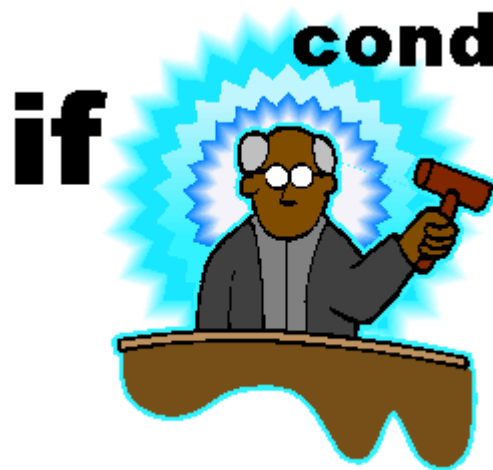


第六章 判斷式

如果沒有控制語法，程式就只能從左至右、自上向下地依序地貫穿這些語法。當然，對單純程式來說，的確是可以用單向流程來編寫，但是對於複雜邏輯的流程，依序執行的程式就無法滿足了。因此，任何的高階語言皆有判斷語法與迴圈結構，來讓執行語法的順序可以滿足複雜的邏輯流程。在本章中，您可以學到：

- if 判斷式
- cond 判斷式與 Select Case 判斷式
- 嵌套選擇判斷



6-1 Visual LISP 的 if 判斷式

在一般的生活邏輯思維上，通常我們總是必須根據特定的條件來作出不同的決定。例如，「如果今天不下雨就去打球」等。程式也是如此，在程式中使用判斷語法，將可用來檢驗條件並根據判斷的結果來指示程式執行的方向。有了判斷式語法結構就可控制程式執行的流程，使程式能以相對較少的指令群來靈活地處理各種情形與各種程式問題。在 Visual LISP 中，if 判斷式的標準語法結構如下：

```
(if condition  
  (statement)  
  ([elsestatement])  
)
```

參數說明：

condition 要判斷的運算式

statement 如果吻合 condition（不為 nil），則執行此運算式

elsestatement 如果不吻合 condition（為 nil），則執行此運算式

與其他高階語言不同的是：Visual LISP 的 if 判斷是只能是雙選擇，而且 statement 和 elsestatement 只能是一個運算式。如果要使用執行多個運算式就要使用 progn 函數。例如：

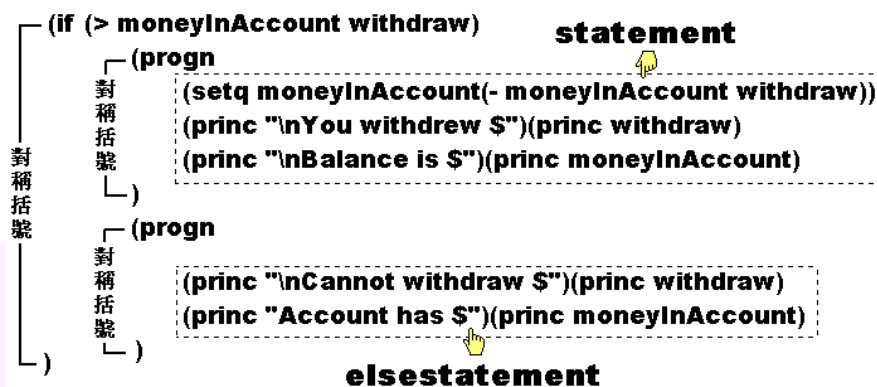


圖 6-1 progn 函數的結構

上面這幾條程式的意思就是說：如果 moneyInAccount 大於 withdraw 的話，就執行第一個 progn 所包含的那三條程式；若 moneyInAccount 不大於 withdraw，就執行第二個 progn 所包含的那二條程式。

在下示範例中，程式將提示您輸入一個字元，然後再根據輸入是字母還是非字母來輸出對應的訊息：

```
(1);;;if judge demo-----if.lsp
(2);;;function:judge if a character is a letter
(3)
(4)(defun c:if (/ ch)
(5)  (setq ch (getstring "請輸入字串:"))
(6)  (setq ch (strcase ch))
(7)  (if (and (>= ch "A") (<= ch "Z"))
(8)      (prompt "沒錯！您輸入的是字元！")
(9)      (prompt "對不起！您輸入的不是字元喔！"))
(10)  )
(11)  (princ)
```

(12))

