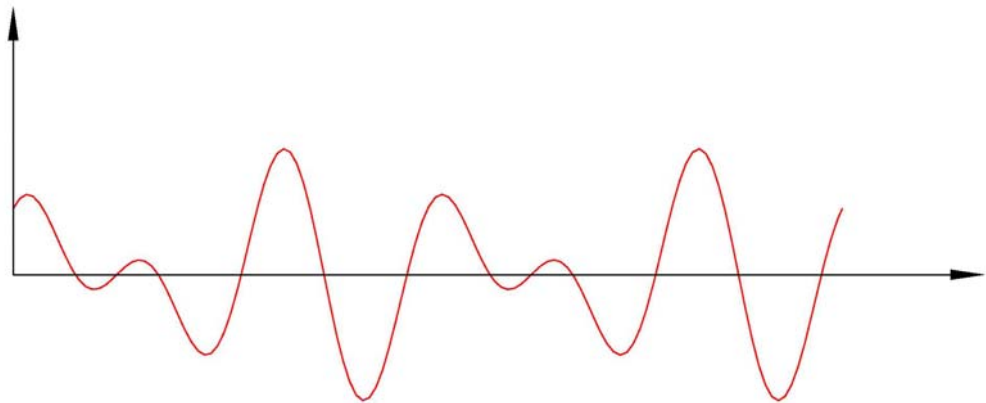


AutoLISP 使用學習經驗談

一、前言：

電腦早已經是大家日常夥伴，各種軟體讓我們的生活與工作變得更輕鬆更多采多姿，尤其網路發達之後以及行動裝置如手機之普及，讓資訊之取得更加方便，社群軟體如 LINE 的崛起，更讓訊息、照片及影片天天如潮水般湧來，也許有人已經快要有招架不住之感覺。

回想比較早期就接觸電腦的年長者，或許還有從 FORTRAN 學習寫程式的痛苦經驗，或者要去記 DOS 指令的回憶，但也慢慢被 Windows 圖形化介面跟操作簡易的套裝軟體沖淡早期對電腦愛恨之情絮。大家都知道電腦很厲害，只用 1 跟 0 就可以靠快速運算變出無限之可能，其中一個關鍵就是軟體。隨著電腦的進步與發展，用過的軟體已經數不清，最普遍的 Word、Excel、PowerPoint 等文書軟體以及影像編輯軟體已幾乎無人不會使用，對工程人員來說 AutoCAD 繪圖軟體則是一個有力的助手。至於 Basic、Fox、C+、VisualBasic 等等程式語言軟體，可能就不是一般使用者都會去接觸的了。不過電腦功能非常強大，各種套裝軟體其實也都還留有一部分可以讓使用者自己去掌控增加其功力的地方，比如 Excel 的 VBA、巨集指令等可以自己編一些簡易執行程式讓工作更符合本身需求，不但有一點創作的樂趣還可以增加工作效率。其中，AutoCAD 中的 AutoLisp 就是類似的工具，多年前在 AutoCAD R12 版時接觸到紙本「AutoLISP 程式員參考手冊」，依樣畫葫蘆也得到一點心得，乃將此分享給有興趣但還在徘徊的朋友。



二、AutoLISP 簡要概述

1. AutoLISP 簡單來說，可以說是將手動操作步驟，一步步寫成批次文字檔，讓軟體讀了去執行，其中可以加迴圈、判斷等等程式手法，利用電腦極高速的執行力去提高效率。
2. 不能免俗的，AutoLISP 程式員參考手冊中一定會有：資料型態(符號、串列、

字串...), 運算符號(加減乘除、次方、Sin、Cos...), 各種指令(畫線 畫圓複製...), 圖元選集(可取得圖元資訊如座標、圖層...), 邏輯(判斷 if、迴圈 while...)等等一大票的資訊。但初學者其實可以不須被沖昏頭, 只要先掌握一小部分必要的指令及寫法, 就可以先試玩了

3. 很特別的一點是, AutoLISP 程式每行指令都是用成對 () 組成。例如:

;簡易範例 執行指令為 text

```
(defun C: text()  
  (setq a 12)  
  (setq b(+ a 5))  
  (print b)  
)
```

上面的例子應該不難猜懂在寫什麼, 當你在 指令行鍵入 text, 按 ENTER, 就會在下一行地方, 顯示出 17。如果程式中括號不成對, 或括錯地方, 不是根本不執行要不就是得到錯誤結果。程式中 defun 是在定義執行指令名稱。

