陣列(變數)概念的產生

在重覆結構單元中，我們可以瞭解到，如果我們所要處理的一系列資料中，存在有一個相同的規律性時，我們就可以應用重覆結構的功能指令，輕鬆的完成該項工作。我們以下表來做一說明：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| $x | $a  =2\*$x-1 | $b  =11-$x | $c  =6-$x | $d  =abs(6-$x) | $sco[$x] | $mda[$x] |
| 1  2  3  4  5  6  7  8  9  10 | 1  3  5  7  9  11  13  15  17  19 | 10  9  8  7  6  5  4  3  2  1 | 5  4  3  2  1  0  -1  -2  -3  -4 | 5  4  3  2  1  0  1  2  3  4 | 86  75  93  80  55  68  72  70  88  61 | 31  29  31  30  31  30  31  31  30  31 |

只要能轉換成1-10這一類型就可以使用陣列應用for搞定，同樣的也可以想成學生的座號資料等等。

ch11-01.php:規律性資料的應用

<?php

Echo "<table border='1'>";

Echo "<tr><td>\$x</td><td>\$a=2\*\$x-1 </td><td>\$b=11-\$x </td><td>\$c=6-\$x </td><td>\$d= abs(6-\$x)</td></tr>";

For($x=1;$x<=10;$x++)

{

$a=2\*$x-1;

$b=11-$x;

$c=6-$x;

$d=abs(6-$x);

Echo "<tr align='center'><td width='30'>".$x."</td><td>" .$a."</td><td>" .$b."</td><td>" .$c."</td><td>" .$d."</td></tr>";

}

Echo "</table>";

?>

在上面ch11-01.php的例子中，我們只執行了，$a , $b , $c , $d 四欄的資料，並没有執行到 $sco[$x] 跟 $mda[$x] 這二欄的資料。因為，要直接從

「86,75,93,80,55,68,72,70,88,61」或「31,29,31,30,31,30,31,31,30,31」這二個資料系列中找出與「1,2,3,4,5,6,7,8,9,10」的數學對應關係是乎有點問題。

不過，從另一個位置的角度來看，我們發現，上面二個資料系列的

1. 資料存放位置有順序性
2. 同一系列的各個資料，相互之間存在有各式各樣相關的處理動作

針對上述性質，php提供了「陣列變數」功能，讓我們能夠很有效率的解決各種相關的工作。

「陣列變數」的方法就是，將存放這些資料的所有變數名稱設成同一名稱，再以[X]來標示資料的存放順序。以上面的資料系列為例，我們設定存放資料的變數名稱如下：

$sco[1]=86

$sco[2]=75

$sco[8]=70

$sco[9]=88

$sco[10]=61

再說一遍，上面實際上是有10個變數，只不過，我們將變數名稱設為相同，再以[x]中的x來做分辨。

Php 提供另一個設定陣列變數的方法,如下：（詳細規則，後面再作敍述）

$sco=array(0, 86,75,93,80,55,68,72,70,88,61);

ch11-02.php 陣列資料的應用

<?php

$sco=array(0, 86,75,93,80,55,68,72,70,88,61);

$mda=array(0, 31,29,31,30,31,30,31,31,30,31);

Echo "<table border='1'>";

Echo "<tr><td>\$x</td><td>\$sco[\$x] </td><td>\$mda[\$x] </td></tr>";

For($x=1;$x<=10;$x++)

{

Echo "<tr align='center'><td width='30'>".$x."</td> <td>" .$sco[$x]."</td> <td>" .$mda[$x]."</td></tr>";

}

Echo "</table>";

?>

ch11-03.php(我們將上面程式再做一些包裝)

<?php

$sco=array(0, 86,75,93,80,55,68,72,70,88,61);

Echo "<table border='1'>";

Echo "<tr><td>座號</td><td>國文 </td></tr>";

$sum=0; //預設總分為0

$avg=0; //預設平均為0

For($x=1;$x<=10;$x++)

{

Echo "<tr align='center'><td width='30'>".$x."</td> <td>" .$sco[$x]."</td> </tr>";

$sum+=$sco[$x];

}

$avg=$sum/10;

Echo "<tr><td>總分</td><td>".$sum."</td></tr>";

Echo "<tr><td>平均</td><td>".$avg."</td></tr>";

Echo "</table>";

?>

陣列(變數)的應用(參考教材)

※ 萬年曆範例(ex10-01-07-1.php)

ch11-04.php (橫列思考)

<?php

$data=array("陳大同",86,75,90,63,0,0); //後5,6兩欄為總分及平均的預設值

Echo "<table border='1'>";

Echo "<tr align='center'><td>座號</td><td>姓名</td><td>國文</td><td>英文 </td><td>數學</td><td>化學</td><td>總分</td><td>平均</td></tr>";

Echo "<tr align='center'><td width='80'>1</td><td width='80' >" .$data[0] . "</td>";

For($x=1;$x<=4;$x++)

{

Echo "<td width='80'> " .$data[$x]."</td> ";

$data[5]+=$data[$x];

}

$data[6]=(int)( $data[5]/4+0.5); //平均為 四拾五入 取整數

Echo "<td width='80' >". $data[5]."</td><td width='80' >". $data[6]."</td></tr>";

Echo "</table>";

?>

**在PHP的陣列應用可以說是非常活潑，下面我們逐項來做說明：**

一維陣列資料設定的方式

格式：

$陣列名稱[索引編號]=資料內容；

說明：

　１．如果没有指定索引編號，PHP如果没有指定編號會自動由０開始編列：

　　　<?Php

$name[]=” 張三”; //與$name[0]=” 張三”; 同義

$name[]=” 李四”; //與$name[1]=” 李四”; 同義

$name[]=” 王五”; //與$name[2]=” 王五”; 同義

?>

　２．若在指定某一個索引編號後，後續卻没有繼續指定編號，PHP會自動遞增：

　　　<?Php

$name[9]=” 張三”;

$name[]=” 李四”; //與$name[10]=” 李四”; 同義

$name[]=” 王五”; //與$name[11]=” 王五”; 同義

?>

　３．索引編號不需依序建立，陣列資料類型不需相同（本功能最好不用）

　　　<?Php

$name[5]=” 張三”; //字串資料

$name[]=” 李四”; //與$name[6]=” 李四”; 同義

$name[9]=86; //數字資料

?>

　４．索引編號可以是字串資料

　　　<?php

　　　　$myArray['myName'] = 'David';

　　　　$myArray['myHeight'] = 181;

