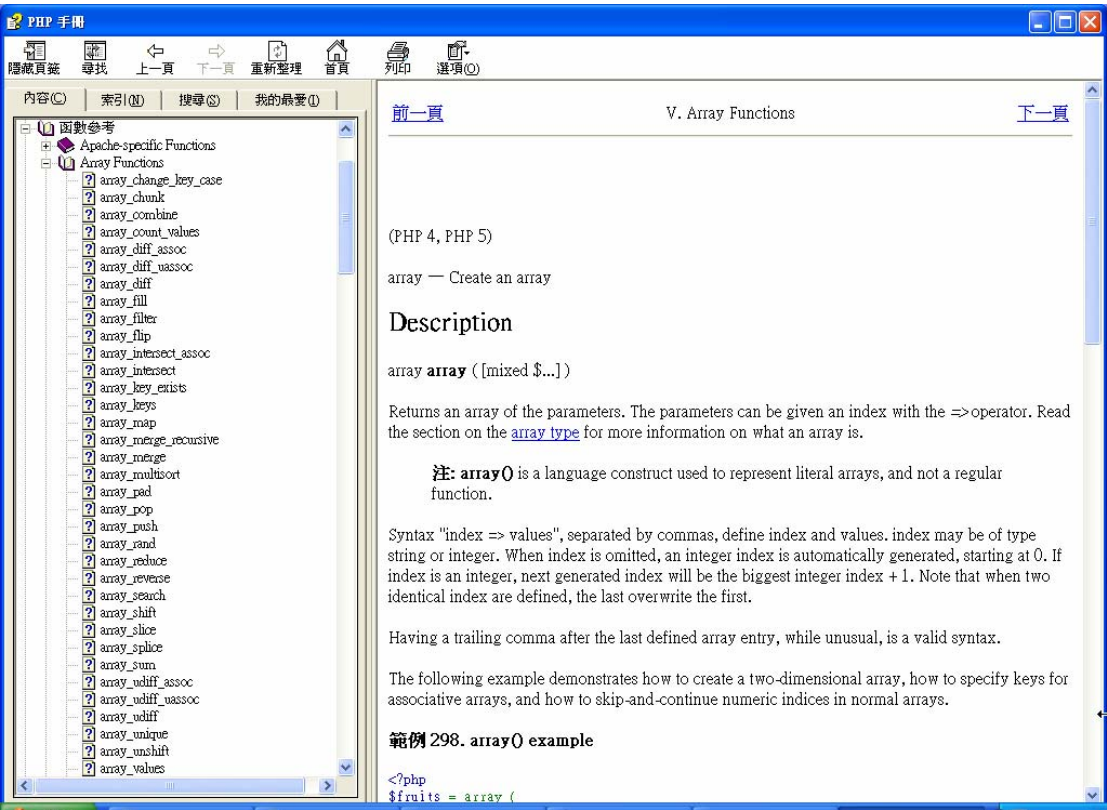
```

        foreach ($avalue as $aavalue)
            echo "<td>".$aavalue;
        echo "<tr>";
    }
?>
</table>

```

陣列函式

PHP 的陣列函式如下圖：



之前我們已經使用 array 與 count 兩個函式，以下我們將再介紹一些常用陣列函式。

array_sum

計算陣列的和。例如

```

$a= Array(0,1,2,3,4);
$sum=array_sum($a);
echo $sum;

```

結果是 10。

sort

將指定陣列由小而大排序。例如，

```

$a= Array(6,1,2,3);

```

```
sort($a);
foreach($a as $avalue)
    echo $avalue;
```

的結果是

1236

rsort

將指定陣列由大而小排序。例如，

```
$a= Array(6,1,2,3);
ksort($a);
foreach($a as $avalue)
    echo $avalue;
```

的結果是

6321

字串則是以 ASCII 的先後順序，ASCII 較小的在前面。例如，

```
$a= Array("aa","cc","BB","AA");
sort($a);
foreach($a as $avalue)
    echo $avalue;
```

的結果是 AABBaabb。中文的排序則以筆畫多寡為排序依據；例如，

```
$a= Array("洪國勝","陳惠敏","王一風");
sort($a);
foreach($a as $avalue)
    echo $avalue;
```

的結果是

王一風洪國勝陳惠敏

array_search()

陣列的搜尋，其語法如下：

```
mixed array_search ( mixed $needle, array $haystack [, bool $strict] )
```

例如，以下敘述可於\$a陣列中搜尋"cc"，若存在則傳回其位置。

```
$a= Array("aa","cc","BB","AA","cc");
$r=array_search("cc",$a) ;
echo $r;    //1
```

array_keys()

陣列的搜尋，剛剛的 array_search()僅傳回符合規定的第一個元素位置，array_keys()則以陣列的形式傳回全部的位置。其語法如下：

array **array_keys** (array \$input [, mixed \$search_value [, bool \$strict]])