4. 使用 AutoCAD 很重要的一個資訊, 就是 點位。在手動操作上, 可以直接用滑鼠去點或輸入相對座標或絕對座標。在 LISP 中點位則是以串列表現。例如:

'(123.45 22.145) 裡邊都是數字可以這樣設, 但要注意括號前面有一小撇
或 (list A B)、(list 3 5) 即將 (A B)或(3 5) 組成一組座標
也都是用 括號 組合起來的。

5. 由前面簡例中可以看到 setq 指令

它就是將內容設定給變數, (setq a 12) 將變數 a 設為 12

(setq pt '(33 11.5)) 將座標 (33 11.5) 設給變數 pt

6. 程式中要加說明讓自己日後容易記得時, 可在最前面加 ;

例如 *;此程式是 1050709 編撰用來練習畫線用的* [此行就不會被執行]

7. 必須先知道的指令還有 command

當你手動在指令行鍵入的所有指令, 幾乎都可以用 command 來執行, 例如:

畫線 (command "line" pt1 pt2 pt3 "") [最後的 ""表示按一下 Enter]

縮放螢幕視野 (command "zoom" "e")、(command "zoom" "s" "0.8x")

畫圓 (command "circle" pt 100)

其中繪圖指令 line、zoom、circle 都是常用指令。即使你平都用滑鼠去點圖標(icon)來繪圖, 不太知道繪圖指令名稱, 一樣可以很容易查得到, 例如去點 繪圖-塑型-擠出 時在指令行也會出現 extrude, 就是該繪圖指令名稱。這裡的**重點**其實是在主指令打完之後, 各個後續動作(如輸入點位)及副指令等等, 甚至若有最後一個按 Enter 動作, 都不得出差錯, 否則就停在那。

8. AutoLISP 的運算式也比較奇怪, 加減乘除是放在前面。例如:

(setq A (+ 3 5)) 即 A=3+5 (setq B (- 6.2 4)) 即 B=6.2-4

(setq C (* 2.1 2)) 即 C=2.1*2

(setq D (/ 2.0 3)) 即 D=2/3

[※ 除法要特別注意，若寫 $(/ 2 3)$ 會用整數計算 $D=0.66667=0$]

9. 設定點位座標，也是比較需要先知道的。前面已經提到，可以用：

(setq pt '(123.45 22.145))、(setq pt (list A B))或(setq pt (list 3 5)) 來設定。

但後續可能會再利用此點位來延伸相對座標。就可用一個極座標指令 polar。

(setq pt1 (polar pt θ d)) 要注意其中 θ 是徑度量

當然也可以用 相對直角座標方式去設定

(setq pt1(list (+ x dx)(+ y dy)))

10. 要由已知座標串列取出其中 x 或 y 數字之方式也蠻有意思

當 pt(x y z) 是一個 3d 座標點串列，你想知道 x y z 數值時

(setq X (car pt)) 此 X 就是截取出 pt 中的第一個變數值 x

在指令欄中可以鍵入 !X 就會回應顯示出其數字可用來比對看有沒抓錯

(setq Y (cadr pt))

其中 cadr 的意義是取出 pt 中除去第一個變數值後之串列 (y z)

再接著取出此串列的第一個變數值 y 方式真的很特別

(setq Z(caddr pt)) 其意義不用再解釋可以自己體會猜測一下

類似指令還有 caar caddr cadar 等等，就等進門後有需要再繼續研究

11. 判斷之指令 if 是寫程式必要之工具，例如：

(if (< a 10) (print "a<0") (print "a >= 0"))

當 a 小於 10 執行第一個括號內容，即印出 a<0 字串

當 a 不小於 10 執行下一個括號內容，即印出 a >= 0 字串

此處要注意的在於如果符合判斷式時要執行的不只一行時

要多一個指令跟一付雙括號，如下例：

```
(if (/= a 0)
  (progn
    (setq c (/ 5.0 a))
    (setq pt (list c 11.5))
  )
  (print "pt 不存在"))
```

也就是當 a 不等於 0 時，要執行兩行指令，就要加一個 progn 來組合

12. 迴圈更是程式最有力工具之一。寫程式常要去執行大量重複或類似工作，就非靠迴圈不可。例如：

```
(while (< x (* 4 pi))
  (setq pt (list x y))
  (setq x(+ x 0.1))
  (setq y1(* 2 (* (sin x) (cos x))))
  (setq y2(- (* 4 (expt (cos x) 3)) (* 3 (cos x)) ))
  (setq y (+ y1 y2)))
```