執行結果：根據輸入字元而定。若輸入的是字元，就輸出 "沒錯！您輸入的是字元！"；如果輸入的非字元，則列印 "對不起！您輸入的不是字元喔！"。

分析：程式中的第(6)行會將輸入字元轉換為大寫字元，而對非字母字元則不起作用。第(7)行～第(10)行是進行判斷並執行對應運算式的程式。

6-2 Visual LISP 的 cond 判斷式

cond 判斷式將提供了一種特殊的多項選擇判斷式。它可用來檢測運算式的各種值，然後再選擇適當的一個來執行。其標準語法如下：

```
(cond
  [((test1) (statement1))]
  [((test2) (statement2))]
  [((test3) (statement3))]
  ...
  [(t (statementn))])
```

cond 函數將對它的判斷運算式 test1、test2、test3...先進行求值，直到其中一個吻合其條件式（不為 nil）為止。然後，再去執行該判斷運算式後的執行運算式（statementx）。如果所有判斷運算式均為 nil（即都不吻合其判斷條件），則執行最後的執行運算式（statementn）。下面，我們就在設計一個簡單的範例來說明。已知函數：

$$f(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

程式本文如下所示：

```
(1);;;cond judge demo-----cond.lsp
(2);;;function: assign y value by x value
(3)
(4)(defun c:cond (/ x y)
(5) (setq x (getreal "\n 請輸入一數字:"))
```

```

(6) (cond
(7)   ((> x 0)(progn(setq y 1) (prompt "x>0")))
(8)   ((= x 0)(progn(setq y 0) (prompt "x=0")))
(9)   (t (progn(setq y -1) (prompt "x<0"))))
(10) )
(11)(princ)
(12))

```

執行結果： 請輸入一數字:-9

x<0

分析：程式的第(6)行～第(10)行是 cond 判斷語法，如果 x>0 則執行第(7)行，如果 x=0 則執行第(8)行，其他情況（即 x<0）執行第(9)行。

6-3 Visual LISP 的嵌套判斷式

在程式編寫的過程中，常常需要使用嵌套判斷構成來處理一些不平常的條件。嵌套判斷結構將讓您可以一層一層地對複雜的條件進行處理。外層結構中的判斷可用於測試基本或一般的條件，而內層結構將用於測試較複雜或特殊的條件。下面的範例是在上一節的 if.lsp 的基礎上所改寫的，其用意將判斷字元是否為字母，若是字母，再判斷所輸入的字母是大寫或是小寫：

```

(1);;;if judge demo-----if2.lsp
(2);;;function: judge if a character is an upper or lower letter or other one
(3)
(4)(defun c:if2 (/ ch)
(5)  (setq ch (getstring "\n 請輸入一個英文字母:"))
(6)  (if(or
(7)      (and (>= ch "A") (<= ch "Z"))
(8)      (and (>= ch "a") (<= ch "z")))
(9)      );end or
(10)  (if(and (>= ch "A") (<= ch "Z"))
(11)      (prompt "\n 您輸入的是大寫字元.")
(12)      (prompt "\n 您輸入的是小寫字元.")
(13)      );end if

```

```
(14) (prompt "您輸入的非字元！")  
(15) );end if  
(16) (princ)  
(17))
```

執行結果：請輸入一個英文字母:R
"您輸入的是大寫字元."

分析：程式中的第(6)行～第(9)行語法的判斷為第一層，用來判斷輸入字元是否為字母。第(10)行～第(13)行語法的判斷為第二層，用來判斷輸入字元是大寫字母還是小寫字母。如果二層的判斷都無法吻合，那就報告說：「您輸入的非字元！」。

6-4 VBA 的 if 判斷式

VBA 中的 if 判斷式與 Visual LISP 有所不同，它可以有單選與多選功能。

6-4-1 單選的 if 語法

單選的 if 標準語法可以有兩種寫法：單行語法和多行塊語法。

● 單行語法

If condition Then statement

● 多行語法

If condition Then
statements
End If

其中，condition 是一條件式，但它是任何可以計算數值的運算式。VBA 會將這個值解釋成 True 或 False：一個為零的數值將被認為是 False，而任何非零數值都會被看成 True。若 condition 為 True，則 VBA 將執行 Then 關鍵字後面的所有 statements。您也可以使用單行或多行語法來有條件地執行一個語法（以

