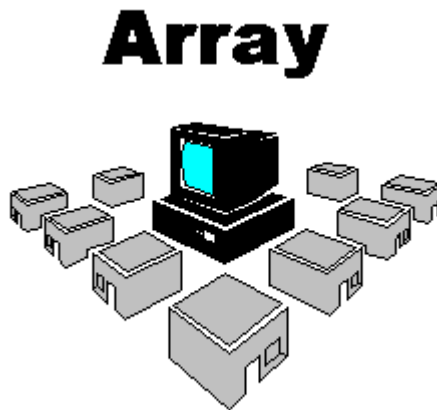


## 第八章 串列與陣列

在程式設計裡，陣列是最常見的資料結構，程式將用它來儲存要進行處理的資料。流行的程式語言都可以執行靜態陣列，但還有許多語言也可執行動態陣列。在本章中，您將可以學到以下的內容：

- 串列（Visual LISP）
- 陣列（VBA）



### 8-1 Visual LISP 的陣列功能

在 Visual LISP 中並沒有陣列這個名辭，但它提供了串列（List）結構來達到一樣的功能。以下，就是各種串列的型態：

```
(1 2 3)
((1 2) 3)
((1 2) (2 4) (3 5))
(1.5 (3.4 2.5 3.4) (34.5 2.2))
(1 . 2)
```

以上這些都是串列。例如，((1 2) (2 4) (3 5)) 經常就是一個三維座標點，最後一個稱為「點對」，第 2、3、4 個串列中還包含著子串列。我們經常將計算得到的數值製成串列並儲存起來，然後再將之拿來做更進一步的後續計算或比較。串列的儲存結構實際上是鏈串，由一組節點組成。這些節點將使用向右指標（指向下一元素的位址）來將它們鏈結在一起，然後用向左指標來指向串列的各元

素。若該元素是原子，則指向該原子的名字，如果該元素是子串列，則指向該子串列。例如，串列 (1 2 3) 與串列 ((1 2) 3) 的儲存結構，如圖 8-1 所示：

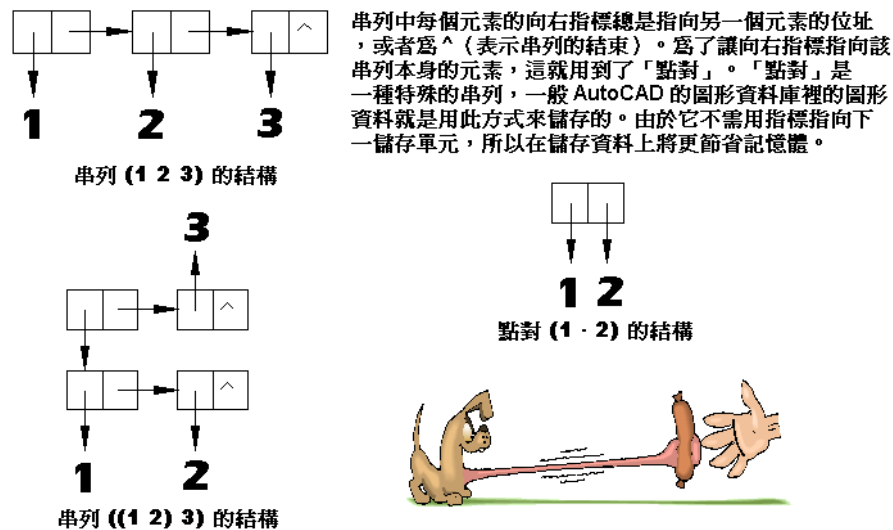


圖 8-1 串列 (1 2 3) 與串列 ((1 2) 3) 的結構

## 8-2 Visual LISP 的串列處理函數

在 Visual LISP 的串列處理函數有下述種類：

- 擷取串列中的部分內容

有 car、cdr 類、nth 與 last 等函數。

- 用於組織串列

有 list、cons、subst 與 append 等函數。

- 用於關聯式串列

有 assoc 函數。

以下將分節詳述。

### 8-2-1 car、cdr 類函數

car, cdr 類的函數將用來擷取串列中的部分內容。首先，car 將傳回串列中的第一個元素，如果第一個元素是子串列則傳回該子串列。例如：

```
_$ (car '(1 2 3))    傳回 1
```

