# Chapter 2

## PANDAS資料處理基礎

# 認識Pandas

Pandas是python的一個數據分析模組，提供高效能、簡易使用的資料格式(Data Frame)讓使用者可以快速操作及分析資料，Pandas除了強化了資料處理的方便性之外，也能與處理網頁資料與資料庫資料等，有點類似於Office的Excel能更加方便的進行運算與分析。所以很多人說Pandas就是python版本的Excel。

# 安裝與匯入Pandas模組

★請於終端機（命令提示字元）輸入

Pip install pandas

進行安裝

★開啟spyder後先匯入模組

Import pandas as pd

# 使用串列建立Series物件

(1) Series 物件：形成一個2陣列的組合，第一行是索引，第二行是我們建立的串列。

(2) 寫入下列程式碼來建立串列，特別注意 s = pd.Series(lst) 之大小寫。

(3) 沒有指定索引，預設從 0 開始自動產生索引。

```
1   # -*- coding: utf-8 -*-
2   """
3   Created on Sat Sep 16 19:52:43 2023
4
5   @author: User
6   """
7
8   import pandas as pd                          # 匯入pandas模組
9
10  lst = ["math" , "english" , "chinese"]  # 建立串列
11
12  s = pd.Series(lst)    # 使用pd.Series( )顯示串列候用變數s代表它
13
14  print(s)
```

```
0       math
1    english
2    chinese
dtype: object
```

# 使用多個Series建立Data Frame物件

(1) Data Frame物件：使用多個Series組合成Excel試算表。

(2) 特別注意 Series 和 DataFrame 的大小寫。

```python
# -*- coding: utf-8 -*-
"""
Created on Sat Sep 16 19:52:43 2023

@author: User
"""

import pandas as pd                          # 匯入pandas模組

lst1 = pd.Series(["Math" , "English" , "Chinese"])
lst2 = pd.Series([6 , 5 , 4])
lst3 = pd.Series([83 , 92 , 88])       # 利用三個Series建立串列

data = {"科目":lst1 , "學分數":lst2 , "得分":lst3} #建立每一行的名稱

df = pd.DataFrame(data)    # 使用pd.DataFrame( )顯示陣列後用變數df代表它

print(df)
```

```
        科目  學分數  得分
0     Math    6  83
1  English    5  92
2  Chinese    4  88
```

# 重新更改列索引及欄索引

(1) 使用 columns 屬性重新定義欄索引

(2) 使用 index 屬性更改列索引

```
8   import pandas as pd                          # 匯入pandas模組
9
10  lst1 = pd.Series(["Math" , "English" , "Chinese"])
11  lst2 = pd.Series([6 , 5 , 4])
12  lst3 = pd.Series([83 , 92 , 88])       # 利用三個Series建立串列
13
14  data = {"科目":lst1 , "學分數":lst2 , "得分":lst3} #建立每一行的名稱
15
16  df = pd.DataFrame(data)    # 使用pd.DataFrame( )顯示陣列後用變數df代表它
17
18  # 上述程式碼已經建立字典了
19
20  labels = ["m" , "e" , "c"]       # 將列索引由0,1,2更改為m,e,c
21  df.columns = ["學科" , "學分" , "班平均"]
22  # 將欄索引由 科目,學分數,得分 更改為 學科,學分,班平均
23  df.index = labels # 使用df.index 顯示更改後的列索引
24  print(df)
```

```
        學科  學分  班平均
m     Math   6   83
e  English   5   92
c  Chinese   4   88
```

# 將結果匯出成.csv檔(1/2)

輸入程式碼

df.to_csv(“檔名.csv” , index = True , encoding = “big5”)

