Python 快速上手

Pandas、matplotlib 繪圖、字串處理與其他常用 Modules

2020/10/07

Pandas

Store data in DataFrame

```
import pandas as pd

df = pd.DataFrame({
                    'names' :  ['Bob', 'Jessica', 'Mary', 'John', 'Mel'],
                    'births' : [968, 155, 77, 578, 973, ]
                  })
```

	names	births
0	Bob	968
1	Jessica	155
2	Mary	77
3	John	578
4	Mel	973

資料載入

- pandas 可以支援多種文字、二進位檔案與資料庫的資料載入，常見的 txt、csv、excel 試算表等等都可以，支援的詳細清單可參考這裡。

載入 csv

```
import pandas as pd

csv_file = “https://storage.googleapis.com/learn_pd_like_tidyverse/gapminder.csv”
gapminder = pd.read_csv(csv_file)
print(type(gapminder))
gapminder.head()
```

載入 excel 試算表

```
xlsx_file = “https://storage.googleapis.com/learn_pd_like_tidyverse/gapminder.xlsx”
gapminder = pd.read_excel(xlsx_file)
print(type(gapminder))
gapminder.head()
```

獲得 DataFrame 資訊

- df.shape：顯示這個 DataFrame 有幾列有幾行。
- df.columns：顯示這個 DataFrame 的變數資訊。
- df.index：顯示這個 DataFrame 的列索引資訊。
- df.info()：顯示關於 DataFrame 的詳細資訊。
- df.describe()：顯示關於 DataFrame 個數值變數的描述統計。

資料整理

- 利用撰寫判斷條件將符合條件的值從資料框中篩選出

gapminder[gapminder[‘country’] == “Taiwan”]

資料整理

- 如果有多個條件，可以使用 | 或是 & 符號連結，例如選出 2007 年的亞洲國家：

```
gapminder[gapminder[(gapminder['year'] == 2007) &
(gapminder['continent'] == 'Asia')]]
```

資料整理

- 用 list 標註變數名稱可以將變數從資料框中選出，例如選出 country 與 continent 變數：

gapminder[[‘country’, ‘continent’]]

資料整理

- 如果只選一個變數且沒有以 list 標註，同樣能選出變數，但是類別會變為 Series：

```
country = gapminder[‘country’]
print(type(country))
```

apply()

- 搭配 apply() 與 lambda 函數將公式應用到每一個值。例如新增一個 country_abb 變數擷取原本 country 變數的前三個英文字母：

```
gapminder[‘country_abb’] = gapminder[‘country’].apple(lambda x: x[:3])
gapminder
```

sum()、mean()

- 呼叫 DataFrame 不同的聚合函數針對欄位計算，例如計算2007年全球人口總數：

```
gapminder[gapminder[‘year’] == 2007][[‘pop’]].sum()
```

- 或者計算 2007 年全球的平均壽命、平均財富：

```
gapminder[gapminder[‘year’] == 2007][[‘lifeExp’, ‘gdpPercap’]].mean()
```

groupby()

- 計算 2007 年各洲人口總數：

```
gapminder[gapminder[‘year’] == 2007].groupby([‘continent’])[‘pop’].sum()
```

- 或者計算 2007 年各洲平均壽命、平均財富：

```
gapminder[gapminder[‘year’] == 2007].groupby([‘continent’])[[‘lifeExp’, ‘gdpPercap’]].mean()
```

Combine DataFrame

- 使用 concat() 可以很容易地將相似的 DataFrame 合併

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

+

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

=

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

matplotlib

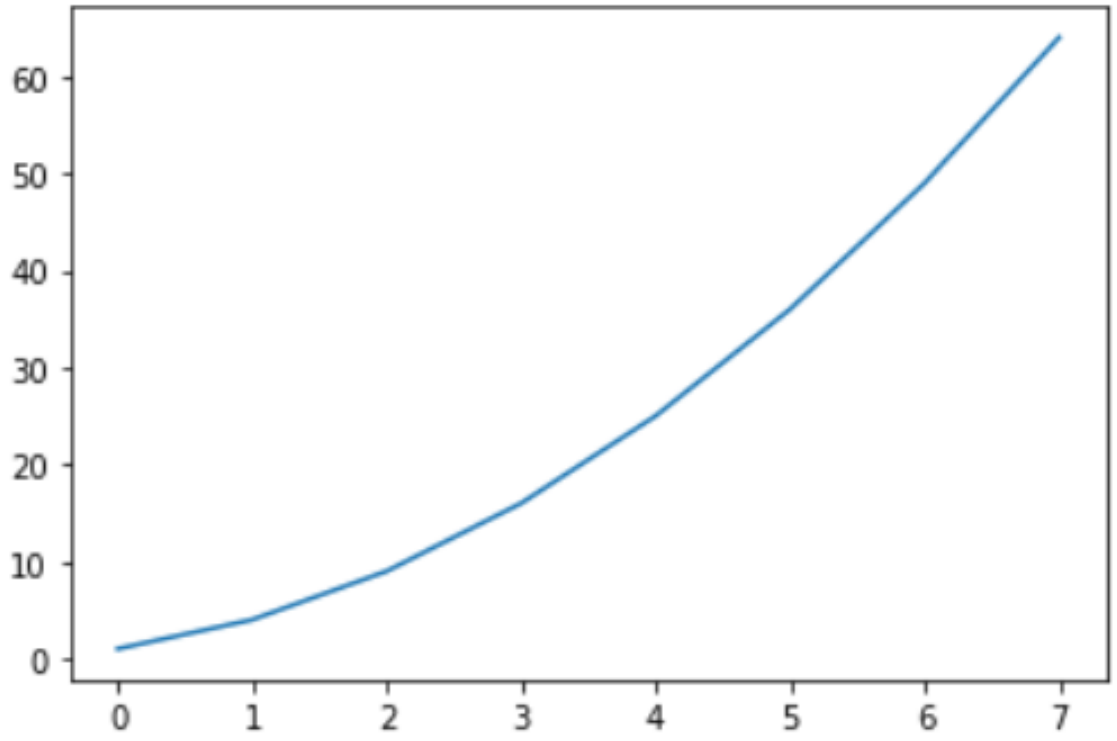
範例取自於 **Python零基礎最強入門之路：王者歸來** (洪錦魁)

安裝套件

- conda install matplotlib
- matplotlib 是一個龐大的繪圖庫模組，其中的 pyplot 子模組就可以完成許多圖表繪製
- import matplotlib.pyplot as plt
- https://matplotlib.org/

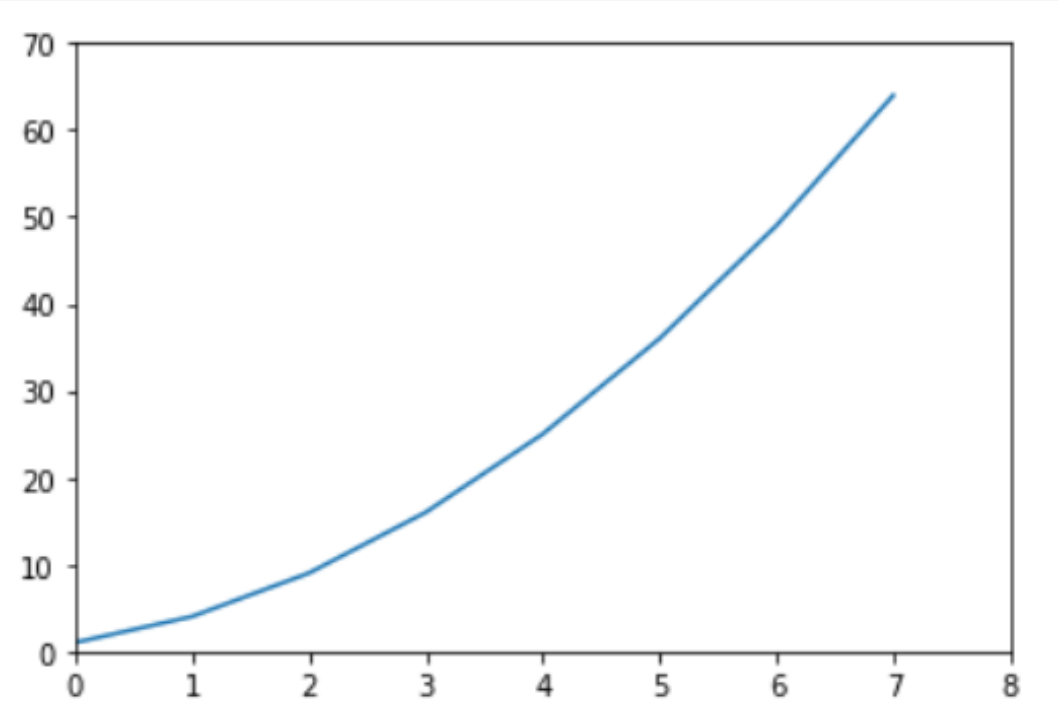
繪製簡單的折線圖

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

squares = [1, 4, 9, 16, 25, 36, 49, 64]
plt.plot(squares)
plt.show()
```



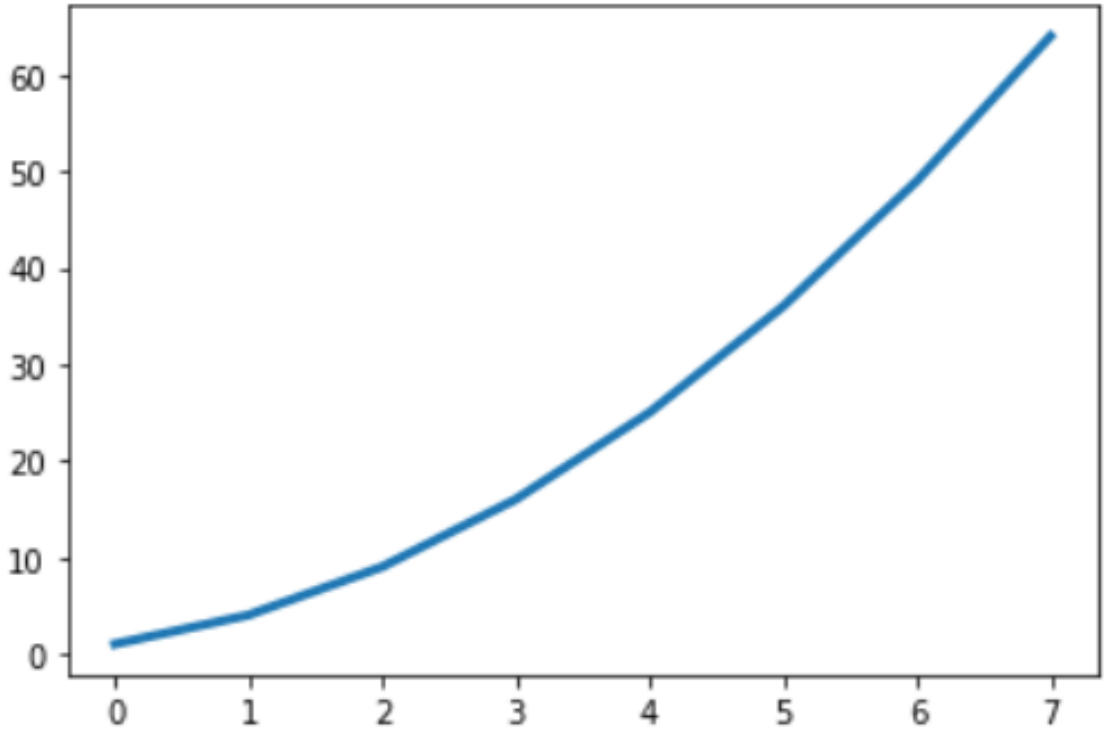
設定 x, y 軸最小和最大刻度

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

squares = [1, 4, 9, 16, 25, 36, 49, 64]
plt.plot(squares)
plt.axis([0, 8, 0, 70])
plt.show()
```