　　　　$myArray['myWeight'] = 78;

　　　　echo "大家好，我的名字叫".$myArray['myName']."。我的身高　　　".$myArray['myHeight']."公分，體重".$myArray['myWeight']."公斤。";

　　　?>

　注意：索引編號如果是用字串，建議用　(‘ ) 來替代　( ＂)。

以array設定陣列初值

PHP提供array指令，用來設定陣列資料內容，有二種用法說明如下：

　＄陣列名稱=array(資料1, 資料2, ……., 資料 n)；

　＄陣列名稱=array(索引1=>資料1, 索引2=>資料2, ……., 索引n=>資料 n)；

例：分別以一般方式和array方式，指定三個陣列，並利用for重覆指令，將陣列資料內容顯示出來。

<html>

<head>

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

<title>陣列資料</title>

</head>

<body>

<?php

//指定第一個陣列

$name[0]="Linda";

$name[1]="Alice";

$name[2]="Elvin";

$name[3]="Peter";

$name[4]="Caroline";

//以array指定第二個陣列

$value=array(123,456,789,566,888);

//以array指定第三個陣列

$address=array(0=>"台北",1=>"新竹",2=>"台中",3=>"台南",4=>"高雄");

for($a=0;$a<5;$a++){

echo "<p><font color='blue'>第".$a."筆資料 : </font>";

echo $name[$a]." , ";

echo $value[$a]." , ";

echo $address[$a]."</p>";

}

?>

</body>

</html>

* 陣列變數在表單功能上的應用(以表單傳送方式設定陣列資料)

我們先修改ch11-04.php 中，陣列資料的設定( $data=array("陳大同",86,75,90,63,0,0); )如下：

$data[0]=$\_POST[“data0”];

$data[1]=$\_POST[“data1”];

$data[2]=$\_POST[“data2”];

$data[3]=$\_POST[“data3”];

$data[4]=$\_POST[“data4”];

$data[0]=”陳大同”;

$data[1]=”86”;

$data[2]=”75”;

$data[3]=”90”;

$data[4]=”63”;

没錯，原始資料應該是由使用者直接輸入，程式才能顯現它的「效率及靈活性」(而不是讓程式設計者在程式中設定)。

所以我們要先設定一個表單輸入程式，如下：

<html>

<head>

<title>成績輸入程式</title>

</head>

<body>

<h2>請輸入下列資料</h2>

<form method="post" action="ex4-5out.php">

<table border="1">

<tr><td>座號</td><td>姓名</td><td>國文</td><td>英文</td><td>數 學</td><td>化學</td></tr><tr>

<td><input type="text" name="data0" size="8"></td>

<td><input type="text" name="data1" size="8"></td>

<td><input type="text" name="data2" size="8"></td>

<td><input type="text" name="data3" size="8"></td>

</tr></table>

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

<input type="reset" name="reset" value="重填">

</form>

</body>

</html>

請注意到，上面程式中標示「紅色」的地方，我們可以應用「廻圈」及「陣列」的功能來簡化程式的篇幅。修改如下：

* 程式中有用到php的指令，所以在存檔時，檔案類型要設定為「\*.php」

ch11-05in.php

<html>

<head>

<title>成績輸入程式</title>

</head>

<body>

<h2>請輸入下列資料</h2>

<form method="post" action="ch11-05out.php">

<table border="1">

<?php

/////////////////////////////////////////////////////////////////////////////////////////

// <tr><td>座號</td><td>姓名</td><td>國文</td><td>英文</td><td>數

// 學</td><td>化學</td></tr><tr>

////////////////////////////////////////////////////////////////////////////////////////

$tit=array("座號","姓名", "國文", "英文", "數學", "化學");

Echo "<tr>";

For($x=0;$x<=5;$x++)

{

Echo "<td>".$tit[$x]."</td>";

}

Echo "</tr><tr><td>1</td>";

/////////////////////////////////////////////////////////////////////////////////////////

// <td><input type="text" name="data0" size="6"></td>

// <td><input type="text" name="data1" size="6"></td>

// <td><input type="text" name="data2" size="6"></td>

// <td><input type="text" name="data3" size="6"></td>

/////////////////////////////////////////////////////////////////////////////////////////

For($x=0;$x<=4;$x++)

{

Echo "<td><input type='text' name='data".$x."' size='6'></td>";

}

?>

</tr></table>

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

<input type="reset" name="reset" value="重填">

</form>

</body>

</html>

ch11-05out.php

<?php

/////////// 接收資料(輸入) //////////////////////////////////////////////////

For($x=0;$x<=4;$x++)

{

$name="data".$x;

$data[$x]=$\_POST[$name]; //$data[$x]=$\_POST["data".$x] (看版本)

}

/////////// 計算總分及平均 (資料處理) /////////////////////////////////

For($x=1;$x<=4;$x++)

{

@$data[5]+=$data[$x];

//php規定所有被用來計算的變數，都要設定初值(看版本)，否則會出現

// notice 訊息，在指令前面加上@符號，即可避免 notice 訊息 的顯示

}

$data[6]=(int)( $data[5]/4+0.5); //平均為 四拾五入 取整數

/////////// 列印報表 (輸出) /////////////////////////////////

Echo "<table border='1'>";

////// 第一列(標題列) //////////////////////////////////////

$tit=array("座號", "姓名", "國文", "英文", "數學", "化學","總分","平均");

Echo "<tr align='center'>";

For($x=0;$x<=7;$x++)

{

Echo "<td>".$tit[$x]."</td>";

}

Echo "</tr>";

////// 第二列(資料列) //////////////////////////////////////

Echo "<tr align='center'><td width='80'>1</td>";

For($x=0;$x<=6;$x++)

{

Echo "<td width='80'> " .$data[$x]."</td> ";

}

Echo "</tr></table>";

?>

　　　程式中，計算總和及平均的工作是常見的功能。例如：銷售業績總和、薪資總和、加班費合計。。。。等

單元作業１：仿上例，將ch11-03.php 修改成由表單輸入資料的程式

**多維陣列(二維以上) 參考教材**

多維陣列格式

＄陣列變數名稱［一維索引］［二維索引］．．．［n維索引］= 資料內容；

例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 姓名 | 身高 | ==> | 住址 |
| 個人(一維) | $psn[‘name’] | $psn[‘hig’] | $psn[‘add’] |
| 整班(二維) | $psn[1][‘name’] | $psn[1][‘hig’] | $psn[1][‘add] |
| 年級(三維) | $psn[‘甲’][1][‘name’] | $psn[‘甲’] [1][‘hig’] | $psn[‘甲’] [1][‘add] |
| 以此類推… | | | | |

