



# 視覺化套件

Matplotlib

# 前言

所謂一圖抵千言，資料視覺化就是將資料以圖表的方式呈現給使用者閱讀，讓使用者能快速了解資料內容

3	1	3	7	9	2	9	3	2	6	4	5	7	2	3	2	3	4	8
8	8	3	9	6	1	3	7	3	5	3	1	3	4	5	4	2	1	6
6	3	8	7	4	1	8	6	3	6	8	8	6	6	3	5	5	5	8
6	8	2	4	3	1	7	5	4	2	7	4	1	9	1	8	2	4	3
2	3	3	8	7	8	6	8	1	4	5	8	7	2	2	5	3	5	2
9	9	2	6	8	1	7	6	3	2	7	6	3	8	3	8	5	1	3
3	5	2	9	2	9	5	8	7	9	3	2	8	8	4	9	5	9	4
5	8	5	6	4	5	8	8	6	1	3	7	8	9	5	5	3	2	5
8	1	3	3	1	5	1	4	2	5	4	1	5	3	6	9	3	2	5
1	6	8	5	1	2	4	3	3	3	2	4	8	8	7	6	4	2	1

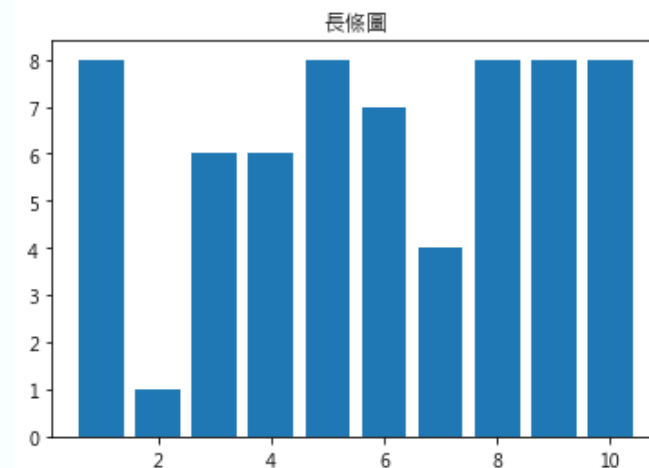
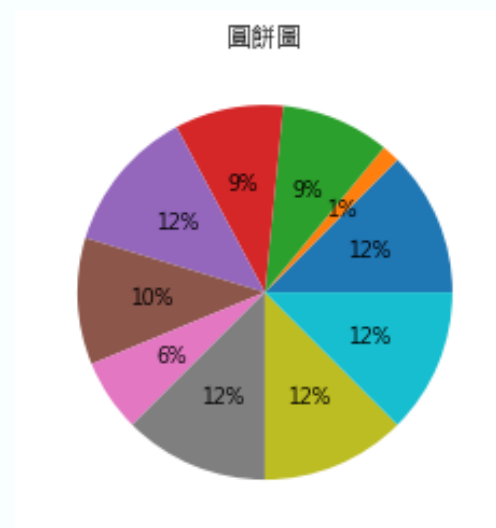
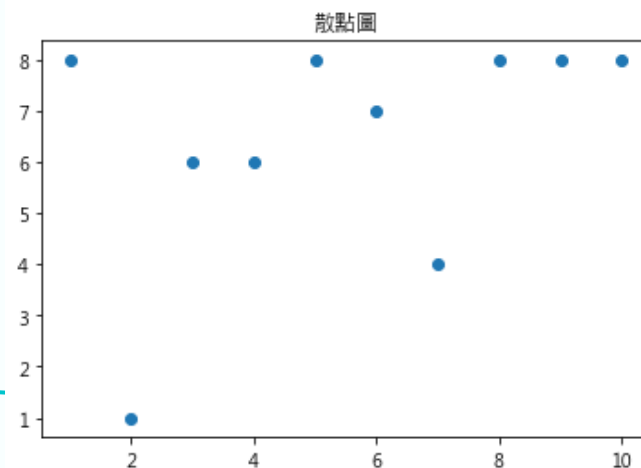
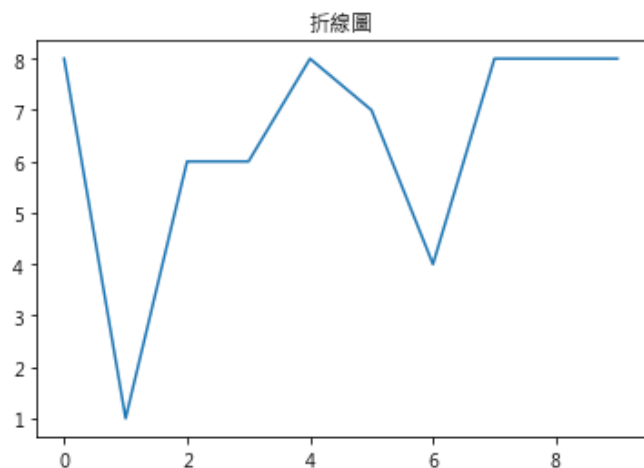
3	1	3	7	9	2	9	3	2	6	4	5	7	2	3	2	3	4	8
8	8	3	9	6	1	3	7	3	5	3	1	3	4	5	4	2	1	6
6	3	8	7	4	1	8	6	3	6	8	8	6	6	3	5	5	5	8
6	8	2	4	3	1	7	5	4	2	7	4	1	9	1	8	2	4	3
2	3	3	8	7	8	6	8	1	4	5	8	7	2	2	5	3	5	2
9	9	2	6	8	1	7	6	3	2	7	6	3	8	3	8	5	1	3
3	5	2	9	2	9	5	8	7	9	3	2	8	8	4	9	5	9	4
5	8	5	6	4	5	8	8	6	1	3	7	8	9	5	5	3	2	5
8	1	3	3	1	5	1	4	2	5	4	1	5	3	6	9	3	2	5
1	6	8	5	1	2	4	3	3	3	2	4	8	8	7	6	4	2	1

28個8

# 圖表類型

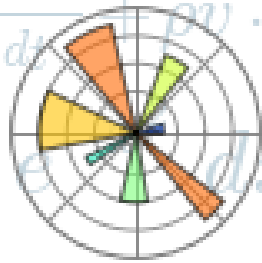
◆ 想要製作一份圖表，首先你要先思考以下的問題：

- ◆ 你擁有什麼樣的資料？
- ◆ 你想表達什麼訊息？
- ◆ 你應該採用什麼樣的視覺化方式？
- ◆ 你從圖表中能獲得什麼樣的資訊？



# matplotlib介紹

- ◆ matplotlib模組主要由python編寫，考慮平台相容特性部份使用C、objective-C和JavaScript編寫
- ◆ 由John D. Hunter開發
- ◆ matplotlib能繪製出2D及3D圖表，能繪製的圖表有折線圖、長條圖、直方圖、圓餅圖、散佈圖、箱線圖、座標圖...等



# matplotlib

# 使用及匯入matplotlib套件

- ◆ 使用matplotlib套件前需要先安裝，語法為：

```
pip install matplotlib
```

- ◆ matplotlib是一個很龐大的繪圖模組，使用時只要匯入其中的pyplot子模組功能即可完成許多圖表，語法為：

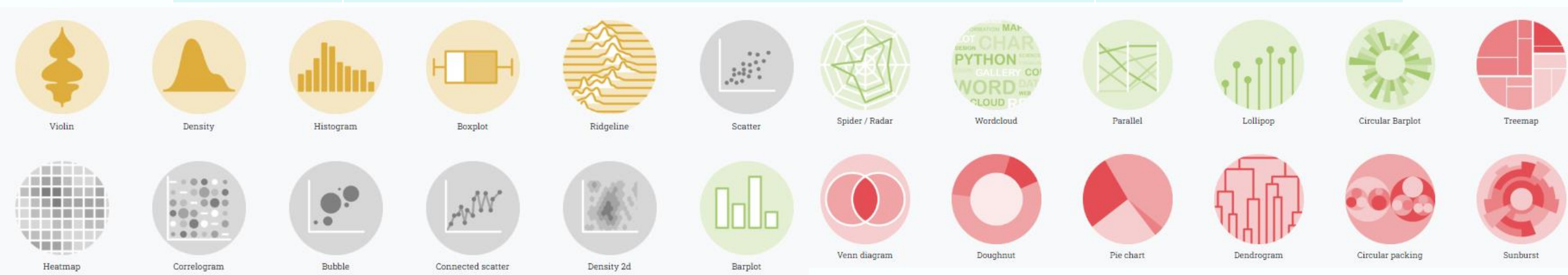
```
import matplotlib.pyplot as plt
```

- ◆ 若想知道matplotlib的版本，可以使用 matplotlib.\_\_version\_\_ 語法查詢

```
import matplotlib  
  
#查看matplotlib版本  
print(matplotlib.__version__)
```

# 常見圖表類型

函數	核心參數說明	圖表類型
plot()	x、y、color(線條顏色)、linestyle(線條類型)、linewidth(線條寬度)、marker(標記類型)、label(標籤)	折線圖
bar()	x、height(柱形高度)、width(柱形寬度)、align(柱形位置)、color(柱形顏色)	長條圖、堆疊長條圖
barh()	x、height(柱形高度)、width(柱形寬度)、align(柱形位置)、color(柱形顏色)	橫條圖、堆疊橫條圖
pie()	colors(填充顏色)、labels(標籤)	圓餅圖
scatter()	x、y、s(散點大小)、c(散點顏色)、label、marker(散點類型)	散點圖、氣泡圖



# 折線圖

- ◆ 使用plot()可以繪製折線圖，語法為：

```
plt.plot(*args [,選擇性參數1=值1, 選擇性參數2=值2, ...])
```

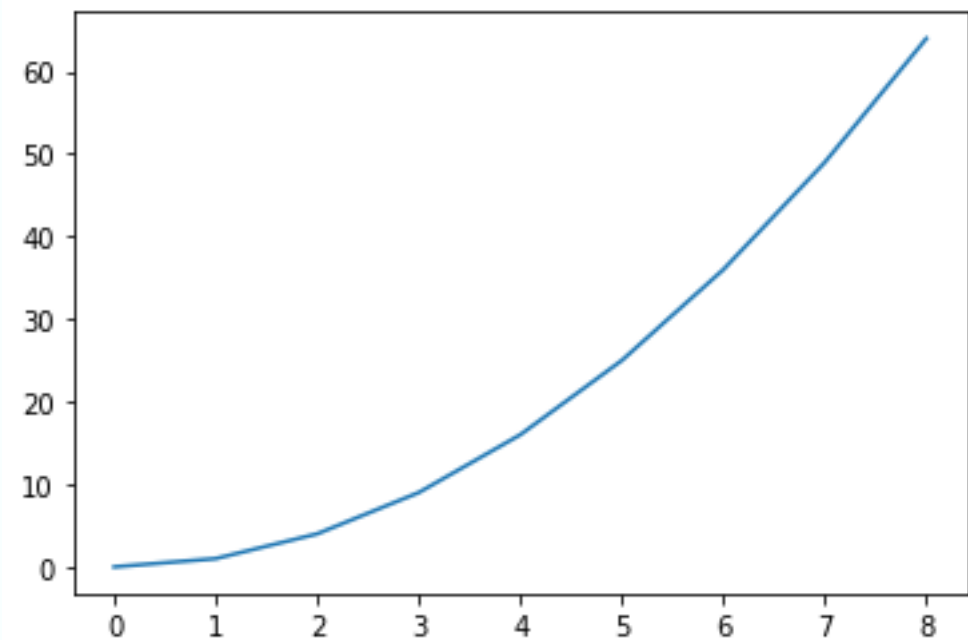
- ◆ 若x軸的數據是0、1、...n時，在使用plot()時可省略x軸不寫

```
import matplotlib.pyplot as plt

# 設定x軸與y軸
x = [0, 1, 2, 3, 4, 5, 6, 7, 8]
y = [0, 1, 4, 9, 16, 25, 36, 49, 64]

# 將資料帶入圖表
plt.plot(x, y)
# plt.plot(y)

# 顯示圖表
plt.show()
```