例如，

```
$a= Array("aa","cc","BB","AA","cc");  
$r=array_keys($a,"cc") ;    //$a 字串的位置與 array_search 相反  
foreach ($r as $key => $rvalue)  
    echo " r['.$key.']='.$rvalue."<br>;
```

的結果如下：

r[0]=1

r[1]=4

共兩個元素。

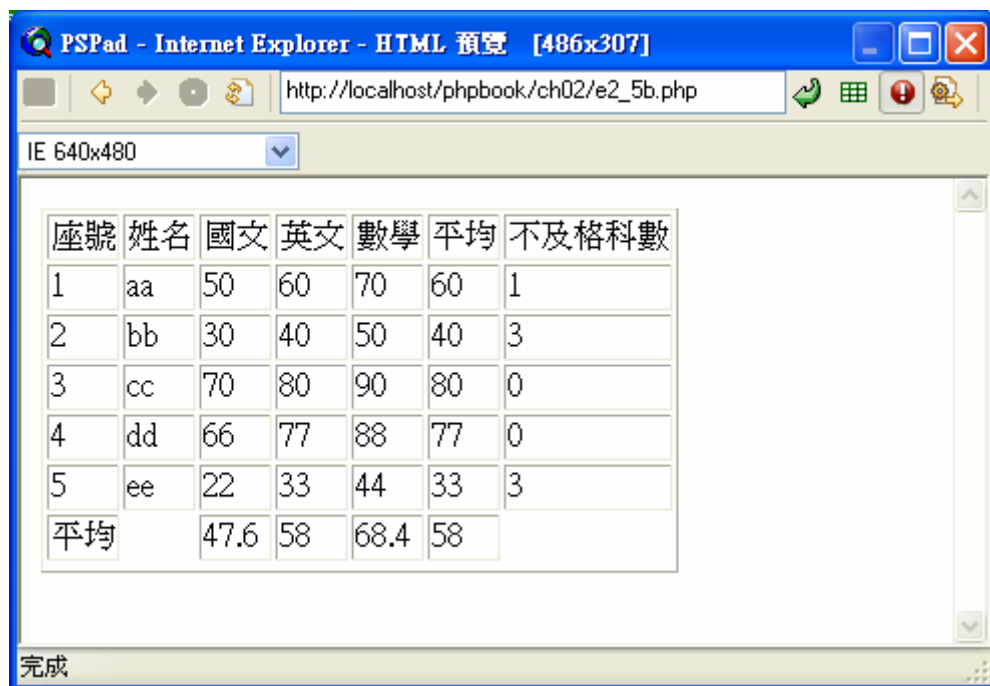
範例 2_5b

假設有學生成績如下：

| 座號 | 姓名 | 國文 | 英文 | 數學 | 平均 | 不及格科數 |
|----|----|----|----|----|----|-------|
| 1 | aa | 50 | 60 | 70 | | |
| 2 | bb | 30 | 40 | 50 | | |
| 3 | cc | 70 | 80 | 90 | | |
| 4 | dd | 66 | 77 | 88 | | |
| 5 | ee | 22 | 33 | 44 | | |
| 平均 | | | | | | |

1. 請以二維陣列儲存以上資料。
2. 計算每人平均。
3. 統計每人不及格科數。
4. 統計各科平均。

執行結果



操作步驟

1、陣列初值設定如下：

```
$a=array(array("座號","姓名","國文","英文","數學","平均","不及格科數"),
array(1,"aa",50,60,70,0,0),
array(2,"bb",30,40,50,0,0),
array(3,"cc",70,80,90,0,0),
array(4,"dd",66,77,88,0,0),
array(5,"ee",22,33,44,0,0),
array("平均"," ",0,0,0,0," "));
```

則其索引值如下：

| 座號 | 姓名 | 國文 | 英文 | 數學 | 平均 | 不及格科數 |
|-----|-----|-----|-----|-----|-----|-------|
| 0,0 | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 |
| 1 | aa | 50 | 60 | 70 | | |
| 1,0 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 |
| 2 | bb | 30 | 40 | 50 | | |
| 2,0 | 2,1 | 2,2 | 2,3 | 2,4 | 2,5 | 2,6 |
| 3 | cc | 70 | 80 | 90 | | |
| 3,0 | 3,1 | 3,2 | 3,3 | 3,4 | 3,5 | 3,6 |
| 4 | dd | 66 | 77 | 88 | | |
| 4,0 | 4,1 | 4,2 | 4,3 | 4,4 | 4,5 | 4,6 |
| 5 | ee | 22 | 33 | 44 | | |
| 5,0 | 5,1 | 5,2 | 5,3 | 5,4 | 5,5 | 5,6 |
| 平均 | | | | | | |

| | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|
| 6,0 | 6,1 | 6,2 | 6,3 | 6,4 | 6,5 | 6,6 |
|-----|-----|-----|-----|-----|-----|-----|