如果要以array來設定多維陣列的資料內容，可以用多重array()設定如下：

＄per=array(1=>array(‘name’=>’姓名資料’ , ‘hig’=>’身高資料’ , ～, ‘add’=>’住址資料’) ,

2=>array(‘name’=>’姓名資料’ , ‘hig’=>’身高資料’ , ～, ‘add’=>’住址資料’) ,

3=>array(‘name’=>’姓名資料’ , ‘hig’=>’身高資料’ , ～, ‘add’=>’住址資料’))；,

例：以一般方式和array方式設定二維陣列資料的初始值，並用for重覆指令，將陣列中的資料顯示出來。

<html>

<head>

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

<title>多維陣列</title>

</head>

<body>

<?php

//指定二維陣列

$price[0][0]=12;

$price[0][1]=25;

$price[1][0]=36;

$price[1][1]=48;

$price[2][0]=11;

$price[2][1]=66;

//指定二維陣列

$data=array("name"=>array("Linda","Alice","Elvin"),

"value"=>array("123","456","789"),

"address"=>array("台北","台中","台南"));

//顯示第一個二維陣列

for($a=0;$a<3;$a++){

echo "<p>";

for($b=0;$b<2;$b++){

echo "\$price[$a][$b] = ".$price[$a][$b]."<br/>";

}

echo "</p>";

}

//以表格顯示第二個二維陣列

echo "<table bgcolor='#FFCCFF' border=2 width='70%'><tr>";

for($a=0;$a<3;$a++){

echo "<td>".$data["name"][$a]."</td>";

}

echo "</tr><tr>";

for($a=0;$a<3;$a++){

echo "<td>".$data["value"][$a]."</td>";

}

echo "</tr><tr>";

for($a=0;$a<3;$a++){

echo "<td>".$data["address"][$a]."</td>";

}

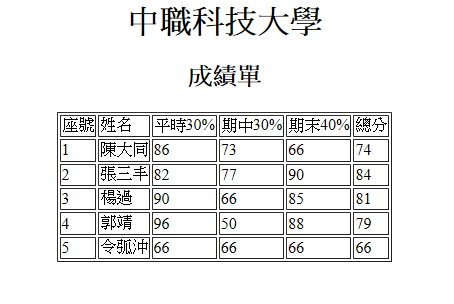
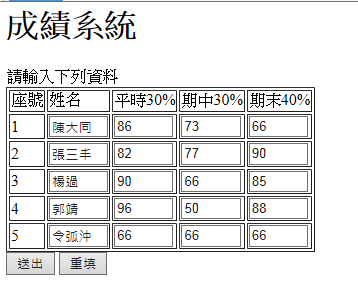
echo "</tr></table>";

?>

</body>

</html>

設計成績系統程式如下：



$x\$y

我們可以應用

二維表單(名稱:scor.$x.$y) 對應到 二維陣列(名稱: $scor[$x][$y] )

來設計程式如下：

ch11-06in.php

<html>

<head>

<title>

成績系統

</title>

</head>

<body>

<h1>成績系統</h1>

<?php

/\*

表單資料設計

表單名稱設為二維陣列變數型態:scor.$x.$y

[$x]代表座號，

[$y]表示項目，列示如后:[1]->姓名、[2]->平時、[3]->期中考、[4]->期末考

\*/

echo "請輸入下列資料<br>";

echo "<form method='post' action=' ch11-06out.php'>";

//列印表格化表單

Echo "<table border='1'>";

Echo "<tr><td>座號</td><td>姓名</td><td>平時30%</td><td>期中30%</td><td>期末40%</td></tr>";

for($x=1;$x<=5;$x++) //外圈為座號

{

echo "<tr><td>".$x."</td>";

for($y=1;$y<=4;$y++) //內圈為項目

{

echo "<td><input type='text' name='scor".$x.$y."' size='4'></td>";

}

echo "</tr>";

}

?>

</table>

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

<input type="reset" name="reset" value="重填">

</form>

</body>

</html>

ch11-06out.php

<?php

/\*

報表資料確認

班級資料設為二維陣列變數:$scor[$x][$y]

[$x]代表座號，

[$y]表示項目，列示如后:[1]->姓名、[2]->平時、[3]->期中考、[4]->期末考、[5]->學期成績

資料獲得方式確認

透過表單接收的資料：

班級各項資料設為二維陣列變數:$scor[$x][$y]，$y為1~4

運算獲得的資料：

總分:$scor[$x][5] (int)(($scor[$x][2]\*0.3+$scor[$x][3]\*0.3+$scor[$x][4]\*0.4)+0.5)

\*/

//////////////// 接收表單資料 及 計算 學期成績 ///////////////////////////

for($x=1;$x<=5;$x++) //$x代表座號

{

　for($y=1;$y<=4;$y++) //$y代表 姓名、平時、期中、期末

{

$midata="scor".$x.$y; //產生對應表單名稱 [scor11]~[scor54]

$scor[$x][$y]=$\_POST[$midata]; //接收對應表單名稱

}

　　　//計算學期分數

$scor[$x][5]=(int)($scor[$x][2]\*0.3+$scor[$x][3]\*0.3+$scor[$x][4]\*0.4+0.5);

}

///////////////// 列印報表

Echo "<center><h1>中職科技大學</h1>";

Echo "<h2>成績單</h2>";

Echo "<table border='1'>";

Echo "<tr><td>座號</td><td>姓名</td> <td>平時30%</td><td>期中30%</td><td>期末40%</td><td>總分</td></tr>";

//1~5號同學

for($x=1;$x<=5;$x++)

{

echo "<tr><td>".$x."</td>";

for($y=1;$y<=5;$y++)

{

if(($y==5) and ($scor[$x][$y]<60))

{

echo "<td><font color='red'>".$scor[$x][$y]."</font></td>";

}

else

{

echo "<td>".$scor[$x][$y]."</td>";

}

}

echo "</tr>";

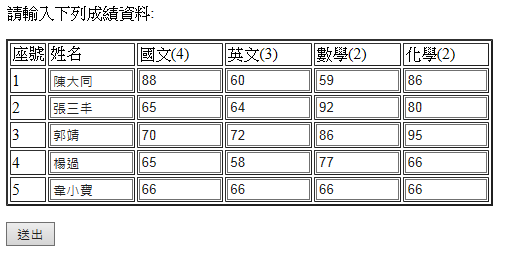
}

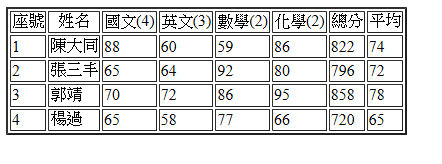
Echo "</table>";

