

碁峯資訊

版權聲明：本教學投影片僅供教師授課講解使用，投影片內之圖片、文字及其相關內容，未經著作權人許可，不得以任何形式或方法轉載使用。

```

import pygame, random, time

class Ball(pygame.sprite.Sprite):
    dx = 0
    dy = 0
    x = 0
    y = 0
    direction = 0
    speed = 0

    def __init__(self, sp, srx, sry, radius, color):
        pygame.sprite.Sprite.__init__(self)
        self.speed = sp
        self.x = srx
        self.y = sry
        self.image = pygame.Surface([radius*2, radius*2])
        self.image.fill((255,255,255))
        pygame.draw.circle(self.image, color, (radius,radius), radius, 0)
        self.rect = self.image.get_rect()
        self.rect.center = (srx,sry)
        self.direction = random.randint(40,70)

    def update(self):
        radian = math.radians(self.direction)
        self.dx = self.speed * math.cos(radian)
        self.dy = -self.speed * math.sin(radian)
        self.x += self.dx
        self.y += self.dy
        self.rect.x = self.x
        self.rect.y = self.y
        if(self.rect.left <= 0 or self.rect.right >= screen.get_width()-10):
            self.bouncelr()
        elif(self.rect.top <= 0 or self.rect.bottom >= screen.get_height()-10):
            self.bounceup()
        else:
            return

    def bouncelr(self):
        self.direction = 360 - self.direction

    def bounceup(self):
        self.direction = (180 - self.direction) % 360

class Brick(pygame.sprite.Sprite):
    def __init__(self):
        self.image = pygame.image.load("media\\brick.png")
        self.image.convert()
        self.rect = self.image.get_rect()
        self.rect.x = int((screen.get_width() - self.rect.width)/2)
        self.rect.y = screen.get_height() - self.rect.height - 20

class Pad(pygame.sprite.Sprite):
    def __init__(self):
        pygame.sprite.Sprite.__init__(self)
        self.image = pygame.image.load("media\\pad.png")
        self.image.convert()
        self.rect = self.image.get_rect()
        self.rect.x = int((screen.get_width() - self.rect.width)/2)
        self.rect.y = screen.get_height() - self.rect.height - 20

    def update(self):
        pos = pygame.mouse.get_pos()
        self.rect.x = pos[0]
        if self.rect.x > screen.get_width() - self.rect.width:
            self.rect.x = screen.get_width() - self.rect.width

def gameover(message):
    global running
    text = font1.render(message, 1, (255,0,255))
    screen.blit(text, (screen.get_width()/2-100,screen.get_height()/2-20))
    pygame.display.update()
    time.sleep(3)
    running = False

pygame.init()
font = pygame.font.SysFont("SimHei", 20)
font1 = pygame.font.SysFont("SimHei", 32)
soundhit = pygame.mixer.Sound("media\\hit.wav")
soundpad = pygame.mixer.Sound("media\\pad.wav")

```

Python

Python Beginner Course

初學特訓班

第四版

07

實戰：一圖勝千言， matplotlib 資訊圖表繪製

[7-1 繪製折線圖：plot](#)

[7-2 繪製長條圖：bar](#)

[7-3 圓餅圖繪製：pie](#)

[7-4 設定圖表區：figure](#)

[7-5 在圖表區加入多張圖表：subplot](#)

[7-6 專題：台灣股市股價走勢圖](#)

7.1 繪製折線圖：plot

7.1.1 Matplotlib 模組的使用

Matplotlib 模組在使用前必須先安裝，(Anaconda 中預設已安裝)，語法如下：

```
pip install matplotlib
```

使用 Matplotlib 繪圖首先要匯入 Matplotlib 模組，由於大部分繪圖功能是在「matplotlib.pyplot」中，因此通常會在匯入「matplotlib.pyplot」時設定一個簡短的別名，方便往後輸入，例如將別名取為「plt」：

```
import matplotlib.pyplot as plt
```

7.1.2 繪製折線圖

折線圖是以 `plot` 函數繪製，語法為：

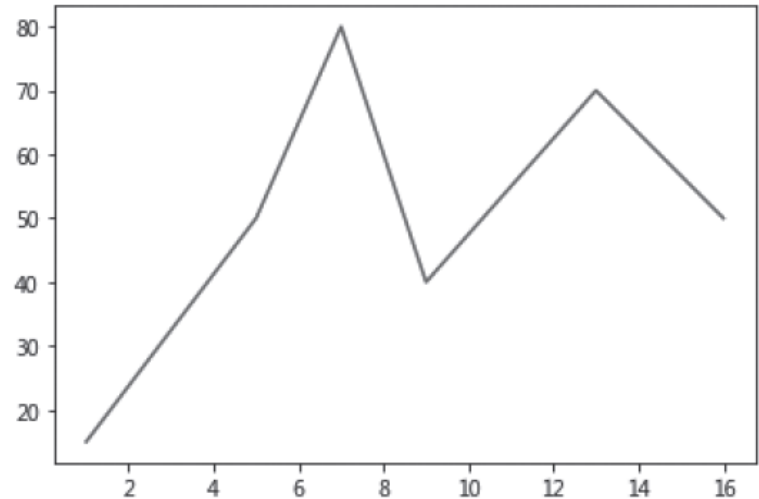
```
plt.plot(x 座標串列 , y 座標串列 [, 其他參數 ])
```

繪圖後不會自動顯示，必須要用 `show` 函數顯示，例如：

```
plt.show()
```

程式碼：`plot1.py`

```
import matplotlib.pyplot as plt  
  
listx = [1,5,7,9,13,16]  
listy = [15,50,80,40,70,50]  
plt.plot(listx, listy)  
plt.show()
```



7.1.3 設定線條及圖例

- **linewidth** or **lw**：設定線條寬度，預設為 1.0，例如設定線條寬度為 5.0：
`linewidth=5.0`。
- **color**：設定線條顏色，預設為藍色，例如設定線條顏色為紅色：`color="r"` 或 `color="red"`。