變更線條寬度 linewidth

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

squares = [1, 4, 9, 16, 25, 36, 49, 64]
plt.plot(squares, linewidth = 3)
plt.show()
```



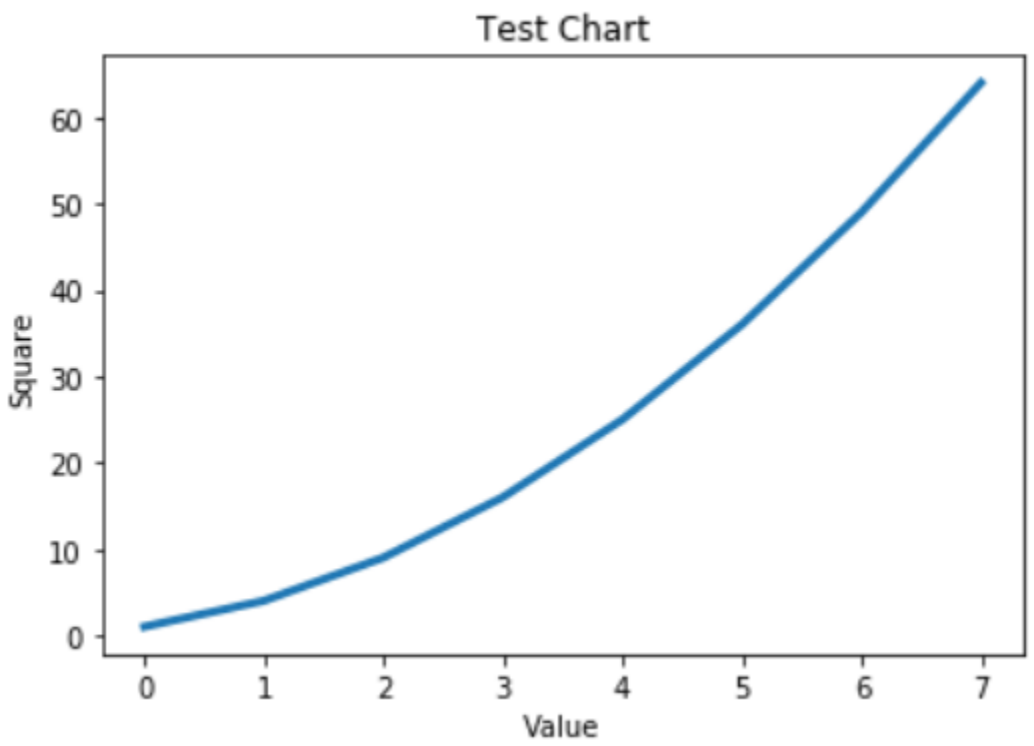
標題的顯示

- title：圖表標題
- xlabel：x 軸標題
- ylabel：y 軸標題
- 語法：title(標題名稱, fontsize=字型大小)

  # xlabel() 與 ylabel() 用法相同

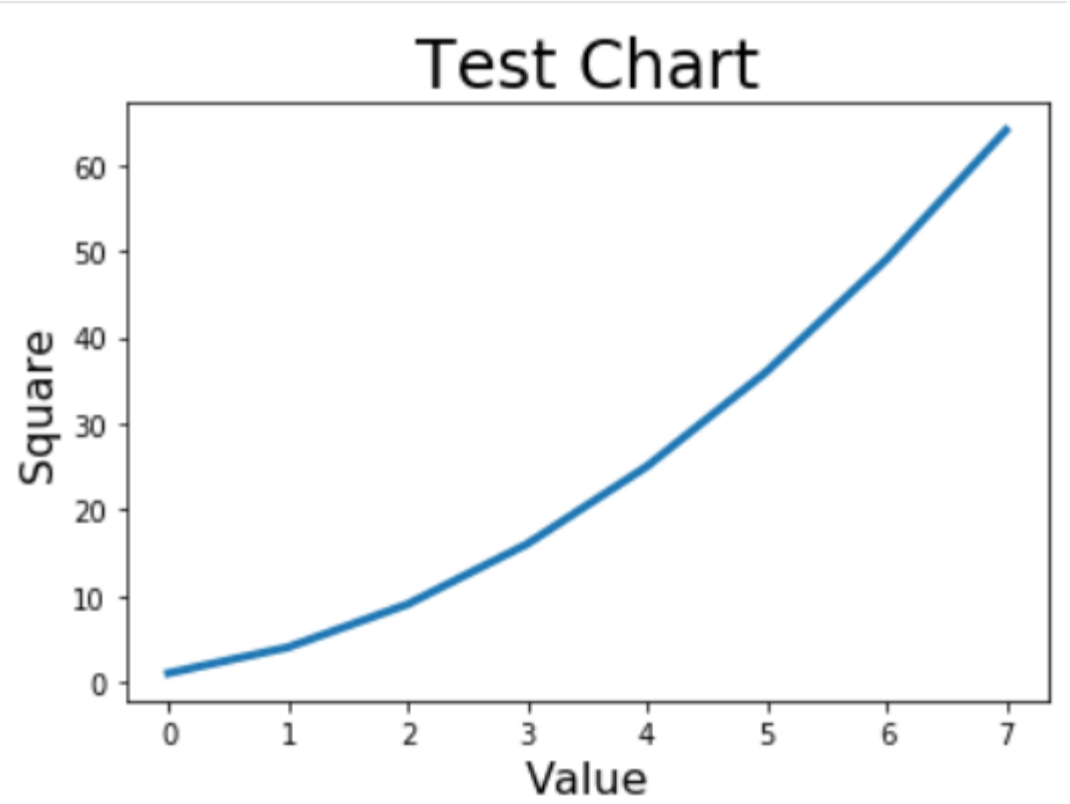
標題的顯示

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

squares = [1, 4, 9, 16, 25, 36, 49, 64]
plt.plot(squares, linewidth = 3)
plt.title("Test Chart")
plt.xlabel("Value")
plt.ylabel("Square")
plt.show()
```