?>

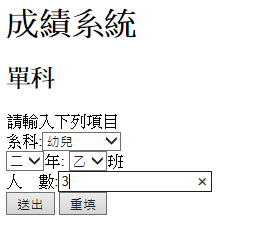
**綜合演練：(單元作業２系列)**

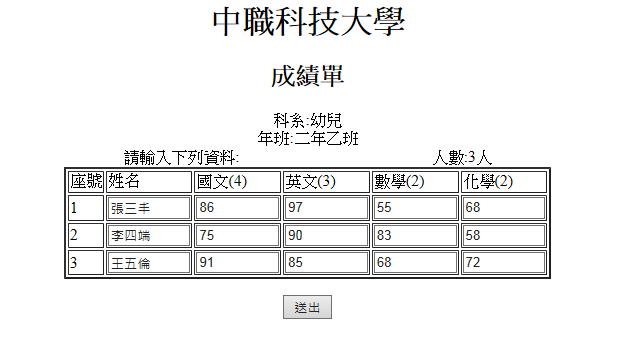
作業2-1：設計表單輸入程式，顯示畫面如下(ex11-02-1in.php、ex11-02-  
1out.php)

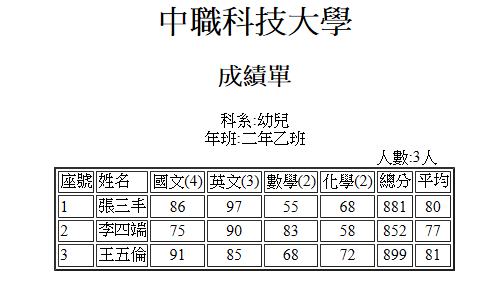




作業2-2：設計表單程式，畫面如下(ex11-02-2-01.php、ex11-02-2-02.php、ex11-02-2-03.php)

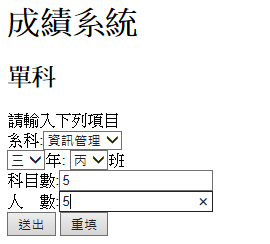


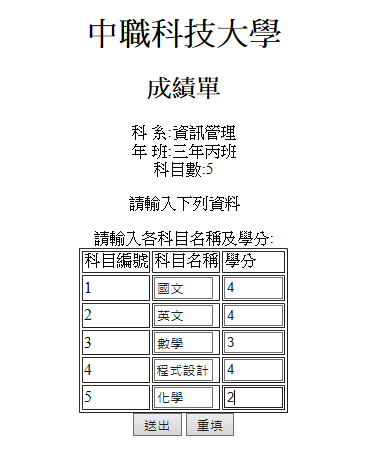


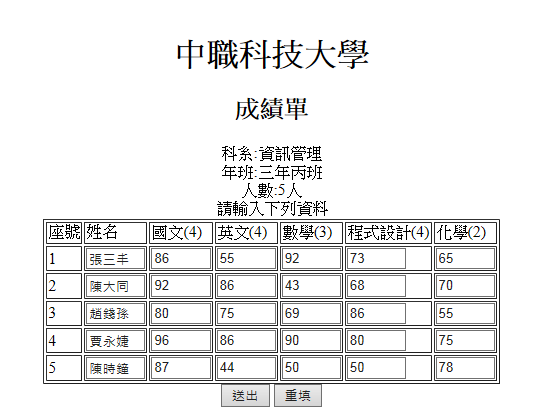


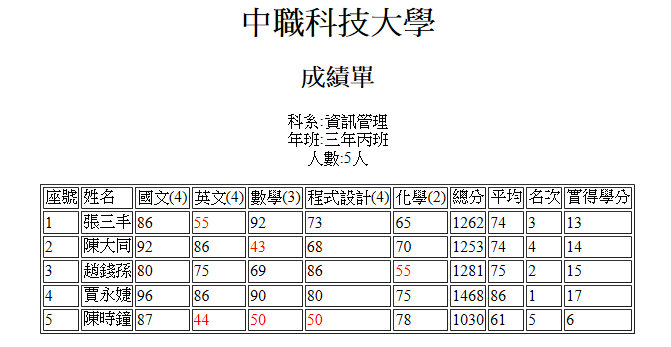
作業2-3：設計表單程式，畫面如下

(cousd3-1.php、cousd3-2.php、cousd3-2b.php、cousd3-3.php)









　作業2-3最後一個報表中，有二個欄位仍是空白。「實得學分」可用有條件的加總獲得。至於「名次」與「排順序」的功能，雖然php已經有提供對應的函數，不過，它的處理邏輯，非常值得我們學習，因此，下面我們稍作一點簡單的說明：

**排順序（泡沫排序法）：由小到大排列**

　針對n(10)個數字，例：86,75,93,80,55,68,72,70,88,61

第1次:拿第1數(86)與其他 (2~10) 數字逐一比較，若前數比後數大，則兩   
 數位置交換，當全部比完後，第1數會變成是最小的數。

第2次:拿第2數(86)與其他 (3~10) 數字逐一比較，若前數比後數大，則兩   
 數位置交換，當全部比完後，第2數會變成是第二小的數。

以此類推:

第9(n-1)次:拿第9(n-1)數(86)與最後一數(n/10)字比較，若前數比後數大，則  
兩數位置交換。

此時，上面數列的順序已經變成由小到大的排列了(55,61,68,70,72,75,80,86,88,93)。

|  |  |  |  |
| --- | --- | --- | --- |
| 原數列 | 第1次 | 第2次 | ……. |
| 86  75  93  80  55  68  72  70  88  61 | ~~86~~ ~~75~~ 55(最小數)  ~~75~~ 86  93  80  ~~55~~ 75  68  72  70  88  61 | 55  ~~86~~ ~~80~~ ~~75~~ ~~68~~ 61(第2小數)  93  ~~80~~ 86  ~~75~~ 80  ~~68~~ 75  72  70  88  ~~61~~ 68 |  |

ch11-07.php(泡沫排序)

<?php

$sco=array( 86,75,93,80,55,68,72,70,88,61); //陣列索引 0 ~ 9

echo "排序結果 (由小到大) 如下：<br>";

For($x=0;$x<=8;$x++) //外圈(0~n-2)，從第1筆到 倒數第2筆

{

For($y=$x+1;$y<=9;$y++) //內圈($x+1~n-1)，從第$x+1筆到最後一筆

{

If($sco[$x]>$sco[$y]) //如果「比較者數字」>「被比較者數字」，則

{

$borrow=$sco[$x]; //進行資料互換

$sco[$x]=$sco[$y];