顏色	代表值	顏色	代表值
藍	b, blue	青	c, cyan
紅	r, red	洋紅	m, magenta
綠	g, green	黑	k, black
黃	y, yellow	白	w, white

- **linestyle** or **ls**：設定線條樣式，設定值有「-」（實線）、「--」（虛線）、「-。」（虛點線）及「:」（點線），預設為「-」。

- **marker**：設定標記樣式，設定值如下：

符號	說明	符號	說明
".", "o", "*"	點、圓、星	"h", "H"	六邊形 1,2
"v", "^\td> <td>正倒三角形</td> <td>"d", "D"</td> <td>鑽石 小, 大</td>	正倒三角形	"d", "D"	鑽石 小, 大
"<", ">"	左右三角形	"+", "x"	十字、叉叉
"s"	矩形	"_", " "	橫線、直線
"p"	五角形	"1", "2", "3", "4"	上下左右人字形

- **markersize** or **ms**：標記大小，例如設定標記為 12 點：`ms=12`。
- **color**、**linestyle**、**marker** 組合字串：這三個設定值的字串可以直接合併設定。
- **label**：設定圖例名稱，例如設定圖例名稱為「label」：`label="label"`。此屬性需搭配 **legend** 函數才有效果。

7.1.4 設定標題

圖形繪製完成後，可對圖表做一些設定，如圖表標題、x 及 y 座標標題等，讓觀看圖表者更容易了解圖表的意義。

設定圖表標題、x 及 y 座標標題的語法如下，如果不設定 `fontsize`，大小會一樣：

```
plt.title( 圖表標題 (,fontsize= 點數 ))
```

```
plt.xlabel(x 座標標題 (,fontsize= 點數 ))
```

```
plt.ylabel(y 座標標題 (,fontsize= 點數 ))
```

例如分別設定了圖表及x、y 座標的標題：

程式碼：plot2.py

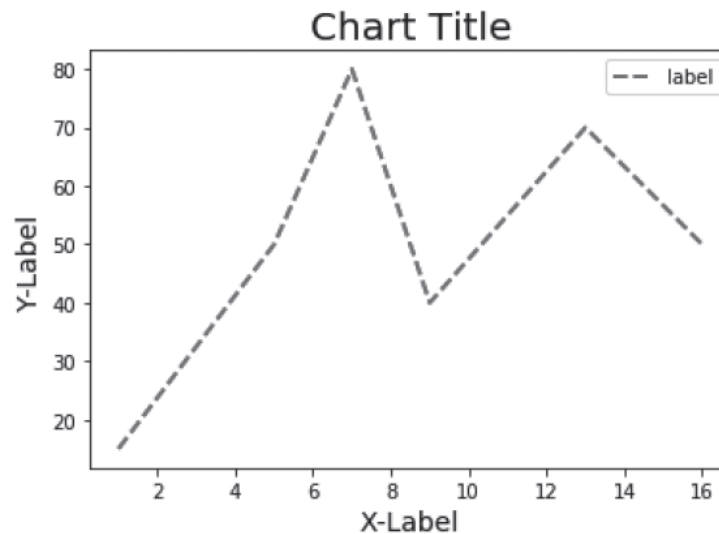
...

```
plt.title("Chart Title", fontsize=20) # 圖表標題
```

```
plt.xlabel("X-Label", fontsize=14) # x 座標標題
```

```
plt.ylabel("Y-Label", fontsize=14) # y 座標標題
```

...



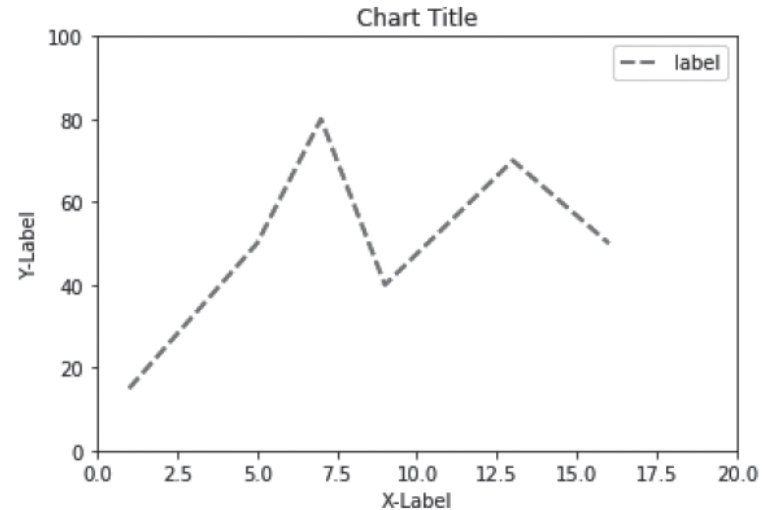
7.1.5 設定座標範圍

如果沒有指定 x 及 y 座標範圍，系統會根據資料判斷最適合的 x 及 y 座標範圍。設計者可以自行設定 x 及 y 座標範圍，語法為：

```
plt.xlim(起始值, 終止值) # 設定 x 座標範圍  
plt.ylim(起始值, 終止值) # 設定 y 座標範圍
```

程式碼：plot3.py

```
...  
plt.xlim(0, 20) # 設定 x 座標範圍  
plt.ylim(0, 100) # 設定 y 座標範圍  
...
```