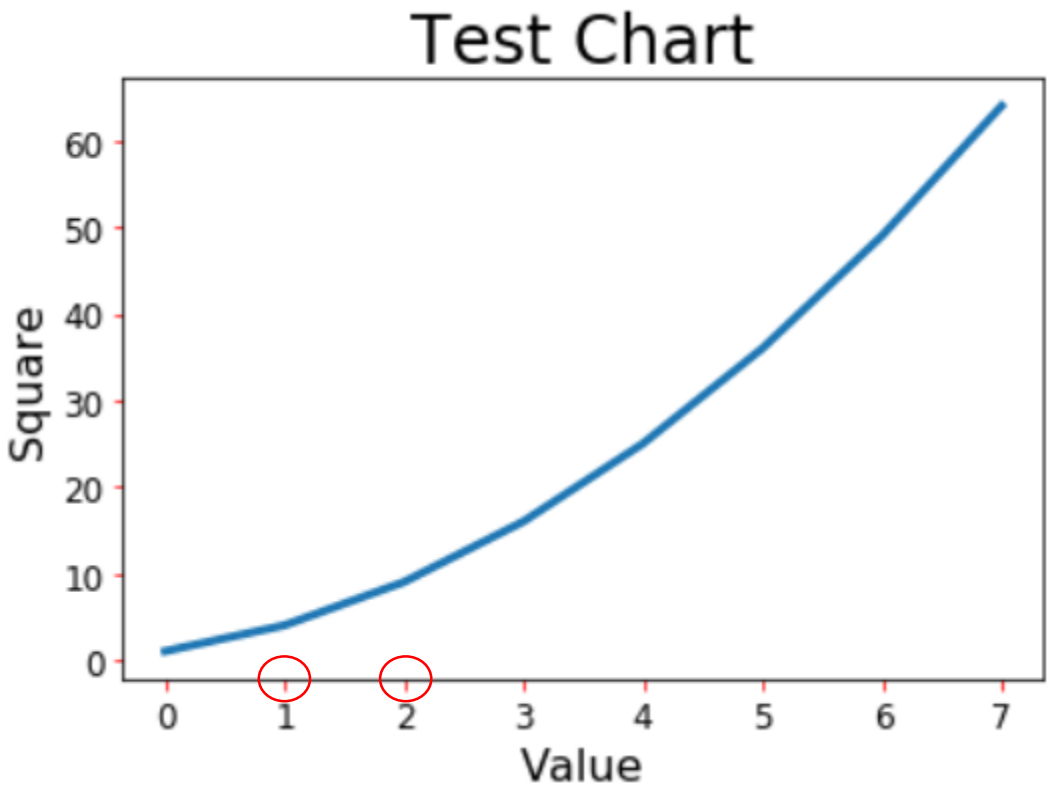
變更標題大小

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

squares = [1, 4, 9, 16, 25, 36, 49, 64]
plt.plot(squares, linewidth = 3)
plt.title("Test Chart", fontsize=24)
plt.xlabel("Value", fontsize=16)
plt.ylabel("Square", fontsize=16)
plt.show()
```



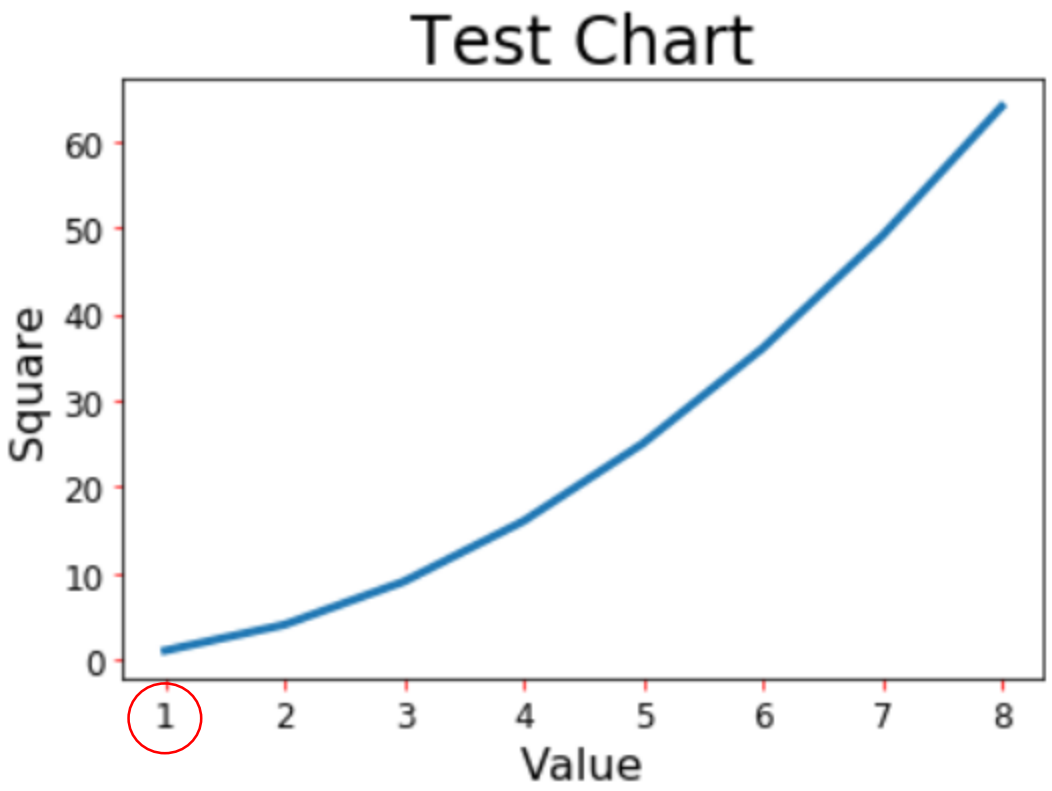
座標軸刻度的設定

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

squares = [1, 4, 9, 16, 25, 36, 49, 64]
plt.plot(squares, linewidth = 3)
plt.title("Test Chart", fontsize=24)
plt.xlabel("Value", fontsize=16)
plt.ylabel("Square", fontsize=16)
plt.tick_params(axis='both', labelsize=12, color='red')
plt.show()
```



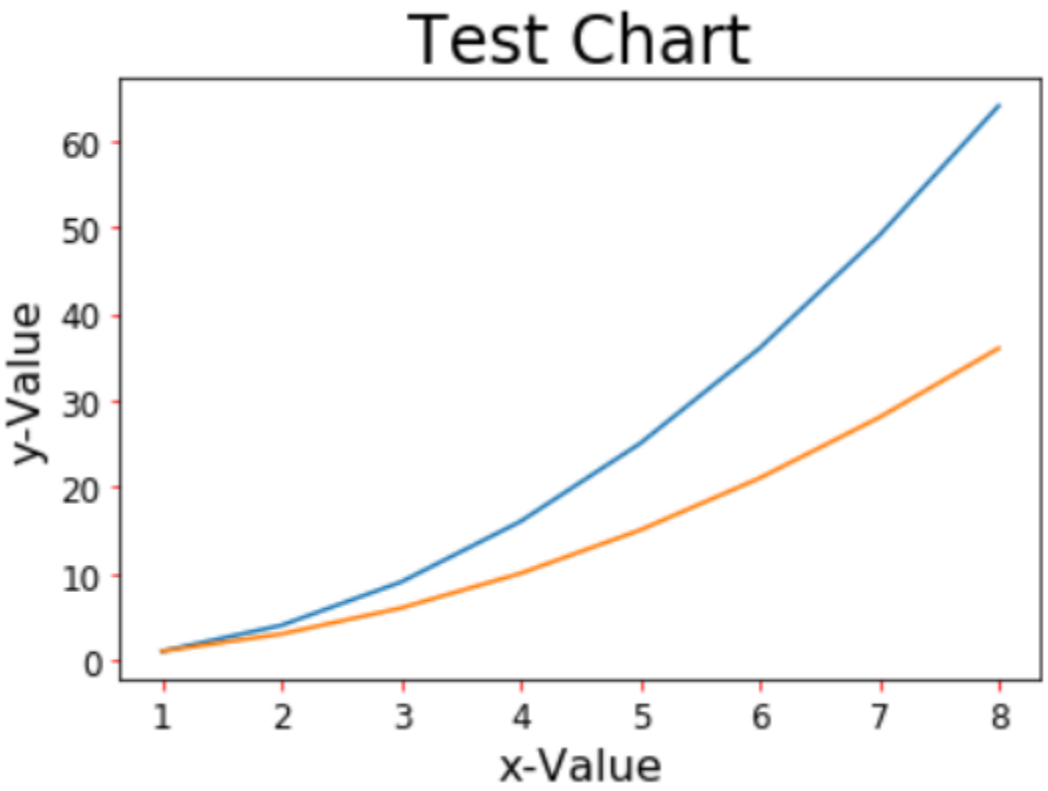
修訂圖表的起始值

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

squares = [1, 4, 9, 16, 25, 36, 49, 64]
seq = [1, 2, 3, 4, 5, 6, 7, 8]
plt.plot(seq, squares, linewidth = 3)
plt.title("Test Chart", fontsize=24)
plt.xlabel("Value", fontsize=16)
plt.ylabel("Square", fontsize=16)
plt.tick_params(axis='both', labelsize=12, color='red')
plt.show()
```



多組數據的應用

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

data1 = [1, 4, 9, 16, 25, 36, 49, 64]
data2 = [1, 3, 6, 10, 15, 21, 28, 36]
seq = [1, 2, 3, 4, 5, 6, 7, 8]
plt.plot(seq, data1, seq, data2)
plt.title("Test Chart", fontsize=24)
plt.xlabel("x-Value", fontsize=16)
plt.ylabel("y-Value", fontsize=16)
plt.tick_params(axis='both', labelsize=12, color='red')
plt.show()
```



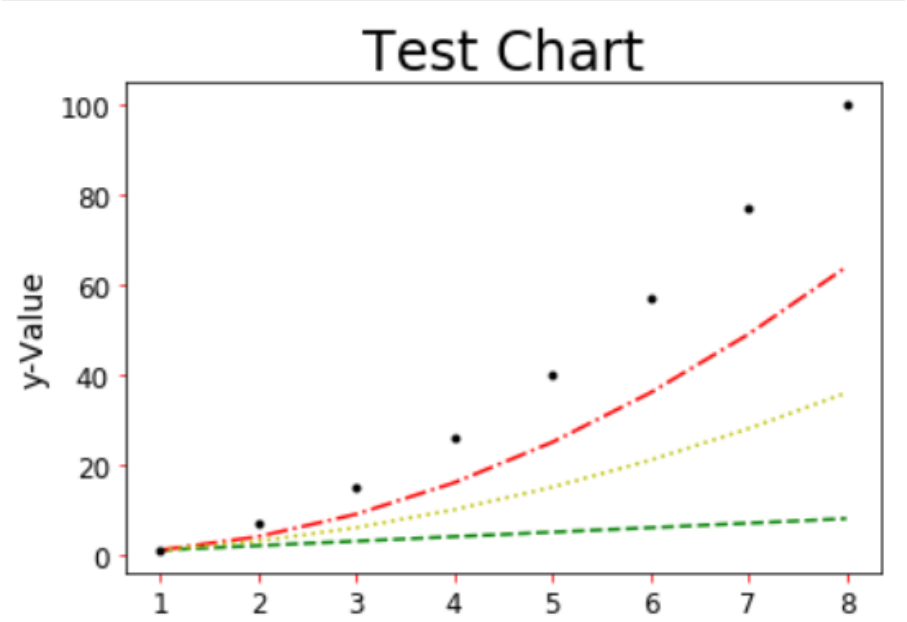
線條色彩與樣式

色彩字元	色彩說明
‘b’	blue (藍色)
‘c’	cyan (青色)
‘g’	green (綠色)
‘k’	black (黑色)
‘m’	magenta (洋紅)
‘r’	red (紅色)
‘w’	white (白色)
‘y’	yellow (黃色)

字元	說明
‘-’ 或 ‘solid’	預設實線
‘--’ 或 ‘dashed’	虛線
‘-.’ 或 ‘dashdot’	虛點線
‘:’ 或 ‘dotted’	點線
‘.’	點標記
‘,’	像素標記
‘o’	圓標記
‘v’	反三角標記
‘^’	三角標記
‘s’	方形標記
‘p’	五角標記
‘*’	星星標記
‘+’	加號標記

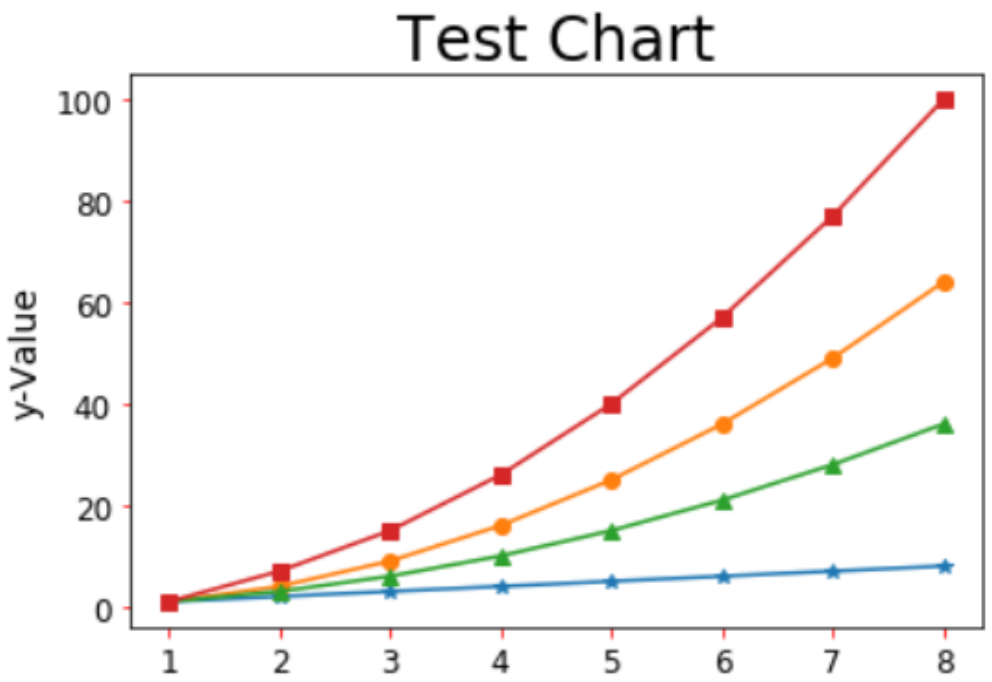
使用不同色彩與線條樣式繪製圖表

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

data1 = [1, 2, 3, 4, 5, 6, 7, 8]
data2 = [1, 4, 9, 16, 25, 36, 49, 64]
data3 = [1, 3, 6, 10, 15, 21, 28, 36]
data4 = [1, 7, 15, 26, 40, 57, 77, 100]
seq = [1, 2, 3, 4, 5, 6, 7, 8]
plt.plot(seq, data1, 'g--', seq, data2, 'r-.', seq, data3, 'y:', seq, data4, 'k.')
plt.title("Test Chart", fontsize=24)
plt.xlabel("x-Value", fontsize=14)
plt.ylabel("y-Value", fontsize=14)
plt.tick_params(axis='both', labelsize=12, color='red')
plt.show()
```



在資料點加上標記

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

data1 = [1, 2, 3, 4, 5, 6, 7, 8]
data2 = [1, 4, 9, 16, 25, 36, 49, 64]
data3 = [1, 3, 6, 10, 15, 21, 28, 36]
data4 = [1, 7, 15, 26, 40, 57, 77, 100]
seq = [1, 2, 3, 4, 5, 6, 7, 8]
plt.plot(seq, data1, '-*', seq, data2, '-o', seq, data3, '-^', seq, data4, '-s')
plt.title("Test Chart", fontsize=24)
plt.xlabel("x-Value", fontsize=14)
plt.ylabel("y-Value", fontsize=14)
plt.tick_params(axis='both', labelsize=12, color='red')
plt.show()
```



