

Chapter 陣 列

4

4-1 認識陣列

4-2 一維陣列

4-3 多維陣列

4-4 不規則陣列

4-5 System.Array 類別

4-6 與陣列相關的函式

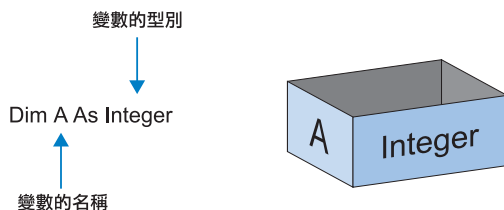
4-1 認識陣列

我們知道電腦可以執行重複的動作，也可以處理大量的資料，但截至目前，我們都只是宣告了極少量的資料，若要宣告成千上百個資料，該怎麼辦呢？難道要寫出成千上百個敘述嗎？當然不是！此時，您應該使用**陣列** (array)，而本章就是要告訴您如何建立及存取陣列。

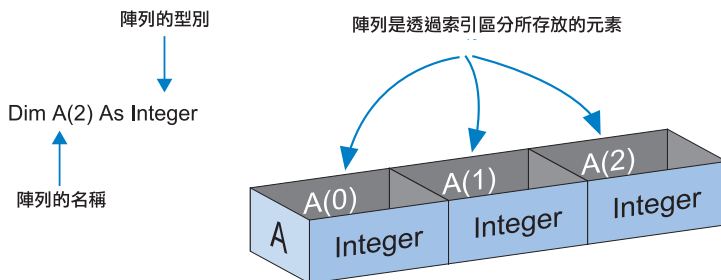
陣列和變數一樣是用來存放資料，不同的是陣列雖然只有一個名稱，卻可以存放多個資料。陣列所存放的資料叫做**元素** (element)，每個元素有各自的**值** (value)，陣列是透過**索引** (index) 區分所存放的元素，在預設的情況下，第一個元素的索引為 0，第二個元素的索引為 1， \dots ，第 n 個元素的索引為 $n - 1$ 。

當陣列的元素個數為 n 時，表示陣列的**長度** (length) 為 n ，合法的長度上限為 $2^{31} - 1$ ，而且除了一維 (one-dimension) 陣列之外，VB 2017 亦支援**多維** (multi-dimension) 陣列，合法的維度上限為 32。

每個變數只能存放一個資料，例如：



陣列雖然只有一個名稱，卻可以用來存放多個資料，例如：



■ 4-2 一維陣列

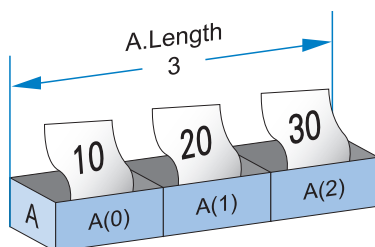
由於 VB 2017 的陣列隸屬於 `System.Array` 類別，因此，在宣告陣列時必須使用關鍵字 `New` 建立 `Array` 物件。以下面的程式碼為例，第 01 行是宣告名稱為 `A`、型別為 `Integer` 的一維陣列變數，此時尚未配置記憶體空間給陣列；第 02 行是配置三個記憶體空間給陣列，每個記憶體空間可以存放一個整數；第 03 ~ 05 行是將陣列內第 1、2、3 個元素的值設定為 10、20、30。

```

01:Dim A() As Integer    —宣告名稱為 A、型別為 Integer 的一維陣列變數
02:A = New Integer(2) {} —配置三個記憶體空間給陣列
03:A(0) = 10
04:A(1) = 20
05:A(2) = 30
    
```

} 將陣列內第 1、2、3 個元素的值設定為 10、20、30

若要存取陣列的元素，可以使用陣列的名稱與索引，例如 `A(0)` 表示陣列 `A` 的第 1 個元素、`A(1)` 表示陣列 `A` 的第 2 個元素；若要取得陣列的元素個數，可以使用陣列的名稱與 `System.Array` 類別的 `Length` 屬性，例如 `A.Length` 會傳回陣列的元素個數為 3。



前面的程式碼可以寫成一行，也就是在宣告陣列的同時指派初始值：

```
Dim A() As Integer = {10, 20, 30} ' 亦可寫成 Dim A() As Integer = New Integer(2) {10, 20, 30}
```