7.1.6 設定格線

為圖表加上格線的語法如下：

```
plt.grid(True)
```

也可以進一步設定格線的顏色、寬度、樣式及透明度，例如：

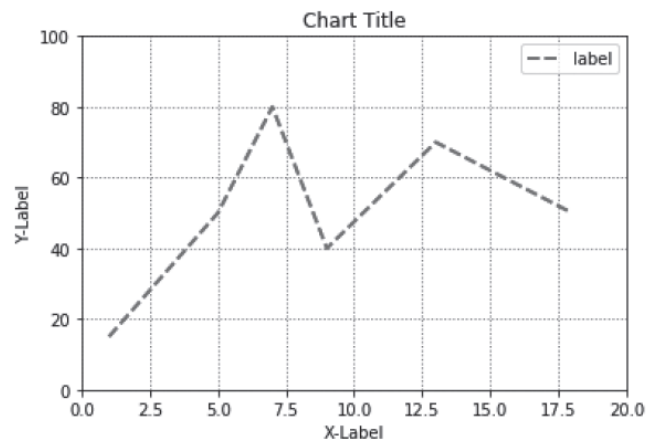
```
plt.grid(color='red', linestyle=':', linewidth=1, alpha=0.5)
```

程式碼：`plot4.py`

```
...
```

```
plt.grid(color='black', linestyle=":", linewidth='1', alpha=0.5)
```

```
...
```

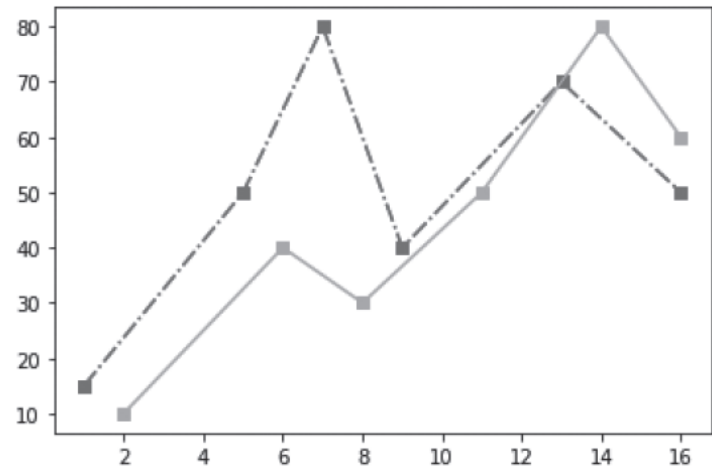


7.1.7 同時繪製多組資料

一個圖表中可以繪製多組資料的線條，如果沒有設定線條顏色，系統會自行設定不同顏色繪圖。例如繪製 2 組數據的線條：

程式碼：plot5.py

```
...  
listx1 = [1,5,7,9,13,16]  
listy1 = [15,50,80,40,70,50]  
plt.plot(listx1, listy1, 'r-.s')  
listx2 = [2,6,8,11,14,16]  
listy2 = [10,40,30,50,80,60]  
plt.plot(listx2, listy2, 'y-s')  
plt.show()
```



其實多組數據可以一起繪圖，因為每個線條的數據、樣式都不同，其語法為：

```
plt.plot(x1 串列 , y1 串列 , 樣式 1, x2 串列 , y2 串列 , 樣式 2, ... )
```

程式碼：plot6.py

```
...  
listx1 = [1,5,7,9,13,16]  
listy1 = [15,50,80,40,70,50]  
listx2 = [2,6,8,11,14,16]  
listy2 = [10,40,30,50,80,60]  
plt.plot(listx1, listy1, 'r-.s', listx2, listy2, 'y-s')  
plt.show()
```

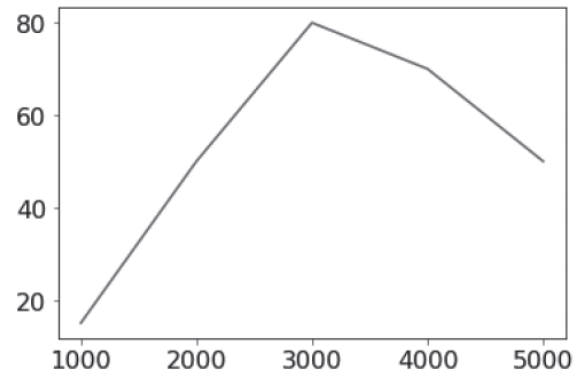
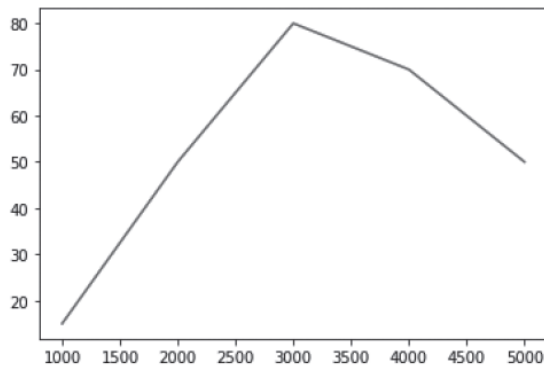
7.1.8 設定座標刻度

在以下的圖表中，x 座標範圍為 0 到 5000，在預設的狀態下 Matplotlib 自動以 500 為間距加上了刻度。但如果想要自訂座標刻度，語法為：

```
plt.xticks( 串列 ) # 設定 x 座標刻度  
plt.yticks( 串列 ) # 設定 y 座標刻度
```