# 折線圖 – 線條寬度

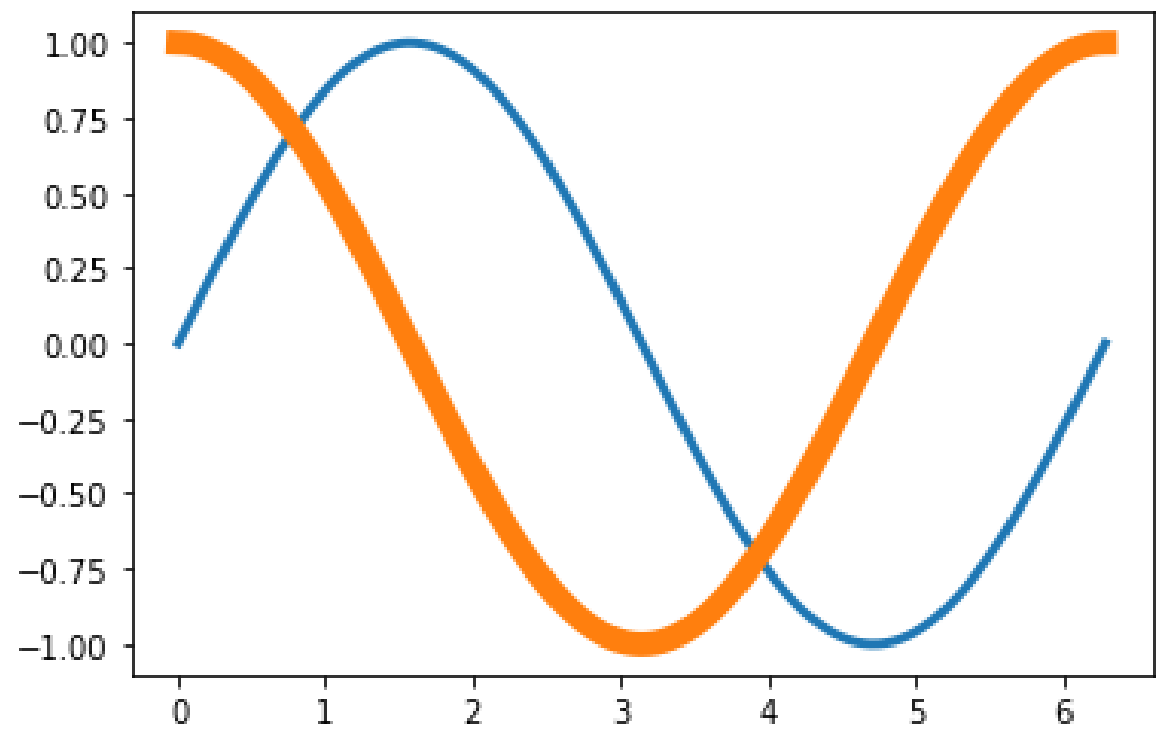
- ◆ 使用參數linewidth(簡寫lw)設定線條寬度

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

# 建立500個元素的陣列
x = np.linspace(0, 2*np.pi, 500)
y1 = np.sin(x)
y2 = np.cos(x)

# 將資料帶入圖表
plt.plot(x, y1, linewidth = 3)    # 線寬度為3
plt.plot(x, y2, lw = 8)          # 線寬度為8

# 顯示圖表
plt.show()
```





# 折線圖 – 線條色彩(一)

- ◆ 若想調整線條顏色，可以在`plot()`內增添`color`參數

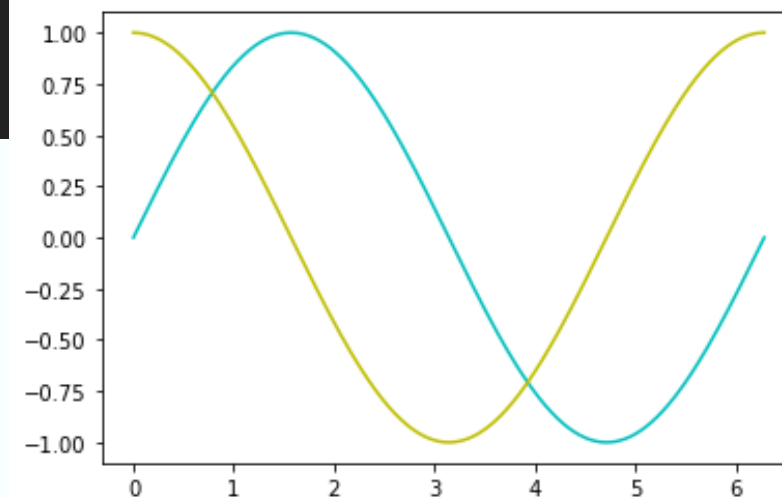
色彩字元	說明
“b”	blue(藍色)
“c”	cyan(青色)
“g”	green(綠色)
“k”	black(黑色)
“m”	magenta(紅色)
“r”	red(紅色)
“w”	white(白色)
“y”	yellow(黃色)

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

# 建立500個元素的陣列
x = np.linspace(0, 2*np.pi, 500)
y1 = np.sin(x)
y2 = np.cos(x)

# 將資料帶入圖表
plt.plot(x, y1, color="c") # 線條顏色為青色cyan
plt.plot(x, y2, color="y") # 線條顏色為黃色yellow

# 顯示圖表
plt.show()
```