您無須在小括號內指定元素個數，編譯器會根據大括號內的元素決定陣列的長度。

隨堂練習

撰寫一個程式，令它宣告一個包含 100 個元素的整數陣列，然後將陣列第 1、2、 \dots 、100 個元素的值設定為 501、502、 \dots 、600。

解答

\MyProj4-1\Module1.vb

Module Module1

Sub Main()

Dim A(99) As Integer — 宣告一個包含 100 個元素的整數陣列

For I As Integer = 0 To 99

A(I) = I + 501

} 使用 For 迴圈設定陣列各個元素的值

Next

End Sub

End Module

這個隨堂練習的 For 迴圈是以變數 I 當作計數器，它的值會從 0 依序遞增到 99，所以在第一次執行時，變數 I 的值為 0，迴圈內的敘述會得到 $A(0) = 501$ ，接著在第二次執行時，變數 I 的值為 1，迴圈內的敘述會得到 $A(1) = 502$ ，依此類推，待迴圈執行完畢時，陣列也跟著定義好了。

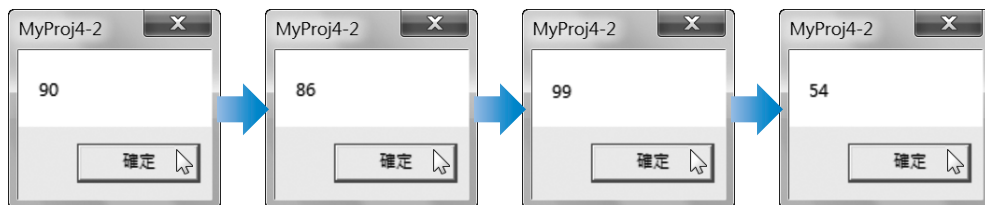


備註

- ▶ 當您存取陣列的元素時，切勿使用超過界限的索引，例如 `Dim A(99) As Integer` 表示陣列 A 的索引為 0 ~ 99，若使用超過界限的索引，例如 `A(100)`，將會在執行時產生 `IndexOutOfRangeException` 例外。
- ▶ 由於 VB 2017 支援隱含型別功能，因此，類似 `Dim A() As Integer = {10, 20, 30}` 的敘述也可以寫成 `Dim A() = {10, 20, 30}`。

隨堂練習

撰寫一個程式，裡面有一個名稱為 **Scores**、包含 4 個元素的整數陣列，用來存放四位學生的分數，分別為 90、86、99、54，而且程式執行完畢後會顯示四位學生的分數。



解答

\MyProj4-2\Module1.vb

Module Module1

Sub Main()

Dim Scores() As Integer = {90, 86, 99, 54}

宣告陣列並指派初始值

For Each Item As Integer In Scores

MsgBox(Item)

Next

使用 For Each 迴圈存取陣列的元素

End Sub

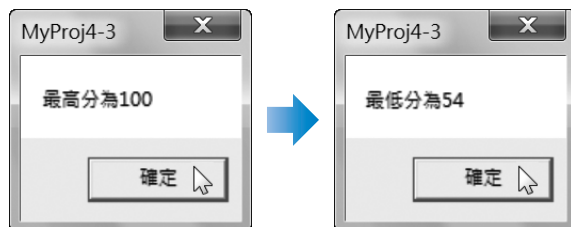
End Module

首先，我們將四個學生的分數存放在整數陣列 **Scores**，然後使用 **For Each** 迴圈讀取各個元素的值。試想，若沒有使用 **For Each** 迴圈，不知道還要多寫幾行程式碼呢！事實上，只要有規律性的變化都可以使用迴圈來執行。

另外要提醒您，盡量不要宣告太大超過實際需要的陣列，以免浪費記憶體空間，同時陣列各個元素的型別必須相同，除非將陣列宣告為 **Object** 型別，才可以指派不同型別的資料給陣列。

隨堂練習

撰寫一個程式，裡面有一個名稱為 **Scores**、包含 6 個元素的整數陣列，用來存放六個學生的分數，分別為 85、60、54、91、100、77，而且程式執行完畢後會顯示最高分及最低分。



解答

\MyProj4-3\Module1.vb

Module Module1

Sub Main()

Dim Scores() As Integer = {85, 60, 54, 91, 100, 77}

Dim MaxScore As Integer = 0, MinScore As Integer = 100

' 使用 For Each 迴圈找出最高分

For Each Item As Integer In Scores

If Item > MaxScore Then MaxScore = Item

Next

MsgBox(" 最高分為 " & MaxScore)