其中，“'”表示要對串列進行操作。

```
_$ (car '((1 2) 3))  傳回 (1 2)
```

接著，cdr 將傳回去掉串列第一個元素的串列。例如：

```
_$ (cdr '(1 2 3))    傳回 (2 3)
```

```
_$ (cdr '((1 2) 3))  傳回 (3)
```

cadr 則可以先對串列進行 cdr 操作，再對所得的串列進行 car 操作。例如：

```
_$ (cadr '((1 2)(3 4)))  傳回 (3 4)
```

而 cdar 則可以先對串列進行 car 操作，再對所得的串列進行 cdr 操作。例如：

```
_$ (cdar '((1 2)(3 4)))  傳回 (2)
```

AutoCAD 將執行類似的 car 和 cdr 的組合函數來達到陣列功能，其組合深度最多可達 4 級。例如，我們也可執行 cdaddr 的串列操作。

## 8-2-2 nth 函數

nth 函數將用來傳回串列中的第 n 個元素，例如：

```
_$ (nth 2 '(11 22 33))  傳回串列中第 3 個元素
```

注意：nth 函數對串列中第 1 個函數的順序是從 0 開始計數的。所以要傳回第 1 個元素，應該使用 (nth 0 list) 語法。

### 8-2-3 last 函數

last 函數將傳回串列中最後一個元素。例如：

```
_$(last '(11 22 33))      傳回 33
```

### 8-2-4 list 函數

list 函數可用於將任意數目的運算式組合成一個串列。其標準語法如下：

```
(list [expression ...])
```

例如：

```
_$(list 100.0 100.0)  傳回 (100.0 100.0)
_$(list 'a ' (b c) 'd)  傳回 (A (B C) D)
```

本函數常用於定義二維或三維的點座標（及由兩個或三個實數所組成的串列）。

### 8-2-5 cons 函數

cons 函數的作用是在串列的起始部分添加一個元素，或構造一個點串列。其標準語法如下：

```
(cons new-first-element list-or-atom)
```

其中，new-first-element 為要添加的新元素，list-or-atom 則為已存在的串列或原子，如果是原子，則構成點對。例如：

```
_$(cons '1 '(2 3 4))      傳回 (1 2 3 4)
_$(cons '(1) '(2 3 4))    傳回 ((1) 2 3 4)
_$(cons '40 100)          傳回 (40 . 100)
```

### 8-2-6 subst 函數

subst 函數的主要用途是在串列中搜索某舊項目，並將串列中出現的每一個舊項目用新項目取代，然後再傳回修改後所得到的新串列。其標準語法如下：

```
(subst newitem olditem lst)
```

其中，newitem 代表要取代的新原子或新串列，olditem 則代表要被取代的舊原子或舊串列。取代完成後，將傳回取代後的新串列。例如：

```
_$ (setq alist ' (1 2 (1 2) 1))      傳回 (1 2 (1 2) 1)  
_$ (subst '100 '1 alist)             傳回 (100 2 (1 2) 100)
```

注意：原串列中的第 1 個和最後一個 1 都被取代，而子串列中的 1 並沒有被取代。

### 8-2-7 append 函數

append 函數的主要作用是可將任意數量的串列組合成一個串列。其標準語法如下：

```
(append [list ...])
```

其中，[list...] 代表任意數量的串列。傳回值為：將所有串列組合到一新串列中。例如：

```
_$ (append ' (1 2) ' (3 4))          傳回 (1 2 3 4)  
_$ (append ' ((1)) ' ((2 3)))       傳回 ((1) (2 3))
```

### 8-2-8 關聯式串列與 assoc 函數

關聯式串列不是函數，它是由「點對」所組成的串列，在 AutoCAD 中多用於儲存圖形資料庫中。例如，一個圓在圖形資料庫中定義為：

```
((-1 . <圖元名: 4005bd58>)  
(0 . "CIRCLE")  
(330 . <圖元名: 4005bcf8>)
```

```
(5 . "2B")  
(100 . "AcDbEntity")  
(67 . 0)  
(410 . "Model")  
(8 . "0")  
(100 . "AcDbCircle")  
(10 209.147 200.271 0.0)  
(40 . 49.3012)  
(210 0.0 0.0 1.0))
```