# 折線圖 – 線條色彩(二)

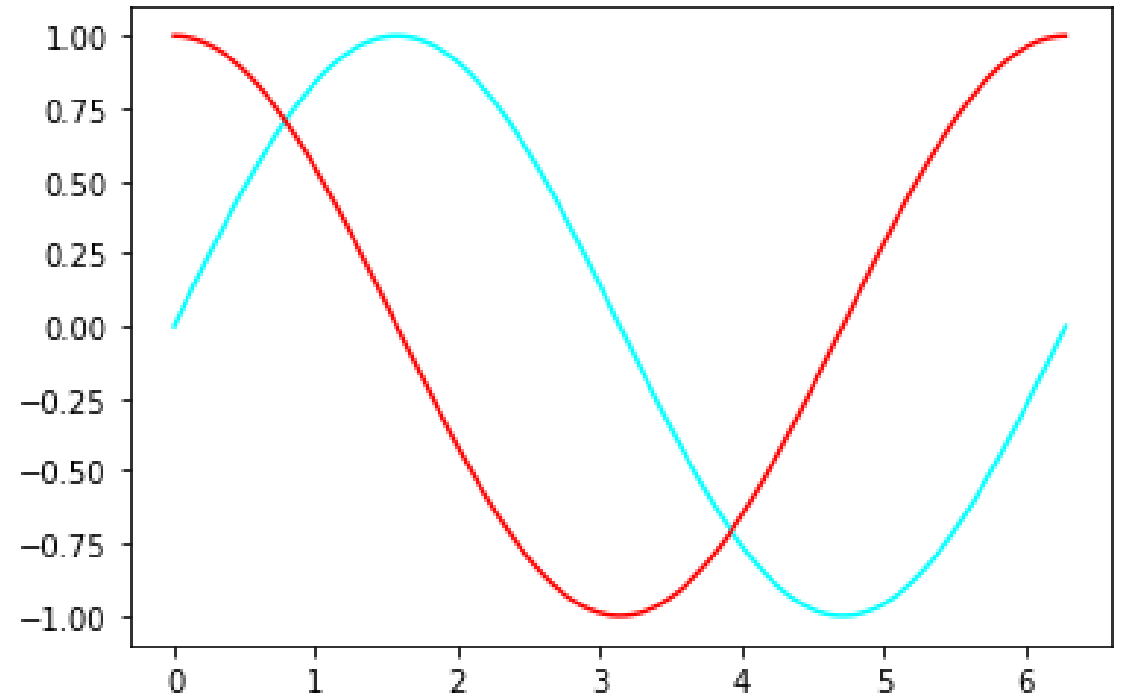
- ◆ color參數也可以使用色碼或css色彩表示

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

# 建立500個元素的陣列
x = np.linspace(0, 2*np.pi, 500)
y1 = np.sin(x)
y2 = np.cos(x)

# 將資料帶入圖表
plt.plot(x, y1, color="#00ffff")
plt.plot(x, y2, color=(255/255,0/255,0/255))

# 顯示圖表
plt.show()
```



# 折線圖 – 線條色彩(三)

## Base Colors



## Tableau Palette



## CSS Colors

black	bisque	forestgreen	slategrey
dimgray	darkorange	limegreen	lightsteelblue
dimgrey	burlywood	darkgreen	cornflowerblue
gray	antiquewhite	green	royalblue
grey	tan	lime	ghostwhite
darkgray	navajowhite	seagreen	lavender
darkgrey	blanchedalmond	mediumseagreen	midnightblue
silver	papayawhip	springgreen	navy
lightgray	moccasin	mintcream	darkblue
lightgrey	orange	mediumspringgreen	mediumblue
gainsboro	wheat	mediumaquamarine	blue
whitesmoke	oldlace	aquamarine	slateblue
white	floralwhite	turquoise	darkslateblue
snow	darkgoldenrod	lightseagreen	mediumslateblue
rosybrown	goldenrod	mediumturquoise	mediumpurple
lightcoral	cornsilk	azure	rebeccapurple
indianred	gold	lightcyan	blueviolet
brown	lemonchiffon	paleturquoise	indigo
firebrick	khaki	darkslategrey	darkorchid
maroon	palegoldenrod	darkslategrey	darkviolet
darkred	darkkhaki	teal	mediumorchid
red	ivory	darkcyan	thistle
mistyrose	beige	aqua	plum
salmon	lightyellow	cyan	violet
tomato	lightgoldenrodyellow	darkturquoise	purple
darksalmon	olive	cadetblue	darkmagenta
coral	yellow	powderblue	fuchsia
orangered	olivedrab	lightblue	magenta
lightsalmon	yellowgreen	deepskyblue	orchid
sienna	darkolivegreen	skyblue	mediumvioletred
seashell	greenyellow	lightskyblue	deeppink
chocolate	chartreuse	steelblue	hotpink
saddlebrown	lawngreen	aliceblue	lavenderblush
sandybrown	honeydew	dodgerblue	palevioletred
peachpuff	darkseagreen	lightslategray	crimson
peru	palegreen	lightslategray	pink
linen	lightgreen	slategray	lightpink

官網色彩說明一

官網色彩說明二

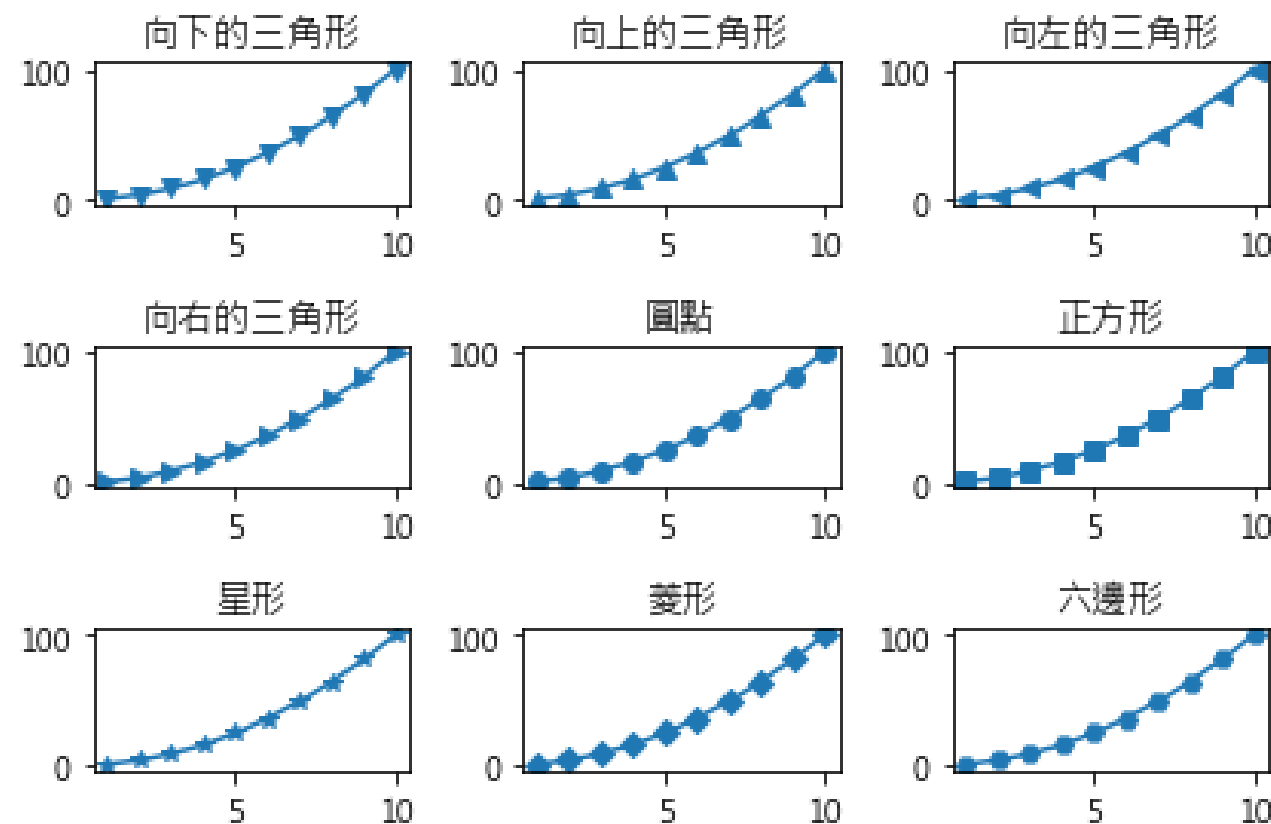
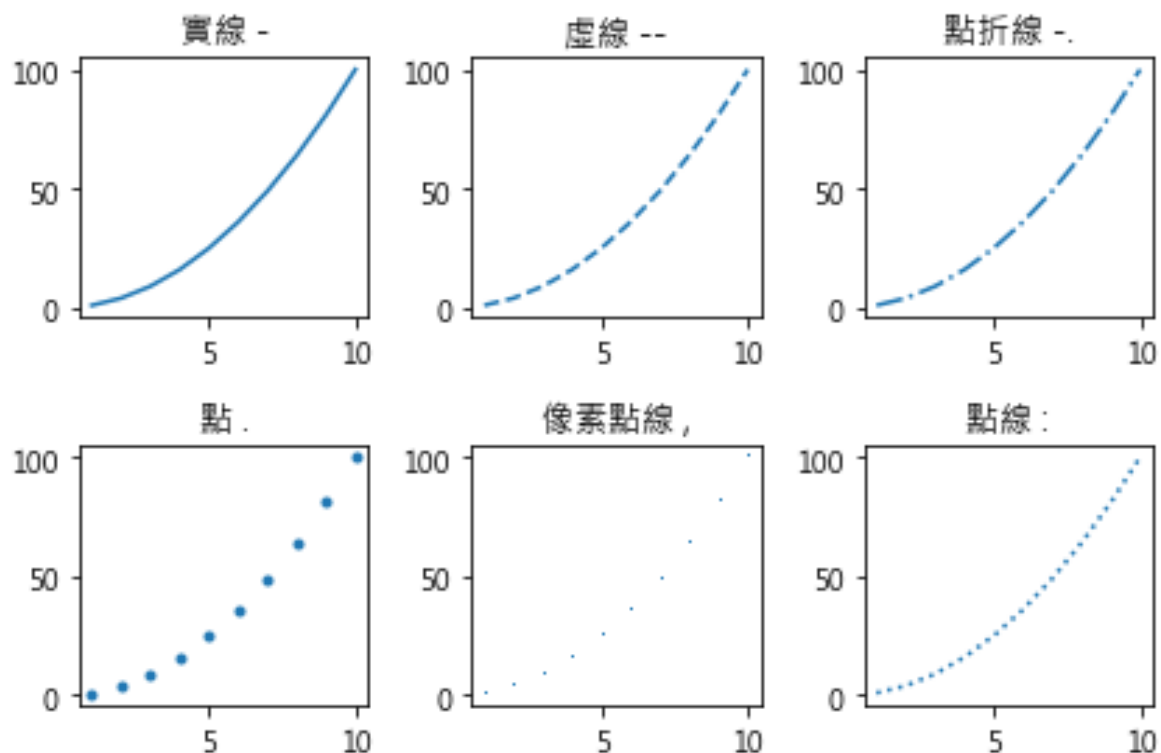
# 折線圖 – 線條樣式(一)

◆ 使用參數linestyle或ls可以設定線條樣式

線條字元	說明
“-”	實線(預設)
“- -”	虛線
“-.”	點折線
“.”	點
“, ”	像素點線
“-.”	點線
“v”	三角形向下標記
“^”	三角形向上標記
“<”	三角形向左
“>”	右三角形
“+”	加號

線條字元	說明
“1”	三叉標記向上
“2”	三叉標記向下
“3”	三叉標記向左
“4”	三叉標記向右
“s”	方形標記
“p”	五角標記
“*”	星星標記
“x”	叉標記
“D”	菱形標記
“d”	細菱形標記
“H”	六邊形標記

# 折線圖 – 線條樣式(二)

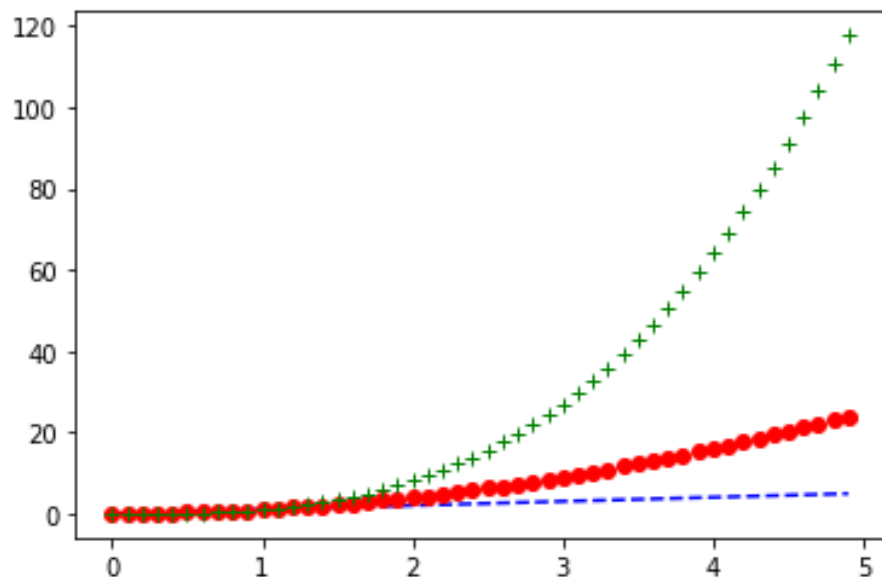


# 折線圖 – 線條樣式(三)

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

x = np.arange(0, 5, 0.1)

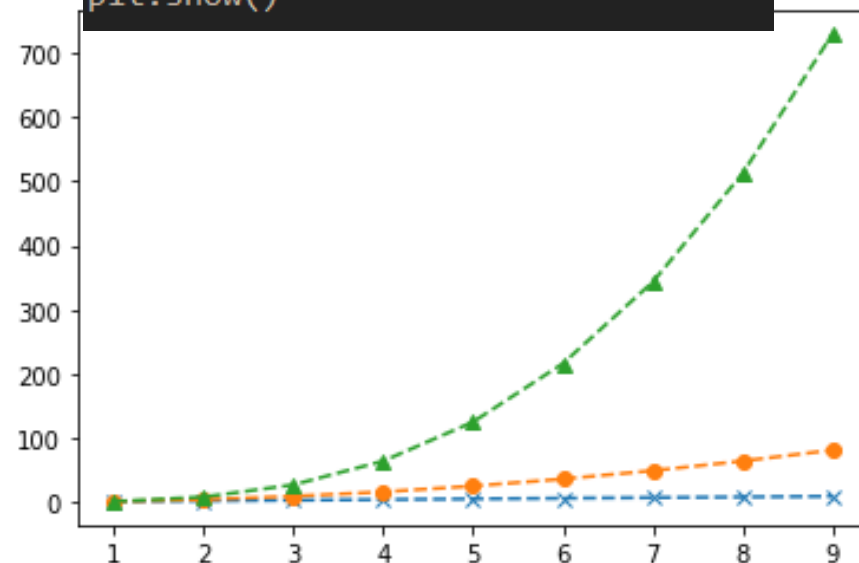
# 第一條線藍色虛線、第二條線紅色圓標記、第三條線綠色+
plt.plot(x, x, "b--", x, x**2, "ro", x, x**3, "g+")
plt.show()
```



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

x = np.arange(1, 10)

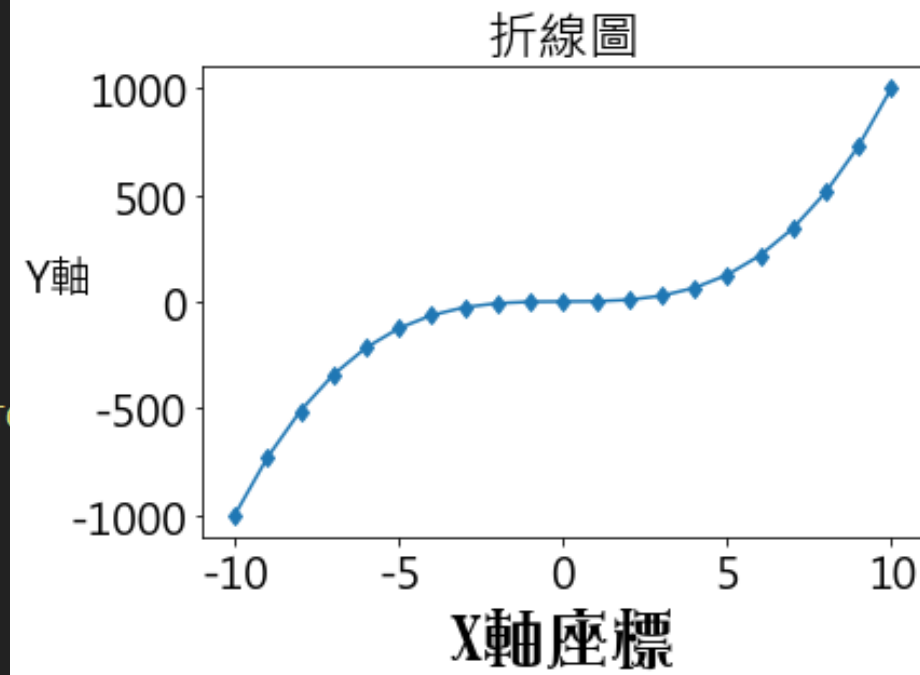
# 使用marker參數設定不同標記
plt.plot(x, x, "--", marker="x")
plt.plot(x, x**2, "--", marker="o")
plt.plot(x, x**3, "--", marker="^")
plt.show()
```



# 設定標題 – 中文字型

- ◆ 標題可分成圖表標題與x軸\y軸標題
- ◆ matplotlib不支援中文字型，若要使用中文字型需先設定

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import matplotlib.font_manager as fm
4
5 # 設定中文(影響整個程式)
6 plt.rcParams["font.family"] = "Microsoft JhengHei"
7 plt.rcParams["font.size"] = 20
8 plt.rcParams["axes.unicode_minus"] = False # 讓正負號能正常顯示
9
10 # 指定位置的中文設定
11 font_path = r"C:\Users\selp\AppData\Local\Microsoft\Windows\Fonts\華康飾藝體W5.TT"
12 font_prop = fm.FontProperties(fname=font_path)
13
14 x = np.arange(-10,11)
15
16 plt.plot(x, x**3, marker="d")
17 plt.title("折線圖")
18 plt.xlabel("X軸座標", fontproperties=font_prop, size=30)
19 plt.ylabel("Y軸", rotation=0)
20 plt.show()
```