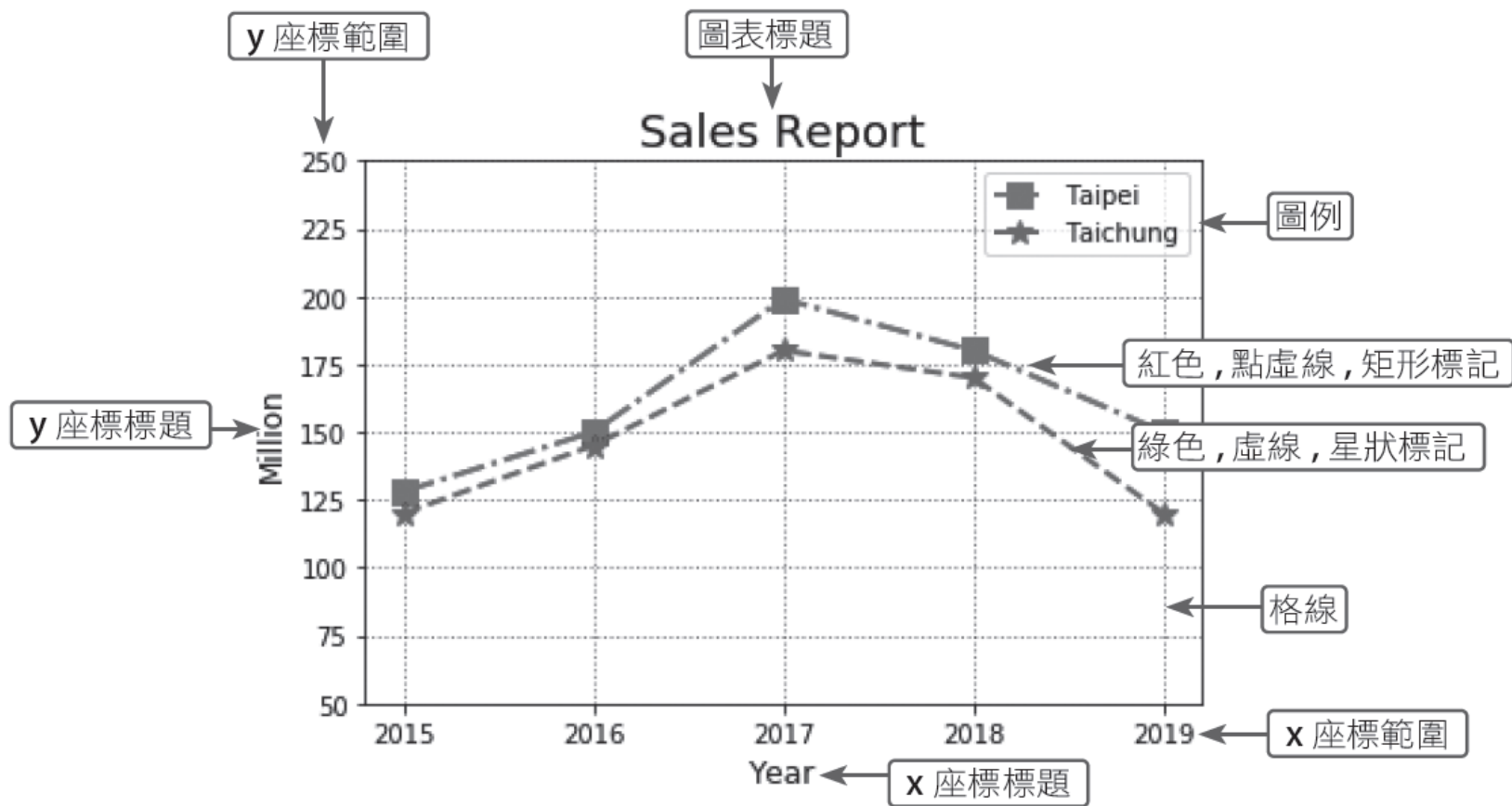
程式碼：plot7.py

```
...  
listx = [1000,2000,3000,4000,5000]  
listy = [15,50,80,70,50]  
plt.plot(listx, listy)  
plt.xticks(listx)  
plt.tick_params(axis='both', labelsize=16, color='red')  
plt.show()
```



範例：繪製折線圖

繪製折線圖並設定各種圖表特性。



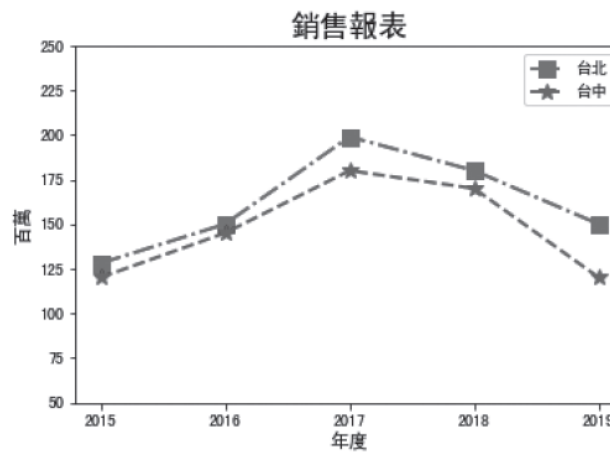
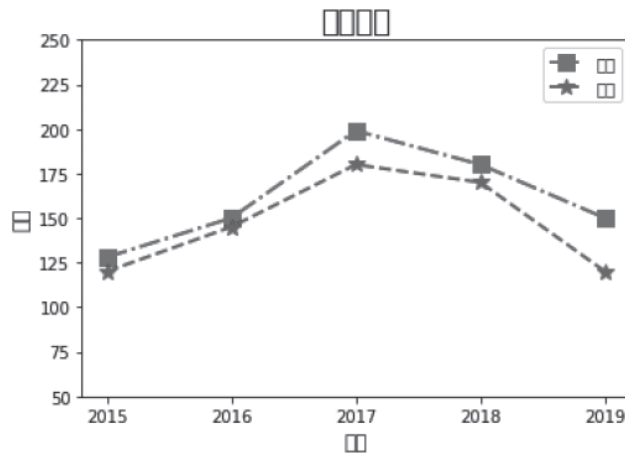
7.1.9 Matplotlib 圖表顯示中文

此時可以使用Matplotlib 的rcParam 的函數修改預設配置，即能讓圖表裡的中文字正常顯示。請在原來的程式碼中加入以下的設定：

程式碼：plot9.py

```
...  
# 設定中文字型及負號正確顯示  
plt.rcParams["font.sans-serif"] = "Microsoft JhengHei"  
# 也可設 mingliu 或 DFKai-SB  
plt.rcParams["axes.unicode_minus"] = False  
plt.show()
```