)

此處要注意的是，迴圈執行的條件為 $x < 4\pi$

故每執行一次，其判斷式之值 x 必須有所改變，如例子中之 增加 0.1

如忘了寫這一步，會永遠執行不完，跌入無窮迴圈之夢魘。

AutoLISP 相關參數使用與指令非常繁多，就本身經驗就剛入門比較需要知道的觀念說明如上，再參考文後所附一些檔案，大致可以先就簡略程式試寫看看囉。

程式之應用，可能需要溝通請使用者輸入一些資訊，可以在執行過程直接請使用者去螢幕上用滑鼠點一個點位，或直接鍵入幾個數字或字串，也可以請使用者輸入所要資訊檔案之路徑檔名，讓程式自己去讀取執行，尤其是後者讓程式讀檔是一個非常有用的模式，例如將懸臂梁各斷面樁號尺寸座標方位等資訊列在一個檔中，讓程式去讀取，幾分鐘內就能繪出一個 3d 圖，不是很酷嗎。

三、AutoLISP 簡單學習案例分享 [會放入附件中]

有了對 AutoLISP 之初步概念，以及須先必備對 ACAD 基本操作的能力，就可以試編幾個簡單的程式，提昇興趣才有動力繼續鑽研。先就幾個簡單例子引導讀者試寫，只要看一下就可很容易了解，並可自己再去延伸編製。

;此程式用來練習設定點位再利用來畫線並做螢幕縮放動作，請關掉抓點模式

```
(defun c:setpt() ;記住執行時要鍵入 setpt
```

```
(setq pt1 '(0 0))
```

;座標為(x y) 如果是單純數字 (0,0) 可以用 '(0 0)來寫

```
(setq pt2 '(1 0))
```

```
(setq pt3 '(1 1))
```

```
(setq pt4 '(0 1))
```

; ;符號表示此行是用來當作寫說明用的 不當作程式碼使用

```
(setq a 12.1) ; 設定變數 也就是 設 a=12.1
```

```
(setq b 21.2)
```

```
(setq pt (list a b))
```

;如果要將變數設為一組集合 就要用 (list a b) 不能用 '()

; ;符號也可以用來做 程式 段落之分隔

; 以下來練習 用 LINE 指令畫線

```
(command "pline" pt1 pt2 pt3 pt4 "c")
```

;也就是在指令上打 PLINE 然後輸入 4 個點座標的意思 最後連到第一點

; 在指令上是打 c 所以程式最後寫 "c"

```
(command "zoom" "e")
```

;就是在指令上 鍵入主指令 zoom 副指令 e 讓畫面放大到最大 以此類推

```
(command "zoom" "s" "0.8x") ;再將畫面縮成現在的 0.8 倍
```

```

; 再畫一條線 不是閉合之模式
(command "line" pt4 pt "")
)

```

;本程式用來使用加減乘除 以及設定相對位置極座標或直角坐標之點位

```

(defun C:calcu()                                ;記住執行時指令為 calcu
  (setq a (+ 30 15))                            ;加法
  (setq d (- 50 12.5))                          ;減法
  (setq pt0 '(10 10))
  (setq ang (* (/ a 180.0) pi))
;利用乘除法轉換 角度為徑度 先計算 a / 180 再去乘以 PI 提醒若是
;除法要小心如果不用 180.0 將計算出整數四捨五入可能會變 0
  (setq pt1 (polar pt0 ang d))
;極座標求相對座標 公式為
; pt1= (polar 第一點 pt 角度[限用徑度量] 距離)
;*****
  (setq xx (* 12 1.5))                          ;即計算出 x1=12*1.5=18.0
  (setq yy (/ 100 2.5))                          ;即計算出 y1=100 / 2.5 = 40.0
  (setq x1 (car pt1)) ;由 P1(x1 x2) 取出 x1 第一個元素用 car 取出
  (setq y1 (cadr pt1))
;由 P1(x1 y1) 取出 y1 用 cdr 取出第一個元素後的所有元素即 (y1)
;再用 car 將 y1 取出 @@@@
  (setq pt2 (list (+ x1 xx) (+ y1 yy)))          ;直角相對座標之設定
;*****
  (command "line" pt0 pt1 Pt2 "")                ;用 LINE 指令畫線
  (command "zoom" "e")                          ;用 zoom e 指令將圖形放至最大
  (command "zoom" "s" "0.7x")                   ;用 zoom s 指令再將圖形縮為某比例
  (command "change" "l" "" "p" "c" 3 "")
;用 change 來改變 最後所繪圖元的顏色 自己比對一下手動在指令欄
; 作業時 按幾次
)