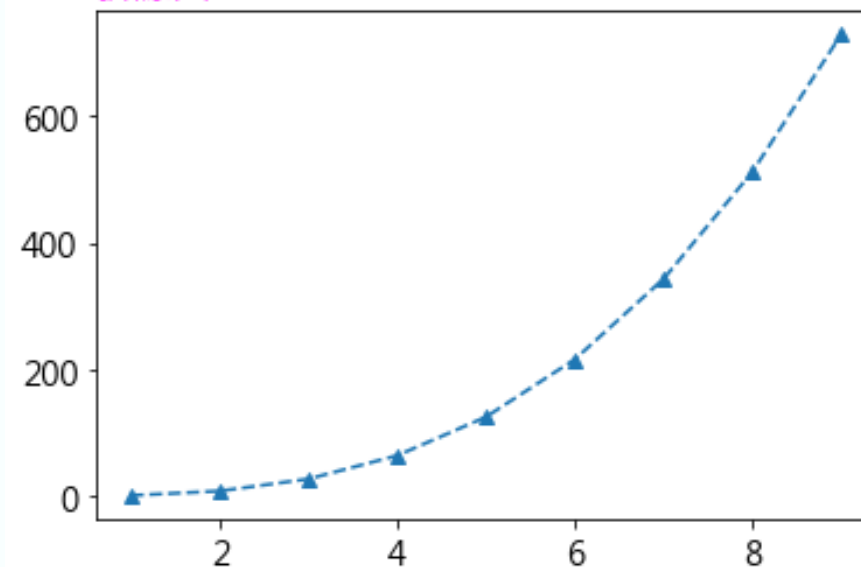


# 設定標題 – 可設定的參數

參數	說明
fontsize	字型大小
fontstyle	設定標題是否傾斜，參數有normal、italic、oblique
loc	設定標題是center(置中)、left(靠左)、right(靠右)，預設為置中
color	設定文字顏色

```
1 import matplotlib.pyplot as plt
2 import matplotlib.font_manager as fm
3 import numpy as np
4
5 # 設定中文(影響整個程式)
6 plt.rcParams["font.family"] = "Microsoft JhengHei"
7 plt.rcParams["font.size"] = 14
8
9 x = np.arange(1, 10)
10 plt.plot(x, x**3, "--", marker="^")
11 plt.title("折線圖", loc="left", color="#ff00ff")
12 plt.show()
```

折線圖





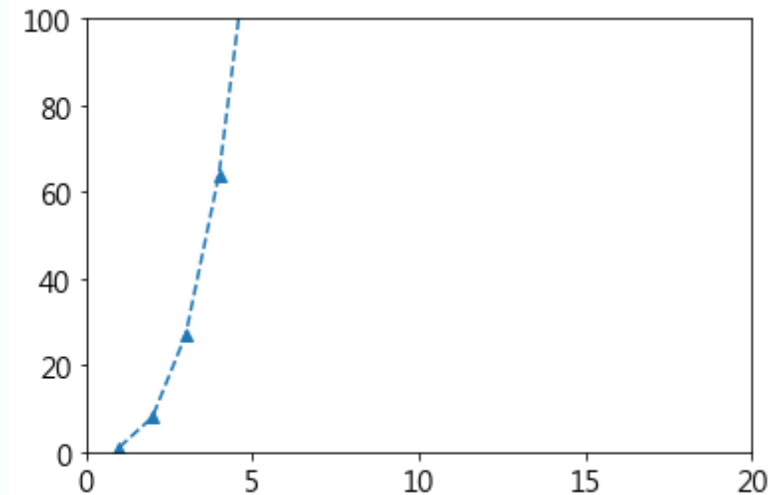
# 設定x軸與y軸的刻度

- ◆ `axis()`可以設定x軸和y軸的範圍，語法為：

```
plt.axis([xmin, xmax, ymin, ymax])
```

- ◆ `xmin`是x軸最小刻度，`xmax`是x軸最大刻度
- ◆ `ymin`是y軸最小刻度，`ymax`是y軸最大刻度

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.arange(1, 10)
5 plt.plot(x, x**3, "--", marker="^")
6 plt.axis([0,20,0,100])
7 plt.show()
```



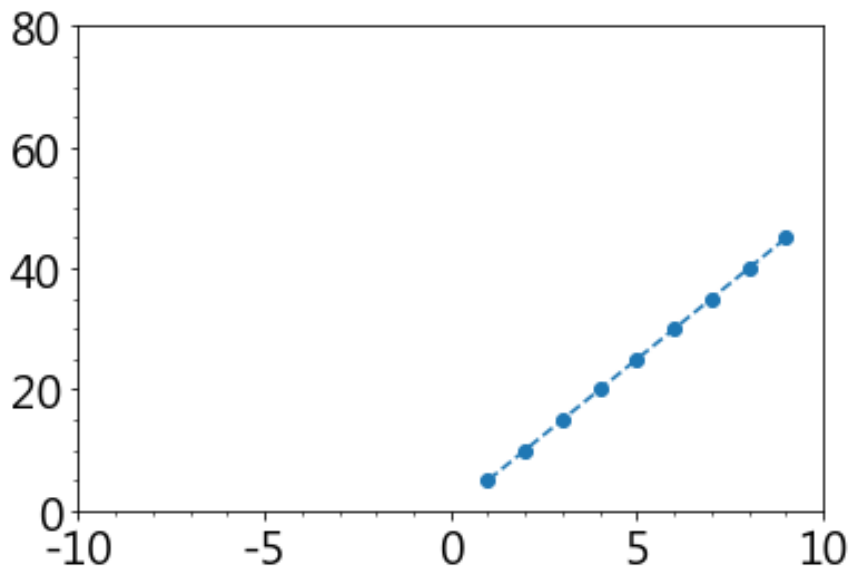
# 設定x軸與y軸的刻度

◆ 使用xlim()、ylim()分別設定x軸與y軸的範圍

◆ xlim(起始值，終止值)

◆ ylim(起始值，終止值)

◆ 次刻度 minorticks\_on()

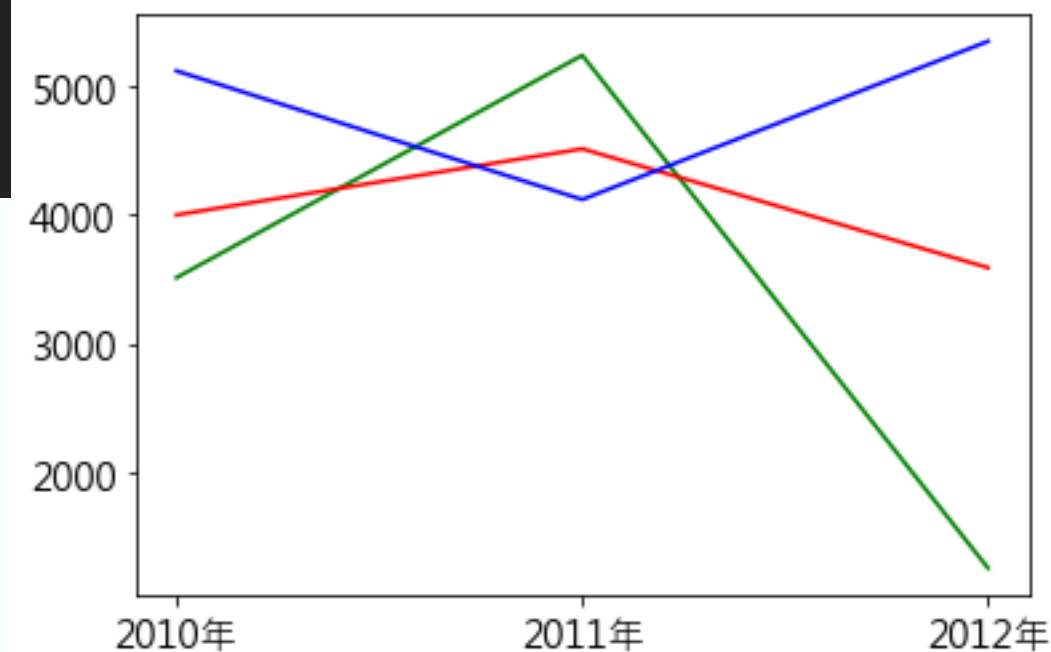


```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 x = np.arange(1,10)
5 plt.plot(x, x*5, "--", marker="o")
6
7 # 將x軸範圍設為-10~10
8 plt.xlim(-10,10)
9 # 將y軸範圍設為0~80
10 plt.ylim(0,80)
11 # 設定次刻度
12 plt.minorticks_on()
13
14 plt.show()
```

# 設定x軸與y軸的標籤

- ◆ 使用xticks()、yticks()可以設定x軸與y軸的標籤

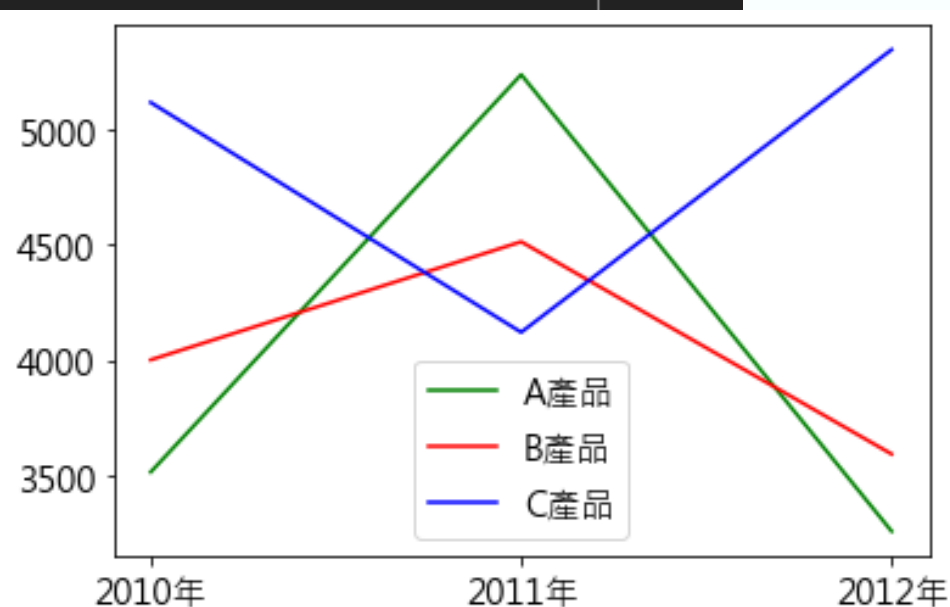
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 data = {"2010年": [3512, 5241, 1254], "2011年": [4000, 4514, 3590], "2012年": [5120, 4120, 5350]}
5 x = [1, 2, 3]
6 y1, y2, y3 = data["2010年"], data["2011年"], data["2012年"]
7 # 設定x軸刻度
8 labels = ["2010年", "2011年", "2012年"]
9 plt.xticks(x, labels)
10 # 將資料帶入圖表
11 plt.plot(x, y1, "g", x, y2, "r", x, y3, "b")
12 plt.show()
```