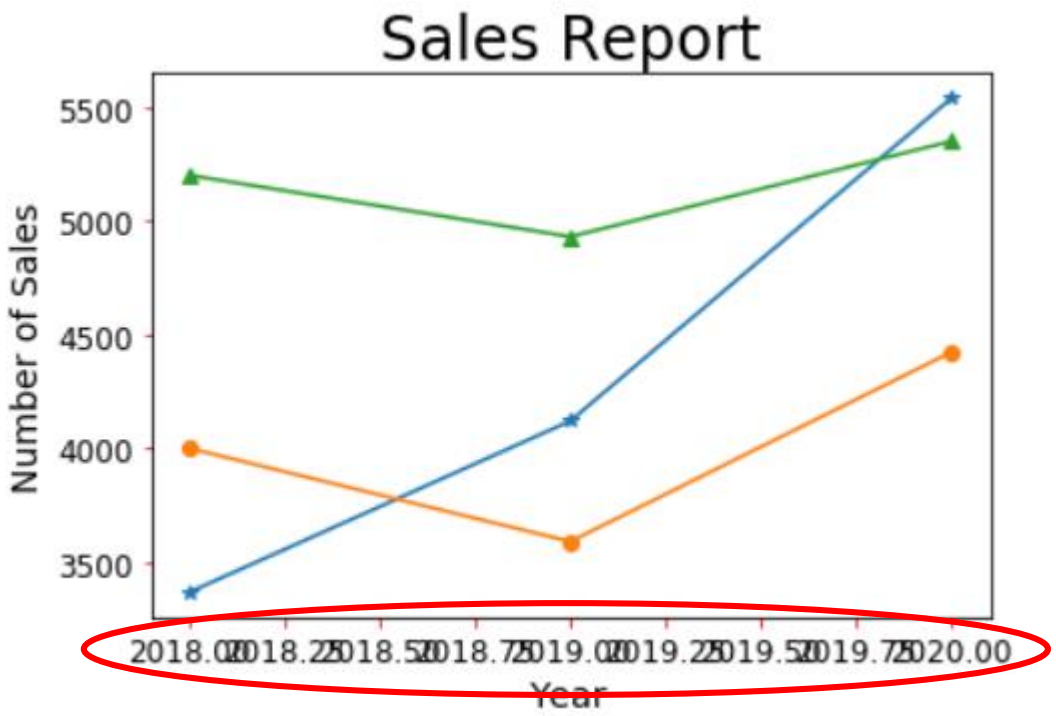
刻度設計

- 三大品牌車輛在 2018-2020 的銷售數據如下：

	2018	2019	2020
Benz	3367	4120	5539
BMW	4000	3590	4423
Lexus	5200	4930	5350

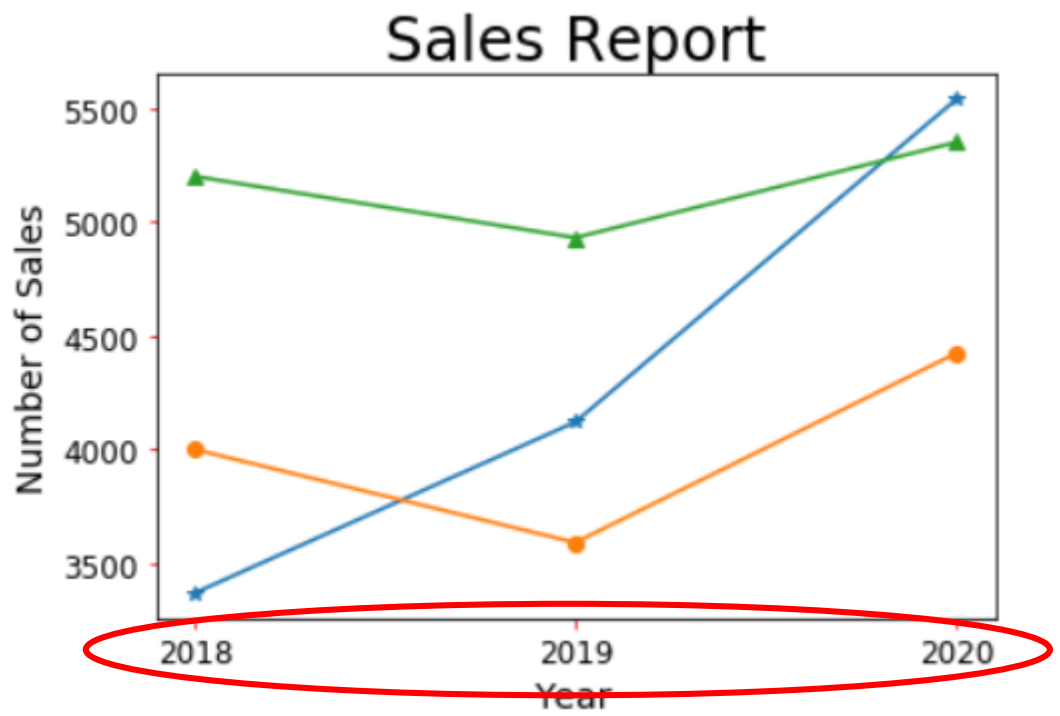
刻度設計

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

Benz = [3367, 4120, 5539]
BMW = [4000, 3590, 4423]
Lexus = [5200, 4930, 5350]
seq = [2018, 2019, 2020]

plt.plot(seq, Benz, '-*', seq, BMW, '-o', seq, Lexus, '-^')
plt.title("Sales Report", fontsize=24)
plt.xlabel("Year", fontsize=14)
plt.ylabel("Number of Sales", fontsize=14)
plt.tick_params(axis='both', labelsize=12, color='red')
plt.show()
```



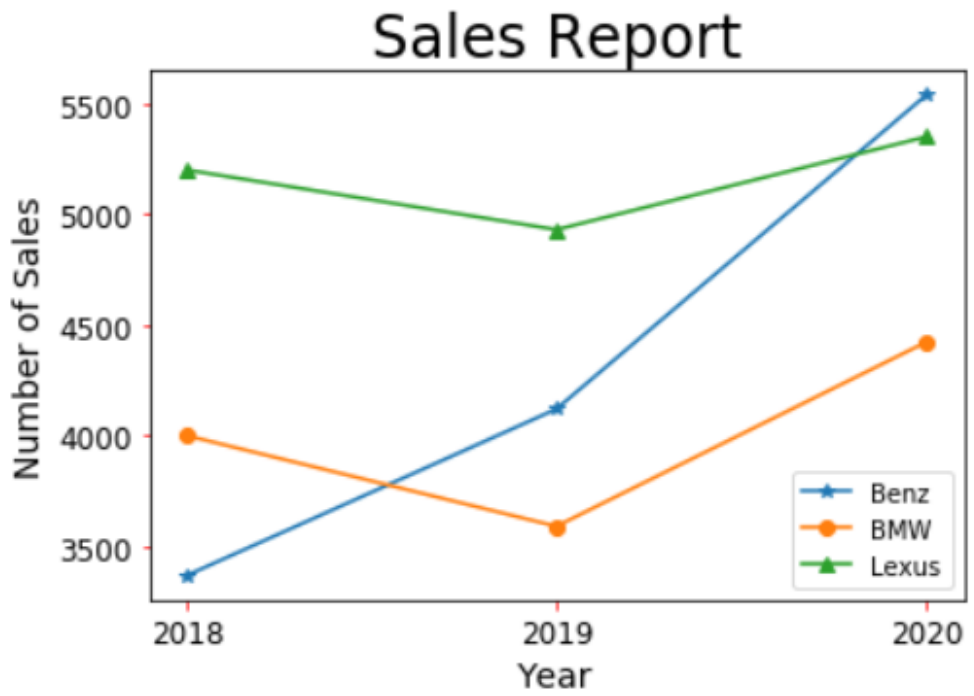
刻度設計

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

Benz = [3367, 4120, 5539]
BMW = [4000, 3590, 4423]
Lexus = [5200, 4930, 5350]
seq = [2018, 2019, 2020]
plt.xticks(seq)
plt.plot(seq, Benz, '-*', seq, BMW, '-o', seq, Lexus, '-^')
plt.title("Sales Report", fontsize=24)
plt.xlabel("Year", fontsize=14)
plt.ylabel("Number of Sales", fontsize=14)
plt.tick_params(axis='both', labelsize=12, color='red')
plt.show()
```



圖例 legend()

