

1. 代數計算： Sympy (如同 mathematica, Maple)
2. 數值計算： matplotlib (如同 matlab)
3. 統計： RPy (用 Python 來執行 R)
4. 音樂作曲： Music21 (MIT 發展的音樂創作及分析工具)
5. 影像處理： PIL, Gimp
6. 立體建模： SuPy (用 Python 執行 SketchUp)
7. 互動幾何模擬： Vpython
8. 立體動畫： Blender
9. 電腦遊戲製作： PyGame
10. 手機 App 製作： Kivy

更重要的是 Python 語法簡潔易懂，接近數學思考方式。

下面介紹基本語法。

一、邏輯條件

注意---冒號：之下同區每列字前須空相同格數

---

### *if* Statements

```
x= input('x=') #執行時由使用者輸入
if x < 0:
    print 'x 小於零'
elif x <60:
    print 'x 不及格'
elif x <90:
    print 'x 及格'
else:
    print 'x 成績很好'
```

### *for* Statements

```
For n in [2,5,7,9]:
    print n
```

```
for n in range(10):
    print n
```

請練習結合上述 **if** 和 **for** 指令敘述。

### *While* Statements

```
k=1
while k<5:
    print 'Now k='+str(k)
    k=k+1
```

請練習比較 **if** 和 **while** 的不同。

## 二、須注意的指令

指令	功能	舉例
%	除法餘數	若令 A=10%3 則 A=1
**	次方	2**3=8
/	整數相除得商數(整數)	若令 A=5/2 則 A=2
/	有小數點的數字相除得一般浮點數字	若令 A=5./2 則 A=2.5
==	條件句中要求判斷是否相等	
!=	條件句中要求判斷是否不相等	
>=	條件句中要求判斷是否不小於	
<=	條件句中要求判斷是否不大於	

## 三、List 觀念 (數學的 Sequence、有序集合)

若令 A=[3,5,7,9] 則 len(A)=4 (有四個元素)

A[0]是 3, A[1]是 5, A[2]是 7, A[3]是 9,

和數學用的足碼相同, A[k]就是  $A_k$ 。

數學談集合所成的集合, Python 的 List 也有

Example:

注意: # 後面是敘述, 不影響程式。

```
B=[[-1, 6], ['李', 3, [11]]] # 注意文字兩邊要加 '
print B[0] # 會顯示出 [-1,6]
print B[0][0] # 會顯示出 -1
print B[1][1] # 會顯示出 3
print B[1][2][0] # 會顯示出 11
print B[1][0].decode('big5') # 會顯示出 李
```

矩陣  $U = \begin{bmatrix} -3 & 4 \\ 7 & 2 \end{bmatrix}$  可寫成  $U=[[-3,4],[7,2]]$  (當然先寫行亦可, 但要一致)

如此  $U_{0,0}$  就是  $U[0][0]$ ,  $U_{1,0}$  就是  $U[1][0]$ , ...

有趣的語法:

```
C=[1, 2]
D=[3,4]
E=C+D
print E ## 會顯示 [1,2,3,4]
E=E+E
print E ## 會顯示 [1,2,3,4,1,2,3,4] , 注意等號右邊 E 是舊的, 等號左邊 E 是新的
E=2*D
print E ## 會顯示 [3,4, 3,4], 注意 E 已改變
```

也可用 `append` 加入元素(加在最後面)

```
F=[1, 2, 3]
F.append(4) #注意有點
print F    #會顯示 [1,2,3,4]
F.append([5])
print F    #會顯示 [1,2,3,4, [5]]
```

移除元素用 `pop`，但要小心。

```
F=[ 2, 3, 4, 5, 6, 7]
F.pop(0) #移除最前面元素
print F  #會顯示 [3, 4, 5, 6, 7]
F.pop(len(F)-1) #注意 len(F)-1 是最後一個的號碼
print F  #會顯示 [3, 4, 5, 6]
G=F
G.pop(1)
print F  #會顯示 [3, 5, 6] 注意：改 G 但 F 也變了。
```

#### 四、其他基礎觀念

1. 每套 Python 程式都有其特殊程式庫，需要 `import` 進來，你的程式才能執行。  
例如使用 `vpypthon` 的程式，一開始都有一行指令：

```
from visual import *
```

`visual` 是其程式庫名稱，\*表示所有程式皆準備使用。

如果你知道那個程式可直接用其名稱取代 \*

例如你整個程式只需要球(`sphere`)及立方體(`box`)，則第一行可改成

```
from visual import sphere, box
```

但如此其他物件都無法使用，好處是電腦不用 `load` 所有程式備用，較省資源。

2. Python 的物件都會有屬性。

例如 `vpypthon` 中的球(`sphere`)，就具備半徑(`radius`)、球心(`pos`)、顏色(`color`)等屬性。  
用 `dir` 指令可看到所有屬性。

例如：

```
from visual import *
A=sphere()
print dir(A)    #會顯示一堆性質，屬性
```

得知屬性後可依其特性直接定義或更改。

例如：

```
From visual import sphere
A=sphere(pos=(0,0,0),radius=5) # 建立一個半徑 5 的球
A.radius=100                  # 半徑改成 100
```

## Vpython 實例

```
from visual import *
a=sphere(color=(1,0,0))
for i in range(30):
    rate(2)
    a.color=(1-i/29., i/29.,0)
```

```
from visual import *
scene.autoscale = 0
a=sphere(color=(1,1,0))
for i in range(30):
    rate(2)
    a.radius=1+i
```

```
from visual import *
import wx
scene.range = 20
a=[]
for i in range(8):
    rate(2)
    a.append(sphere(pos=(2*i,0)))
wx.Exit()
```

```
from visual import *
import wx
scene.range = 20
a=[];ac=[]
for i in range(4):
    rate(2)
    x=10*cos(i*pi/2); y=10*sin(i*pi/2)
    a.append(sphere(pos=(x,y)))
    ac.append(curve(pos=[(x,y)]))
for j in range(500):
    rate(30)
    for i in range(4):
        u=a[(i+1)%4].pos-a[i].pos
        v=0.01*u
        a[i].pos=a[i].pos+v
        ac[i].append(pos=a[i].pos)
wx.Exit()
```