下的兩條程式是相等的)：

```
If ssetObj.Count=1 Then lineObj=ssetObj.item(0)
```

```
If ssetObj.Count=1 Then  
    lineObj=ssetObj.item(0)  
End If
```

注意：If...Then 的單行語法格式並不需用到 End If 語法。如果 condition 為 True 時，要執行多行程式碼，就必須使用多行語法 If...Then...End If 語法。

6-4-2 多選的 if 語法

當被測條件可能有一種以上的結果時，就可以使用多選的 if 語法。多選 if 語句中將使用 If...Then...Else 這樣的語法結構，從而根據條件來執行其中的一個語法。其標準語法如下：

```
If condition1 Then  
    [statementblock-1]  
[ElseIf condition2 Then  
    [statementblock-2]]  
    [ElseIf condition3 Then  
    [statementblock-3]] ...  
[Else  
    [statementblock-n]]  
End If
```

在上述這樣的語法結構中，VBA 首先將測試 condition1。如果它為 False，VBA 就測試 condition2...餘此類推，直到測試出一個為 True 的條件為止。當它找到一個為 True 的條件時，VBA 就會執行對應的運算式群，然後再執行 End If 後面的程式。如果所有的測試都不是 True，那麼 VBA 將執行 Else 下的運算式群。您可以使用任意數量的 ElseIf 子句，或者一個也不用。Else 子句是可有可無的。此外，您可以有一個 Else 子句，而不管有沒有 ElseIf 子句。例如：以下的程式範例，我們將根據分數 (score) 來給等級 (Mark) 賦予一值：

```
If score<60 Then  
    Mark="failed"
```



```
    ElseIf score<70 Then
Mark="D"
    ElseIf score<80 Then
Mark="C"
    ElseIf score<90 Then
Mark="B"
    Else
        Mark="A"
End If
```

注意：您可以添加更多的 ElseIf 子句到 If...Then 結構中去。但是，當每個 ElseIf 都將相同的運算式比作不同的數值時，這個結構編寫起來就很乏味。在這種情況下，我們建議您可以使用 Select Case 的判斷結構。

6-5 VBA 的 Select Case 判斷式

VBA 提供 Select Case 結構來替代 If...Then...Else 語法結構，從而可以在多個運算式群中有選擇地執行其中一個。Select Case 語法的能力與 If...Then...Else 語法類似，但在多選的情況下，Select Case 語法將使程式更加易讀。

Select Case 將在結構的上方處理一個測試運算式並只計算一次。然後，VBA 將運算式的值與結構中的每個 Case 的值進行比較。如果相等，就執行與該 Case 相關聯的運算式群。其標準語法如下：

```
Select Case testexpression
    [Case expressionlist1
    [statementblock-1]]
    [Case expressionlist2
    [statementblock-2]]
[Case Else
    [statementblock-n]]
End Select
```

在上述這樣的語法結構中，每一個 expressionlist 都可以是一個或幾個值的運算式串列。如果在一個串列中有多個值，就必須使用逗號來將值分隔開。每一個 statementblock 中都可能含有零個或多個運算式。如果不止一個 Case 與測試運算式相匹配，則系統將只對第一個匹配的 Case 執行與之相關聯的運算式群。

如果在運算式列表中沒有一個值與測試運算式相匹配，則 VBA 就會執行 Case Else 子句（此項是可有可無的）中的語法。例如：

```
Select Case WeekDay(Date)
Case=1
    DayName="Monday"
    Message="Wish you have a nice week! "
Case=6
    DayName="Saturday"
    Message="Wish you have a nice weekend! "
Case=7
    DayName="Sunday"
    Message="Did you have a nice weekend? "
Case Else
    Message="Please work studios!"
```

注意：Select Case 結構每次都會在開始處計算運算式的值。而 If...Then...Else 結構將為每個 ElseIf 語法來計算不同的運算式。只有在 If 語法和每一個 ElseIf 語法來計算相同運算式時，才能用 Select Case 結構來替換 If...Then...Else 結構。

