



## 第四章 陣列

4-1 陣列

4-2 多維陣列與結合陣列

4-3 陣列元素應用函數

備註：可依進度點選小節

## 4-1 陣列



- 陣列介紹
- 建立一維陣列
- 存取陣列元素
- 陣列應用
- foreach陣列迴圈

# 陣列介紹(1/3)

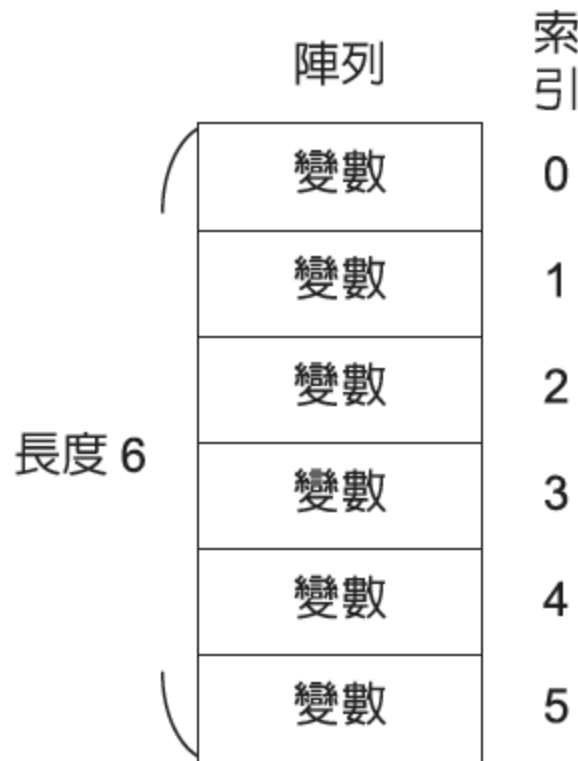


- 在宣告相同屬性變數名稱的情況下，例如記錄班級中每個人的科目成績，必須為每位同學的成績各別建立不同的變數來記錄
- 在這種情況下，需要使用大量程式碼來建立變數及儲存資料
- 這個章節便是帶領讀者瞭解如何有效的建立相同屬性的資料型態，以便在變數名稱上的管理與使用

## 陣列介紹(2/3)



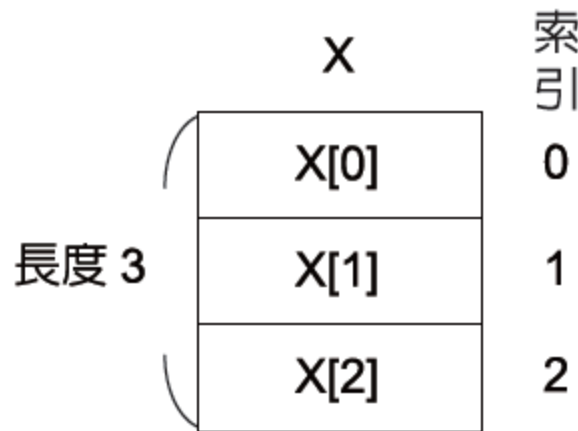
- 陣列(Array)是在記憶體中的一塊連續記憶體空間，且具有相同名稱與資料型態的集合



# 陣列介紹(3/3)



- 陣列名稱搭配對應的索引我們稱為元素，其表示方式為「陣列名稱[索引]」
- 舉例說明，一個陣列名稱為X，其長度為3



# 建立一維陣列(1/3)



- 在PHP 中會用下面兩種寫法來使用陣列變數：
  - \$陣列名稱[索引] = 陣列元素的值;
  - \$陣列名稱 = array(陣列元素的值1,...陣列元素的值N);

# 建立一維陣列(2/3)



- 第一種寫法有沒有指定索引都可以使用，如果沒有索引，則會自動地從0遞增索引來指定陣列元素的值

無索引的宣告方式

```
<?php
    $name[] = "張曉明";           //陣列索引為0，name[0]元素為張曉明
    $name[] = "陳大章";           //陣列索引為1，name[1]元素為陳大章
?>
```

有索引的宣告方式

```
<?php
    $name[0] = "張曉明";           //陣列索引為0，name[0]元素為張曉明
    $name[1] = "陳大章";           //陣列索引為1，name[1]元素為陳大章
?>
```