' 使用 For Each 迴圈找出最低分

For Each Item As Integer In Scores

If Item < MinScore Then MinScore = Item

Next

MsgBox(" 最低分為 " & MinScore)

End Sub

End Module

除了使用 For Each 迴圈之外，您也可以改用 For 迴圈，此時須藉助於 System.Array 類別的 Length 屬性做為計數器，如下：

```
Module Module1
    Sub Main()
        Dim Scores() As Integer = {85, 60, 54, 91, 100, 77}
        Dim MaxScore As Integer = 0, MinScore As Integer = 100

        ' 使用 For 迴圈找出最高分
        For I As Integer = 0 To Scores.Length - 1
            If Scores(I) > MaxScore Then MaxScore = Scores(I)
        Next
        MsgBox(" 最高分為 " & MaxScore)

        ' 使用 For 迴圈找出最低分
        For I As Integer = 0 To Scores.Length - 1
            If Scores(I) < MinScore Then MinScore = Scores(I)
        Next
        MsgBox(" 最低分為 " & MinScore)

    End Sub
End Module
```

或者，您可以直接呼叫 System.Array 類別的 Sort() 方法進行排序，如下：

```
Module Module1
    Sub Main()
        Dim Scores() As Integer = {85, 60, 54, 91, 100, 77}

        System.Array.Sort(Scores)

        MsgBox(" 最高分為 " & Scores(5)) } 排序完畢後陣列的元素會由小到大排列，故
        MsgBox(" 最低分為 " & Scores(0)) } Scores(5) 為最高分，Scores(0) 為最低分。

    End Sub
End Module
```

固定大小陣列 V.S. 動態陣列

前面幾個隨堂練習所宣告的陣列都是屬於**固定大小陣列** (fixed-size array)，雖然方便使用 For 迴圈加以存取，但有時我們可能不想在一開始設定陣列的大小或一開始無法判斷陣列的大小，必須等到程式執行階段再視實際情況做設定，即**動態陣列** (dynamic array)，此時，我們可以使用 **Dim** 陳述式宣告一個尚未定義大小的陣列名稱，其語法如下，注意小括號內不要輸入數值：

```
Dim arrayname() [As type] '宣告一個尚未定義大小的陣列
```

目前還無法使用這個動態陣列，因為尚未定義陣列的大小，若要重新定義，可以使用 **ReDim** 陳述式，其語法如下：

```
Redim arrayname(n) '重新定義陣列的大小，共 n+1 個元素，索引從 0 到 n
```

在使用 **ReDim** 陳述式重新定義陣列的大小後，才能存取陣列。若之後又覺得陣列太大或太小，可以再使用 **ReDim** 陳述式重新定義陣列的大小，但要注意的是在重新定義陣列的大小後，陣列之前存放的資料都會被清除掉，若要予以保留，必須加上 **Preserve** 關鍵字，如下：

```
Redim Preserve arrayname(n) '重新定義陣列的大小並保留陣列之前的資料
```



注意

- ▶ **ReDim** 陳述式不僅能重新定義動態陣列的大小，也能重新定義固定大小陣列的大小，但不能宣告陣列。
- ▶ **ReDim** 陳述式不能重新定義陣列的維度或宣告成其它型別。
- ▶ 雖然 **Preserve** 關鍵字可以使動態陣列在重新定義大小後保留之前的資料，但若之後的大小比之前的大小還小，那麼少掉的資料還是會被清除掉。
- ▶ 還有一個與陣列相關的陳述式叫做 **Erase**，它可以釋放陣列變數及其元素所佔用的記憶體空間，例如 **Erase A** 可以釋放陣列 **A**。



備註

若要取得陣列的大小、最大索引、最小索引、某個值首次或最後一次出現在陣列的哪個位置等資訊、將陣列排序、將陣列反轉、二元搜尋法…，可以使用 `System.Array` 類別提供的屬性與方法，例如 `GetLength()`、`GetUpperBound()`、`GetLowerBound()`、`IndexOf()`、`LastIndexOf()`、`Sort()`、`Reverse()`、`BinarySearch()`…，第 4-5 節會介紹 `System.Array` 類別的成員。