$sco[$y]= $borrow;

}

}

Echo $sco[$x]." < ";

}

Echo $sco[$x];

?>

**設定名次**

我們常說：1.「只有你一個人(没跟別人比)，當然是第1名」

2.如果還有其他人，就要和其他所有人比過才能知道名次

所以，我們先設定一個二維陣列分別存放「分數 及 名次」如下：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 分數 | 86 | 73 | 90 | 55 | 81 | 70 | 66 | 93 | 68 | 70 |
| 名次 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

※還没比較前，每一位都是第1 名

接下來，開始進行比較(每1位，都要和其他所有人比過)，當有人分數高於比較者時，比較者的名次就要加1

程式如下：

ch11-08.php(設定名次)

<?php

// 設定陣列資料

$sco=array(0=>array(86,73,90,55,81,70,66,93,68,70),

1=>array(1,1,1,1,1,1,1,1,1,1)); //第2維 陣列索引 0 ~ 9

// 進行分數比較 及 名次比較

For($x=0;$x<=9;$x++) //外圈為比較者

{

For($y=0;$y<=9;$y++) //內圈為被比較者

{

If($sco[0][$x]<$sco[0][$y]) //如果比較者分數較小，則

{

$sco[1][$x]++; //名次加1

}

}

}

//列印報表資料

Echo "名次排定如下：<br>";

Echo "<table border='1' width='450'>";

For($x=0;$x<=1;$x++)

{

If($x==0)

{ Echo "<tr align='center'><td>分數</td>"; }

Else

{ Echo "<tr align='center'><td>名次</td>"; }

For($y=0;$y<=9;$y++)

{

Echo "<td>". $sco [$x][$y]."</td>";

}

Echo "</tr>";

}

Echo "</table>";

?>

單元作業3：運用設定名次的技巧，完成上面的習作2-3

**列印陣列的全部資料內容**

如果我們想列出某一 陣列 的全部資料內容時，可以使用下列的一些方法：2

１.當陣列索引值為數字，且由0開始(中間也没有缺漏資料)(1維陣列)

方法：先用count($陣列名稱)指令，計算陣列的長度，再用for廻圈  
列印陣列資料。

ch11-09.php

<?php

$sco=array( 86,75,93,80,55,68,72,70,88,61);

Echo "陣列資料列印如下：<br>";

For($x=0;$x<count($sco);$x++)

{ Echo $sco[$x]." ： "; }

?>

２.當陣列索引值不是數字時，可用Foreach指令來執行，有2種格式如下：

格式1：將陣列「資料值」放到自訂變數中，依序列出

　　Foreach(陣列名稱　as 自訂變數名稱 )

　　{

指令動作；

　　｝

格式2：陣列「索引值」放到 $key變數中，「資料值」放到 $value變數中列出

　　Foreach(陣列名稱　as $key => $value )

　　{

指令動作；

　　｝

　Foreach指令，會自動依陣列變數的元素數量做為執行次數，依序執行完畢才會跳出廻圈。

ch11-10.php

<html>

<head>

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

<title>foreach重覆結構</title>

</head>

<body>

<?php

//定義名稱為$price的一維陣列

$price=array(12,23,34,40,66);

echo "陣列中所含的資料有 : <br/>";

//以foreach結構列出所有的陣列值

foreach($price as $number)

{

echo $number." , ";

}

//列出索引值和相關資料

echo "<p>";

echo "另一種顯示陣列的方式 : <br/>";

foreach($price as $key => $number)

{

echo "\$price[$key] => ".$number."<br/>";

}

echo "</p>";

?>

</body>

</html>

本例中，$price=array(12,23,34,40,66); 指令可改成

$price=array("one"=>12,"two"=>23,"three"=>34,"fourth"=>40,"five"=>66);

再試看看程式執行結果。

３.多維陣列資料的列印，可用print\_r(陣列變數名稱)指令，本指令會以  
 「array設定格式」的方式列印陣列資料

ch11-11.php

<?php

$class=array("score" => array(70,95,70,60,88,78,95,88,92,100),

//$class[score[0]=70]

//$class[score[1]=[95]

//$class[score[2]=70]

//$class[score[3]=60]

"name" => array("Alice","Peter","Elvin","Sindy","Simon",

"Bob","Hank","Charles","Caroline","Linda"));

//$class[score,name]類似這種的感覺

//將索引值改成大寫英文字母

$class\_new=array\_change\_key\_case($class, CASE\_UPPER);

//使用print\_r函數以「array設定格式」的模式列印

print\_r($class\_new);

?>

* Php是一個開放的程式碼，因此，有很多的使用者陸續的研發設計各種方便好用的指令功能，而隨著版本的更新，PHP官網也將這些較成熟且好用的功能收錄，同學可自行上「官網」或應用「關鍵字」上網查詢。

下面，我們就一些較常用的功能做簡單介紹 (**參考教材**)

* 多維陣列資料排序功能，我們在後面資料庫單元，會再以實際範例再來做演練。

單元作業４：應用陣列功能搭配for廻圈指令，修改cssstyle銷售畫面的程式

(purses-arr.php、toy-arr.php、ex11-04-01.htm)

提示：產品資料可如下二維陣列編排

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| $y  $x | 產品編號 | 品名 | 圖案檔名 | 規格  …… | 單價 | 表單名稱 | … |
| 0 |  |  |  |  |  |  |  |
| 1 |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

圖案檔：bank-1.wmf、bank-11.wmf、bank-13.wmf、bank-37.wmf、bank-56.wmf、bank-123.wmf、bank-150.wmf、bank-152.wmf、bank-164.wmf、bank-176.wmf、01.jpg、02.jpg、03.jpg、04.jpg、05.jpg、06.jpg、07.jpg、08.jpg、09.jpg、10.jpg

與陣列變數處理相關的各項指令

List指令

List指令只適用於數值索引的陣列，且陣列的索引值必須是由0開始；list指令是另外設定變數，並將陣列變數的資料容，依序填入list 指令所指定的變數之中，下面直接以例子來做說明：

<?php

$info=array(“張三” , “經理” , 45000);

list ($name , $emp , $pay)=$info; //將陣列$info的資料依序放入變數$name , $emp , $pay中

echo “姓名：”.$name.”<br/>”;

echo “職務：”.$emp.”<br/>”;

echo “薪資：”.pay.”<br/>”;