## 建立一維陣列(3/3)



- 第二種寫法是使用array() 建構子來建立陣列，中間則以「,」號來區隔多個陣列元素的值

```
<?php
    $name = array ("張曉明","陳大章"); //使用建構子來建立陣列
?>
```

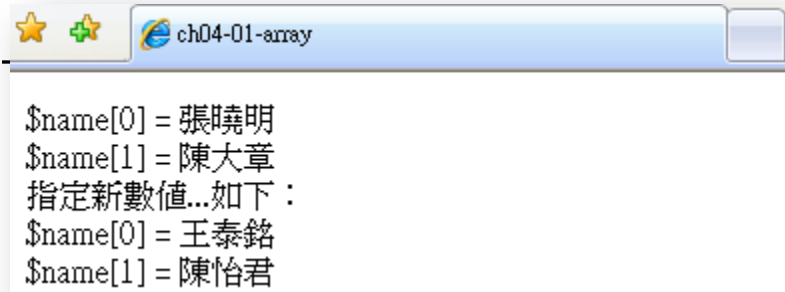


# 存取陣列元素



- 使用方式與變數存取方式大同小異，只需要透過索引值便可存取資料

```
<?php
$name[] = "張曉明"; //陣列索引為0，name[0]元素為張曉明
$name[] = "陳大章"; //陣列索引為1，name[1]元素為陳大章
echo "\$name[0]=".$name[0]."<BR>\$name[1]=".$name[1]."<BR>";
//讀取name陣列
echo "指定新數值...如下:<BR>";
$name[0] = "王泰銘";
$name[1] = "陳怡君";
echo "\$name[0] = ".$name[0]."<BR>\$name[1] = ".$name[1];
?>
```



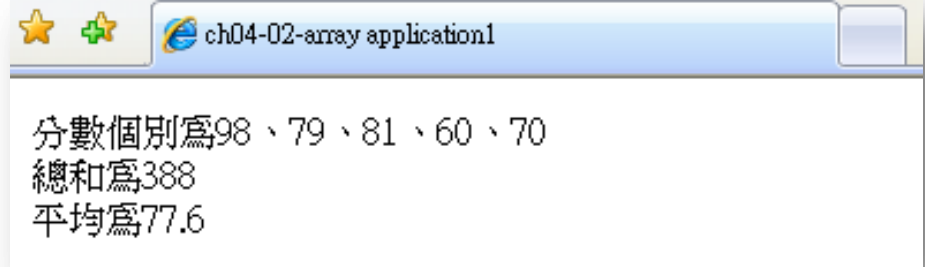
# 陣列應用(1/2)



- 使用迴圈來做陣列資料的讀取

```
<?php
    $score = array(98,79,81,60,70);    //宣告陣列數值
    $sum = 0;
    echo "分數個別為";
    for($i=0;$i<count($score);$i++)    //迴圈應用
    {
        if($i<count($score)-1)
        { echo $score[$i]."、"; }
        else
        { echo $score[$i]."<BR>"; }
        $sum += $score[$i];
    }
    echo "總和為".$sum."<BR>";
    echo "平均為".($sum/count($score));
?>
```

//條件敘述應用



# 陣列應用(2/2)



- 範例：使用迴圈將資料指定於陣列中

```
<?php
    for($i=0;$i<=10;$i++){
        $sum[$i] = $i*2;
    }
    for($i=0;$i<count($sum);$i++){
        echo "sum[$i] = ".$sum[$i]. "<BR>";
    }
?>
```



ch04-03-array application2

```
sum[0] = 0
sum[1] = 2
sum[2] = 4
sum[3] = 6
sum[4] = 8
sum[5] = 10
sum[6] = 12
sum[7] = 14
sum[8] = 16
sum[9] = 18
sum[10] = 20
```

# foreach 陣列迴圈(1/2)



- PHP 提供foreach 陣列迴圈來方便讀取陣列，不需要設定初始值與判斷陣列長度，便可一一走訪整個陣列

```
<?php
    foreach($陣列名稱 as $陣列元素變數)
    {
        echo $陣列元素變數; //顯示陣列元素數值
    }
?>
```