隨堂練習

撰寫一個程式，令它重複出現對話方塊要求使用者輸入班上學生的分數，若要結束，可以輸入 "Quit"，然後按 [確定]，就會出現另一個對話方塊顯示剛才輸入的每筆分數，如下（提示：由於事先並不知道使用者會輸入幾個學生的分數，所以必須使用動態陣列存放學生的分數）。



- A** 程式開始執行時會出現對話方塊要求輸入分數。
- B** ——輸入分數，結束時輸入 "Quit"，就會出現對話方塊顯示每筆分數。
- C** 若一開始就直接輸入 "Quit" 或按 [取消]，則會出現對話方塊顯示尚未輸入資料。

解答**\MyProj4-4\Module1.vb**

Module Module1

Sub Main()

```
Dim Scores() As Integer      ' 宣告用來存放分數的動態陣列名稱 ( 尚未定義大小 )
Dim Number As Integer = 0    ' 這個變數用來記錄總共輸入幾筆分數
Dim Data As Object           ' 這個變數用來存放使用者在對話方塊輸入的單筆分數
Dim Item As Integer          ' 這個變數用來做為 For Each 迴圈的變數
Dim Result As String = ""    ' 這個變數用來存放最後要顯示的分數字串
Data = InputBox(" 請輸入學生的分數 (0-100) : ")
' 使用迴圈要求使用者輸入分數並存放在動態陣列
' 首先要檢查是否輸入 "Quit" 或按 [ 取消 ] , 否的話才執行迴圈內的敘述
```

```
Do While (UCase(Data) <> "QUIT") And (Data <> "")
```

```
    ReDim Preserve Scores(Number)      ' 根據分數筆數重新配置陣列大小並保留資料
    Scores(Number) = Data               ' 將最新輸入的分數放在陣列最後一個元素
    Number = Number + 1                 ' 分數筆數加一
    Data = InputBox(" 請輸入學生的分數 (0-100) : ")
```

```
Loop
```

```
' 在顯示每筆分數前先判斷有無輸入分數
```

```
If Number = 0 Then
```

```
    MsgBox(" 您尚未輸入任何資料 ! ")
```

```
Else
```

```
' 使用 For Each 迴圈找出每筆分數並存放在字串變數 Result
```

```
For Each Item In Scores
```

```
    Result = Result & Item & " "
```

```
Next
```

```
MsgBox(" 您輸入的分數有 " & Result)
```

```
End If
```

```
End Sub
```

```
End Module
```

■ 4-3 多維陣列

除了一維陣列，VB 2017 亦支援多維陣列，其中以二維陣列最常見。舉例來說，下面是一個 m 列、 n 行的成績單，那麼我們可以使用 Dim 陳述式宣告一個 $m \times n$ 的二維陣列存放這個成績單：

```
Dim arrayname(m-1, n-1) [As type]
```

	第 0 行	第 1 行	第 2 行	……	第 $n-1$ 行
第 0 列		國文	英文	……	數學
第 1 列	王小美	85	88	……	77
第 2 列	孫大偉	99	86	……	89
……	……	……	……	……	……
第 $m-1$ 列	張婷婷	75	92	……	86

$m \times n$ 的二維陣列有兩個索引，第一個索引是從 0 到 $m-1$ (共 m 個)，第二個索引是從 0 到 $n-1$ (共 n 個)，當我們要存取二維陣列時，就必須同時使用這兩個索引，上面的成績單為例，我們可以使用兩個索引將它表示成如下：

	第 0 行	第 1 行	第 2 行	……	第 $n-1$ 行
第 0 列	A(0, 0)	A(0, 1)	A(0, 2)	……	A(0, $n-1$)
第 1 列	A(1, 0)	A(1, 1)	A(1, 2)	……	A(1, $n-1$)
第 2 列	A(2, 0)	A(2, 1)	A(2, 2)	……	A(2, $n-1$)
……	……	……	……	……	……
第 $m-1$ 列	A($m-1$, 0)	A($m-1$, 1)	A($m-1$, 2)	……	A($m-1$, $n-1$)

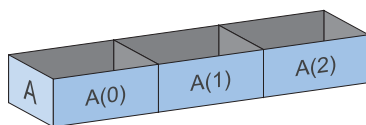
根據上表可知，「王小美」是存放在二維陣列內索引為 (1, 0) 的位置，而「王小美」的國文分數是存放在二維陣列內索引為 (1, 1) 的位置，數學分數則是存放在二維陣列內索引為 (1, $n-1$) 的位置。