# 圖例

- ◆ 若想在圖表上增加圖例，必需在`plot()`函數內增加`label`標籤設定

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 data = {"2010年": [3512, 5241, 3254], "2011年": [4000, 4514, 3590], "2012年": [5120, 4120, 5350]}
5 x = [1, 2, 3]
6 y1, y2, y3 = data["2010年"], data["2011年"], data["2012年"]
7 # 設定x軸刻度
8 labels = ["2010年", "2011年", "2012年"]
9 plt.xticks(x, labels)
10 # 將資料帶入圖表
11 plt.plot(x, y1, "g", label="A產品")
12 plt.plot(x, y2, "r", label="B產品")
13 plt.plot(x, y3, "b", label="C產品")
14 # 設定圖例
15 plt.legend()
16 plt.show()
```



# 圖例

◆ 圖例位置可以修改，語法為`plt.legend(loc=數值)`

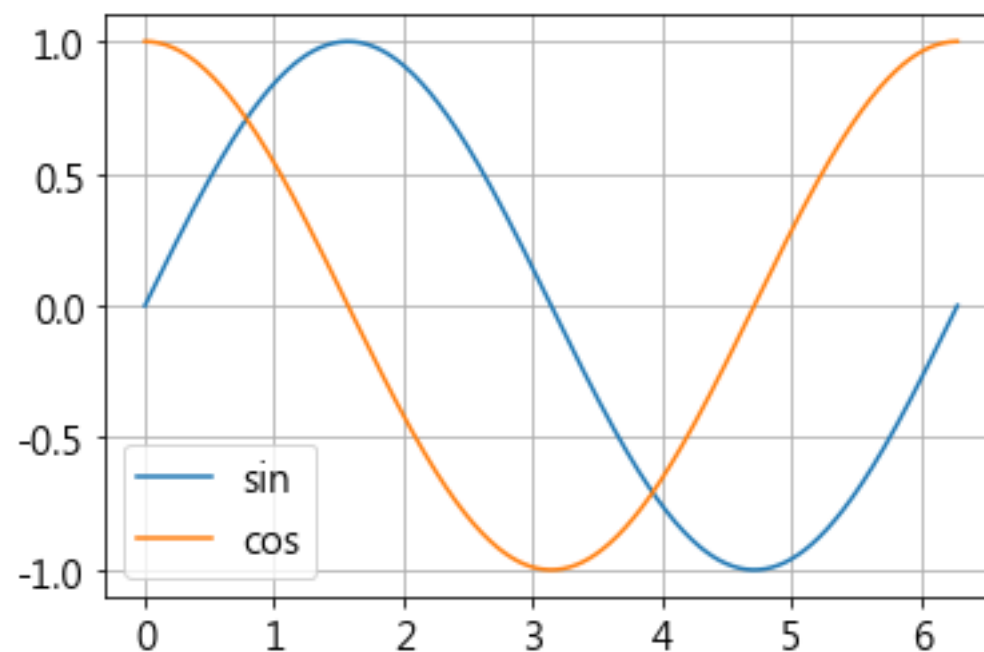
loc編號	說明
0	best
1	upper right
2	upper left
3	lower left
4	lower right
5	right
6	center left
7	center right
8	lower center
9	upper center
10	center

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 data = {"2010年": [3512, 5241, 3254], "2011年": [4000, 4514, 3590], "2012年": [5120, 4120, 5350]}
5 x = [1, 2, 3]
6 y1, y2, y3 = data["2010年"], data["2011年"], data["2012年"]
7 # 設定x軸刻度
8 labels = ["2010年", "2011年", "2012年"]
9 plt.xticks(x, labels)
10 # 將資料帶入圖表
11 plt.plot(x, y1, "g", label="A產品")
12 plt.plot(x, y2, "r", label="B產品")
13 plt.plot(x, y3, "b", label="C產品")
14 # 設定圖例
15 plt.legend(loc=5)
16 plt.show()
```

# 網格

- ◆ 圖表預設是不顯示網格，但若想讓圖表更容易閱讀，可以使用 `grid()` 函數顯示網格

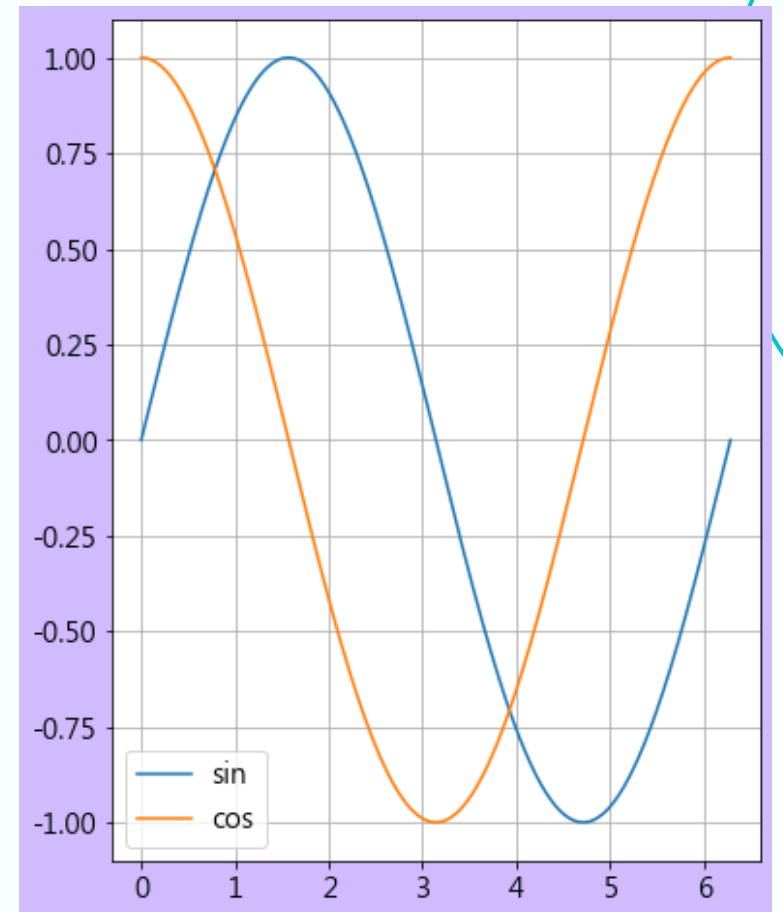
```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.linspace(0, 2*np.pi, 500)
5 y1 = np.sin(x)
6 y2 = np.cos(x)
7 plt.plot(x, y1, label="sin")
8 plt.plot(x, y2, label="cos")
9 # 設定圖例
10 plt.legend()
11 # 設定格線
12 plt.grid()
13 plt.show()
```



# 設定畫布

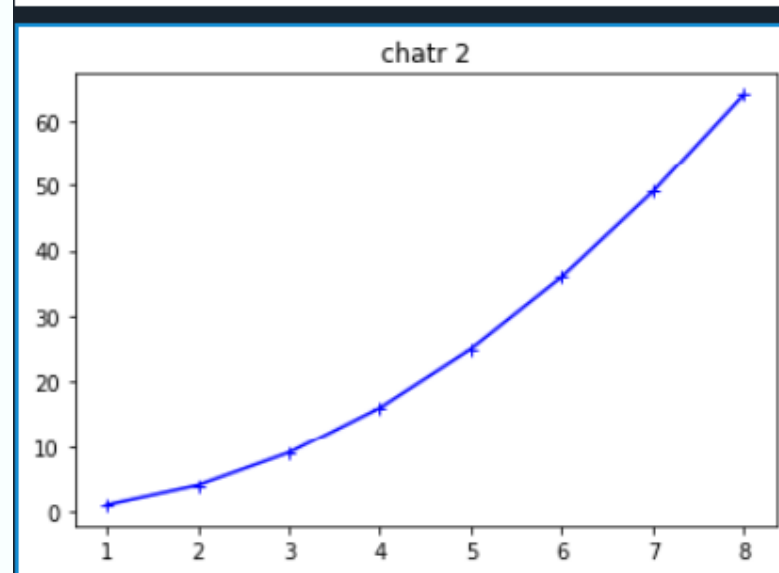
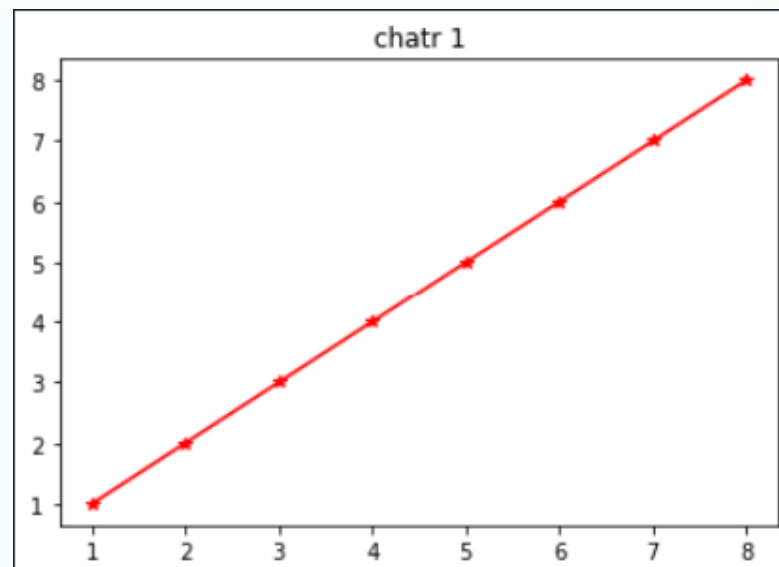
- ◆ 可以使用figure()函數建立畫布
- ◆ figure()函數一定要在plot()函數前先設定

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = np.linspace(0, 2*np.pi, 500)
5 y1 = np.sin(x)
6 y2 = np.cos(x)
7
8 # 建立畫布，大小為6x8，背景色為#D1BBFF
9 plt.figure(figsize=(6,8), facecolor="#D1BBFF")
10 plt.plot(x, y1, label="sin")
11 plt.plot(x, y2, label="cos")
12 # 設定圖例
13 plt.legend()
14 # 設定格線
15 plt.grid()
16 plt.show()
```



# 建立多個畫布

```
1  import matplotlib.pyplot as plt
2
3  y1 = x = [i for i in range(1,9)]
4  y2 = [i**2 for i in range(1,9)]
5
6  # 建立第一個圖表
7  plt.figure(1)
8  plt.plot(x, y1, "r-*")
9  plt.title("chatr 1")
10 # 建立第二個圖表
11 plt.figure(2)
12 plt.plot(x, y2, "b-+")
13 plt.title("chatr 2")
14 plt.show()
```





# 繪製多個圖表 – 子圖表

- ◆ 函數subplot()可以在視窗圖表(figure)內建立子圖表(axes)，語法為：  
`plt.subplot(*args, **kwargs)`
- ◆ \*args的參數為(nrows, ncols, index)，三個參數皆為整數，預設為(1,1,1)

範例2

subplot(1,2,1)

subplot(1,2,2)

範例1

subplot(2,1,1)

subplot(2,1,2)

範例3

subplot(2,3,1)

subplot(2,3,2)

subplot(2,3,3)

subplot(2,3,4)

subplot(2,3,5)

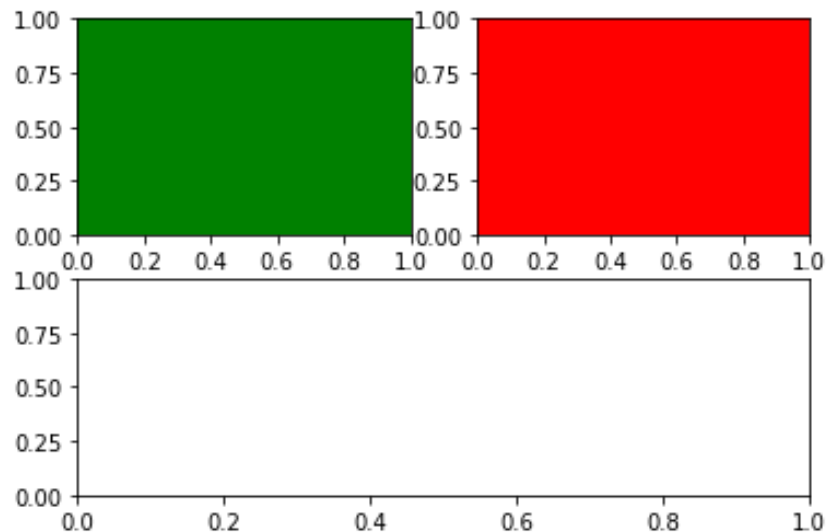
subplot(2,3,6)

# 子圖表 – subplot()

## ◆ plt.subplot(列數, 欄數, 子圖序號)

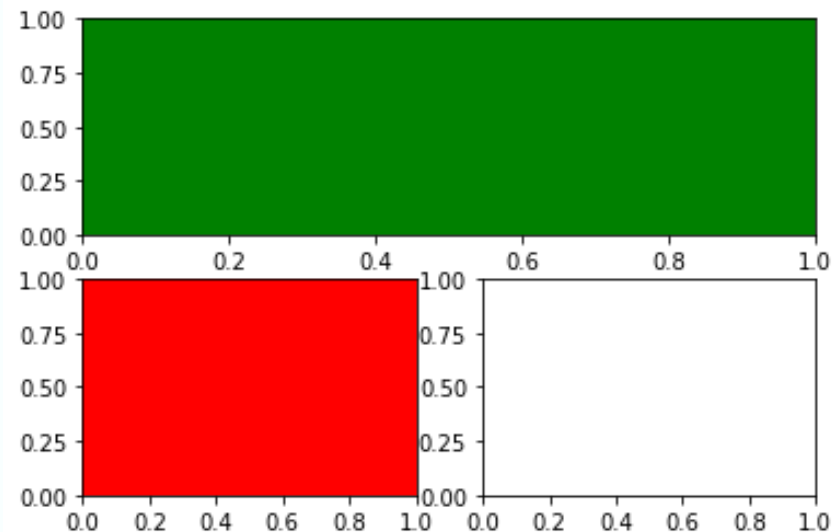
```
import matplotlib.pyplot as plt