# foreach 陣列迴圈(2/2)



- 範例

```
<?php
    $score = array(98,79,81,60,70); //宣告陣列數值
    $sum = 0;
    echo "分數個別為\t";
    foreach($score as $selement){ //foreach迴圈
        echo $selement."\t";
        $sum += $selement;
    }
    echo "<BR>總和為".$sum."<BR>";
    echo "平均為".($sum/count($score));
?>
```



ch04-04-foreach

分數個別為 98 79 81 60 70  
總和為388  
平均為77.6

## 4-2 多維陣列與結合陣列

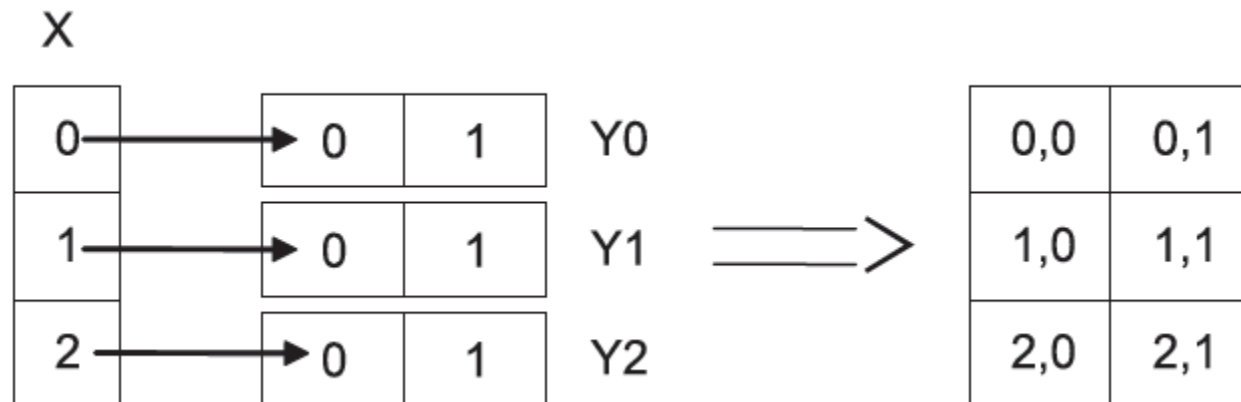


- 二維陣列
- 多維陣列
- 結合陣列
- 陣列相關函數

# 二維陣列(1/3)



- 是擁有兩個索引的陣列，屬於一維陣列的延伸
- 由一個X的一維陣列與多個Y一維陣列所組成，每個X的陣列元素都會指向對應Y的一維陣列，透過這樣的指向對應來合併成一個二維陣列



# 二維陣列(2/3)



## • 建立二維陣列

```
<?php
    $變數名稱0 = array(陣列元素的值,...); //為Y的一維陣列
    $變數名稱1 = array(陣列元素的值,...); //為Y的一維陣列
    $變數名稱2 = array(陣列元素的值,...); //為Y的一維陣列
    $變數名稱3 = array($變數名稱0,$變數名稱1,$變數名稱2,...);
    //為X的一維陣列並指向為Y的一維陣列，合併成二維陣列
?>
```

```
<?php
    $變數名稱3 = array(
        $變數名稱0 = array(陣列元素的值,...),
        $變數名稱1 = array(陣列元素的值,...),
        $變數名稱2 = array(陣列元素的值,...),
        ...);
?>
```

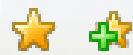


# 二維陣列(3/3)



## • 範列

```
<?php
    $row0 = array(1,2);           //為列的一維陣列
    $row1 = array(3,4);           //為列的一維陣列
    $row2 = array(5,6);           //為列的一維陣列
    $array = array($row0, $row1, $row2); //為行的一維陣列→組成二
                                     //維陣列
    foreach ($array as $row)       //行的陣列迴圈
    {
        foreach ($row as $element) //列的陣列迴圈
        {
            echo $element."\t";
        }
        echo "<BR>";
    }
?>
```