每個點對前的數字是一群組碼，每一個群組碼都有它們固定的意義。例如，40 就代表圓的半徑值。其中，還有一些是所有類型的圖素都有的，如 -1 代表串列圖素名，0 代串列圖素類型，8 代串列圖層名稱，100 代表串列的資料標記等。

例如，當您想要直接到此圓的圖形資料庫中更改此圓的半徑值時，就要想辦法使用 cons 函數來將 40 與新的半徑值組成新的點對，再使用其他的串列函數來將新點對取代上示圓圖形資料庫裡的 (40 . 49.3012) 這個點對項目。

assoc 則是關聯式串列的檢索函數，它的作用是可於關聯式串列中搜尋一個元素，如果找到，就傳回該關聯式串列項目。其標準語法如下：

```
(assoc element alist)
```

其中，element 為您要搜尋的元素，alist 則為要搜尋的關聯式串列。例如：我們先在 AutoCAD 中畫一個圓，然後輸入如下語法：

```
_$ (setq b (entsel))
```

然後請點取一圓。接著，系統將傳回：

```
(<圖元名: 4005bd58> (234.779 158.854 0.0))
```

請接著鍵入：

```
_$ (setq a (entget (car b)))
```

將傳回：

```
((-1 . <圖元名: 4005bd58>) (0 . "CIRCLE") (330 . <圖元名: 4005bcf8>)  
(5 . "2B") (100 . "AcDbEntity") (67 . 0) (410 . "Model") (8 . "0")
```

```
(100 . "AcDbCircle")(10 209.147 200.271 0.0) (40 . 49.3012)
(210 0.0 0.0 1.0))
```

此時，接著再鍵入：

```
_$(assoc -1 a)
```

就可傳回： (-1 . <圖元名: 4005bd58>)

下面，我們就將之前所說的：直接到圓的圖形資料庫中更改圓半徑值的程式寫法示範給各位。首先，如果您圖面上的圓，其尺寸或位置要修改時，您設計的 AutoLISP/VLISP 程式會如何來處理？若以前面的觀念來撰寫這一個 AutoLISP/VLISP 程式，相信大多數的寫法是將此圓刪除，然後再重新畫一個。

我們將在這個範例裡，將直接進到圓圖素的資料庫裡，將代表半徑與圓心座標的資料抽出並改成新的，再塞回去來直接瞬間更改圓的大小或位置。程式本文如下：

```
(1)(defun c:alist(/ se r ed pp cx cy ed1 pp1 c bp)
(2)(setvar "cmdecho" 0)
(3)(setvar "blipmode" 0)
```

說明：設定本程式的名稱與一些環境系統變數，以利稍後程式的運行。

```
(4)(graphscr)
```

說明：在單螢幕系統裡，graphscr 函數可將螢幕由文字螢幕轉換成繪圖螢幕（就好像按下 <F2> 功能鍵）。而在這裡下 graphscr 指令是要保證目前的螢幕畫面要回到繪圖螢幕上來。

```
(5)(setq se (entsel "\n 請選取欲修改半徑或圓心位置的圓圖素: "))
(6)(initget 1 "M C B")
(7)(setq a (getkeyword "\n 移動(M)/改變半徑(C)/兩者都要(B):"))
(8)
(9)(if (= a "M") (cp))
(10)
(11)(if (= a "C") (cr))
(12)
(13)(if (= a "B")
```



```

(14)    (progn
(15)      (cr)
(16)      (cp)
(17)    )
(18));endif

```

說明：這是開始要使用者點取圓與回答要執行「移動圓」、「改變半徑」或是「兩者都做」的功能。由於操作者可有 3 種選擇；所以，我們特別使用 `initget` 函數來限制操作者鍵入 M、C 與 B 3 個獨立的字元；然後，再使用 `if` 函數來判斷使用者所鍵入的字元。它們是：如果輸入「M」，就執行 `cp` 副程式；如果輸入「C」，就執行 `cr` 副程式；如果輸入「B」，就先執行 `cr` 副程式，後執行 `cp` 副程式。如果您要讓操作者輸入文字來指定程式執行的方向，那麼這整段程式也是一個可以讓您套用的標準模式。注意：最後一系列的「；」符號後所表示的是註解，不會執行的。

```

(19))