如下原來圖表中無法正確顯示的文字，都成功的變成中文了喔！



7.2 繪製長條圖：bar

7.2.1 繪製長條圖

長條圖是以 `bar` 函數繪製，語法為：

```
plt.bar(x 座標串列, y 座標串列, width=0.8, bottom=0[, 其他參數])
```

繪圖時除了 `x`、`y` 座標串列參數之外，呈現每個項目的矩形也是重點，常用參數有：

- **width**：設定每個項目矩形的寬度。以二個刻度之間的距離為基準，用百分比為單位來設定。不設定時預設值為0.8。
- **bottom**：設定每個項目矩形 `y` 座標的起始位置，不設定時預設值為0。
- **color**：設定每個項目矩形的顏色，設定值與折線圖相同，預設為藍色。例如設定紅色可以為"r" 或"red"。如果設定值為"rgb"，代表會以紅、綠、藍依序循環顯示每個項目矩形的顏色。
- **label**：設定每個項目圖例名稱，此屬性需搭配 `legend` 函數才有效果。

本範例將要用長條圖呈現每個課程的選修人數：

程式碼：bar1.py

...

```
listx = ['c', 'c++', 'c#', 'java', 'python']
```

```
listy = [45, 28, 38, 32, 50]
```

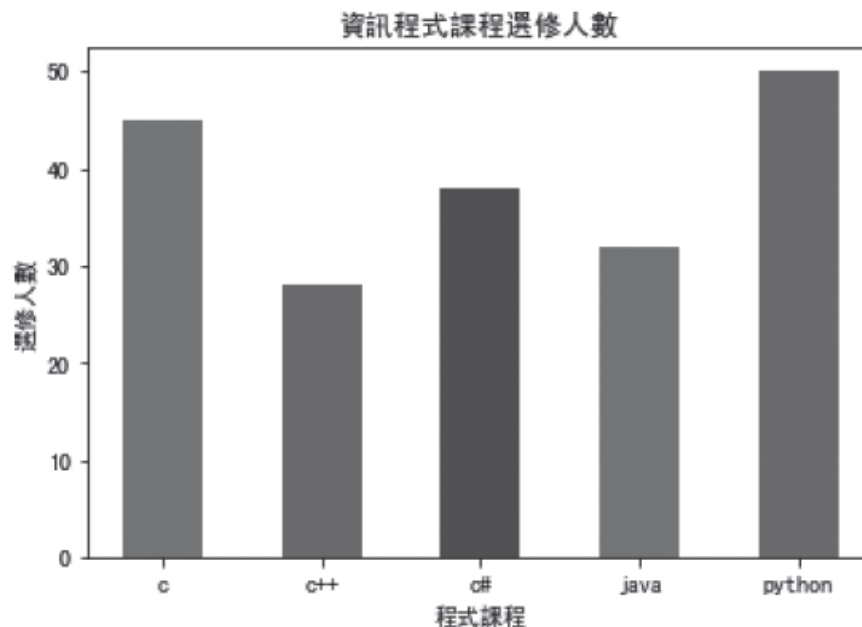
```
plt.bar(listx, listy, width=0.5, color='rgb')
```

```
plt.title(" 資訊程式課程選修人數 ")
```

```
plt.xlabel(" 程式課程 ")
```

```
plt.ylabel(" 選修人數 ")
```

...



7.2.2 繪製橫條圖

橫條圖是以 `barh` 函數繪製，語法為：

```
plt.barh(y 座標串列, x 座標串列, height=0.8, left=0[, 其他參數])
```

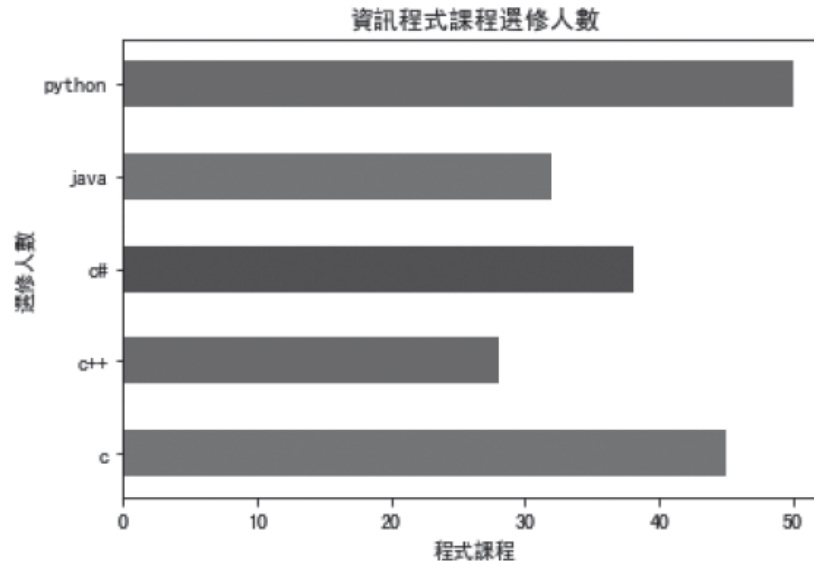
橫條圖基本上與長條圖相似，但因為方向不同，所有參數就必須倒過來。繪圖時除了設定矩形樣式的參數與長條圖相同外，還需特別注意：

- **y座標串列**：顯示每個項目的名稱串列或是序列串列。
- **x座標串列**：顯示每個項目的數值串列。
- **height**：設定每個項目矩形的高度。以二個刻度之間的距離為基準，用百分比為單位來設定。不設定時預設值為0.8。
- **left**：設定每個項目矩形 x 座標的起始位置，不設定時預設值為0。