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

Benz = [3367, 4120, 5539]
BMW = [4000, 3590, 4423]
Lexus = [5200, 4930, 5350]
seq = [2018, 2019, 2020]
plt.xticks(seq)
lineBenz, = plt.plot(seq, Benz, '-*', label="Benz")
lineBMW, = plt.plot(seq, BMW, '-o', label="BMW")
lineLexus, = plt.plot(seq, Lexus, '-^', label="Lexus")
plt.legend(handles=[lineBenz, lineBMW, lineLexus], loc='best')
plt.title("Sales Report", fontsize=24)
plt.xlabel("Year", fontsize=14)
plt.ylabel("Number of Sales", fontsize=14)
plt.tick_params(axis='both', labelsize=12, color='red')
plt.show()
```



“best” : 0

“upper right” : 1

“upper left” : 2

“lower left” : 3

“lower right” : 4

“right” : 5 (與 “center right” 相同)

“center left” : 6

“center right” : 7

“lower center” : 8

“upper center” : 9

“center” : 10

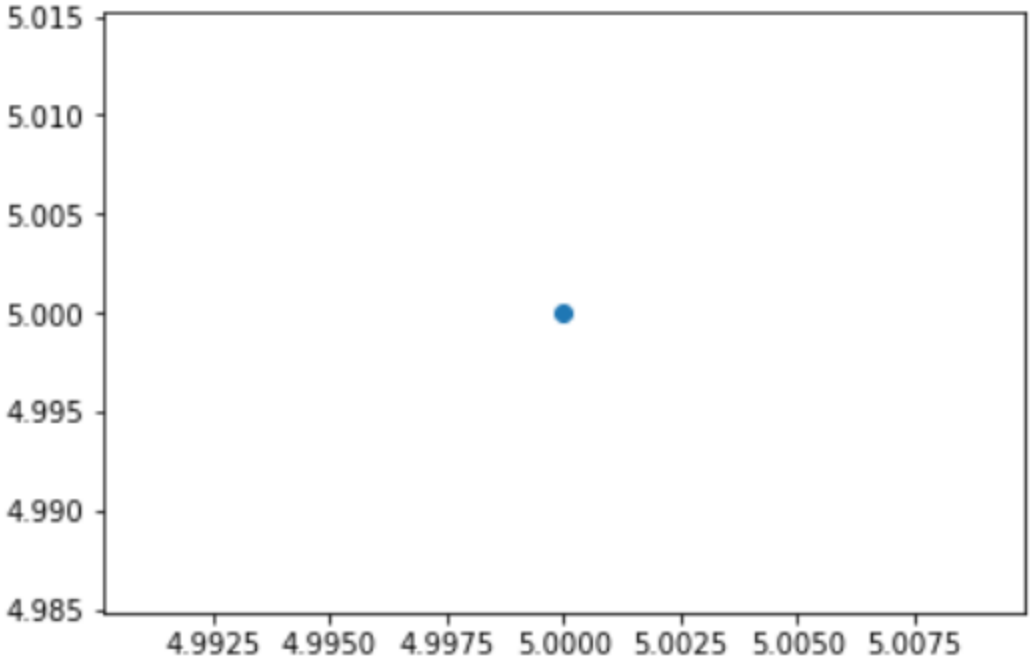
保存圖檔

- 可以使用 savefig() 保存圖檔，必須放在 show() 的前方。

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

Benz = [3367, 4120, 5539]
BMW = [4000, 3590, 4423]
Lexus = [5200, 4930, 5350]
seq = [2018, 2019, 2020]
plt.xticks(seq)
lineBenz, = plt.plot(seq, Benz, '-*', label="Benz")
lineBMW, = plt.plot(seq, BMW, '-o', label="BMW")
lineLexus, = plt.plot(seq, Lexus, '-^', label="Lexus")
plt.legend(handles=[lineBenz, lineBMW, lineLexus], loc='best')
plt.title("Sales Report", fontsize=24)
plt.xlabel("Year", fontsize=14)
plt.ylabel("Number of Sales", fontsize=14)
plt.tick_params(axis='both', labelsize=12, color='red')
plt.savefig('sales report.png')
plt.show()
```

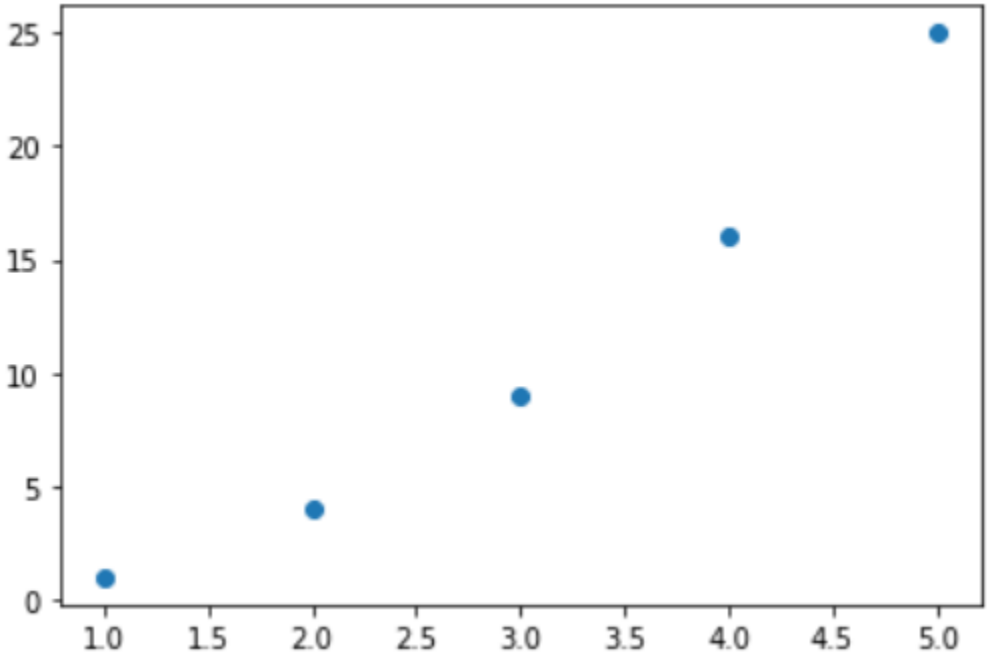
繪製散點圖 scatter()

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

plt.scatter(5, 5)
plt.show()
```



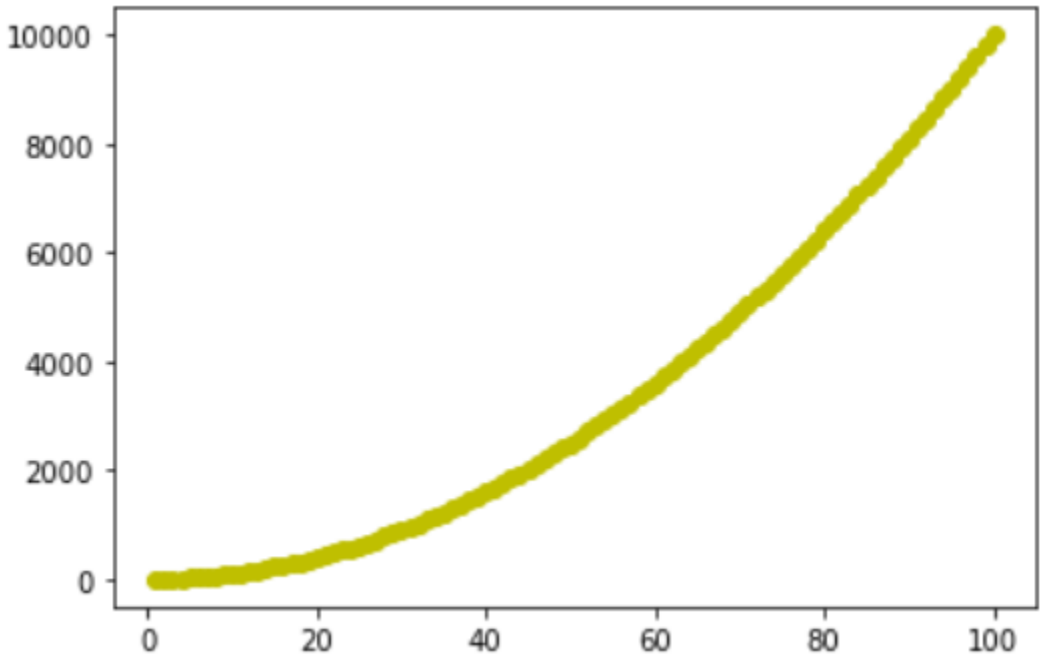
繪製系列點

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

xpt = [1, 2, 3, 4, 5]
ypt = [1, 4, 9, 16, 25]
plt.scatter(xpt, ypt)
plt.show()
```



繪製系列點

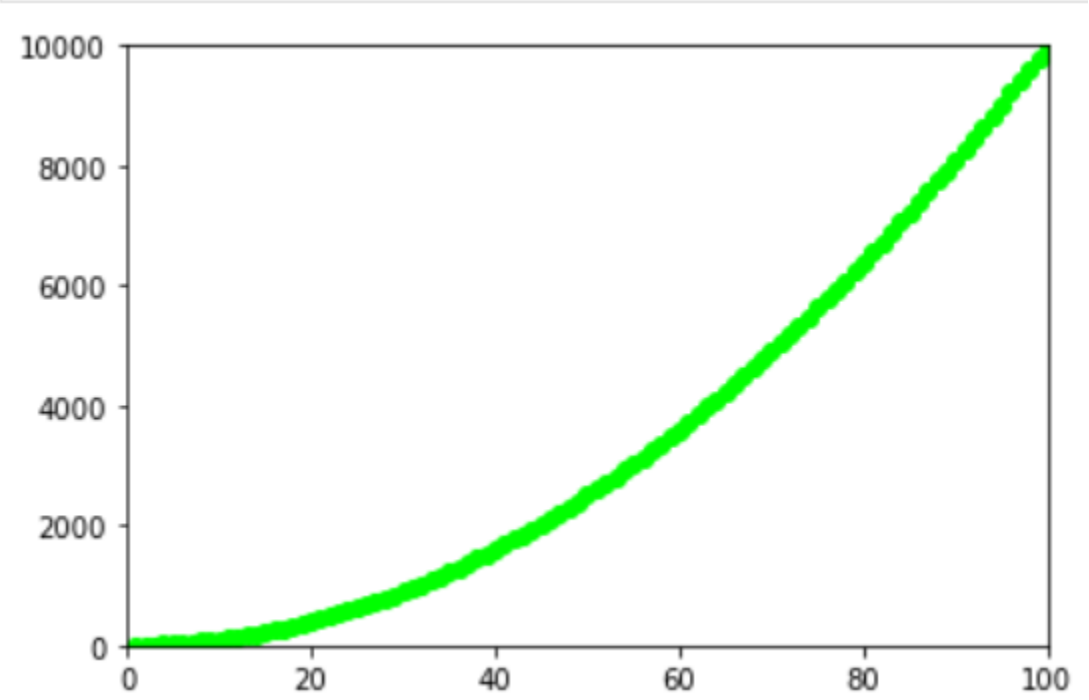
```
%matplotlib inline
import matplotlib.pyplot as plt

xpt = list(range(1,101))
ypt = [x**2 for x in xpt]
plt.scatter(xpt, ypt, color='y')
plt.show()
```



設定繪圖區間

- axis([xmin, xmax, ymin, ymax])

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

xpt = list(range(1,101))
ypt = [x**2 for x in xpt]
plt.axis([0, 100, 0, 10000])
plt.scatter(xpt, ypt, color=(0, 1, 0))
plt.show()
```



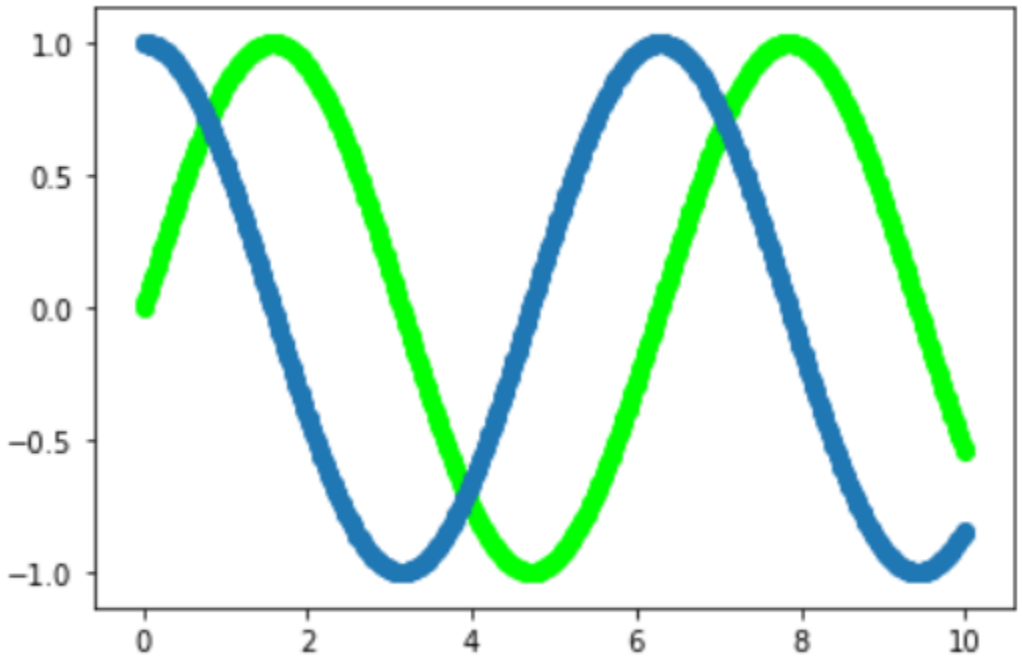
利用 numpy 產生數列