?>

例：設定一個陣列，以list指令將陣列中的每一個元素，轉存到list指令所指定的變數，另外設定一個陣列，然後再以while 及list指令，將陣列的資料顯示出來。

ch11-12.php

<html>

<head>

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

<title>list 結構</title>

</head>

<body>

<?php

//list第一種應用

$info=array(“張三” , “經理” , 45000);

list ($name , $emp , $pay)=$info; //將陣列$info的資料依序放入變數$name , $emp , $pay中

echo “姓名：”.$name.”<br/>”;

echo “職務：”.$emp.”<br/>”;

echo “薪資：”.pay.”<br/>”;

//list第二種應用

$number=array(12,34,45,56,78,89);

echo "<p>陣列中的所有值:<br/>";

//while搭配list結構 ， $index陣列的索引值 ， $value對應索引編號的資料

while(list($index,$value)=each($number))

{

echo "\$number[$index] => ".$value."<br/>";

}

echo "</p>"

?>

</body>

</html>

指標控制指令

|  |  |
| --- | --- |
| 函數指令 | 功能說明 |
| reset($陣列名稱) | 將陣列指標移動到陣列的第一個元素位置 |
| prev($陣列名稱) | 將陣列指標往前移動一個元素位置 |
| next($陣列名稱) | 將陣列指標往後移動一個元素位置 |
| end($陣列名稱) | 將陣列指標移動到陣列的最後一個元素位置 |
| key($陣列名稱) | 讀取目前指標位置的陣列元素索引編號值 |
| current($陣列名稱) | 讀取目前指標位置的陣列元素資料內容 |
| count($陣列名稱) | 計算陣列的長度 |

下面的例子直接應用 指標控制指令 來做說明，範例中尚缺 prev($陣列名稱)指令，請自行加入。

ch11-13.php

<html>

<head>

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

<title>陣列指標控制函數</title>

</head>

<body>

<?php

$info=array("Linda","Alice","Peter","Elvin","Caroline");

echo "此陣列有 <font color='red'>".count($info)." </font>個元素<br/>";

for($a=0;$a < count($info);$a++)

{

echo "\$info[$a]=>".$info[$a] ."<br/> ";

}

echo "<p>目前指標的位置 ： ".key($info)."<br/>";

//將指標移到下一個

next($info);

echo "指標移到下一個位置 : ".key($info)."</p>";

//將指標移到下一個

next($info);

echo "<p>指標移到下一個位置 : ".key($info)."<br/>";

echo "對應的陣列內容 : ".current($info)."</p>";

//指標移到最後一個

end($info);

echo "<p>最後一個位置 : ".key($info)."<br/>";

echo "對應的陣列內容 : ".current($info)."</p>";

//將指標指到第一個位置

reset($info);

echo "<p>第一個位置 : ".key($info)."<br/>";

echo "對應的陣列內容 : ".current($info)."</p>";

?>

</body>

</html>

陣列資料排序指令

格式:

排序函數($陣列名稱[,排序型別]);

說明: 排序型別有三種說明如下:

1. SORT\_REQULAR ： 預設的排序型別，不改變陣列元素的資料型態。
2. SORT\_NUMERIC ： 以數值模式來排序陣列元素。
3. SORT\_STRING ： 以字串模式來排序陣列元素。

陣列排序相關指令

|  |  |
| --- | --- |
| 函數指令 | 功能說明 |
| sort($陣列變數) | 以遞增方式排列陣列的元素 |
| rsort($陣列變數) | 以遞減方式排列陣列的元素 |
| asort($陣列變數) | 以遞增方式排列陣列的元素，對應的索引編號也跟著變動。 |
| arsort($陣列變數) | 以遞減方式排列陣列的元素，對應的索引編號也跟著變動。 |
| ksort($陣列變數) | 以遞增方式排列陣列的索引編號值 |
| krsort($陣列變數) | 以遞減方式排列陣列的索引編號值 |
| usort($陣列變數) | 以使用者自訂的函數來排列陣列元素 |
| uasort($陣列變數) | 以使用者自訂的函數來排列陣列元素，對應的索引編號也跟著變動 |
| uksort($陣列變數) | 以使用者自訂的函數來排列陣列的索引編號值 |

例：設計兩個陣列，一個以編號為索引編號，另一個以字串為索引編號，以sort函數來排序第一個陣列，以asort來排列第二個陣列。

ch11-14.php

<html>

<head>

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

<title>陣列排序</title>

</head>

<body>

<?php

//第一個陣列

$number=array(123,345,23,66,7,821,10,33,90,100);

echo "原始的陣列順序 : <br/>";

for($a=0;$a<count($number);$a++){

echo $number[$a]." , ";

}

echo "<p>排序後的陣列 : <br/>";

sort($number);

for($a=0;$a<count($number);$a++){

echo $number[$a]." , ";

}

echo "</p>";

//第二個陣列

$fruits = array("first" => "apple","second" =>"lemon" ,"third" =>"orange" , "fourth" =>"banana" );

echo "<p>第二個陣列 : <br/>";

while (list($key, $val) = each($fruits)) {

//列出索引值和陣列元素

echo "$key => $val<br/>";

}

//使用arsort遞減排序,索引也一併被調整

asort($fruits);

//將指標指回陣列的第一個位置

reset($fruits);

echo "<p>asort之後 : <br/>";

while (list($key, $val) = each($fruits)) {

//列出索引值和陣列元素

echo "$key => $val<br/>";

}

?>

</body>

</html>

請應用上例，自行練習 rsort , arsort , ksort , krsort , usort , uasort , uksort 各函數。

多(維)陣列排序

多陣列排序指令，提供同時對多個陣列同時進行排序工作，而且，不同陣列可設定不同的排序型態和方式，格式如下：

array\_multisort ( $陣列１[ ,排序型態 ][ ,排序方法] ,陣列2,……,陣列n );

說明: 排序型別有三種說明如下:

1. SORT\_REQULAR ： 預設的排序型別，不改變陣列元素的資料型態。
2. SORT\_NUMERIC ： 以數值模式來排序陣列元素。
3. SORT\_STRING ： 以字串模式來排序陣列元素。

排列方法有二種：

1. SORT\_ASC：遞增排序
2. SORT\_DESC：遞減排序

例：設計一個包括成績和姓名的二維陣列，利用array\_multisort指令，設定成績子陣列是依成績遞減排序；成績相同者依字串遞增排序。

ch11-15.php

<html>

<head>

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

<title>array\_multisort函數</title>

</head>

<body >

<h2>array\_multisort 函數</h2>

<?php

$class=array("score" => array(70,95,70,60,88,78,95,88,92,100),

"name" => array("Alice","Peter","Elvin","Sindy","Simon",

"Bob","Hank","Charles","Caroline","Linda"));

