MAPLE 編程基本要點

[稱爲執行區，> 稱爲 “prompt”，勉強翻做 “提示符號”。

撰寫、編輯 執行區內容

游標停在執行區任何一處，

按 Enter，是 將這個執行區的內容 載入 並 執行

按 Shift + Enter，是 在游標後換列或新增一列

按 ctrl + j，是 在執行區下方 新增一執行區

按 ctrl + k，是 在執行區上方 新增一執行區

按 ctrl + Delete，是刪除當前執行區

按 F3 是將游標後的內容 分成下一執行區

按 F4 是當前的執行區與下方執行區合併

“ 隔開 ”，無論是一格、多格、或換列，執行效果相同。一個好的 programmer，爲了程式除錯、後續修改、功能擴充，必需採取 縮排 以及撰寫必要的 註解 使自己 或後繼者 容易了解程式的目的與結構。這是一開始就必須養成好習慣！

```
# 爲註解的開端(# 後 同一列的內容 電腦不編譯)
% 代表最新的執行結果
; (可執行) 陳述的結尾，執行後將顯示於屏幕
: (可執行) 陳述的結尾，執行後不顯示於屏幕(不顯示 不表示未執行)
:= 賦值或覆值，例如： [> L := R: # L 被覆值爲 R
    [> L = R; # 屏幕顯示 L = R 字樣，無意義
    [> L = R: # 無意義
```

括號

凡是程式、函數，名稱後一定都要緊跟著圓括號（ 或有 或無 傳遞參數 ）配對。
進行二元運算 + - * / 時使用的結合分配律 僅可使用配對的圓括號（ ），不可使用其他括號。
名稱後若使用配對的方括號 [] 則是用來表示某成員的。例如：

```
> A := [2,0,1,6]: # A 是個一維陣列，A[3] 的值爲 1
> B := [[1,2,3],[4,5,6]]: # B 是個二維陣列，B[2,2] 的值爲 5
> C := [[[1,2],[3,4]],[[5,6],[7,8]]]: # C 是個三維陣列，C[2,1,2] 的值爲 6
> S := {2,0,1,6} : # S 是個集合，S[1],S[2],S[3],S[4] 分別爲 0,1,2,6
> L := [2,0,1,6] : # L 是個一維陣列，L[1],L[2],L[3],L[4] 分別爲 2,0,1,6
> v := <2,0,1,6> : # v 是個四維行向量，v[1],v[2],v[3],v[4] 分別爲 2,0,1,6
```

長度相同的一維陣列彼此可做 + - 運算，長度相同的行向量彼此可做 + - 運算，
但 長度相同的一維陣列 與 行向量 並不通用。

再簡單整理一次：

F 後緊跟有 (則 F 是程式或函數。

A 後緊跟有 [是用來表示 A 的某成員。

R 後緊跟有 { 是錯誤的，花括號 {x,y,z} 只能用來定義集合。