下面是幾個不同維度的陣列，您可以比較看看，其中第一個敘述是宣告一個型別為 **Integer** 的一維陣列，索引分別為 0 ~ 2，總共可以存放 3 個元素；第二個敘述是宣告一個型別為 **Integer** 的二維陣列，索引分別為 0 ~ 1、0 ~ 2，總共可以存放 $2 \times 3 = 6$ 個元素；第三個敘述是宣告一個型別為 **Integer** 的三維陣列，索引分別為 0 ~ 1、0 ~ 1、0 ~ 2，總共可以存放 $2 \times 2 \times 3 = 12$ 個元素。

◎一維陣列

Dim A(2) As Integer

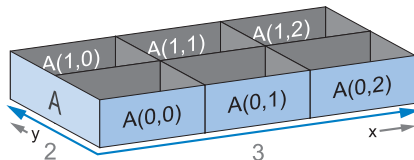
↑
元素個數



◎二維陣列

Dim A(1,2) As Integer

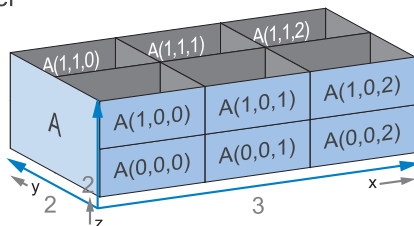
↑↑
y方向的元素個數
x方向的元素個數



◎三維陣列

Dim A(1,1,2) As Integer

↑↑↑
z方向的元素個數
y方向的元素個數
x方向的元素個數

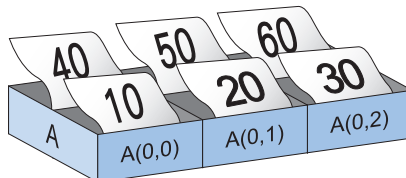


我們還可以宣告超過三維的陣列（維度上限為 32），只是陣列的維度愈多，佔用的記憶體空間就愈多，也愈不容易管理。

此外，我們可以在宣告多維陣列的同時指派初始值，例如：

```
Dim A(,) As Integer = {{10,20,30},{40,50,60}}
```

元素	值
A(0,0)	10
A(0,1)	20
A(0,2)	30
A(1,0)	40
A(1,1)	50
A(1,2)	60



我們同樣可以使用 `System.Array` 類別的 `Length` 屬性取得多維陣列的元素個數，以上面的 `Dim A(,) As Integer = {{10, 20, 30}, {40, 50, 60}}` 為例，`A.Length` 將會傳回 6。



注意

- ▶ 您可以使用 **ReDim** 陳述式重新定義多維陣列的大小，無論是固定大小陣列或動態陣列，但請注意，您只能改變多維陣列最後一維的上限，若改變其它維或最後一維的下限，將會發生錯誤。
- ▶ 當您使用 **ReDim** 陳述式重新定義動態多維陣列的大小時，也可以加上 **Preserve** 關鍵字保留之前的資料。
- ▶ 您不能重新定義多維陣列的維數。
- ▶ 陣列的型別取決於陣列的維度及各個元素的型別，兩個陣列只有在維度相同且各個元素的型別相同時，才會被視為相同型別，至於各個維度的長度則對陣列的型別沒有影響。

隨堂練習

假設有 8 位學生各自舉行三輪比賽，得分如下，試撰寫一個程式，令它重複出現對話方塊提示使用者輸入每位學生在每一輪比賽的得分，輸入完畢後再顯示每位學生的總得分。

	第 1 輪	第 2 輪	第 3 輪
學生 1	5	7.7	8
學生 2	8.8	5.8	8
學生 3	6	9	8.1
學生 4	7.6	8.5	9.5
學生 5	9	9	9.2
學生 6	4	6.3	7.9
學生 7	8.2	7	9.6
學生 8	9.1	8.5	8.9



- A** 程式開始執行時會出現對話方塊要求使用者輸入每位學生在每一輪比賽的得分。
- B** 最後一位學生在最後一輪比賽的得分輸入完畢後，會出現對話方塊顯示總得分。

解答

\MyProj4-5\Module1.vb

Module Module1

Sub Main()