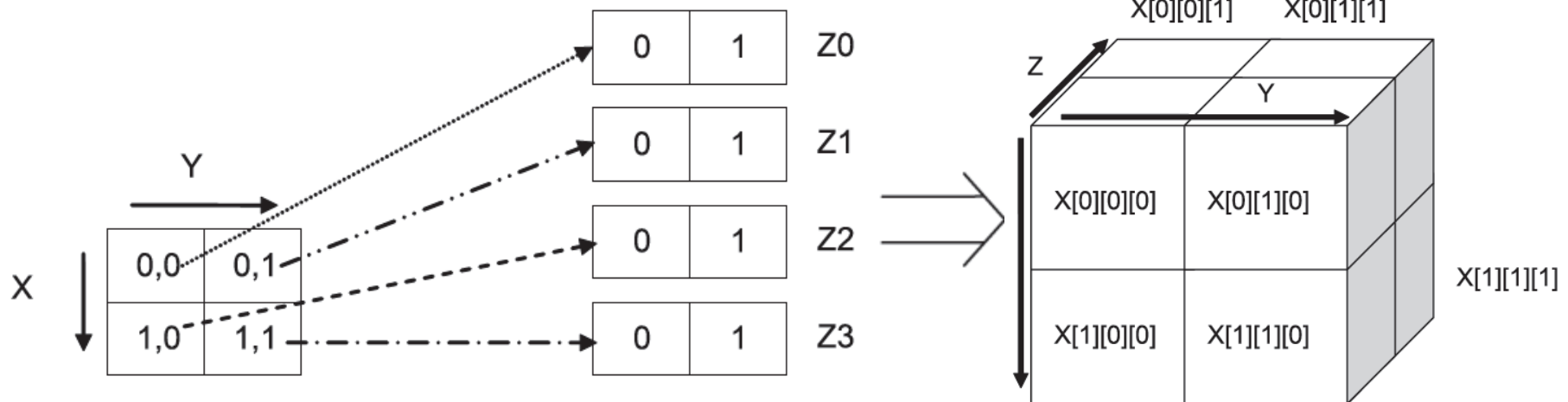
ch04-05-two dimension array

```
1 2
3 4
5 6
```

# 多維陣列(1/3)



- 多維陣列即是擁有多個索引的陣列
- 以三維陣列為例，即是有3個索引的陣列，由一個二維陣列和多個一維陣列的組合，二維陣列中的每個陣列元素值指向對應的一維陣列



## 多維陣列(2/3)



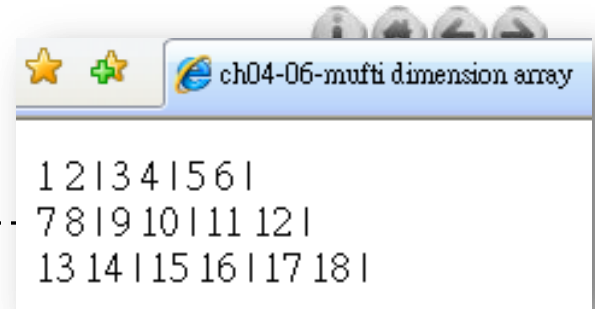
- 四維陣列我們也可以想像成三維陣列中每個元素值指向對應的一個一維陣列，後面的多維陣列以此類推

# 多維陣列(3/3)

- 範例：三維陣列

```
<?php
$array0 = array($row0= array(1,2), $row1= array(3,4),
                $row2= array(5,6)); //二維陣列
$array1 = array($row3= array(7,8), $row4= array(9,10),
                $row5= array(11,12)); //二維陣列
$array2 = array($row6= array(13,14), $row7= array(15,16),
                $row8= array(17,18)); //二維陣列
$cubical = array($array0,$array1,$array2);

//三維陣列
foreach ($cubical as $array) //三維的陣列迴圈
{
    foreach ($array as $row) //二維的陣列迴圈
    {
        foreach ($row as $element) //一維的陣列迴圈
        {
            echo $element."\t"; }
        echo "\n"; }
    echo "<BR>"; }
?>
```



# 結合陣列(1/2)



- 是指陣列元素的值對應陣列中所指定的鍵值，也可以說是替索引值命名指定名稱
  - 其鍵值可以是正整數或字串，但不可重複

```
<?php
```

```
    $變數名稱["指定名稱"] = 陣列元素的值;
```