```

說明：`alist` 主程式的開頭 `defun` 函數對稱括號。

```

(20)(defun cr()
      (setq r (getreal "\n 請鍵入圓的新半徑: "))

```

說明：`cr` 副程式的開頭語法，並要求操作者輸入圓的新半徑。

```

(20)(setq ed (entget (car se)))
(21)(setq pp (cons (car (nth 10 ed)) r))
(22)(setq ed (subst pp (assoc 40 ed) ed))
(23)(entmod ed)

```

說明：首先，在這段程式裡，我們先使用 `entget` 函數來將選到的圓圖素資料的第一位資料抓出來（使用 `car` 函數），並放到 `ed` 變數裡；此時的 `ed` 變數內容將類似如：

```

((-1 . <圖元名稱: 4005ed58>) (0 . "CIRCLE") (330 . <圖元名稱:
4005ecf8>) (5 . "2B") (100 . "AcDbEntity") (67 . 0)
(410 . "Model") (8 . "0") (100 . "AcDbCircle") (10 240.49 159.471)
(40 . 18.0878) (210 0.0 0.0 1.0))

```



然後，我們將 ed 變數內，排列第 11 位（英文的表示即為 10th；所以，使用 nth 函數時，要指定 10。但是，請注意：不同版本的 AutoCAD，此圖素資料庫內的元素順序可能不同，所以要測試一下來算出真正的排列位置）的半徑數字抓出來；然後，再與新的半徑 r（假設，操作者輸入 50），使用 cons 函數來將它們組合成新的串列，並放到 pp 變數裡。此時的 pp 變數內容為：

```
(40 . 50.0)
```

接著，精采的是：我們去查 CIRCLE 圖素的群組碼，原來，40 是代表圓半徑；所以，我們就使用 assoc 函數來將搜尋此 40 群組碼，找到後，再使用 subst 函數來以新的串列（即 pp）取代舊的串列（即 ed），並將結果取代原 ed 變數；至此，ed 變數就已經被換成新的資料串列了。此時的新 ed 變數內容為：

```
((-1 . <圖元名稱: 4005ed58>) (0 . "CIRCLE") (330 . <圖元名稱:  
4005ecf8>) (5 . "2B") (100 . "AcDbEntity") (67 . 0)  
(410 . "Model") (8 . "0") (100 . "AcDbCircle") (10 240.49 159.471)  
(40 . 50.0) (210 0.0 0.0 1.0))
```

最後，我們再使用 entmod 函數來顯示此新的圓圖素資料。換句話說，entmod 函數會去查此圓的資料庫，並將更改的結果顯示出來。執行至此，您會發現圓已經依據操作者的指定而更改過了。

```
(24))
```

說明：cr 副程式的 defun 函數對稱括號。

```
(25)(defun cp()
```

```
(26)(initget 32)
```

說明：cp 副程式的開頭語法。(initget 32) 的語法將產生一條自圓心而起的橡皮拉線。

```
(27)(setq ed1 (entget (car se)))
```

```
(28)(setq bp (list (cadr (assoc 10 ed1)) (caddr (assoc 10 ed1))))
```

```
(29)(setq c (getpoint bp "\n 請鍵入圓的新圓心位置:"))
```

```
(30)(setq cx (car c))  
(31)(setq cy (cadr c))  
(32)(setq ed1 (entget (car se)))  
(33)(setq pp1 (list (car (nth 9 ed1)) cx cy))  
(34)(setq ed1 (subst pp1 (assoc 10 ed1) ed1))  
(35)(entmod ed1)
```

說明：這段程式的設計原理與第 (21) 條程式一樣，只是圓心的資料是排在串列的第 10 位，而圓心的群組碼為 10 而已。

(36))

說明：cp 副程式的 defun 函數對稱括號。

**載入並執行此程式後的操作：**

請選取欲修改半徑或圓心位置的圓圖素: (請點取一圓)

移動(M)/改變半徑(C)/兩者都要(B): B <Enter>

請鍵入圓的新半徑: (請輸入新半徑)

請鍵入圓的新圓心位置: (請輸入圓心位置)