本範例將要用橫條圖呈現每個課程的選修人數：

```
程式碼：bar2.py
```

```
...  
listy = ['c', 'c++', 'c#', 'java', 'python']  
listx = [45, 28, 38, 32, 50]  
plt.barh(listy, listx, height=0.5, color='rgb')  
plt.title(" 資訊程式課程選修人數 ")  
plt.xlabel(" 程式課程 ")  
plt.ylabel(" 選修人數 ")  
...
```



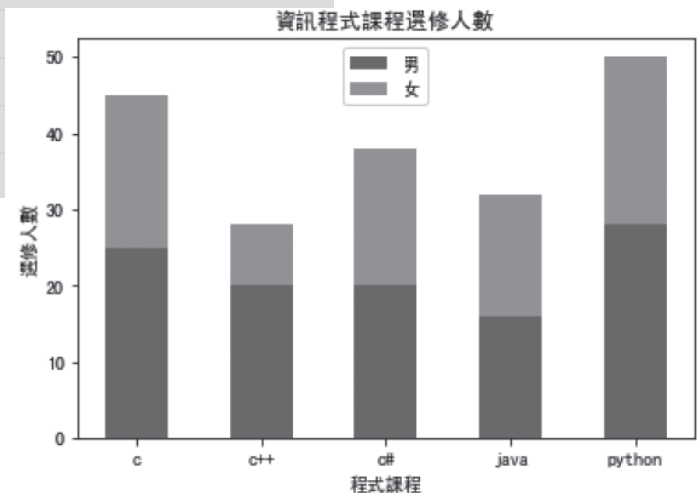
7.2.3 繪製堆疊長條圖

這時就必須要應用到 `bottom` 屬性，完成第一組長條圖後，在繪製第二組長條圖時，可將 `y` 軸的起點設定為第一組資料的 `y` 軸高度。

在範例中將要用堆疊長條圖來表現每個課程中選修人數，並顯示男女的比重：

程式碼：bar3.py

```
...
listx = ['c','c++','c#','java','python']
listy1 = [25,20,20,16,28]
listy2 = [20,8,18,16,22]
plt.bar(listx, listy1, width=0.5, label='男')
plt.bar(listx, listy2, width=0.5, bottom=listy1, label='女')
plt.legend()
plt.title(" 資訊程式課程選修人數 ")
plt.xlabel(" 程式課程 ")
plt.ylabel(" 選修人數 ")
...
```



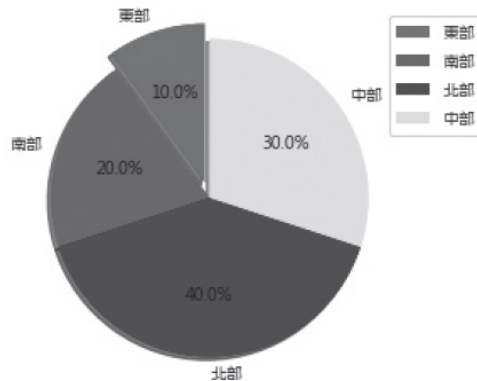
7.3 圓餅圖繪製：pie

圓餅圖是以 pie 函數繪製，語法為：

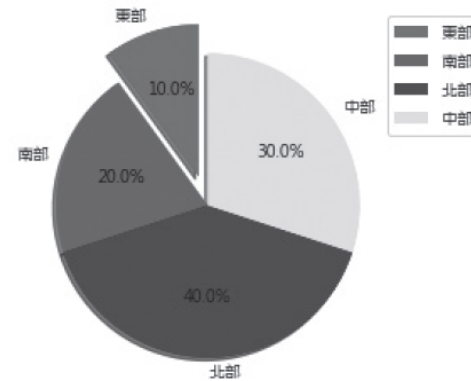
```
plt.pie(資料串列[, 其他串列參數])
```

資料串列是數值串列，為圓餅圖的資料，為必要參數。其他常用的參數有：

- **labels**：每一個項目標題組成的串列。
- **colors**：每一個項目顏色字元組成字串或是串列，如'rgb' 或 ['r', 'g', 'b']。
- **explode**：每一個項目凸出距離數字組成的串列，「0」表示正常顯示。下圖顯示第一部分不同凸出值的效果。