plt.figure()
plt.subplot(2,2,1, facecolor="g")
plt.subplot(2,2,2, facecolor="r")
plt.subplot(2,1,2)
plt.show()
```



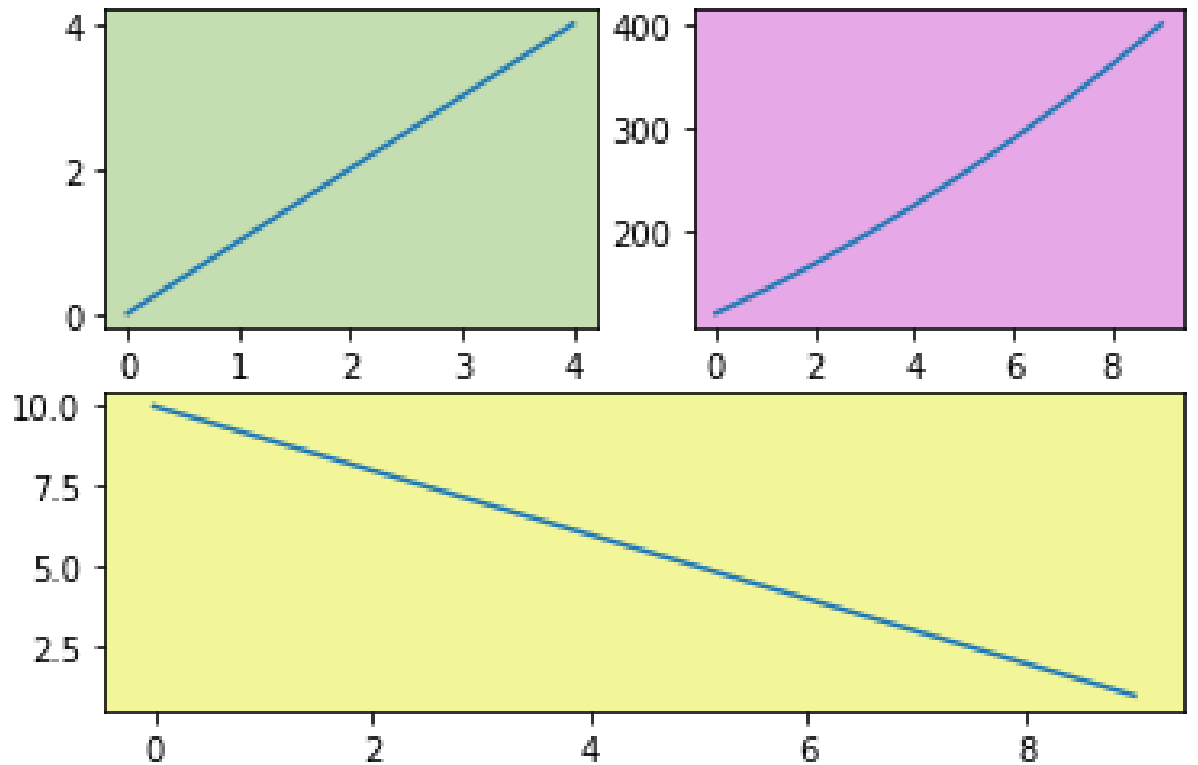
```
import matplotlib.pyplot as plt

plt.figure()
plt.subplot(2,1,1, facecolor="g")
plt.subplot(2,2,3, facecolor="r")
plt.subplot(2,2,4)
plt.show()
```



# 子圖表 – subplot()

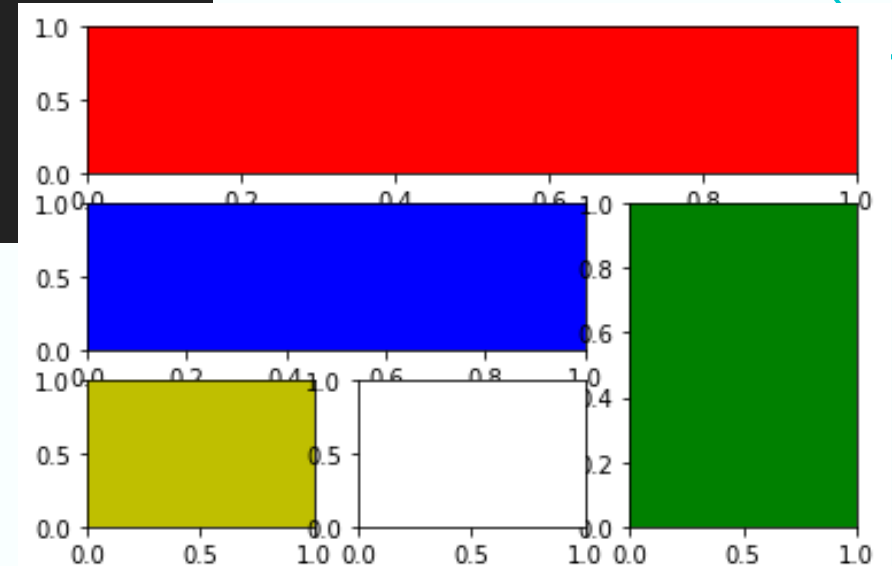
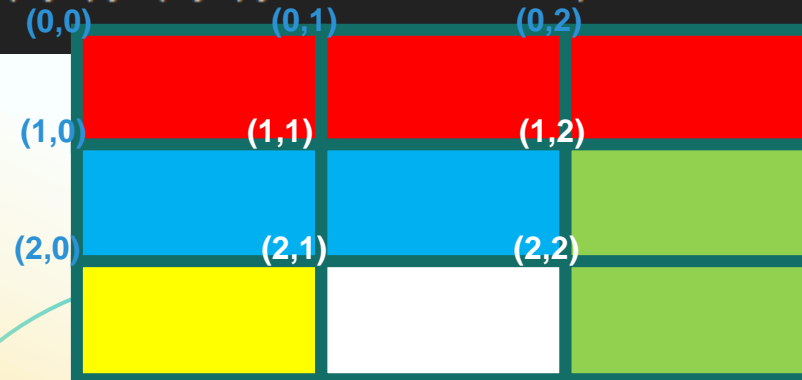
```
1  import matplotlib.pyplot as plt
2
3  x = [x for x in range(5)]
4  y = [y**2 for y in range(11,21)]
5  z = [z for z in range(10,0,-1)]
6
7  plt.figure()
8  plt.subplot(2,2,1,facecolor="#C4DDB1")
9  plt.plot(x)
10 plt.subplot(2,2,2,facecolor="#E6A8E6")
11 plt.plot(y)
12 plt.subplot(2,1,2,facecolor="#F3F599")
13 plt.plot(z)
14 plt.show()
```



# 子圖表 – subplot2grid()

## ◆ plt.subplot2grid(表格數, 位置, 列, 欄)

```
1 import matplotlib.pyplot as plt
2
3 plt.figure()
4 # 總共三列三欄，從第1列第1欄開始，大小為1列3欄
5 plt.subplot2grid((3,3), (0,0), rowspan=1, colspan=3, facecolor="r")
6 # 總共三列三欄，從第2列第1欄開始，大小為2列1欄
7 plt.subplot2grid((3,3), (1,0), rowspan=1, colspan=2, facecolor="b")
8 # 總共三列三欄，從第2列第2欄開始，大小為2列1欄
9 plt.subplot2grid((3,3), (1,2), rowspan=2, colspan=1, facecolor="g")
10 # 總共三列三欄，從第3列第1欄開始
11 plt.subplot2grid((3,3), (2,0), facecolor="y")
12 # 總共三列三欄，從第3列第2欄開始
13 plt.subplot2grid((3,3), (2,1), facecolor="w")
14 plt.show()
```



# 儲存圖表

- ◆ 使用savefig()函數可以儲存圖表，語法為：

```
plt.savefig(檔名, [dpi=120, facecolor="w", edgecolor="w"])
```

- ◆ dpi：解析度，預設為80
- ◆ facecolor：圖表表面的顏色
- ◆ edgecolor：圖表邊緣的顏色