```
import numpy as np

x1 = np.linspace(0, 10, num=11)
print(type(x1), x1)
x2 = np.arange(0, 11, 1)
print(type(x2), x2)
x3 = np.arange(11)
print(type(x3), x3)
```

```
<class 'numpy.ndarray'> [ 0.  1.  2.  3.  4.  5.  6.  7.  8.  9. 10.]
<class 'numpy.ndarray'> [ 0  1  2  3  4  5  6  7  8  9 10]
<class 'numpy.ndarray'> [ 0  1  2  3  4  5  6  7  8  9 10]
```

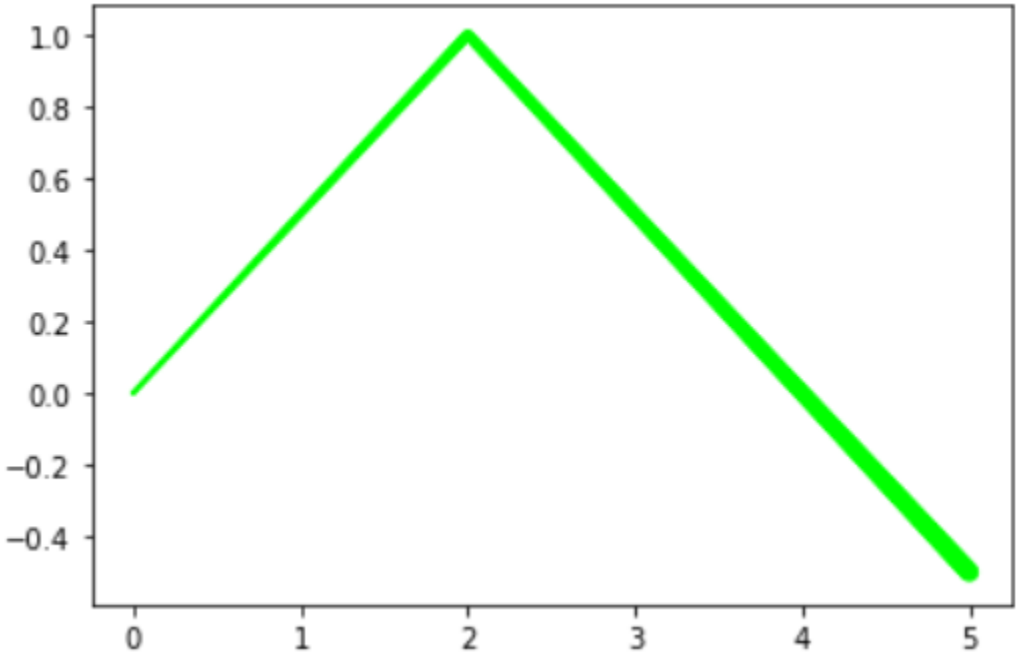
繪製波形

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

xpt = np.linspace(0, 10, 500)
ypt1 = np.sin(xpt)
ypt2 = np.cos(xpt)
plt.scatter(xpt, ypt1, color=(0, 1, 0))
plt.scatter(xpt, ypt2)
plt.show()
```



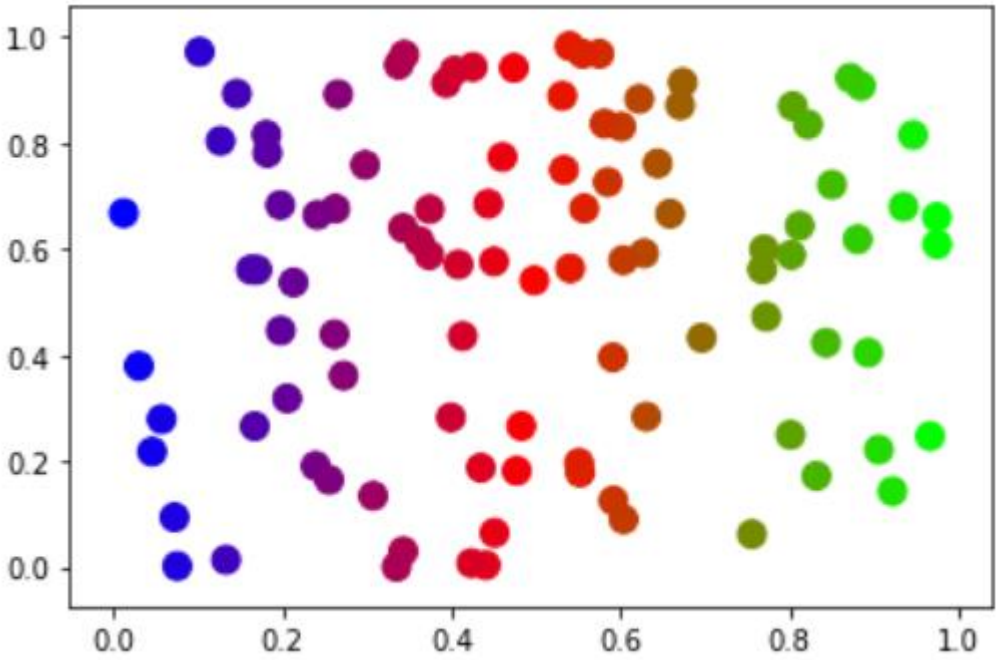
建立不等寬度的散點圖

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

xpt = np.linspace(0, 5, 500)
ypt = 1- 0.5*np.abs(xpt-2)
lwidths = (1+xpt)**2
plt.scatter(xpt, ypt, s=lwidths, color=(0, 1, 0))
plt.show()
```



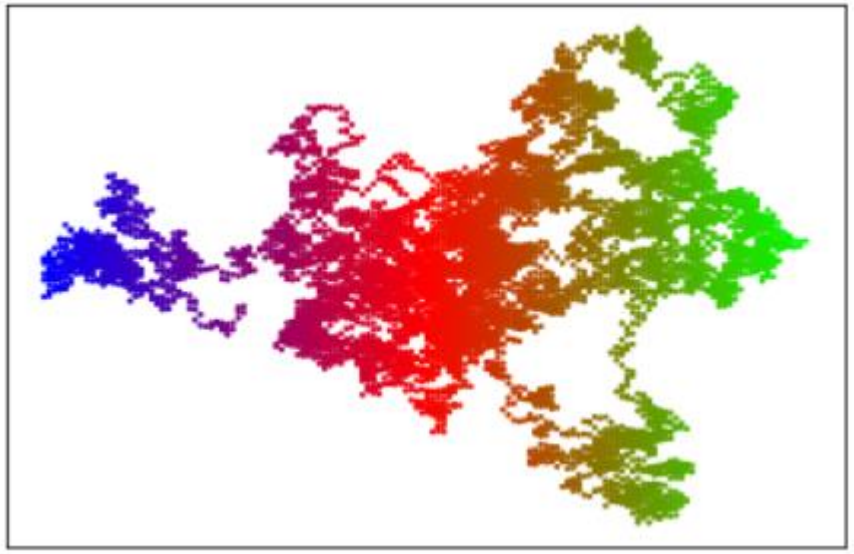
隨機數的應用(補充)

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

num = 100
while True:
    x = np.random.random(100)
    y = np.random.random(100)
    t = x
    plt.scatter(x, y, s=100, c=t, cmap='brg')
    plt.show()
    yORn = input("是否繼續?(y,n) ")
    if yORn == 'n' or yORn == 'N':
        break
```



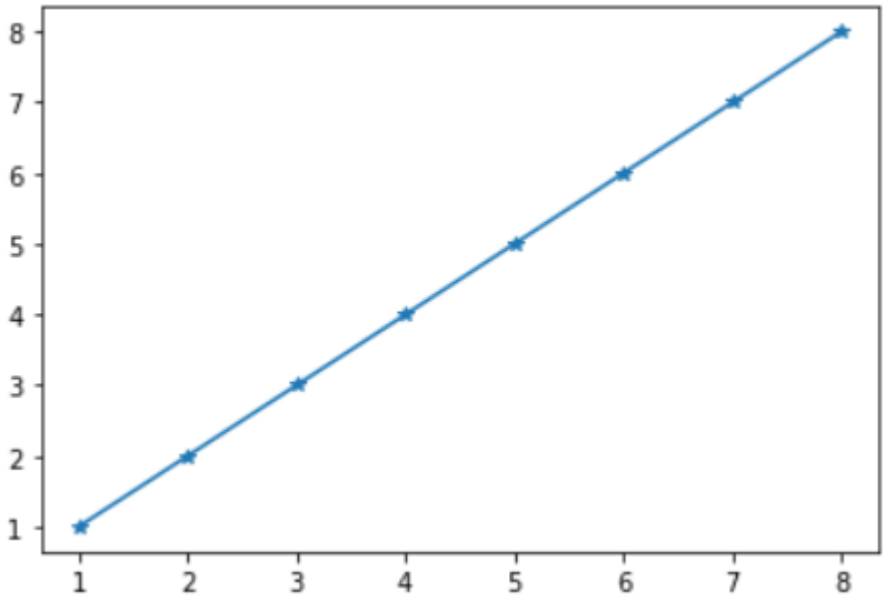
隨機數的移動(補充)