// 將分數作為數值，由高到低排序

array\_multisort($class["score"], SORT\_NUMERIC, SORT\_DESC,

// 將名字作為字串，由小到大排序

$class["name"], SORT\_STRING, SORT\_ASC);

echo "<table bgcolor='#FFFF99' border=2 width='80%'>";

echo "<tr><th>score</th>";

for($a=0;$a<10;$a++)

{

echo "<td>".$class["score"][$a]."</td>";

}

echo "</tr><tr><th>name</th>";

for($a=0;$a<10;$a++)

{

echo "<td>".$class["name"][$a]."</td>";

}

?>

</body>

</html>

練習：修改上例，先依原始資料順序列印，再列印排序後資料

陣列分割與合併

　　PHP也提供一些指令，可以將一個大的陣列分割成幾個小陣列的組合，也可以將數個陣列合併成一個大陣列，或是以一個陣列當作索引，另一個陣列當作資料，合併成一個新的陣列，指令格式及功能說明如下：

|  |  |
| --- | --- |
| 指令格式 | 功能說明 |
| array\_chunk($陣列,size) | 將陣列依指定的數量分割成多個小陣列，這些小陣列包括在原列之中。  (維度加1) |
| array\_combine($key陣列,$value陣列) | 一個陣列當索引值，另一個陣列為資料，組成一個新陣列。 |
| array\_merge($陣列1,$陣列2) | 將多個陣列合併成為一個單一陣列。 |
| array\_merge\_recursive($陣列1,$陣列2) | 將陣列2接到陣列1之後，如果索引值有相同，這些相同索引資料，會以遞迴方式合併起來。(一樣的時候會產生下一個陣列 簡單來說就是會變成2微陣列) |

下面直接以範例來驗證並瞭解各個指令的功能：

例：先設定一個含有１０個元素的陣列，以array\_chunk( ) 指令，將陣列分割成5個元素的小陣列。

程式執行後，會變成一個二維新陣列

ch11-16a.php

<html>

<head>

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

<title>陣列分割</title>

</head>

<body>

<?php

$score=array(70,95,70,60,88,78,95,88,92,100);

echo "原始score陣列:<br/>";

foreach($score as $value){

echo $value." , ";

}

//分割score陣列

echo "<p>分割後的score陣列:<br/>";

print\_r(array\_chunk($score,5));

?>

</body>

</html>

例：設定二個陣列，利用array\_merge( )指令，將二個陣列合併成一個新陣列，再用array\_combine( )，將二個陣列合併成另一個新陣列，比較二個指令的執行結果。

ch11-16b.php

<html>

<head>

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

<title>陣列合併</title>

</head>

<body>

<?php

$score=array(70,95,70,60,88,78,95,88,92,100);

$name =array("Alice","Peter","Elvin","Sindy","Simon",

"Bob","Hank","Charles","Caroline","Linda");

//單純合併兩個陣列

echo "</p><p>單純合併score和name陣列:<br/>";

print\_r(array\_merge($score,$name));

//以$name為索引,$score為值,合併為一個陣列

$class=array\_combine($name,$score);

//顯示新陣列

echo "</p><p>以name為索引,score為值,合併成新陣列:<br/>";

while(list($index,$value)=each($class))

{

echo "\$class[$index] => ".$value."</br>";

}

?>

</body>

</html>

例：設定二個陣列，用array\_merge\_recursive( )指令，合併這二個陣列。

ch11-16c.php

<html>

<head>

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

<title>array\_merge\_recursive函數</title>

</head>

<body>

<?php

//原始兩個陣列

$info1 = array("color" => array("red","blue"),"Linda",5,50000);

$info2 = array("color" => array("green","blue"),"Alice","Hank","Mary");

//列印原$info1

echo "原始 \$info1 陣列 : <br/>";

print\_r($info1);

//列印原$info2

echo "<p>原始 \$info2 陣列 : <br/>";

print\_r($info2);

//merge\_recursive後的陣列

$result = array\_merge\_recursive($info1, $info2);

echo "</p><p>array\_merge\_recursive後的陣列 : <br/>";

print\_r($result);

?>

</body>

</html>

上述程式中，索引資料有( red,blue )及( green,blue )，合併後，會以遞迴方式( red,blue,green,blue)方式合併

上面各項範例中，可以看到一個指令:

Print\_r( $陣列變數 )

是將陣列變數內容以下列方式列印出來

Array([索引1] => 值1 [索引2] =>值2…..[索引n] =>值n)

其他陣列資料處理指令

1. 指定將陣列的 字串索引值 以 大寫 或 小寫 的型式

array\_change\_key\_case($陣列變數，型態)；

註：CASE\_UPPER　(大寫)，CASE\_LOWER (小寫)

例：將字串索引值設定為大寫

ch11-17.php

<html>

<head>

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

<title></title>

</head>

<body>

<?php

$class=array("score" => array(70,95,70,60,88,78,95,88,92,100),

"name" => array("Alice","Peter","Elvin","Sindy","Simon",

"Bob","Hank","Charles","Caroline","Linda"));

$class\_new=array\_change\_key\_case($class, CASE\_UPPER);

//使用print\_r函數以人類易於了解的模式列印

print\_r($class\_new);

?>

</body>

</html>

1. 計算陣列中，每一個資料出現的次數

array\_count\_values($陣列變數)；

例：設定一個陣列，計算陣列中各項資料出現的次數

ch11-18.php

<html>

<head>

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

<title>array\_count\_values 函數</title>

</head>

<body>

<?php

$score =array(70,95,70,60,88,78,95,88,92,100);

//計算陣列各元素出現的次數

print\_r(array\_count\_values($score));

?>

</body>

</html>

1. **查詢**原始陣列中的資料是否在指定陣列中也有出現，並將結果顯示出來

array\_diff ($原始陣列 , $指定陣列) //没有出現

array\_interset ($原始陣列 , $指定陣列)　　　　//有出現

例：設定二個陣列，分別用array\_diff 和 array\_interset來查詢第一個陣列資料　( 有 / 没有）出現在第二個陣列中。

ch11-19.php

<html>

<head>

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

<title>array\_diff與array\_intersect</title>

</head>

<body>

<?php

$class1=array("Linda","Tina","Peter","Charles","Jason","Coco","Ben","Alphalli","Bob","May");

$class2=array("Lkk","Tina","Hank","Peter","Charles","Caroline","Coco","Ben","Bert","Any");

echo "使用array\_diff 函數<br/>";