程式列印

```
<?php
```

```
//二維陣列初值
```

```
$a=array(array("座號","姓名","國文","英文","數學","平均","不及格科數"),
          array(1,"aa",50,60,70,0,0),
          array(2,"bb",30,40,50,0,0),
          array(3,"cc",70,80,90,0,0),
          array(4,"dd",66,77,88,0,0),
          array(5,"ee",22,33,44,0,0),
          array("平均"," ",0,0,0,0," "));
```

```
//計算每人平均
```

```
for ($i=1;$i<=5;$i++) {
    for ($j=2;$j<=4;$j++)
        $a[$i][5]+=$a[$i][$j];
    $a[$i][5]/=3;
}
```

```
//計算每人不及格科數
```

```
for ($i=1;$i<=5;$i++)
    for ($j=2;$j<=4;$j++)
        if ($a[$i][$j] <60)
            $a[$i][6]++;
```

```
//計算每科平均
```

```
for ($i=2;$i<=5;$i++) {
    for ($j=1;$j<=5;$j++)
        $a[6][$i]+=$a[$j][$i];
    $a[6][$i]/=5;
}
```

```
//以表格輸出結果
```

```
echo"<table border=1>";
foreach($a as $avalue ){
    foreach ($avalue as $aavalue)
        echo "<td>".$aavalue;
    echo "<tr>";
}
echo "</table>";
```

?>

習題

1、同上範例，但增加計算每人平均、每科不及格人數、每科最高分、每科最低分。

2、假設某一公司有三個業務員，其每季的業績如下：

編號	第一季	第二季	第三季	第四季	和	零的個數
1	0	2	4	6		
2	8	0	2	1		
3	7	0	0	9		
和						
零的個數						

(1)請以陣列儲存以上資料。

(2)計算每一個人的業績總和及零的個數。

(3)計算每一季的業績總和及零的個數。

3、請寫一網頁，可以模擬樂透開獎。也就是電腦可輸出七個 1 到 42 的不重複亂數，第七個亂數稱為特別號。

4、假設以下資料是張三預先買的八張樂透彩券號碼：

編號	1	2	3	4	5	6
1	2	9	17	18	22	34
2	2	9	15	17	22	34
3	2	9	17	19	22	34
4	1	2	9	18	22	34
5	2	3	9	17	18	19
6	9	15	17	18	22	28
7	1	2	3	9	17	18
8	17	18	19	20	21	22

A、請以二維陣列儲存以上資料。

B、開獎的七個號碼分別是 2,9,17,18,22,34,15，15 是特別號，請以一維陣列儲存。

C、請核對每張彩券的得獎情形，並以表格輸出結果。

2-6 檔案

PHP 的檔案操作均經由檔案函式完成，本節將介紹這些檔案相關函式，線上查詢如下圖所示：



\$_SERVER

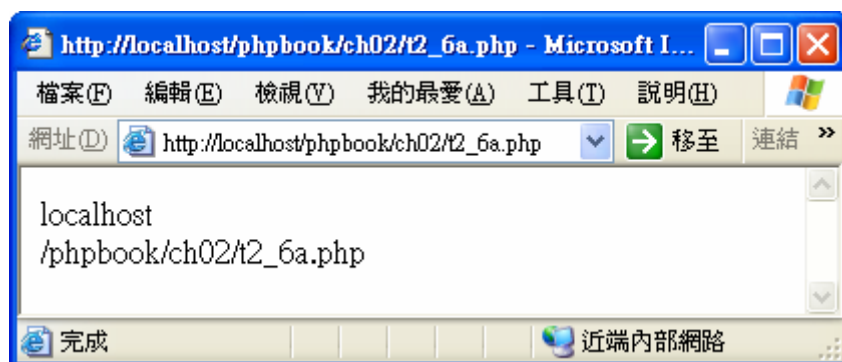
PHP 的全域陣列變數，用來取得伺服器的系統資訊。例如，

```
$name=$_SERVER["SERVER_NAME"];
echo $name."<br>";
```

可取得伺服器網域名稱或 IP 位址。又例如，

```
$path=$_SERVER["PHP_SELF"];
echo $path."<br>";
```

可取得本身 PHP 程式的路徑與名稱。以上程式(t2_6a)執行如下圖。



file_exists()

檢查檔案是否存在，其語法如下：

```
bool file_exists ( string filename)
```

例如，以下敘述可檢查 counter.txt 是否存在。(本例 counter.txt 必須與程式放在同一資料夾，才可得 true)

```
$file="counter.txt";
echo file_exists($file)."<br>";
```

copy()