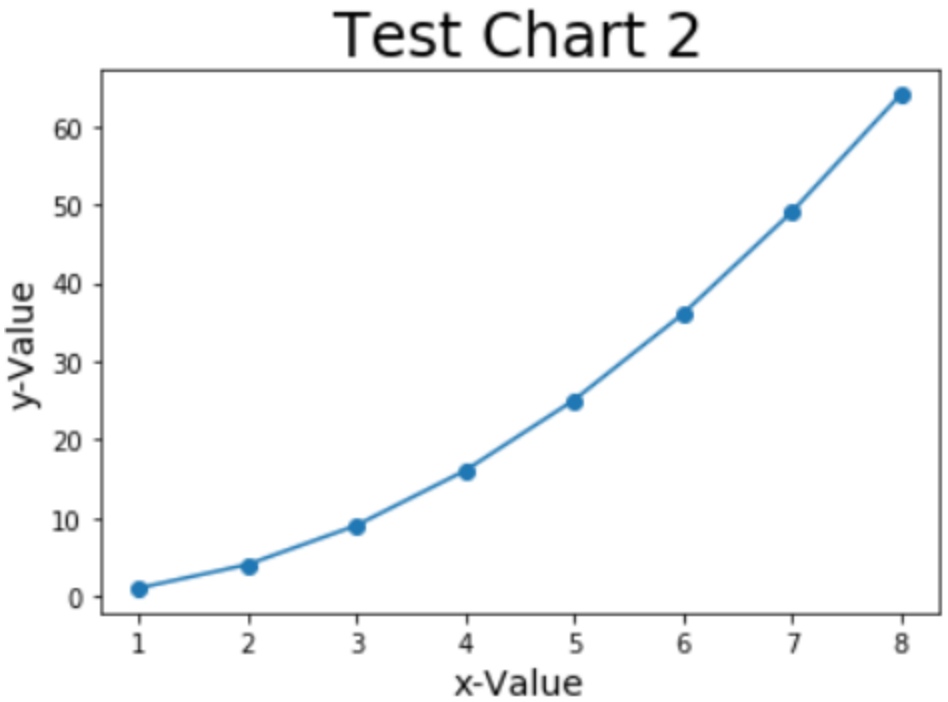
```
import matplotlib.pyplot as plt
import random

def loc(index):
    '''處理座標的移動'''
    x_mov = random.choice([-3, 3])          #隨機x軸移動值
    xloc = x[index-1] + x_mov               #計算x軸新位置
    y_mov = random.choice([-5, -1, 1, 5])   #隨機y軸移動值
    yloc = y[index-1] + y_mov               #計算y軸新位置
    x.append(xloc)                          #x軸新位置加入串列
    y.append(yloc)                          #y軸新位置加入串列

num =  10000                                #設計隨機點的數量
x = [0]                                     #設定第一次執行x軸座標
y = [0]                                     #設定第一次執行y軸座標
while True:
    for i in range(1, num):                 #建立點的座標
        loc(i)
    t = x                                   #色彩隨著x軸變化
    plt.scatter(x, y, s=2, c=t, cmap='brg')
    plt.axes().get_xaxis().set_visible(False) #隱藏x軸座標
    plt.axes().get_yaxis().set_visible(False) #隱藏y軸座標
    plt.show()
    yORn = input("是否繼續?(y,n) ")         #詢問是否繼續
    if yORn == 'n' or yORn == 'N':          #輸入n或N則程式結束
        break
    else:
        x[0] = x[num-1]                     #上次結束x座標成新的起點x座標
        y[0] = y[num-1]                     #上次結束y座標成新的起點y座標
        del x[1:]                           #刪除舊串列x座標元素
        del y[1:]                           #刪除舊串列y座標元素
```



繪製多個圖表

```
import matplotlib.pyplot as plt

data1 = [1, 2, 3, 4, 5, 6, 7, 8]
data2 = [1, 4, 9, 16, 25, 36, 49, 64]
seq = [1, 2, 3, 4, 5, 6, 7, 8]
plt.figure(1)
plt.plot(seq, data1, '-*')
plt.figure(2)
plt.plot(seq, data2, '-o')
plt.title("Test Chart 2", fontsize=24)
plt.xlabel("x-Value", fontsize=14)
plt.ylabel("y-Value", fontsize=14)
plt.show()
```





含有子圖的圖表 subplots()

subplot(2, 1, 1)

subplot(2, 1, 2)

subplot(1, 2, 1)

subplot(1, 2, 2)

subplot(2, 2, 1)

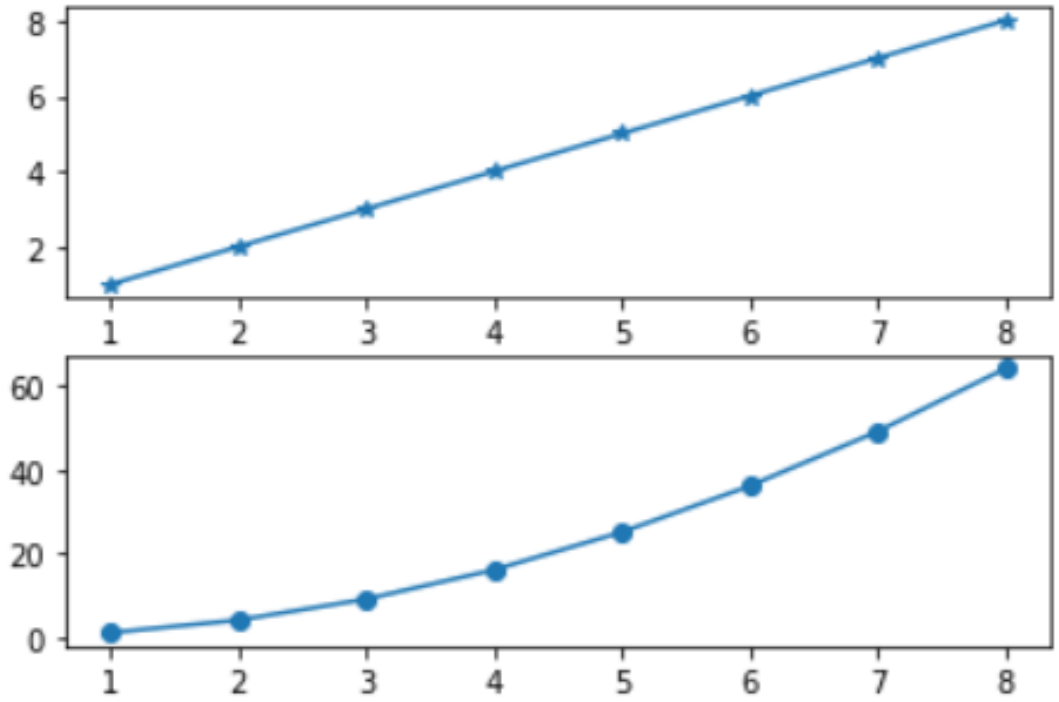
subplot(2, 2, 2)

subplot(2, 2, 3)

subplot(2, 2, 4)

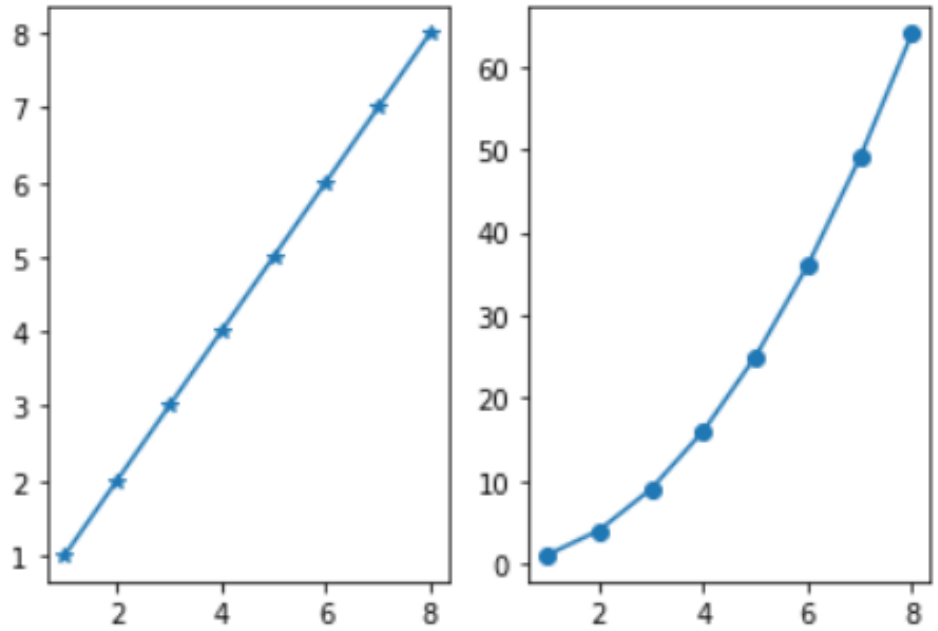
含有子圖的圖表 subplots()

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

data1 = [1, 2, 3, 4, 5, 6, 7, 8]
data2 = [1, 4, 9, 16, 25, 36, 49, 64]
seq = [1, 2, 3, 4, 5, 6, 7, 8]
plt.subplot(2,1,1)
plt.plot(seq, data1, '-*')
plt.subplot(2,1,2)
plt.plot(seq, data2, '-o')
plt.show()
```



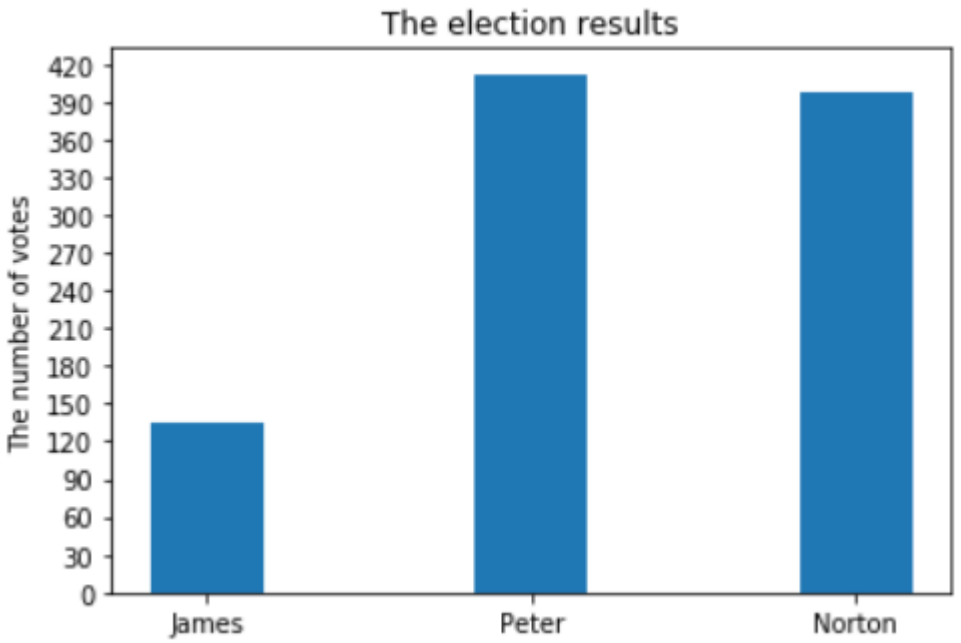
含有子圖的圖表 subplots()

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

data1 = [1, 2, 3, 4, 5, 6, 7, 8]
data2 = [1, 4, 9, 16, 25, 36, 49, 64]
seq = [1, 2, 3, 4, 5, 6, 7, 8]
plt.subplot(1,2,1)
plt.plot(seq, data1, '-*')
plt.subplot(1,2,2)
plt.plot(seq, data2, '-o')
plt.show()
```