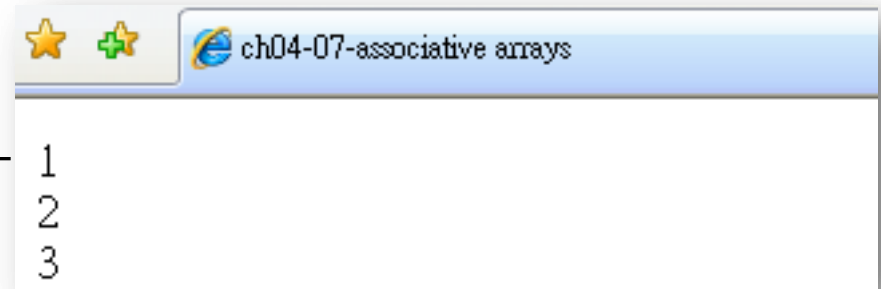
```
?>
```

## 結合陣列(2/2)



- 範例

```
<?php
    $array["number1"] = 1; //新增結合陣列元素
    $array["number2"] = 2; //新增結合陣列元素
    $array[] = 3; //新增陣列元素
    foreach($array as $element)
    {    echo $element."<BR>"; }
?>
```



- 這裡須注意，使用結合陣列後，若無使用結合陣列來新增陣列元素，則陣列元素的索引值會以0 開始遞增

# 陣列相關函數(1/3)



函數名稱	說明
<code>print_r(\$array)</code>	輸出陣列中每個元素索引或鍵值與元素內的值
<code>array_keys(\$array)</code>	回傳陣列中所有元素的索引或鍵值
<code>array_values(\$array)</code>	回傳陣列中所有元素的值
<code>current(\$array)</code>	回傳目前指標所在陣列元素的值
<code>key(\$array)</code>	回傳目前指標所在陣列元素的索引或鍵值
<code>each(\$array)</code>	回傳目前陣列元素的索引或鍵值和元素內的值，並移動自下一個元素
<code>reset(\$array)</code>	陣列指標回到第一個陣列元素
<code>end(\$array)</code>	陣列指標指到最後一個陣列元素
<code>next(\$array)</code>	陣列指標指向下一個陣列元素
<code>prev(\$array)</code>	陣列指標指向前一個陣列元素

# 陣列相關函數(2/3)



## • 範例

```
<?php
    $array["a"] = 1;           //新增結合陣列元素
    $array[] = 2;              //新增陣列元素
    $array["b"] = 3;          //新增結合陣列元素
    $array[] = 4;              //新增陣列元素
    echo "1.\t";
    print_r($array);           //輸出的陣列中每個元素索引或鍵值與元素內的值
    echo "<BR>2.\t";
    print_r(array_keys($array)); //輸出回傳陣列中所有元素的索引或鍵值
    echo "<BR>3.\t";
    print_r(array_values($array)); //輸出回傳陣列中所有元素的值
    echo "<BR>4.\t";
    echo current($array);       //回傳目前指標所在陣列元素的值
    echo "<BR>5.\t";
    echo key($array);           //回傳目前指標所在陣列元素的索引或鍵值
```



# 陣列相關函數(3/3)



```
echo "<BR>6.\t";  
print_r(each($array));           //回傳目前的陣列元素的索引或鍵值和元素內  
//的值，並移動至下一個元素  
echo "<BR>7.\t";  
echo reset($array);              //陣列指標回到第一個陣列元素  
echo "<BR>8.\t";  
echo end($array);                //陣列指標指到最後一個陣列元素  
echo "<BR>9.\t";  
echo prev($array);               //陣列指標指向前一個陣列元素  
echo "<BR>10.\t";  
echo next($array);              //陣列指標指向下一個陣列元素
```

?>

```
1. Array ( [a] => 1 [0] => 2 [b] => 3 [1] => 4 )  
2. Array ( [0] => a [1] => 0 [2] => b [3] => 1 )  
3. Array ( [0] => 1 [1] => 2 [2] => 3 [3] => 4 )  
4. 1  
5. a  
6. Array ( [1] => 1 [value] => 1 [0] => a [key] => a )  
7. 1  
8. 4  
9. 3  
10. 4
```

## 4-3 陣列元素應用函數



- 陣列元素計算與加總
- 陣列元素排序與搜尋

# 陣列元素計算與加總(1/2)



- 陣列元素計算

count (陣列變數名稱)