//列出陣列1中沒有出現在陣列2的元素和對應的索引值

print\_r(array\_diff($class1,$class2));

echo "<p>使用array\_intersect函數<br/>";

//列出陣列1中沒有出現在陣列2的元素和對應的索引值

print\_r(array\_intersect($class1,$class2));

echo "</p>";

?>

</body>

</html>

1. 填滿　陣列資料

array\_fill(begin,size,”字串資料”)；

說明：begin 開始填寫的位置

size 要填寫的數量

例：設定一個陣列，從索引位置1開始，填入10筆字串，字串值為”Holiday”

ch11-20.php

<html>

<head>

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

<title>array\_fill 函數</title>

</head>

<body>

<h2>array\_fill 函數</h2>

<?php

//以填滿的方式產生新陣列

$info=array\_fill(1,10,"Holiday");

//從1開始填,陣列大小是10

for($a=0;$a<count($info);$a++) {

echo "\$info[$a] => ".$info[$a]."<br/>";

}

?>

</body>

</html>

1. 陣列資料篩選

PHP提供array\_filter( )指令用來從陣列中篩選出符合條件的資料。格式如下：

　 array\_filter( $陣列,自訂函數)；

例：分別設定一個數值陣列　及一個字串陣列，應用array\_filter(　)指令，在數值陣列篩選出>50的資料，在字串陣列篩選長度為3的資料。

ch11-21.php

<html>

<head>

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

<title>array\_filter 函數</title>

</head>

<body>

<h2>array\_filter 函數</h2>

<?php

//自訂傳回大於50的函數

function Greater($nm){

return($nm>50);

}

//自訂傳回字串長度等於3的字串

function Len3($str){

if(strlen($str)==3){

return($str);

}

}

$number=array(12,56,34,126,288,566,33,78,90,16);

$class=array("Linda","Tina","Peter","Charles","Jason","Coco","Ben","Alphalli","Bob","May");

echo "原數值陣列 : ";

foreach($number as $value){

echo $value." , ";

}

echo "<p>篩選>50的數值陣列 : ";

//執行篩選,結果存到$number2陣列

$number2=array\_filter($number,"Greater");

foreach($number2 as $value){

echo $value." , ";

}

echo "</p><p>原字串陣列 : ";

foreach($class as $value){

echo $value." , ";

}

echo "</p><p>篩選長度=3的字串陣列 : ";

//執行篩選,結果存到$class2陣列

$class2=array\_filter($class,"Len3");

foreach($class2 as $key => $value){

echo $value." , ";

}

echo "</p>";

?>

</body>

</html>

註：自訂函數指令在下一單元中再做說明。

1. 陣列的　索引與資料　對調

相關操作指令格式及功能說明如下表：

|  |  |
| --- | --- |
| 指令 | 功能 |
| array\_keys($陣列) | 傳回陣列中的所有索引值 |
| array\_values($陣列) | 傳回陣列中的所有資料值 |
| array\_flip($陣列) | 將 索引 和 資料 對調 |
| array\_reverse($陣列) | 將陣列資料順序反向調整 |
| array\_key\_exists(key , $陣列) | 查詢指定的索引是否存在(true/false) |
| array\_search(字串 , $陣列) | 查詢資料是否出現在陣列中，傳回(索引值/null) |

例：設計一個陣列，依序列出所有陣列資料、索引值，再將索引值和資料對調。

ch11-22.php

<html>

<head>

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

<title>索引和元素互調</title>

</head>

<body>

<?php

//原始陣列,預設索引是由0開始的整數

$class=array("Linda","Tina","Peter","Charles","Jason","Coco","Ben","Alphalli","Bob","May");

//列出所有的索引

print\_r(array\_keys($class));

echo "<p>";

//列出所有的元素

print\_r(array\_values($class));

echo "<p></p>";

//互調後的結果

print\_r(array\_flip($class));

?>

</body>

</html>

練習：在上例程式中，將陣列資料反向排列，並分別查詢　索引10 及　資料"Peter"是否存在。

1. 擴充陣列資料

相關操作指令格式及功能說明如下表：

|  |  |
| --- | --- |
| 指令 | 功能說明 |
| array\_push(＄陣列,val1,val2) | 將一個或多個資料依序加到陣列的最後面 |
| array\_pad(＄陣列,size,value) | 以指定的value加到指定的size位置，  size為正是將資料加到原陣列之後  size為負是將資料加到原陣列之前 |

說明：array\_pad( );指令結果必須放在新陣列中

　　　array\_push( )指令只能將資料加在陣列後面

　　　array\_push( )指令所加入的資料可以不同，

　　　array\_pad( ); 指令所加入的所有資料必須不同

例：設計一個陣，用array\_push( )指令在陣列最後加入三個新資料，再利用array\_pad( );指今在陣列最前面加入兩個相同資料。

ch11-23.php

<html>

<head>

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

<title>擴充陣列</title>

</head>

<body>

<?php

//原始陣列

$number=array(12,56,34,126,288,566,33,78,90,16);

//以array\_push()新增三筆新資料

//結果會變更到原來的陣列

array\_push($number,111,222,333);

print\_r($number); //顯示陣列

//以array\_pad()往前擴充到15(即新增兩筆資料在前面)

//結果會產生新的陣列

$info=array\_pad($number,-15,"Happy");

echo "<p>";

print\_r($info); //顯示陣列

?>

</body>

</html>

1. 刪除陣列資料

相關操作指令格式及功能說明如下表：

|  |  |
| --- | --- |
| 指令 | 功能說明 |
| array\_pop(＄陣列) | 將陣列的最後一個資料刪除 |
| array\_shift(＄陣列) | 將陣列的第一個資料刪除 |
| array\_sum(＄陣列) | 計算陣列中所有數值資料的總和 |

例：設定一個數字陣列，分別用上述指令來處理陣列資料。

ch11-24.php

<html>

<head>

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

<title>刪除陣列元素</title>

</head>

<body>

<?php

//原始陣列

$number=array(12,56,34,126,288,566,33,78,90,16);

echo "陣列數值的總合 : ".array\_sum($number)."<p>";

echo "原始陣列:<br/>";

print\_r($number);

echo "</p><p>";

//刪除最後一個元素

array\_pop($number);

echo "刪除最後一個:<br/>";

print\_r($number);

//刪除第一個元素

array\_shift($number);

echo "</p><p>";

echo "刪除第一個:<br/>";

print\_r($number);

?>

</body>

</html>

練習:將上個中，array\_sum( )指令放在後面，看結果如何。