' 宣告一個 8×4 的二維陣列以存放每位學生在三輪比賽中的得分與總得分

Dim Scores(7, 3) As Double

Dim I As Integer, J As Integer ' I、J 為 For 迴圈的計數器

Dim Data As Double ' 這個變數用來暫存使用者輸入的得分

Dim subTotal As Double ' 這個變數用來暫存每位學生的總得分

Dim Result As String = "" ' 這個變數用來存放最後要顯示的總得分字串

For I = 0 To 7

 subTotal = 0 ' 將用來暫存每位學生總得分的變數歸零

 For J = 0 To 2

 Data = InputBox(" 請輸入第 " & I + 1 & " 位學生在第 " & J + 1 & " 輪的得分 ")

Scores(I, J) = Data ' 將使用者輸入的分數存放在二維陣列

 subTotal = subTotal + Data ' 將得分累計暫存在 subTotal 變數

 Next

Scores(I, 3) = subTotal ' 將累計出來的總得分存放在二維陣列

Next

For I = 0 To 7

 Result = Result & " 第 " & I + 1 & " 個學生的總得分為 " & **Scores(I, 3)** & Chr(13)

Next

MsgBox(Result)

End Sub

End Module

這個程式使用一個 8×4 的二維陣列，除了存放 8 位學生在 3 輪比賽中的得分，還多加一個欄位存放各自的總得分，因此，巢狀迴圈內的 **Scores(I, 3) = subTotal** 就是將第 I 位學生的總得分存放在陣列的 **Scores(I, 3)** 位置。

4-4 不規則陣列

除了一般的資料，陣列的元素也可以是另一個陣列，即所謂的**不規則陣列** (jagged array)，以下面的敘述為例，Arr 陣列的第一個元素是 Arr1 陣列，第二個元素是 Arr2 陣列：

```
Dim Arr1() As Integer = {10, 20, 30, 40, 50}
Dim Arr2() As Integer = {100, 200}
Dim Arr(1,0) As Integer
Arr(0) = Arr1
Arr(1) = Arr2
```

若要存取 Arr 陣列的元素，可以寫成如下：

```
Arr(0)(0)      ' 傳回值為 10
Arr(0)(1)      ' 傳回值為 20
Arr(1)(0)      ' 傳回值為 100
```

我們也可以在宣告不規則陣列的同時指派初始值，例如：

```
Dim Arr(0) As Integer = {New Integer() {10, 20, 30, 40, 50}, New Integer() {100, 200}}
```



在您将一个数组指派给另一个数组时，请勿违反下列事项，以免产生错误：

- ▶ 两个数组必须同为实值型别 (value type) 或同为参考型别 (reference type)，若同为实值型别，那么必须为相同型别，若同为参考型别，那么来源数组到目的数组之间必须存在著广义型别转换。
- ▶ 两个数组的维度必须相同。

4-6 與陣列相關的函式

陣列判斷函式 IsArray()

❖ 語法

```
IsArray(varName) As Boolean
```

❖ 說明

若參數 *varName* 指向一個陣列，就傳回 True，否則傳回 False。

取得陣列最小索引函式 LBound()

❖ 語法

```
LBound(arrName, arrRank) As Integer
```

❖ 說明

這個函式會傳回陣列某個維度的最小索引，第一個參數 *arrName* 為陣列的名稱，第二個參數 *arrRank* 為陣列的第幾個維度，1 表示第一維，2 表示第二維，…，依此類推，若第二個參數省略不寫，表示為預設值 1，例如 `LBound(A, 2)` 將傳回陣列 A 第二維的最小索引。由於 VB 2017 不允許變更陣列的最小索引，一律從 0 開始，故 `LBound()` 函式的傳回值恆為 0。



備註

您也可以使用 `System.Array` 類別提供的 `GetLowerBound()` 方法取代 `LBound()` 函式，不過，這個方法的參數為陣列的第幾個維度，0 表示第一維，1 表示第二維，…，依此類推。舉例來說，假設 `Dim A(3, 5) As Integer`，則 `A.GetLowerBound(0)` 就相當於 `LBound(A, 1)`，即取得陣列 A 第一維的最小索引。

取得陣列最大索引函式 UBound()