```
1 import matplotlib.pyplot as plt
2 import matplotlib.font_manager as fm
3 import numpy as np
4
5 x = np.arange(1, 10)
6 plt.plot(x, x**3, "--", marker="^")
7 # 儲存函數要在show()之前先寫，支援jpg/png/pdf三種
8 plt.savefig("123.jpg")
9 plt.show()
```

# 圖表常用元素

函數	核心參數說明	功能
figure()	figsize(圖表尺寸)、dpi(解析度)	設定圖表的大小與解析度
title()	str(圖名)、fontdict(文字格式)	設定標題
xlabel()、ylabel()	xlabel(x軸名)、ylabel(y軸名)	設定x軸與y軸的標題
axis()、xlim()、ylim()		設定x軸與y軸的範圍
xticks()、yticks()	ticks(刻度數值)、labels(刻度名稱)	設定x軸與y軸的刻度
grid()	b(有無格線)、axis(x軸和y軸格線)、color(格線顏色)、alpha(透明度)	設定x軸與y軸的主要和次要格線
legend()	loc(位置)	控制圖例顯示

# 長條圖

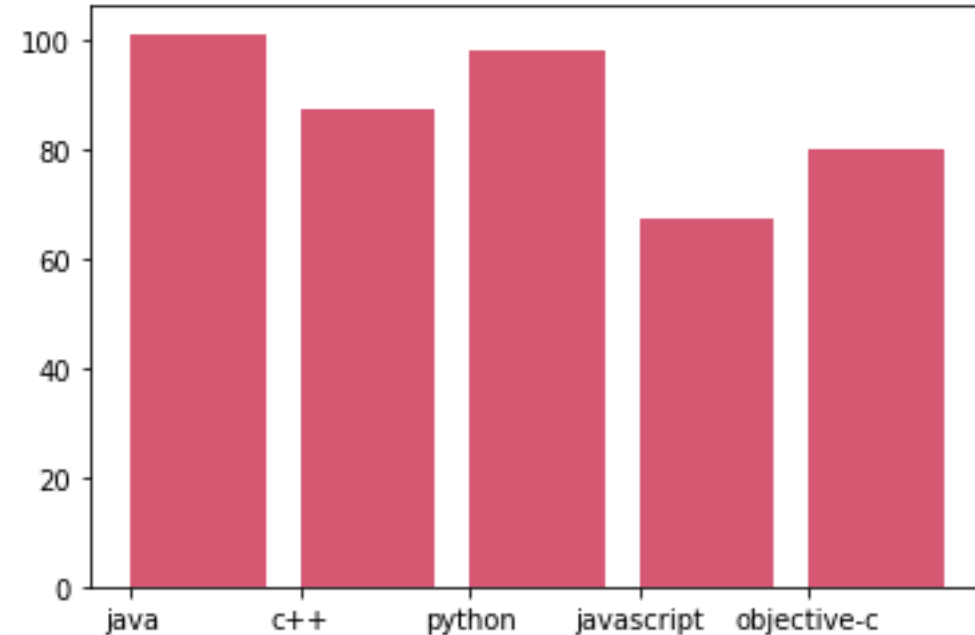
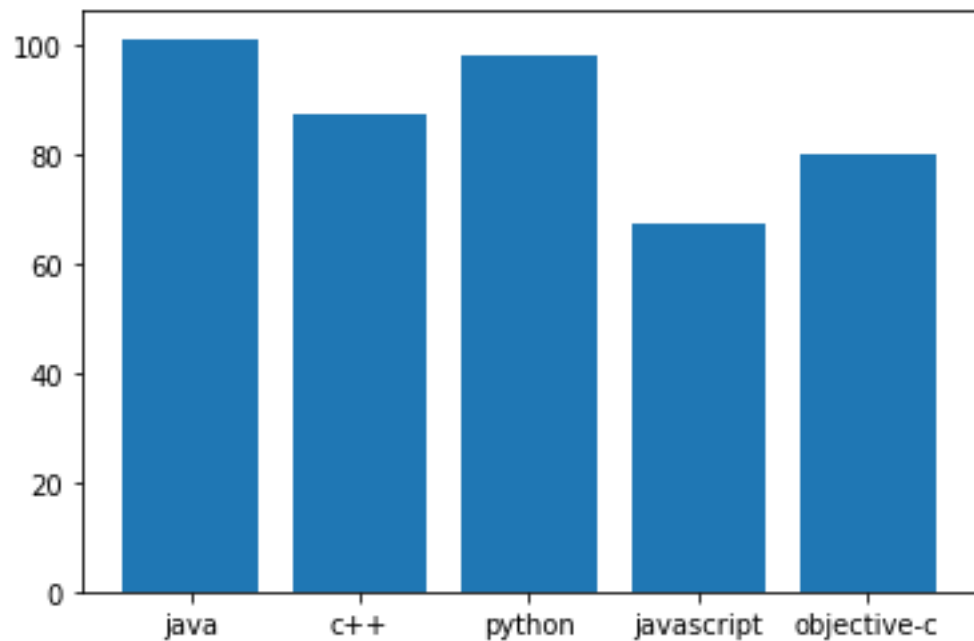
- ◆ 長條圖是統計常用的圖表，長條的高度和數據的多寡成正比，預設是垂直顯示，也可以改為水平顯示，語法為：

```
plt.bar(x, height, width, bottom, align, **kwargs)
```

參數	說明
x	x座標的序列值
height	y座標的序列值，也是長條高度
width	長條寬度
bottom	長條底部座標
align	標籤對齊方式，有center、edge，預設為center
color	長條顏色
label	長條標籤
linewidth	長條邊框寬度
hatch	長條內部造型，例如"/"、"\"," "、"o"、"+"

# 長條圖

```
1 import matplotlib.pyplot as plt
2
3 # 設定資料
4 name = ["java", "c++", "python", "javascript", "objective-c"]
5 student = [101, 87, 98, 67, 80]
6 # 將資料帶入圖表，標籤靠左，長條顏色為粉紅
7 plt.bar(name, student, align="edge", color="#D55772")
8 # 顯示圖表
9 plt.show()
```

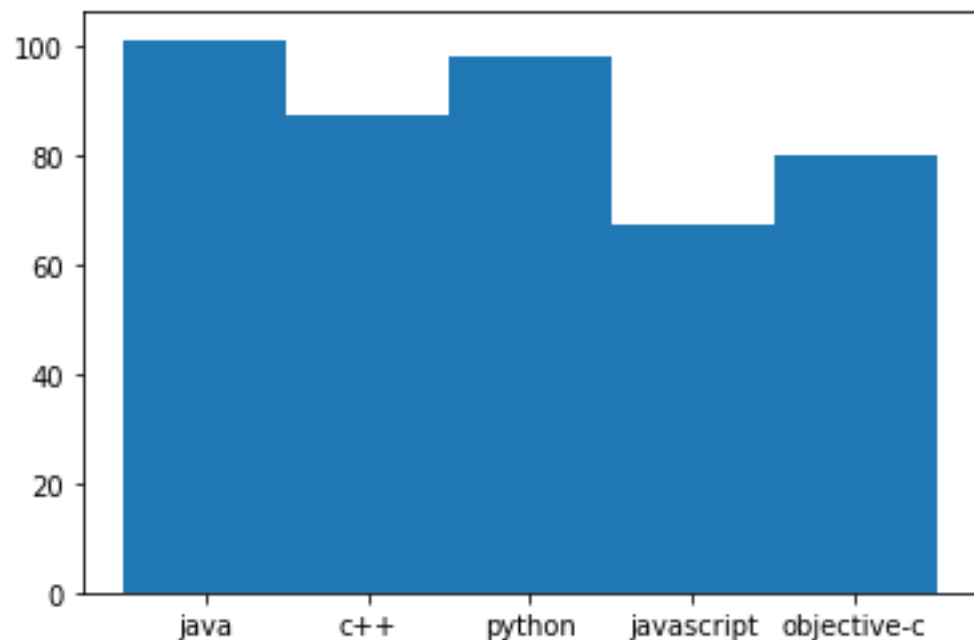


`plt.bar(name, student)`

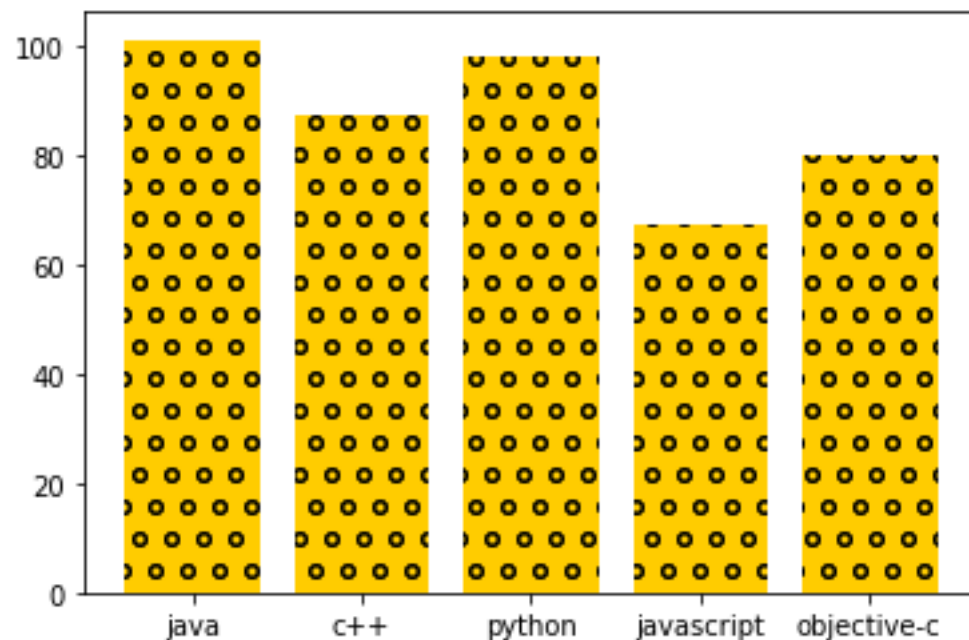


# 長條圖

```
1 import matplotlib.pyplot as plt
2
3 # 設定資料
4 name = ["java", "c++", "python", "javascript", "objective-c"]
5 student = [101, 87, 98, 67, 80]
6 # 將資料帶入圖表，長條寬度為1(預設為0.8)
7 plt.bar(name, student, width=1.0)
8 # 顯示圖表
9 plt.show()
```

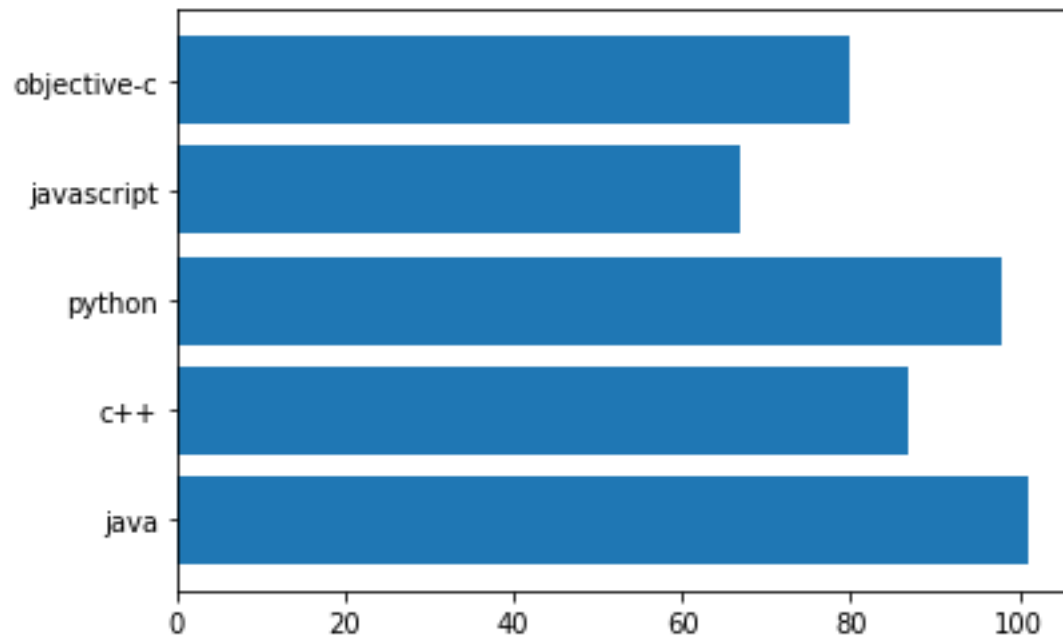


```
1 import matplotlib.pyplot as plt
2
3 # 設定資料
4 name = ["java", "c++", "python", "javascript", "objective-c"]
5 student = [101, 87, 98, 67, 80]
6 # 將資料帶入圖表，長條顏色為黃、內部造型為圓圈
7 plt.bar(name, student, hatch="o", color="#FFCC00")
8 # 顯示圖表
9 plt.show()
```



# 橫長條圖

```
1 import matplotlib.pyplot as plt
2
3 # 設定資料
4 name = ["java", "c++", "python", "javascript", "objective-c"]
5 student = [101, 87, 98, 67, 80]
6 # 將資料帶入圖表
7 plt.barh(name, student)
8 # 顯示圖表
9 plt.show()
```



# 圓餅圖

- ◆ 圓餅圖是一種統計圖表，使用`pie()`函數，圓餅圖可使用百分比描述數據之間的相對關係，語法為：