```

;1050109 利用幾個常用指令繪製基礎長方形基礎及其下面之 pc 3d 圖

;實務上將利用讀檔功能將數百個基礎資訊 (長寬高 xy 座標 方位 及基礎高
; 程) 一氣呵成 全部擺到定位去

```

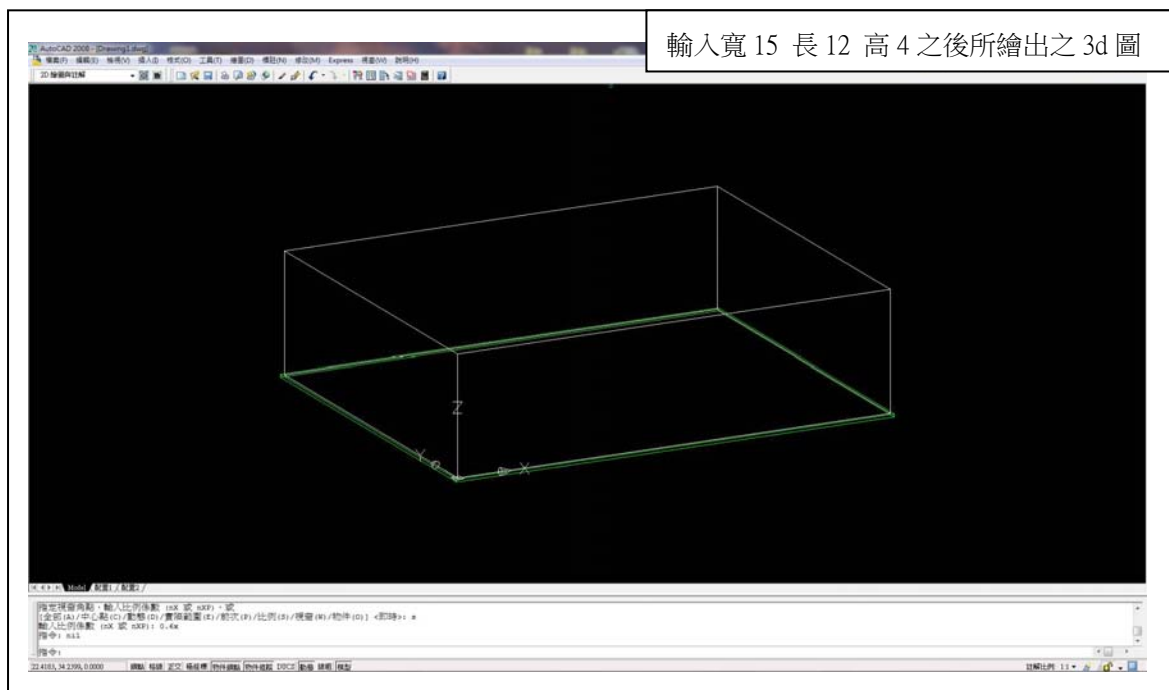
(defun C:FTGpc()                                ;記住執行時要鍵入 ftgpc

```

```

;讓使用者輸入要繪圖之資訊 其中 getreal 是請取得輸入實數之指令
(setq W (getreal "\n 請輸入基礎寬度" ))
(setq L (getreal "\n 請輸入基礎里程向長度"))
(setq H (getreal "\n 請輸入基礎高度"))
;繪製基礎及 pc pc 固定加寬 0.1 厚度 0.1
(command "_rectangle" '(0 0) (list W L)) ;繪製基礎矩形
(command "plan" "w") ;為抓點方便先設定上視圖並同時將圖放至最大
(command "zoom" "s" "0.5x") ;調整螢幕之大小以利正確抓點
(command "offset" 0.1 '(0 0) '(10000 0) "")
;利用基礎斷面用偏移指令畫出 PC 斷面
(command "_extrude" "1" "" -0.1)
;利用最後所繪圖元 (即 pc 的框) 先擠出 PC 實體
(command "change" "1" "" "p" "c" "g" "")
;將 pc 改為另一顏色較好區別 g 為 green 綠色
(command "_extrude" '(0 0) "" H)
;利用 '(0 0)點抓基礎框來擠出實體
(command "_vpoint" '(-1 -2 0.7)) ;設定視角觀看成果
(command "zoom" "s" "0.6x")
)