- 陣列元素加總

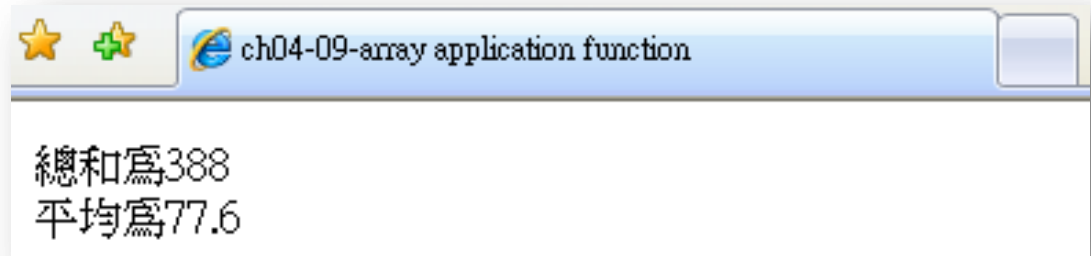
array\_sum(陣列變數名稱)

# 陣列元素計算與加總(2/2)



- 範例

```
<?php
    $score = array(98,79,81,60,70); //宣告陣列數值
    echo "總和為".array_sum($score)."<BR>";
    echo "平均為".(array_sum($score)/count($score));
?>
```



# 陣列元素排序與搜尋(1/4)



- 若需要將陣列元素內的數值按照由大到小或由小到大來排序，PHP 提供了六種函數來使用

函數使用格式	說明
sort(陣列變數名稱)	陣列元素的值從小到大排序
rsort(陣列變數名稱)	陣列元素的值從大到小排序
asort(陣列變數名稱)	結合陣列元素的值由小到大排序
arsort(陣列變數名稱)	結合陣列元素的值由大到小排序
krsort(陣列變數名稱)	結合陣列索引或鍵值由小到大排序
krsort(陣列變數名稱)	結合陣列索引或鍵值由大到小排序

- sort/rsort 與 asort/arsort 的差別，在於 sort/rsort 排序時，元素的索引或鍵值並不會保留下來，並從 0 開始重新排序，而 asort/arsort 則會保留對應的索引或鍵值

## 陣列元素排序與搜尋(2/4)



- 若想搜尋數值是否在陣列中，PHP 也提供了兩種函數

函數使用格式	說明
<code>in_array(數值,陣列變數名稱)</code>	檢查值是否在陣列元素中，傳回布林值
<code>array_search(數值,陣列變數名稱)</code>	檢查數值是否在陣列中，如果是，傳回鍵值，若沒有則傳回false

# 陣列元素排序與搜尋(3/4)



## • 範例

```
<?php
    $score = array(98,79,81,60,70);    //建立陣列數值
    sort($score); print_r($score);
    echo "<BR>";
    rsort($score); print_r($score);
    echo "<BR>";
    $score = array(98,79,81,60,70);    //重新建立陣列元素的值
    asort($score); print_r($score);
    echo "<BR>";
    arsort($score); print_r($score);
    echo "<BR>";
    $score = array(98,79,81,60,70);    //重新建立陣列元素的值
    ksort($score); print_r($score);
    echo "<BR>";
    krsort($score); print_r($score);
    echo "<BR>";
```

# 陣列元素排序與搜尋(4/4)



```
if (in_array(30,$score))  
{ echo "數值30有在陣列中<BR>"; }  
else{ echo "數值30沒有在陣列中<BR>"; }  
echo "70在陣列第".array_search("70",$score). "元素內";  
?>
```

```
Array ( [0] => 60 [1] => 70 [2] => 79 [3] => 81 [4] => 98 )  
Array ( [0] => 98 [1] => 81 [2] => 79 [3] => 70 [4] => 60 )  
Array ( [3] => 60 [4] => 70 [1] => 79 [2] => 81 [0] => 98 )  
Array ( [0] => 98 [2] => 81 [1] => 79 [4] => 70 [3] => 60 )  
Array ( [0] => 98 [1] => 79 [2] => 81 [3] => 60 [4] => 70 )  
Array ( [4] => 70 [3] => 60 [2] => 81 [1] => 79 [0] => 98 )  
數值30沒有在陣列中  
70在陣列第4元素內
```



本章結束

Q&A 討論時間