

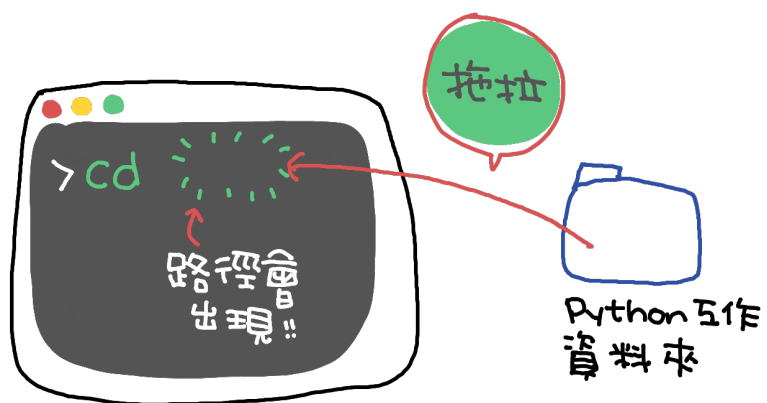
Python (Jupyter Notebook) 快速入門 1/3

政治大學應用數學系 蔡炎龍

開始

在終端機 (Anaconda Prompt) 下用

```
cd 工作資料夾路徑
```

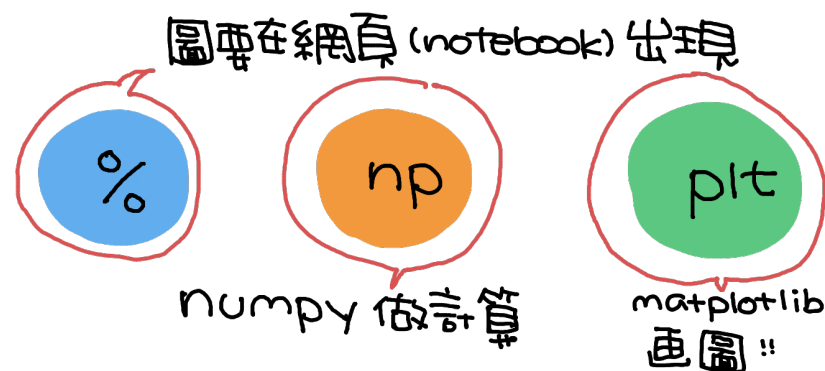


然後打入

```
jupyter notebook
```

數據分析的標準魔術

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
```



變數

Python 使用變數不需事先宣告, 直接設就可以。字串可以用單引號或雙引號。

```
a = 94.87
st = "Python 很好玩"
```

條件判斷

等於要最小心。

```
a = 3
a > 5 # False
a <= 7 # True
a == 3 # True
a != 3 # False
```

輸入和字串的相加

```
name = input("請問您的大名: ")
```

```
print("你好, " + name + "!")
```

寫個 while 迴圈

注意 Python 深愛的縮排。

```
c = 1
while c <= 10:
    print("*" * c)
    c = c + 1
```

快速生成 list

事實上在 Python 3 中要用 list() 才能真的變成 list。

```
range(5) # 0, 1, 2, 3, 4
range(1,5) # 1, 2, 3
range(1,10,2) # 1, 3, 5, 7, 9
```

for 迴圈是把 list 中的元素一一拿出來

```
for k in range(1, 10):
    print("*" * k)
```

字串 format 法

字串用「加」的有時有點麻煩, 我們可以有個簡單的 format 法。

```
name = "炎龍"
age = 18
"{}現年{}歲。".format(name, age) # '炎龍現年18歲。'
```

在 Python 3.6 有個更方便的「f-字串」法。

```
f"{name}現年{age}歲。"
```

。

定義函數

```
def sq(x):
    return x**2
```

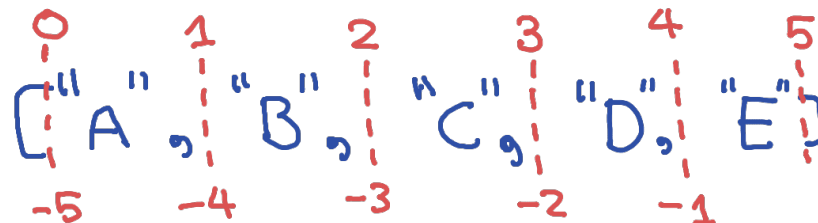
執行範例:

```
>>> sq(2)
4
```

list 切割刀

這裡學學我們有個 list 時, 怎麼取出其中的一部份。

```
L = ["A", "B", "C", "D", "E"]
```



Python 的索引編號是如圖這樣在間隔與間隔之間, 會取出編號線「右邊」的元素。