```



看過以上三個簡單程式寫法，應該已經有一點寫 AutoLISP 觀念了。至於執行方式，是先載入 LISP 程式，然後在指令行鍵入該程式執行指令就可以看到結果在螢幕中顯示，去操作看看吧。

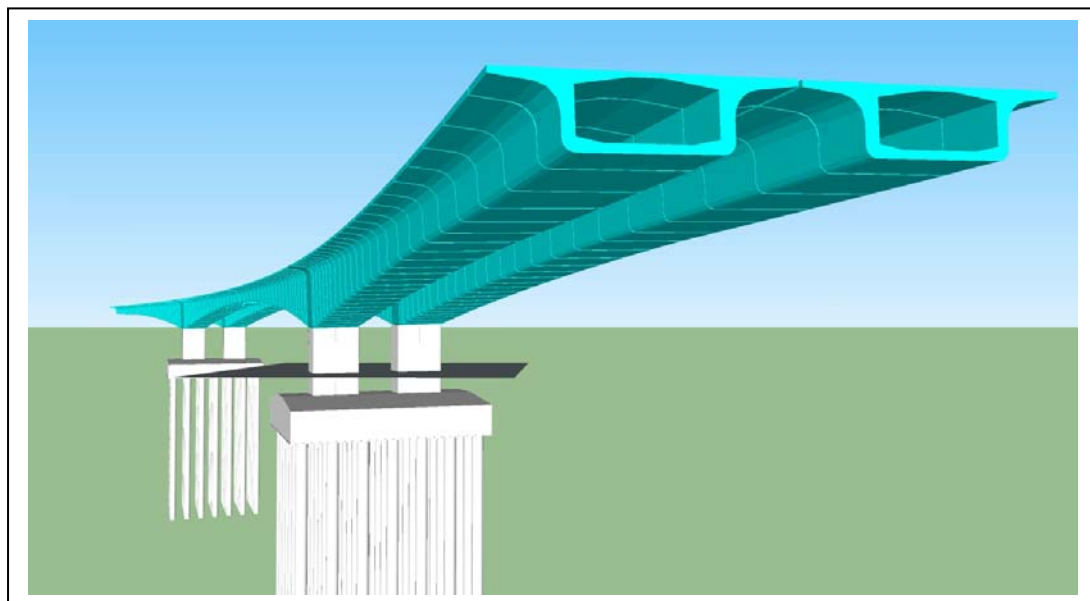
四、結語

探頭進入 AutoLISP 領域後，有興趣的朋友就可以一頭栽進來，拿著類似 0 與 1 的簡單利器，發展出無限空間的夢想，也許不見得全然是在工作上，也可以當作娛樂或用來跟小孩課業學習上互動的工具，比如跟小孩討論多項式或三角函數的圖形時，都可以增加許多樂趣。

附件中也附上 cos.lsp sin.lsp sinexp2.lsp sin2cos3.lsp 等幾個三角函數圖形小程序供參考，設定繪圖範圍在 $0 \leq x \leq 4\pi$ ，由於程式每人編製習慣可能各異，僅供參考。下面是這些小程序的成果圖。



下面這個圖，則是以前依據台北港懸臂橋資訊利用 AutoLSIP 繪出的 3d 圖（不包括墩柱部分），再轉為 sketchup 圖檔來表現的。



其用 LISP 繪製的截錄短片錄影檔亦放在附錄中有興趣者可參考一下。

附錄檔案：

1. 1050106 教學 1 setpt.lsp 執行指令 setpt
2. 1050107 教學 2 calcu.lsp 執行指令 calcu
3. 1050109 教學 3 ftgpc.lsp 執行指令 ftgpc
4. 1050109 起 常用指令.pdf
5. sin.lsp 執行指令 drawsine
6. cos.lsp 執行指令 drawcos
7. sinexp2.lsp 執行指令 drawsine²
8. sin2cos3.lsp 執行指令 sin2cos3
9. TP02 懸臂梁.avi