長條圖的製作 bar()

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

votes = [135, 412, 397]        #得票數
N = len(votes)                 #計算長度
x = np.arange(N)               #長條圖 x 軸座標
width = 0.35                   #長條圖寬度
plt.bar(x, votes, width)       #繪製長條圖

plt.ylabel('The number of votes')
plt.title('The election results')
plt.xticks(x, ('James', 'Peter', 'Norton'))
plt.yticks(np.arange(0, 450, 30))
plt.show()
```



字串處理

字串處理

- 字串處理技巧用在檔名處理時非常實用。
- Python 提供許多內建 function 方便處理字串

```
>>> String = “hello world”
>>> print(String)
hello world
>>> print(String[:4])
hell
>>> print(String.find(“he”))
0
```

字串處理

```
>>> print(String.capitalize())
Hello world
>>> print(String.upper())
HELLO WORLD
>>> print(String.endswith(“d”))
True
>>> print(String.split(“ ”))
[“hello”, “word”]
>>> a = “1”
>>> print(a.zfill(3))
001
```

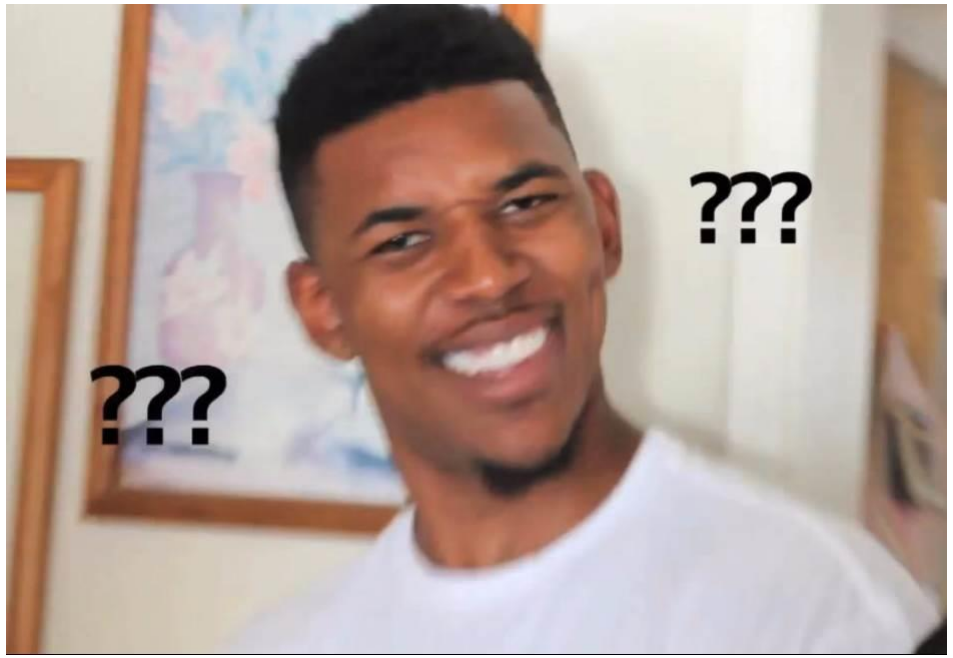
字串處理

```
>>> print(String.replace(“hello”, “Nihao”))
Nihao world

>>> print(String.strip())
helloword
```

正則表達式 (regular expression)

- 甚麼式 regular expression

- [A-Za-z0-9._]+@[A-Za-z.]+\.(com|edu)+\.tw



強大的 regular expression

REGULAR EXPRESSION

3 matches, 1967 steps (~4ms)

/ [A-Za-z0-9._+]+@[A-Za-z\.]+\.(com|edu)+\.tw / g

TEST STRING

SWITCH TO UNIT TESTS

```
"""
swc@iis.sinica.edu.tw02-27883799#1712Big Data Analytics/ Deep
LearningSocial Computing / Computational Social Science /
Crowdsourcing
Multimediaand Network SystemsQuality of ExperienceInformation
SecurityPh.D. candidate at NTU EEchihfan02-27883799#1602Camera
CalibrationComputer VisionData
Analysiscmchang02-27883799#1671System OptimizationMachine
LearningyusraBig data
analysiscclin02-27883799#1668Data Analysisrusi02-
27883799#1668Government Procurement ActFinancial
AnalysisPsychology & NeuroscienceMarketingxinchinchenEmbedded
Systemkyoyachuan062602-27883799
#1601FinTechActuarial ScienceData Analysiskai0604602-
27883799#1601Data AnalysisMachine Learningchloe02-
27839427Accountingafun02-27883799 afun@iis.sinica.edu.tw
#1673Data AnalysisWeb developmentyunhsu198902-
27883799#1668MarketingTIGP Ph.D. Fellow at Academia Sinica &
NCCUbaowalyMachine LearningData AnalysisSocial
Computingchangyc1427883799#1678
Data Analysisjimmy1592302-2788379
jimmy15923@iis.sinica.com.tw#1688Data
AnalysisjasontangAnalysisMachine Learninguchen02-27883799#1668Deep
Learningpjwu02-27883799#1604Computational PhotographyData Analysis
"""
```

regular expression 常用符號

regular expression 使用許多符號來訂定搜尋規則，
要學會使用就必須知道符號的意義。

符號	意義	範例	符合範例的字串
*	前一字元或括號內字元出現0次或多次	a*b*	aaaab、aabb、bbb
+	前一字元或括號內字元出現1次或多次	a+b+	aaabb、abbbb、abbbbb
{m, n}	前一字元或括號內字元出現 m 次到 n次 (包含 m, n)	a{1,2}b{3,4}	abbb、aabbbb、aabbb
[]	符合括號內的任一字元	[A-Z]+	APPLE、QWER
\	跳脫字元	\.|\\	.|
.	符合任何單一字元(符號、數字、空格)	a.c	auc、abc、a c

Python 的 re

Python 有內建的 regular expression 函數

```
Signature: re.findall(pattern, string, flags=0)
Docstring:
Return a list of all non-overlapping matches in the string.

If one or more capturing groups are present in the pattern, return
a list of groups; this will be a list of tuples if the pattern
has more than one group.
```

```
# 找出所有內容等於 python_crawler 的文字
pattern = "我寫好的 regular expression"
string = "我想要搜尋的字串"
re.findall(pattern, string)
```

範例 1：*,+,{} 的用法

* 代表前面的字元可出現零次以上

+ 代表前面的字元至少要出現一次以上

{m, n}代表前面的字元可出現 m 次 ~ n 次

```
import re
pattern = "a+b*c"
test_string = 'find aabc, ac, skip abb, dd'
re.findall(pattern, test_string)
```

```
['aabc', 'ac']
```

範例 2：找到數字

[ ] 代表的意思是這個字元可以是括號內的任何一個

[0-9] 代表可以是 0 ~ 9 之間的任意數字

[a-z] 代表可以是 a ~ z 之間的任意文字

```
pattern = "[0-9]+"
test_string = '12 drummers drumming, 11 pipers piping, 10 loard a-leaping'
re.findall(pattern, test_string)
```

```
['12', '11', '10']
```

範例 3：找到文字

當有指定的文字需要搜尋，可透過 [ ] 搭配 *, +, {} 進行搜尋

```
import re

pattern = "[cmf]an"
test_string = 'find: can, man, fan, skip: dan, ran, pan'
re.findall(pattern, test_string)
```

```
['can', 'man', 'fan']
```

範例 4：跳脫符號

萬一我今天想要找到的就是 + 這個符號怎麼辦?

ANS : 在前面加上一個跳脫符號 \

```
import re

pattern = ". {3}\."
test_string = 'find: 591., dot., yes., skip: non!'
re.findall(pattern, test_string)
```

```
['591.', 'dot.', 'yes.']
```

範例 5：條件式搜尋

| 代表左右邊只要任一符合條件即可

```
import re

pattern = "I love cats|I love dogs"
test_string = 'find: I love cats, I love dogs, skip: I love logs, I love cogs'
re.findall(pattern, test_string)
```

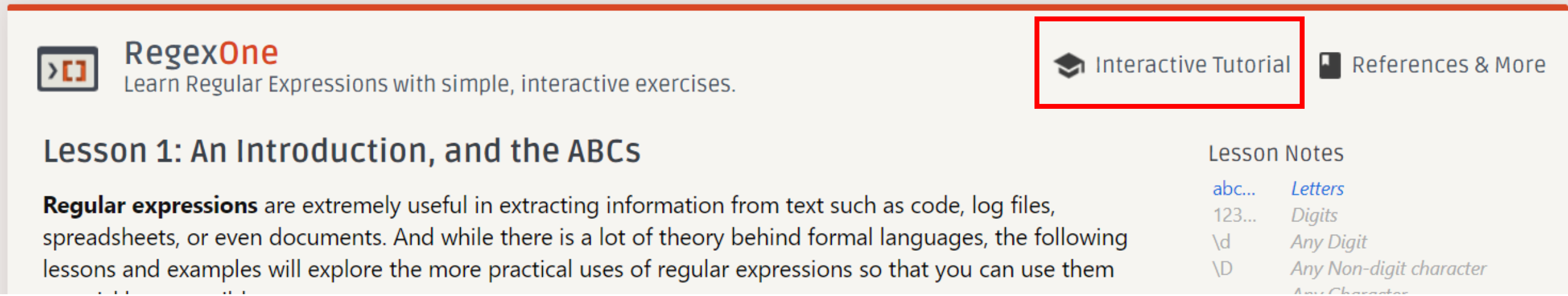
```
['I love cats', 'I love dogs']
```

Email 的 regular expression (1/2)

[A-Za-z0-9._]+@[A-Za-z.]+(com|edu)\.tw

練習時間

請到[RegexOne](#)網站右上方的Interactive Tutorial 完成總共 15 個練習



其他常用 Modules

os – Get the full path

```
>>> import os

>>> related_path = ‘src’
>>> absolute_path = os.path.abspath(related_path)
>>> print(‘absolute_path’)
/home/afun/Desktop/src
```

os – os.path.join

```
>>> ‘/’.join(‘path’, ‘result’, ’a.csv’)
path/result/a.csv

>>> os.path.join(‘path’, ‘result’, ‘a.csv’)
path/result/a.csv
```

os – Check the directory is exist

If the filename does not exist, there always have a method to create new file implicitly. However, if the directory doex not exist, it raise exception.

```
import os
path = ‘/home/afun/src/utils’
# check the path
if not os.path.exists(path):
    # invoke
    os.makedirs(path)
```

os – Other useful function

os.remove()

os.rename()

os.listdir() # list file & folders in the directory

os.getcwd() # get current directory

os.chmod() # change mode

os.path.split() # return (dirname(), basename())

os.path.basename()

glob – List the files in condition

Normally, there’s a lot of different type of file under the same directory but use os.listdir() will return all the file.

What we need is list file flexibly, including specifying extension and the level of structure

glob – List the files by specifying extension

```
import os
import glob

path = os.getcwd()

print(os.listdir(path))

print(glob.glob(‘*.py’))
```