```
L[2] # 'C'
L[-2] # 'D'
```

某個範圍。

```
L[1:4] # ['B', 'C', 'D']
```

要從最前或到最後不需標出。

```
L[2:] # ['C', 'D', 'E']
L[:3] # ['A', 'B', 'C']
```

字串和串列切割法基本上是一樣的。

切換資料形態

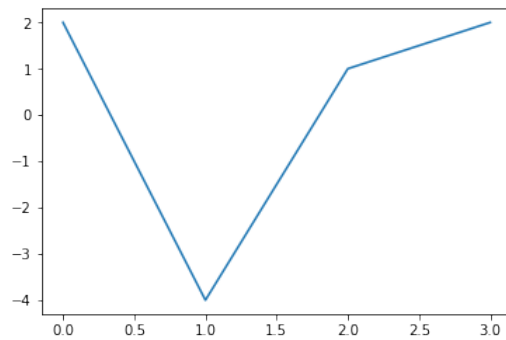
```
s = '3'  
a = int(s) # 3  
b = float(s) # 3.0  
c = 87  
cs = str(c) # '87'
```

if 條件判斷式

```
g = float(input("請輸入你的成績: "))  
  
if g >= 60:  
    print("過了!")  
elif g > 55:  
    print("可惜, 差一點點...")  
else:  
    print("拍拍")
```

用 matplotlib 畫個圖

```
plt.plot([2,-4,1,2])
```



Python 事實上是畫出 (0,2), (1,-4), (2,1), (3,2) 四個點, 再連起來。下面的畫法是等價的。

```
plt.plot([0, 1, 2, 3], [2, -4, 1, 2])
```

一般的畫法是

```
plt.plot(X, Y)
```

其中

* X 為點的 X 座標所成的 list (或 array)

* Y 為點的 Y 座標所成的 list (或 array)

numpy 的向量導向型運算

把一個 list 變成 numpy 的 array 很容易。

```
p = np.array([199, 299, 399])
```

假設上面是三件物品的美金價格, 我們要一次換算成台幣可以嗎? (假設匯率是 30.05)。

```
p*30.05 # array([ 5979.95,  8984.95, 11989.95])
```

也可以一次做更複雜的運算。

```
np.log(p) + p**2 # array([ 39606.29330482,  89406.70044357, 159206.98896142])
```

畫函數的圖

畫函數的圖其實就是描點法。假設我們要畫

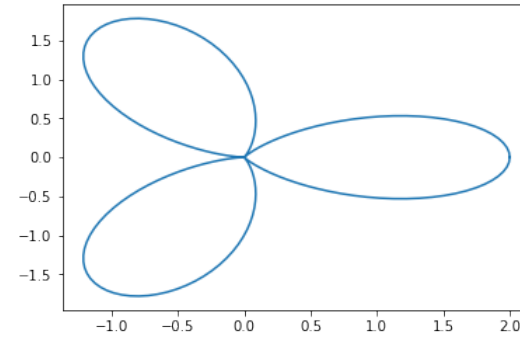
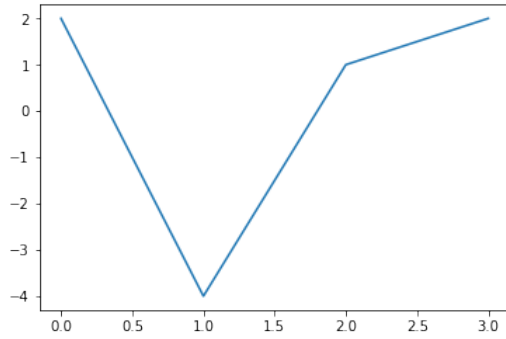
$$f(x) = \sin(x) + x$$

假設我們由 0 到 10 這個範圍, 取 100 個點。

```
x = np.linspace(0, 10, 100)
```

畫出這個函數。

```
y = np.sin(x) + x  
plt.plot(x, y)
```



參數式的畫圖

其實參數式就是把 (x,y) 一一算出, 然後連起來, 所以指令基本上是一樣的。

比如說我們要畫

$$r = 1 + \cos(3\theta)$$

我們為方便用 t 取代 θ 。

```
t = np.linspace(0, 2*np.pi, 200)
```

我們的極座標方程式。

```
r = 1 + np.cos(3*t)
```

畫出來!

```
plt.plot(r*np.cos(t), r*np.sin(t))
```