```
plt.pie(x, [選擇性參數1=值1,選擇性參數2=值2, ...])
```

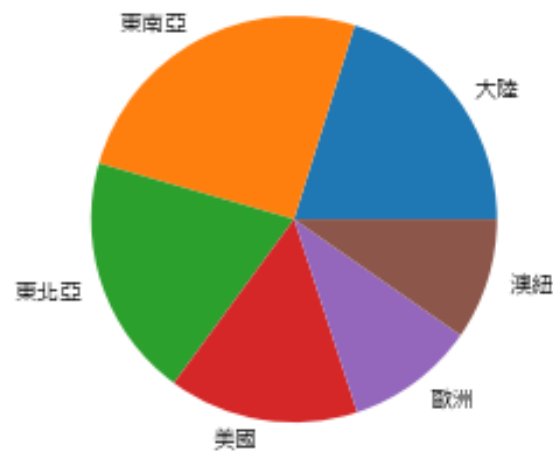
- ◆ `explode`：設定圓餅圖分離，數值越大分離越遠，預設是0
- ◆ `labels`：圓餅標籤
- ◆ `colors`：圓餅顏色
- ◆ `autopct`：項目的百分比格式，語法為“%格式%%”
- ◆ `pctdistance`：圓餅標籤與圓心之間距離的比率，預設0.6

# 圓餅圖 – 百分比

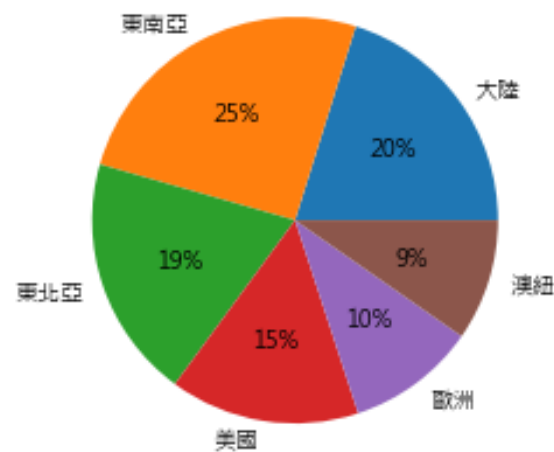
```
1 import matplotlib.pyplot as plt
2 # 設定字型
3 plt.rcParams["font.family"] = ["Microsoft JhengHei"]
4 # 設定資料
5 area = ['大陸', '東南亞', '東北亞', '美國', '歐洲', '澳紐']
6 people = [10000, 12600, 9600, 7500, 5100, 4800]
7 # 將資料代入圓餅圖內
8 plt.pie(people, labels=area)
9 # 設定標題
10 plt.title('五月份國外旅遊調查表', fontsize=16)
11 # 顯示圖表
12 plt.show()
```

```
1 import matplotlib.pyplot as plt
2 # 設定字型
3 plt.rcParams["font.family"] = ["Microsoft JhengHei"]
4 # 設定資料
5 area = ['大陸', '東南亞', '東北亞', '美國', '歐洲', '澳紐']
6 people = [10000, 12600, 9600, 7500, 5100, 4800]
7 # 將資料代入圓餅圖內，並顯示百分比
8 plt.pie(people, labels=area, autopct="%d%")
9 # 設定標題
10 plt.title('五月份國外旅遊調查表', fontsize=16)
11 # 顯示圖表
12 plt.show()
```

五月份國外旅遊調查表



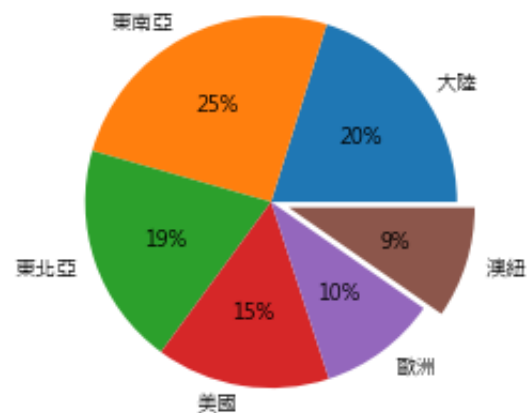
五月份國外旅遊調查表



# 圓餅圖 – 數據分離與陰影

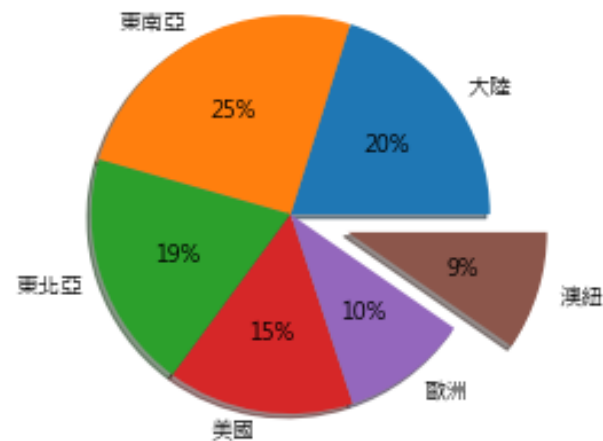
```
1 import matplotlib.pyplot as plt
2 # 設定字型
3 plt.rcParams["font.family"] = ["Microsoft JhengHei"]
4 # 設定資料
5 area = ['大陸', '東南亞', '東北亞', '美國', '歐洲', '澳紐']
6 people = [10000, 12600, 9600, 7500, 5100, 4800]
7 exp = [0.0, 0.0, 0.0, 0.0, 0.0, 0.1]
8 # 將資料代入圓餅圖內，設定數據分離(explode參數)
9 plt.pie(people, labels=area, autopct="%d%", explode=exp)
10 # 設定標題
11 plt.title('五月份國外旅遊調查表', fontsize=16)
12 # 顯示圖表
13 plt.show()
```

五月份國外旅遊調查表



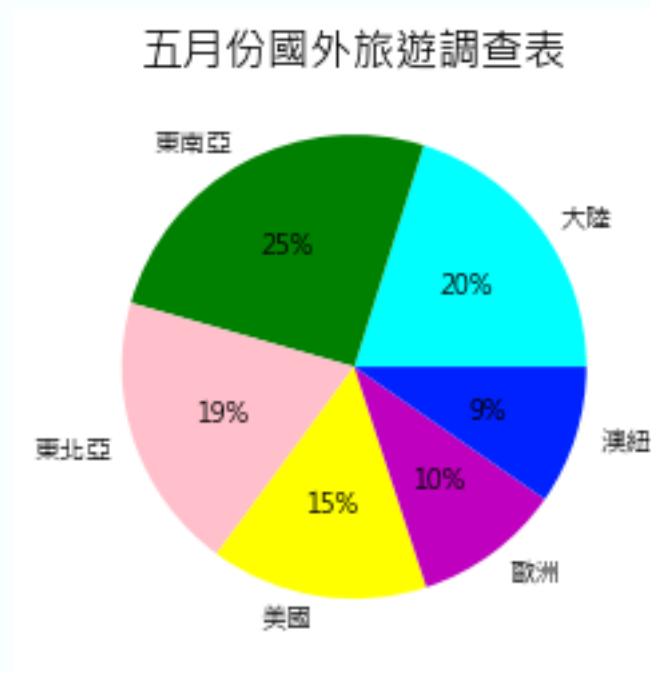
```
1 import matplotlib.pyplot as plt
2 # 設定字型
3 plt.rcParams["font.family"] = ["Microsoft JhengHei"]
4 # 設定資料
5 area = ['大陸', '東南亞', '東北亞', '美國', '歐洲', '澳紐']
6 people = [10000, 12600, 9600, 7500, 5100, 4800]
7 exp = [0.0, 0.0, 0.0, 0.0, 0.0, 0.3]
8 # 將資料代入圓餅圖內，設定數據分離(explode參數)與圓餅陰影(shadow參數)
9 plt.pie(people, labels=area, autopct="%d%", explode=exp, shadow=True)
10 # 設定標題
11 plt.title('五月份國外旅遊調查表', fontsize=16)
12 # 顯示圖表
13 plt.show()
```

五月份國外旅遊調查表



# 圓餅圖 – 圓餅顏色

```
1 import matplotlib.pyplot as plt
2 # 設定字型
3 plt.rcParams["font.family"] = ["Microsoft JhengHei"]
4 # 設定資料
5 area = ['大陸', '東南亞', '東北亞', '美國', '歐洲', '澳紐']
6 people = [10000, 12600, 9600, 7500, 5100, 4800]
7 color = ['aqua', 'g', 'pink', 'yellow', 'm', '#0022ff']
8 # 將資料代入圓餅圖內，自訂圓餅顏色
9 plt.pie(people, labels=area, autopct="%d%", colors=color)
10 # 設定標題
11 plt.title('五月份國外旅遊調查表', fontsize=16)
12 # 顯示圖表
13 plt.show()
```



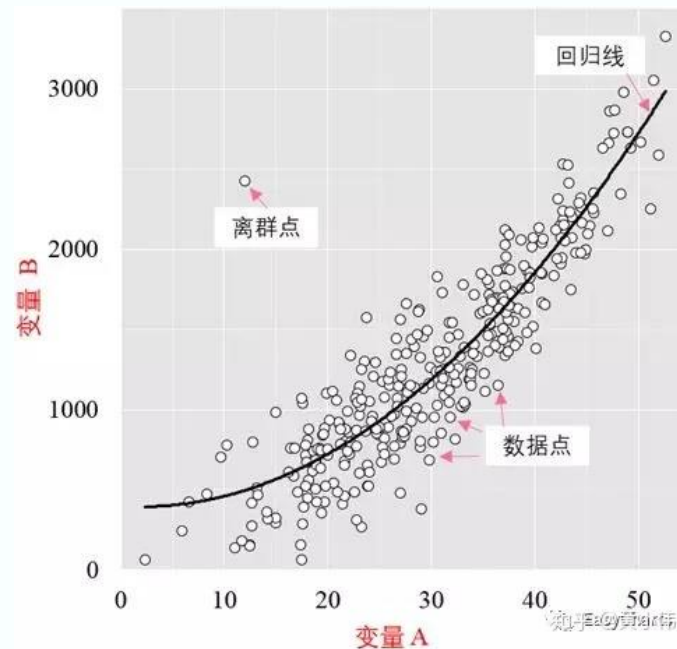
# 散點圖

- ◆ 雖然`plot()`也可以繪製散點圖，但`plot()`主要用於做折線圖，而`scatter()`則是專門繪製散點圖，語法為：

`plt.scatter(x, y, [選擇性參數1=值1, 選擇性參數2=值2, ...])`，

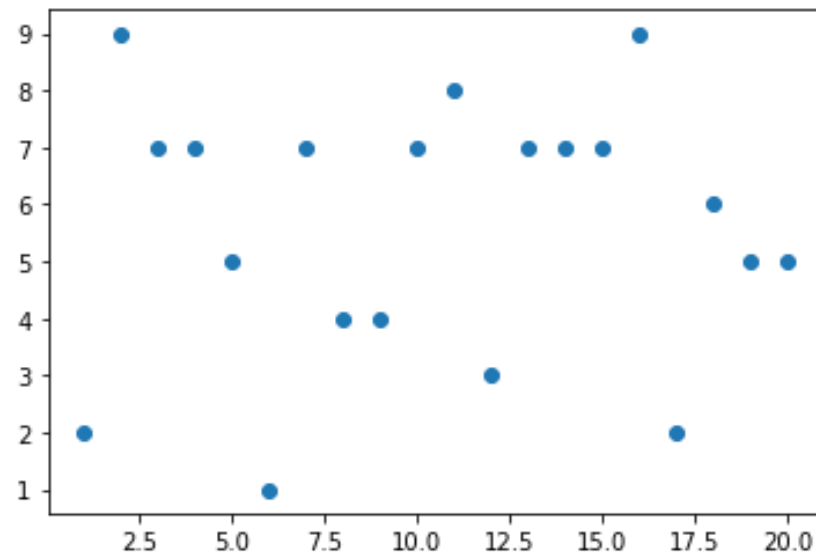
- ◆ 散點圖使用一系列的散點在直角座標中展示變數的數值分佈

- ◆ `x`、`y`：散點圖的位置，`x`與`y`必需數量一致
- ◆ `s`：繪製點的大小，預設20
- ◆ `c`：點的颜色
- ◆ `marker`：點的樣式，預設為"o"



# 散點圖

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 # 設定x與y軸資料
5 x = np.arange(1,21)
6 y = np.random.randint(1,10,20)
7 # 將資料代入圖表
8 plt.scatter(x, y)
9 # 顯示圖表
10 plt.show()
```



```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 # 設定x與y軸資料
5 x = color = np.arange(1,11)
6 y = np.random.randint(1,10,10)
7
8 # 將資料代入圖表，自訂顏色與點的大小
9 plt.scatter(x, y, c=color, s=100)
10 # 顯示圖表
11 plt.show()
```