**分析：**

1. 這個範例是我以前上 AutLISP 課的標準範例，講解完後，很少有人不懂的。您可以於執行本程式後，於 AutoCAD 的指令提示號下鍵入：

指令: !se <Enter>

指令: !ed <Enter>

指令: !pp <Enter>

即在變數前加上「！」符號，就可以要 AutoCAD 立刻顯示該變數的內容。如此，就可以清查所擷取到的串列長什麼樣子，來完全了解所需資料是位於串列裡的位置以及資料串列的結構。

但是請注意：要如此做之前，這些 se、ed 或 pp 等變數都不能先設為臨時變數；亦即，我們一般都在程式測試並執行無誤後，才會將：defun c:alist()這條語法改成：

```
defun c:alist(/ se r ed pp cx cy ed1 pp1 c bp)
```

這樣，就可以包含臨時變數（局部變數）的定義。

2. 現在，您應該恍然大悟了，原來 AutoCAD 存檔的就是這些資料串列，而不是影像。AutoCAD 在讀檔時，會將這些資料串列轉成「向量」後，再重新顯示於螢幕上，讓您可以再繼續編輯...
3. 除了圖素資料的直接擷取與抽換技巧外，使用 `initget` 函數來限制操作者的鍵入文字也是新的技巧。

## 8-3 VBA 的陣列功能

在陣列的處理上，VBA 將有：一維陣列、多維陣列與動態陣列等功能。以下，就分節詳述之。

### 8-3-1 一維陣列

在 VBA 裡，陣列是必須先宣告的。其陣列的宣告方式與其他的變數是一樣的，它可以使用 `Dim`、`Static`、`Private` 或 `Public` 等語法來宣告。例如：

<code>Dim str(50) As String</code>	'宣告可存儲 51 個字元的一維陣列
<code>Dim dbl(9) As Double</code>	'宣告可存儲 10 個雙精度浮點數的一維陣列
<code>Dim pt(0 to 2) As Double</code>	'宣告可存儲 3 個雙精度浮點數的一維陣列，在 AutoCAD 中，經常用它來表示一個點。

一般，陣列的第 1 個元素的預設索引號碼為 0，這和 C 語言類似。不過，陣列的第 1 個元素索引也可以不為 0，您可以在 `Dim` 語法中指定其上下限：

<code>Dim str(1 To 50) As String</code>	'宣告可存儲 50 個字元的一維陣列
<code>Dim dbl(2 To 9) As Double</code>	'宣告可存儲 8 個雙精度浮點數的一維陣列

您可以引用陣列索引來直接使用其中的元素。例如：

```
str(1) = "The first string"  
str(2) = "The second string"  
pt(0) = 100.0
```

以下，就是一個輸出 5 個訊息框來顯示 1~5 的平方根的範例：

```
(1);;;array demo-----arr.dvb
(2);;;function: calculate square of 1 to 5
(3)
(4)Option Explicit
(5)Public Sub arr()
(6)Dim x(5) As Double
(7)Dim i As Integer
(8)For i = 1 To 5
(9)  x(i) = Sqr(CDbl(i))
(10) MsgBox "x(" & i & ") = " & x(i) & vbCr
(11)Next
(12)End Sub
```

**分析：**在程式的第(9)行裡，我們使用了雙精度浮點陣列，將 1~5 的平方根值賦予陣列，然後再於第(10)行中使用 MsgBox 將它們顯示出來。整個過程都是使用 for...next 迴圈來處理的。

### 8-3-2 多維陣列

對多維陣列來說，下面是幾個多維陣列宣告的範例：

```
Dim Matrix(9)(9)(9) As Double
```

以下就是實際的使用範例：

```
Matrix(0)(0)(0)=0
Matrix(9)(9)(9)=999
```

### 8-3-3 動態陣列

有時候，我們可能不知道需要為數多大的陣列，那怎麼辦？別擔心，您仍可以不提供放置最大資料個數的陣列（以免陣列大部分空著，浪費資源），因為 VBA 可以支援動態陣列。動態陣列的長度可以在程式執行時自動視情況改變。

同時，也有可能要當操作者輸入許多資料，且應用程式要處理與顯示時才需要陣列，無論如何，這個時候都是使用動態陣列的時機。

要產生動態陣列，就如同一般的 Dim 語法一樣來宣告，但不要指定陣列維數，如下所示：

```
Dim Matrix() As Double
```

當在以後的程式中知道了要在陣列中存放多少元素時，再使用 ReDim 語法來產生確定的陣列維數。此時，宣告實際維數的語法為：

```
ReDim Matrix(UserCount)
```