6-6 VBA 的嵌套判斷式

與 Visual LISP 一樣，在程式編寫的過程中，常常需要使用嵌套判斷構成來處理一些不平常的條件。嵌套判斷結構將讓您可以一層一層地對複雜的條件進行處理。外層結構中的判斷可用於測試基本或一般的條件，而內層結構將用於測試較複雜或特殊的條件。下面的範例是以 6-3 節的範例為基礎來改寫的，其用意將判斷字元是否為字母，若是字母，再判斷所輸入的字母是大寫或是小寫：

```
(1)'if judge demo-----if2.dvb
(2)'function: judge if a character is an upper or lower letter or other one
(3)
(4)Option Explicit
(5)Public Sub if2()
(6)Dim ch As String
(7)Ch = ThisDrawing.Utility.GetString(False, "Enter a letter:")
```



```
(8)If (((ch >= "A") And (ch <= "Z"))) Or _  
(9)  ((ch >= "a") And (ch <= "z"))) Then  
(10) If ((ch >= "A") And (ch <= "Z")) Then  
(11)     ThisDrawing.Utility.Prompt "You entered an upper case letter."  
(12) Else  
(13)     ThisDrawing.Utility.Prompt "You entered an lower case letter."  
(14) End If  
(15)Else  
(16) ThisDrawing.Utility.Prompt "Your input was not a letter."  
(17)End If  
(18)End Sub
```

執行方式：Enter a letter:b

You entered an lower case letter.

分析：第(8)行中最後的“_”為續行符號，表示下一行的內容將接續在此行之後。這對於無法在螢幕全部顯示的長語法是很好用的。第(8)、(9)行將開始判斷的第一層，用意在於判斷其是否為字母。第(10)行～第(15)行就是判斷的第二層，用來判斷字母是大寫或小寫，並執行對應的運算式。

啟發性習題

一.選擇題(單複選混合)

- 1.() cond 語法與 if 語法的關係是：
 - (a) if 將替 cond 語法鋪設多個執行運算式
 - (b) cond 將替 if 語法鋪設多個執行運算式
 - (c) 兩者各自獨立並無關聯
 - (d) 以上皆非

- 2.() 有關 VBA 中的 if 語法，以下敘述何者為真？
 - (a) 多行語法必須以 End If 結尾
 - (b) 多行語法必須以 End 結尾
 - (c) 不論是 statement 或 elstatement 都可直接寫入多個運算式

(d) 以上皆非

3.() 以下說法何者正確？

- (a) VBA 的 Select Case 結構每次都會在開始處計算運算式的值。而 If...Then...Else 結構將為每個 ElseIf 語法來計算不同的運算式。所以，只有在 If 語法和每一個 ElseIf 語法來計算相同運算式時，才能用 Select Case 結構來替換 If...Then...Else 結構
- (b) 多選的 if 語法必須用到 ElseIf 或 Else 來居中轉折，然後仍以 End If 結尾
- (c) 嵌套判斷式結構將讓您可以一層一層地對複雜的條件進行處理。外層結構中的判斷可用於測試基本或一般的條件，而內層結構將用於測試較複雜或特殊的條件。
- (d) 以上皆真

二.實作問答題

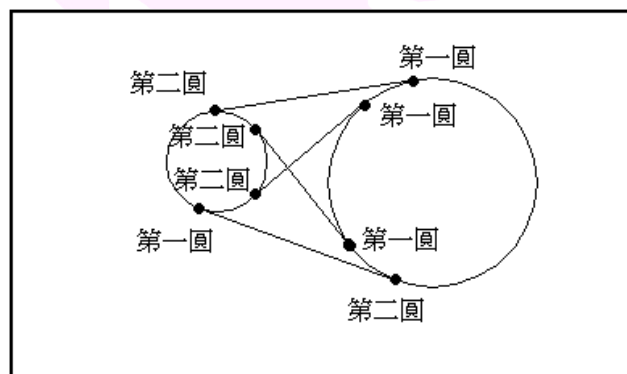
1. 在 AutoCAD 中並沒有兩圓切線功能，請設計之（以 AutoLISP/VLISP 撰寫）。

設計詢問句：

- (1) 請選取第一圓:
- (2) 請選取第二圓:

解答檔案名稱：CTAN.LSP，V-CTAN.LSP

完成圖例：



2. 請設計一個自動繪出機械正齒輪圖形的程式（以 AutoLISP/VLISP/VBA 撰寫）。

設計詢問句：

- (1) 正齒輪齒數:<24>
- (2) 徑節:<4>
- (3) 壓力角:<14.5>
- (4) 圓弧精確度:<10>
- (5) 齒輪的中心點:
- (6) 要繼續完成繪製嗎?:<Y>

解答檔案名稱：GEAR.LSP，V-GEAR.LSP，GEAR.DVB

完成圖例：