複製檔案，其語法如下：

```
bool copy ( string source, string dest)
```

source 是來源字串，**dest** 是目的字串。例如，以下敘述可將 **counter.txt** 複製成 **counter.bak**。

```
$file="counter.txt";  
echo(copy($file,"counter.bak"));
```

其次，檔案的上傳即是使用 **copy()** 函式。例如，以下敘述可將使用者電腦的檔案 "d:\\test\\aa.txt" 上傳到伺服器，且其目的資料夾同程式的資料夾。

```
copy ("d:\\test\\aa.txt","aa.txt");
```

rename()

檔案更名，其語法如下：

```
bool rename ( string oldname, string newname )
```

例如，以下敘述可將 **counter.bak** 更名為 **coun.txt**。

```
echo(rename("counter.bak","coun.txt")."<br>");
```

unlink()

刪除檔案，其語法如下：

```
bool unlink ( string filename)
```

例如，以下敘述可將 **coun.txt** 刪除。

```
echo(unlink("coun.txt")."<br>");
```

fopen()

開啓檔案，其語法如下：

```
resource fopen ( string filename, string mode)
```

resource 是一種檔案串流，您可以想像成水流一樣，**fopen** 就如同打開水龍頭開關。**mode** 的模式如下表：

<i>mode</i>	Description
'r'	Open for reading only; place the file pointer at the beginning of the file.
'r+'	Open for reading and writing; place the file pointer at the beginning of the file.
'w'	Open for writing only; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.
'w+'	Open for reading and writing; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.
'a'	Open for writing only; place the file pointer at the end of the file. If the file does not exist, attempt to create it.
'a+'	Open for reading and writing; place the file pointer at the end of the file. If the file does not exist, attempt to create it.

例如，以下敘述可以以可讀可寫的狀態開啓 **aa.txt**，且檔案指標在最前面，也就是之前的檔案將會被清除。

```
$file="aa.txt";
```

```
$resource=fopen($file,"w+");
```

以下敘述則是以唯寫的方式開啓 aa.txt，且檔案指標放在檔案的結尾，也就是資料將接續放在原檔案後面。

```
$file="aa.txt";
```

```
$resource=fopen($file,"a");
```

```
fclose()
```

關閉串流檔案，其語法如下：

```
bool fclose ( resource handle)
```

例如，以下敘述可關閉已經開啓的 aa.txt 檔案。

```
fclose($resource);
```

```
fwrite()
```

將字串寫入指定的檔案串流。其語法如下：

```
int fwrite ( resource handle, string string [, int length])
```

handle 是串流名稱，string 是即將寫入的字串，length 是寫入的長度，若 length 大於 string 則以 length 的長度為限。最後 PHP 傳回寫入資料的長度。例如，以下程式(t2_6b)可將”專題製作”寫入檔案。

```
<?php
```

```
    $file="aa.txt";
```

```
    $string="專題製作";
```

```
    $resource=fopen($file,"w+");
```

```
    fwrite($resource,$string);
```

```
    fclose($resource);
```

```
?>
```

以上程式執行後，以記事本開啓 aa.txt 結果如下：



以下程式(t2_6c)則可將” 專題製作” 寫入兩次。其中，\r\n 表示換行。

```
<?php
```

```
    $file="aa.txt";
```

```
    $string="專題製作\r\n"; //\r\n 表示換行
```

```
    $resource=fopen($file,"w+");
```

```
    fwrite($resource,$string);
```

```
    fclose($resource);
```

```
    $resource=fopen($file,"a");
```

```
    fwrite($resource,$string);
```

```
fclose($resource);  
?>
```

fread()

檔案讀取函式，其語法如下：

string **fread** (resource handle, int length)

length 為檔案長度，可由 filesize()函式取得。例如，以下程式(t2_6d)可取得 aa.txt 的內容。

```
<?php
```

```
    $file="aa.txt";  
    $length=filesize($file);  
    $resource=fopen($file,"r");  
    $string=fread($resource,$length);  
    echo nl2br($string);  
    fclose($resource);
```

```
?>
```

nl2br()可將原來 \r\n 的換行資料轉為 HTML 的換行
，所以執行結果如下：



範例 2-6a 請寫一網頁，此網頁可顯示所有使用者的瀏覽次數，也就是訪客計數器。

執行結果



程式列印

```
<?php
```

```
    $file="counter.txt";  
    if ( file_exists($file)) {
```

```
$resource=fopen($file,"r");
$counter=fread($resource,filesize($file));
fclose($resource);
}
else {
    $counter=0;
    $resource=fopen($file,"w");
    fwrite($resource,$counter);
    fclose($resource);
}
//已經取得$counter
$counter++;
//顯示
echo($counter);
//存回
$resource=fopen($file,"w");
fwrite($resource,$counter);
fclose($resource) ;
```

?>