其中，UserCount 也可以是操作者所輸入的數值。例如：

```
ReDim Matrix(9)
```

```
ReDim Matrix(9)(9)(9)
```

注意：ReDim 只能改變陣列的維數，而不能改變陣列的類型。

以下，我們根據操作者所輸入的整數 n，計算從 1~n 每個數的平方根值，並顯示於訊息框中。程式本文如下：

```
(1);;;dynamic array demo-----dynarr.dvb  
(2);;;function: calculate square of 1 to n  
(3)  
(4)Option Explicit  
(5)Public Sub dynarr()  
(6)Dim x() As Double  
(7)Dim i As Integer, n As Integer  
(8)n = InputBox("Input array number:")  
(9)ReDim x(n)  
(10)For i = 1 To n  
(11)  x(i) = Sqr(CDbl(i))  
(12)  MsgBox "x(" & i & ") = " & x(i) & vbCr  
(13)Next  
(14)End Sub
```

分析：在上示程式中，我們將於第(6)行中定義動態陣列，然後在第(9)行處

確定了它的維數。

## 啓發性習題

### 一.選擇題(單複選混合)

- 1.(    ) 有關 AutoLISP/VLISP 「串列」的敘述，何者爲真？
  - (a) 可用來代表座標點
  - (b) 可用來代表圖形資料庫裡的組成元素
  - (c) 可用來將需要的文數字元組合並儲存成我們所希望的內容，方便後續再拿出來運用
  - (d) 以上皆真
- 2.(    ) 以下敘述，何者爲真？
  - (a) 不同版本的 AutoCAD，於圖素資料庫內的元素順序可能不同
  - (b) subst 函數能用來組合新的串列
  - (c) assoc 函數能用來搜尋群組碼
  - (d) cons 函數能以新的串列（即 pp）取代舊的串列
- 3.(    ) 有關 VBA 「陣列」的敘述，何者爲非？
  - (a) Dim Matrix(9)(9)(9) As Double 是動態陣列的宣告語法
  - (b) Dim Matrix() As Double 是多維陣列的宣告語法
  - (c) Dim x(5) As Double 是一維陣列的宣告語法
  - (d) 以上皆非

### 二.實作問答題

1. 當我們要以 CHAMFER 指令來繪製倒角圖形時，一般是要回答其第一條線與第二條線的截角距離，才可以繪製。但有時候，我們只知道其中一條線的截角距離與其水平或垂直夾角，而不知道第二條線的截角距離，您可以設計如此條件的功能嗎？（使用 AutoLISP/VLISP 撰寫）

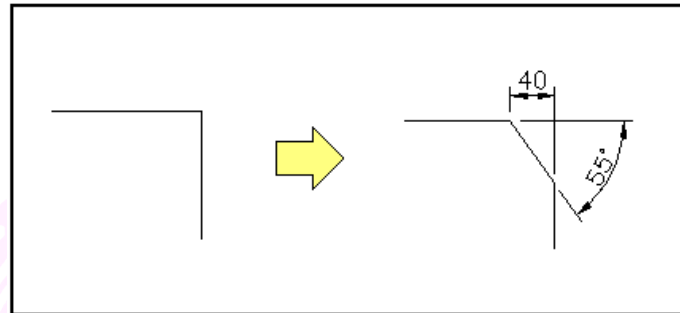


設計詢問句：

- (1) 第一條線的截角距離: 40 <Enter>
- (2) 與第一條線的角度: 55 <Enter>
- (3) 請選取第一條線:
- (4) 請選取第二條線:

解答檔案名稱：DCHAMFER.LSP，V-DCHAMFER.LSP

完成圖例：



2. 請設計一個可自動繪製截斷線的功能（以 AutoLISP/VLISP/VBA 撰寫）。

設計詢問句：

- (1) 請輸入比例: 3
- (2) 請輸入插入點與角度: (請選取線上二點)

解答檔案名稱：IB.LSP，V-IB.LSP，IB.DVB

配合檔案名稱：BREAK.DWG (截斷線圖塊圖形)

完成圖例：