❖ 語法

```
UBound(arrName, arrRank) As Integer
```

❖ 說明

這個函式會傳回陣列某個維度的最大索引 (**Integer** 型別)，第一個參數 **arrName** 為陣列的名稱，第二個參數 **arrRank** 為陣列的第幾個維度，1 表示第一維，2 表示第二維，…，依此類推，若第二個參數省略不寫，表示為預設值 1。舉例來說，假設 `Dim A(3, 5) As Integer`，`UBound(A)` 將傳回陣列 **A** 第一維的最大索引為 3，`UBound(A, 2)` 將傳回陣列 **A** 第二維的最大索引為 5。

有了 `LBound()` 和 `UBound()` 函式，我們就可以使用這兩個函式找出陣列的上限及下限，進而決定陣列的大小。使用這兩個函式的好處是不用隨時記住陣列的大小，有需要的時候再去取得其上限或下限即可。



備註

您也可以使用 **System.Array** 類別提供的 `GetUpperBound()` 方法取代 `UBound()` 函式，不過，這個方法的參數為陣列的第幾個維度，0 表示第一維，1 表示第二維，…，依此類推。舉例來說，假設 `Dim A(3, 5) As Integer`，則 `A.GetUpperBound(1)` 就相當於 `UBound(A, 2)`，即取得陣列 **A** 第一維的最大索引為 5。



注意

當我們撰寫如下敘述時，表示宣告一個尚未指向任何陣列的二維陣列變數 **A**，它的值為 **Nothing**，日後還要建立一個非空的二維陣列並指派給它：

```
Dim A( , ) As Integer
```

若我們想宣告的是沒有包含元素的二維陣列，而不是不存在的二維陣列，可以在宣告時將其中一個維度宣告為 -1，例如 `Dim A(-1, 5) As Integer`。

隨堂練習

使用 LBound() 和 UBound() 函式將隨堂練習 <MyProj4-3> 的 For Each…Next 迴圈改寫成 For…Next 迴圈。

解答

```
Module Module1
    Sub Main()
        Dim Scores() As Integer = {85, 60, 54, 91, 100, 77}
        Dim MaxScore As Integer = 0, MinScore As Integer = 100
        ' 使用迴圈找出最高分
        For I As Integer = LBound(Scores) To UBound(Scores)
            If Scores(I) > MaxScore Then MaxScore = Scores(I)
        Next
        MsgBox(" 最高分為 " & MaxScore)
        ' 使用迴圈找出最低分
        For I As Integer = LBound(Scores) To UBound(Scores)
            If Scores(I) < MinScore Then MinScore = Scores(I)
        Next
        MsgBox(" 最低分為 " & MinScore)
    End Sub
End Module
```

由於 LBound(Scores) 的傳回值恆為 0，故可直接以 0 取代。此外，我們也可以使用 System.Array 類別提供的方法來改寫 For…Next 迴圈，如下：

```
For I As Integer = Scores.GetLowerBound(0) To Scores.GetUpperBound(0)
    If Scores(I) > MaxScore Then MaxScore = Scores(I)
Next
```


二、練習題

1. 撰寫一個程式，令它宣告一個陣列 {5, 8, 2, 3, 7, 6, 9, 1, 4, 8, 3, 0}，然後在陣列內搜尋最大值及最小值，並在對話方塊中顯示其索引。
2. 撰寫一個程式，令它宣告一個陣列 {5, 8, 2, 3, 7, 6, 9, 1, 4, 8, 3, 0}，然後計算這些元素的平均值，並在對話方塊中顯示結果。
3. 撰寫一個程式，令它宣告一個陣列 {5, 8, 2, 3, 7, 6, 9, 1, 4, 8, 3, 0}，然後在陣列內搜尋第一個值為 6 的元素，並在對話方塊中顯示其索引，若找不到，就在對話方塊中顯示 -1。
4. 假設在縣市長選舉中，候選人 A ~ D 於選區 1 ~ 5 的得票數如下，試撰寫一個程式，令它使用二維陣列存放如下的得票數，然後在對話方塊中顯示每位候選人的總得票數。

	第 1 選區	第 2 選區	第 3 選區	第 4 選區	第 5 選區
候選人 A	1521	3002	789	2120	1786
候選人 B	522	765	1200	2187	955
候選人 C	2514	2956	1555	1036	4012
候選人 D	1226	1985	1239	3550	781

5. 撰寫一個程式，令它使用二維陣列存放如下的元素，然後在二維陣列內搜尋最大值及最小值，並在對話方塊中顯示其索引。

21	22	23	24	25	26
11	12	13	14	15	16
1	2	3	4	5	6