▲ explode=0.1

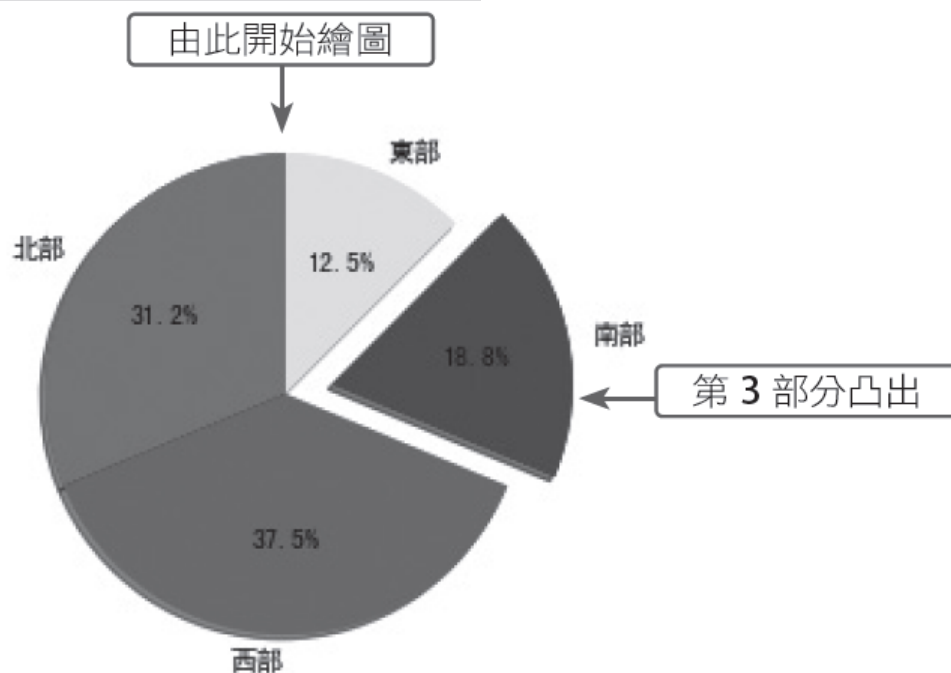


▲ explode=0.2

- **labeldistance**：項目標題與圓心的距離是半徑的多少倍，例如「1.1」表示項目標題與圓心的距離是半徑的 1.1 倍。
- **autopct**：項目百分比的格式，語法為「% 格式 %%」，例如「%2.1f%%」表示整數 2 位數，小數 1 位數。
- **pctdistance**：百分比文字與圓心的距離是半徑的多少倍。
- **shadow**：布林值，True 表示圖形有陰影，False 表示圖形沒有陰影。
- **startangle**：開始繪圖的起始角度，繪圖會以逆時針旋轉計算角度。

程式碼：pie.py

```
...
3     sizes = [25, 30, 15, 10]
4     labels = ["北部", "西部", "南部", "東部"]
5     colors = ["red", "green", "blue", "yellow"]
6     explode = (0, 0, 0.2, 0)
7     plt.pie(sizes,
8             explode = explode,
9             labels = labels,
10            colors = colors,
11            labeldistance = 1.1,
12            autopct = "%2.1f%%",
13            pctdistance = 0.6,
14            shadow = True,
15            startangle = 90)
...
```



7.4 設定圖表區：figure

圖表區是以 **figure** 類別來建立，語法為：

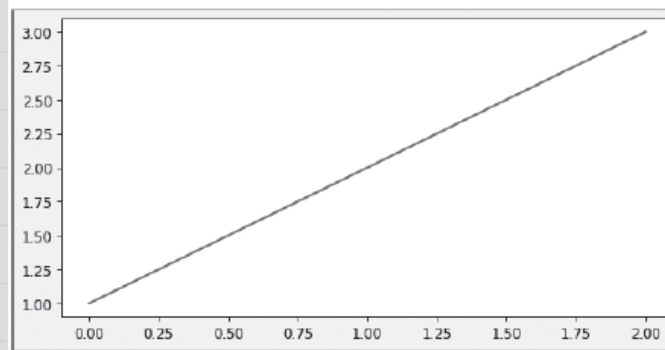
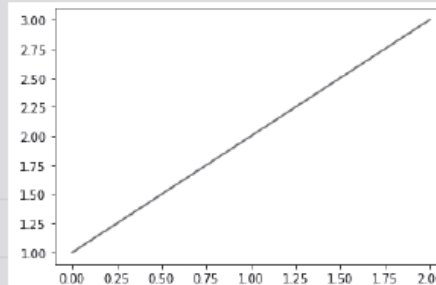
```
plt.figure([ 設定屬性參數 ])
```

如果沒有設定參數則會以預設值建立圖表區，以下為常用的參數：

- **figsize**：設定方式為串列：[寬, 高]，單位為英吋，預設值為[6.4, 4.8]。
- **dpi**：設定解析度，單位為每英吋的點數(Dots per inch)。
- **facecolor**：設定背景顏色，預設值為白色(white)。
- **edgecolor**：設定邊線顏色，預設值為白色(white)。
- **frameon**：布林值，設定是否有邊框，預設值為True。
- **tight_layout**：布林值，設定多個圖表間是否有邊界，預設值為False。

程式碼：figure.py

```
...
import matplotlib.pyplot as plt
# 新增圖表區
plt.figure()
plt.plot([1,2,3])
# 新增圖表區並設定屬性
plt.figure( figsize=[10,5],
            dpi=96,
            facecolor="whitesmoke",
            edgecolor="r",
            linewidth=1,
            frameon=True)
plt.plot([1,2,3])
plt.show()
```



7.5 在圖表區加入多張圖表：subplot

在圖表區用欄列方式加入多張圖表可以使用subplot 函數，語法為：

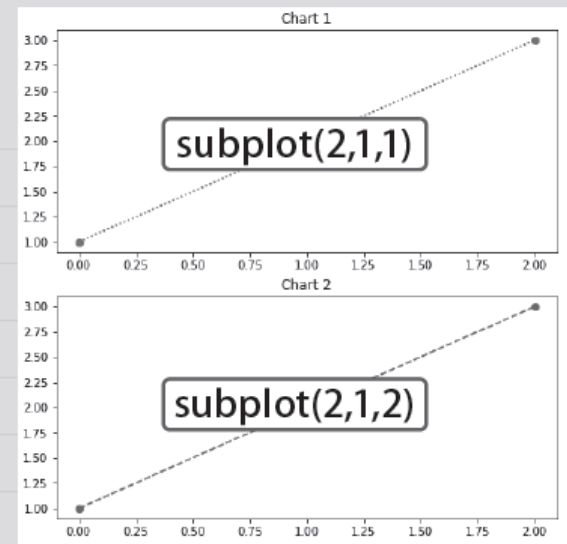
```
plt.subplot( 橫列數 , 直欄數 , 圖表索引值 )
```

例如要在圖表區加入2 列1 欄的二張圖表：

程式碼：subplot1.py

```
import matplotlib.pyplot as plt
plt.figure(figsize=[8,8])
plt.subplot(2,1,1)
plt.title(label='Chart 1', fontsize=20)
plt.plot([1,2,3], 'r:o')

plt.subplot(2,1,2)
plt.title(label='Chart 2', fontsize=20)
plt.plot([1,2,3], 'g--o')
plt.show()
```