### Project-1

- 探討 2 維 3 維空間中多個點或追逐或互斥時運行的軌跡。
- 以逆時鐘方向追逐和以順時鐘方向追逐最後收斂點是否相同？
- 初時點位置有對稱性時(例如都在同一圓上)是否最後收斂點可確定？

## Vpython 互動式實例

```
from visual import *
scene.range = 10
while True:
    rate(30)
    if scene.mouse.clicked:
        mo = scene.mouse.getclick()
        sphere(pos=mo.pos, radius=0.2, color=(1,1,0))
```

```
from visual import *
scene.range = 8
scene.fov=0.0001
A = []
for i in range(2):
    A.append([])
    for j in range(2):
        x=3*i;y=3*j
        A[i].append(box(pos=(x,y),i=i,j=j))

while 1:
    rate(30)
    if scene.mouse.clicked:
        mo = scene.mouse.getclick()
        mo.pick.color=(1,1,0)
        print [mo.pick.i, mo.pick.j]
```

```
from visual import *
scene.range = 5
def f(x):
    return (x-2)**2-5
p=[] #建立點座標的 List
for i in range(20):
    x=10*i/20.-5
    y=f(x)
    p.append((x,y))
c=curve(pos=p) #建立曲線
while True:
    rate(30)
    if scene.mouse.clicked:
        mo = scene.mouse.getclick()
        scene.center=(mo.pos[0],mo.pos[1],0)
        scene.range=0.5*scene.range
```

```

from visual import *
scene.range = 8

A = []
for i in range(5):
    A.append(arrow(pos=(0,0),axis=(0.01,0),visible=0))
A0=arrow(pos=(0,0),axis=(1,0),visible=1)
j=0
while True:
    rate(30)
    A0.axis=scene.mouse.pos
    if scene.mouse.clicked:
        mo = scene.mouse.getclick()
        A[j].axis=vector(A0.axis[0],A0.axis[1])
        A[j].pos=(0,0)
        A[j].visible=1
        j=(j+1)%len(A)
    for i in range(len(A)):
        if mag2(A[i].pos)>100:
            A[i].pos=(0,0)
            A[i].axis=vector(0.01,0)
            A[i].visible=0
        else:
            A[i].pos=A[i].pos+0.1*A[i].axis

```

上述發射箭頭的程式可改成鍵盤控制，請參考 vpython 的 help 檔案。  
組合以上程式可思考下列專題。

## Project-2

Conway's game of life

## Project-3

探討函數極限定義 ( $\epsilon$ ,  $\delta$  Argument)

例如使用者按一個點，此點便會動態逼進函數曲線上某個點，同時使用者可放大畫面同步觀察是否有極限？

注意：piecewise defined function 可用條件句方式定義之。

## Project-4

給定平面上一曲線，從曲線上朝切向量方向發射飛箭或射線，整個平面是否皆被覆蓋，有沒有那裡射不到？如果稱此區域為安全區，那我們隨便畫一塊區域  $E$  是否存在一條曲線  $\gamma$  其安全區就是  $E$ 。設計遊戲探討此類性質，並從網路尋找相關性質 (關鍵字： family of curves, family of tangents, envelope)

## Project-5

互動式積分觀念展示，解釋放大縮小時整個區域面積比是線量比的平方，然後證明畢氏定理及相關性質(例如：兩股與斜邊上 3 個相似形的面積關係)

## PIL (python image library) 實例

### 1. 顏色變化

```
from PIL import Image
a=Image.open('boat.jpg') #找張 jpg 照片放在此程式檔之檔案夾內
m=a.size[0]
n= a.size[1]
new=Image.new('RGB', (m,n), (0, 0, 0)) #建立一張同大小全黑的暫時圖檔
for i in range(m):
    for j in range(n):
        r,g,b=a.getpixel((i,j))          #注意 r g b 為 0 到 255 之間的整數
        R=g;G=b; B=r
        new.putpixel((i,j),(R,G,B))
new.save('t_1.jpg')          #此檔案會存在此程式檔之檔案夾內
print 'ok'
```

### 2. 位置變化

```
a=Image.open('boat.jpg')
m=a.size[0]
n=a.size[1]
new=Image.new('RGB', (m,n), (0, 0, 0))
k=0.5
for i in range(m):
    for j in range(n):
        x=i/(m-1.) # 轉換點 pixel 位置為 x 座標在 0, 1 之間的點
        y=j/(n-1.) # 轉換點 pixel 位置為 y 座標在 0, 1 之間的點
        xx=x**0.5
        yy=y**0.5
        ii=int(xx*(m-1))%m #把 xx 座標換回 pixel
        jj=int(yy*(n-1))%n #把 yy 座標換回 pixel
        r,g,b=a.getpixel((ii,jj)) # 取顏色
        new.putpixel((i,j),(r,g,b)) #把顏色畫到新圖上
new.save('t_1.jpg')
print 'ok'
```

## Project-6

化圓為方，化方為圓。探討 conformal mapping (保角映射)

## Project-7

運用複變函數轉換圖片

(例如： $f(z) = z^2$  就是把 $(x, y)$ 對應到 $(x^2 - y^2, 2xy)$  )。

Möbius transformation 也可試試