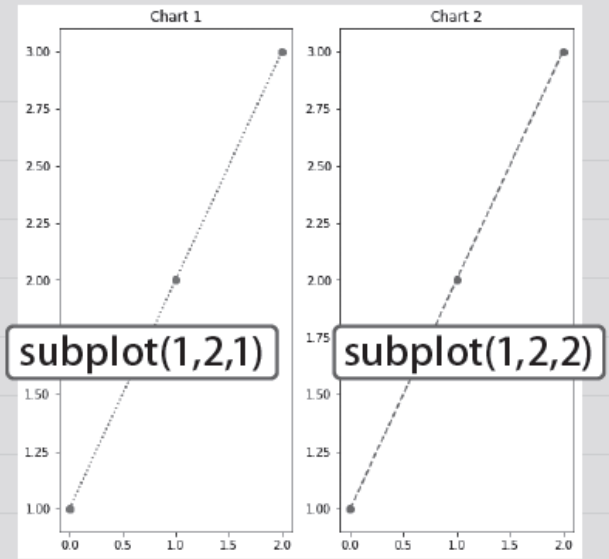


例如要在圖表區加入1列2欄的二張圖表：

程式碼：`subplot2.py`

```
import matplotlib.pyplot as plt
plt.figure(figsize=[8,8])
plt.subplot(1,2,1)
plt.title(label='Chart 1', fontsize=20)
plt.plot([1,2,3], 'r:o')

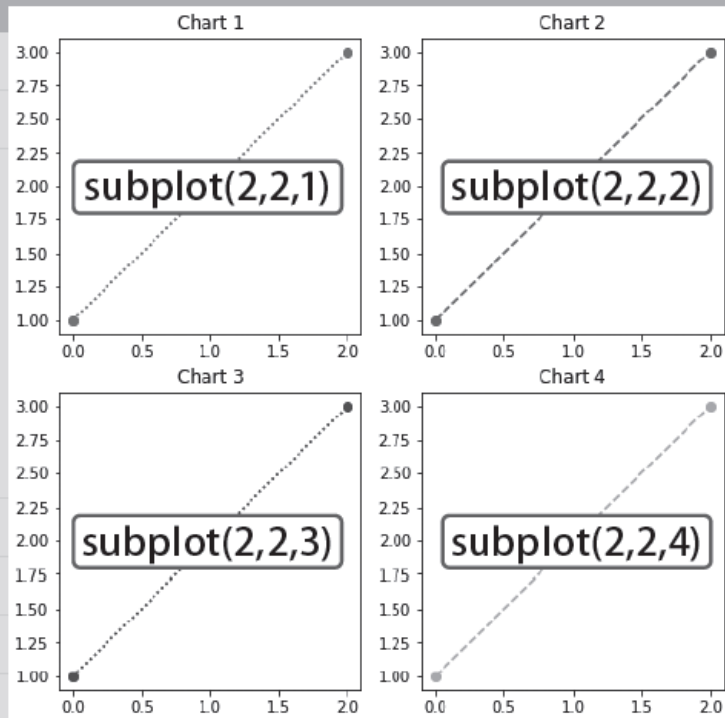
plt.subplot(1,2,2)
plt.title(label='Chart 2', fontsize=20)
plt.plot([1,2,3], 'g--o')
plt.show()
```



再多張的圖表也沒問題，例如要在圖表區加入2列2欄的四張圖表：

程式碼：subplot3.py

```
import matplotlib.pyplot as plt
plt.figure(figsize=[8,8])
plt.subplot(2,2,1)
plt.title(label='Chart 1')
plt.plot([1,2,3], 'r:o')
plt.subplot(2,2,2)
plt.title(label='Chart 2')
plt.plot([1,2,3], 'g--o')
plt.subplot(2,2,3)
plt.title(label='Chart 3')
plt.plot([1,2,3], 'b:o')
plt.subplot(2,2,4)
plt.title(label='Chart 4')
plt.plot([1,2,3], 'y--o')
plt.show()
```



7.6 專題：台灣股市股價走勢圖

7.6.1 取得個股月成交資訊

以Chrome 瀏覽器開啟「<http://www.twse.com.tw>」台灣證券交易所網頁，點選交易資訊 / 盤後資訊 / 個股月成交資訊。



▲ 台灣證券交易所網站：<https://www.twse.com.tw/zh/>

在資料日期欄位下拉式選單中選擇民國108年、股票代碼欄位中輸入2330，然後按查詢鈕，即會顯示「108年2330台積電月成交資訊」。

按 列印/ HTML 鈕即可以表格方式顯示下載的資訊，也可以透過網址直接下載資料。



108年2330 台積電 月成交資訊

年度	月份	最高價	最低價	加權(A/B)平均價	成交筆數	成交金額(A)	成交股數(B)	週轉率(%)
108	1	229.50	206.50	218.60	241,579	162,731,372,580	744,403,635	2.87
108	2	240.00	226.00	232.09	130,403	92,471,159,656	398,420,368	1.53
108	3	248.50	227.50	238.08	161,696	115,730,790,581	486,084,900	1.87
108	4	270.00	245.00	258.72	205,989	169,010,058,727	653,253,770	2.51
108	5	265.00	227.00	242.02	281,361	218,717,277,224	903,700,933	3.48
108	6	248.50	229.50	239.31	266,583	168,973,111,961	706,079,172	2.72
108	7	266.50	240.00	253.60	270,831	173,426,841,378	683,857,533	2.63
108	8	259.00	240.00	250.90	236,552	164,256,059,178	654,659,478	2.52
108	9	272.50	253.00	264.00	211,892	157,033,967,680	594,814,567	2.29
108	10	301.50	273.00	288.94	293,469	227,751,124,384	788,214,849	3.03
108	11	315.00	296.50	307.10	222,914	192,686,573,818	627,432,788	2.41
108	12	345.00	304.00	328.08	298,188	242,147,622,040	738,054,599	2.84

7.6.2 實戰：個股年度統計圖

本專題分析台灣證券交易所月成交資料網頁後，擷取整個年度個股每月最高成交價、最低成交價，繪製個股每月統計圖。