(1) 第一個參數為檔名。

(2) 第二個參數index是決定是否寫入索引，True是寫入，False是不寫入。

(3) 第三個參數encoding為編碼，因我們有中文字所以使用 big5 或 utf8。

(4) 將反綠的地方同時改成 excel，就是匯出excel檔案。

# 將結果匯出成.csv檔(2/2)

參照第24行程式碼

```
8   import pandas as pd                          # 匯入pandas模組
9
10  lst1 = pd.Series(["Math" , "English" , "Chinese"])
11  lst2 = pd.Series([6 , 5 , 4])
12  lst3 = pd.Series([83 , 92 , 88])       # 利用三個Series建立串列
13
14  data = {"科目":lst1 , "學分數":lst2 , "得分":lst3} #建立每一行的名稱
15
16  df = pd.DataFrame(data)    # 使用pd.DataFrame( )顯示陣列後用變數df代表它
17
18  # 上述程式碼已經建立字典了
19
20  labels = ["m" , "e" , "c"]       # 將列索引由0,1,2更改為m,e,c
21  df.columns = ["學科" , "學分" , "班平均"]
22  # 將欄索引由 科目,學分數,得分 更改為 學科,學分,班平均
23  df.index = labels # 使用df.index 顯示更改後的列索引
24  df.to_csv("excel123.csv" , index = True , encoding = "big5")
25  print(df)
```

# 匯入DataFrame物件

| 程式碼 | 說明 |
| --- | --- |
| df.read_csv(檔名) | 匯入 CSV 格式的檔案 |
| df.read_json(檔名) | 匯入 JSON 格式的檔案 |
| df.read_html(檔名) | 匯入 HTML 網頁的 <table> 標籤的資料 |
| df.read_excel(檔名) | 匯入 Excel 檔案 |

| | |
| --- | --- |
| pd.read_csv(“檔名” , “編碼”) | 有中文欄名時使用 |

# Pandas 常用的資料處理小方法（一）

一、新增日期範圍：

用Excel開啟2330.TW後可以看到裡面沒有日期範圍，我們使用pd.date_range( )新增日期。

```
 8   import pandas as pd
 9
10   dates_d = pd.date_range("20230918", periods=5, freq="D")
11   # 使用pd.date_range()新增日期 , 從20230918開始 , period 表示要產生的個數 , freq表
12   # 示D為日 , M是月
13
14   print(dates_d)  # 輸出日期
15
16   df = pd.read_csv("2330.TW.csv") #匯入檔案2330.TW
17   df["Date"] = dates_d  # 輸出dates_d設定的相關資料
18   print(df)
```

```
In [9]: runfile('C:/Users/User/.spyder-py3/untitled0001.py', wdir='C:/Users/User/.spyder-py3')
DatetimeIndex(['2023-09-18', '2023-09-19', '2023-09-20', '2023-09-21',
               '2023-09-22'],
              dtype='datetime64[ns]', freq='D')
    Open   High    Low  Close  Adj Close    Volume       Date
0  184.0  185.0  183.0  184.0      184.0  18569000 2023-09-18
1  184.5  185.5  183.5  184.0      184.0  20198000 2023-09-19
2  185.0  185.0  181.5  182.0      182.0  29107000 2023-09-20
3  182.5  185.5  182.5  184.5      184.5  41130000 2023-09-21
4  180.5  182.5  180.5  181.5      181.5  52352000 2023-09-22
```

# Pandas 常用的資料處理小方法（二）

二、統計每個值出現的次數：

| 程式碼 | 說明 |
| --- | --- |
| unique( ) | 找出該欄中的不同值 |
| nunique( ) | 該欄不同值有幾種 |
| value_counts( ) | 該欄每個不同值出現的次數 |

# Pandas 常用的資料處理小方法（二）

以 2330.TW 之csv格式檔為例：

```
 8  import pandas as pd
 9
10  df = pd.read_csv("2330.TW.csv")    # 匯入2330.TW 的csv檔
11  print(df["Close"].unique())        # Close欄出現哪些不同值
12  print(df["Close"].nunique())       # Close欄共有幾個不同值
13  print(df["Close"].value_counts()) # Close每個不同值出現的次數
```

```
[184.  182.  184.5 181.5]
4
Close
184.0    2
182.0    1
184.5    1
181.5    1
Name: count, dtype: int64
```

# Pandas 常用的資料處理小方法（三）

計算平均數mean( )、計數count( )、中位數median( )：使用groupby( )函式

```
 8  import pandas as pd
 9
10  df = pd.read_csv("groupex.csv")    # 匯入groupex檔案
11  print(df)                          # 輸出表格
12  df2 = df.groupby("Good").mean()    # 輸出平均數
13
14
15  print(df2)
```

```
  Good  Price  number
0    a    100   32434
1    a     80   16543
2    b    120    1564
3    b    130   16543
4    b    200    5000
5    c    360   32434
6    c    250    3456
       Price         number
Good
a       90.0  24488.500000
b      150.0   7702.333333
c      305.0  17945.000000
```

# 習題一

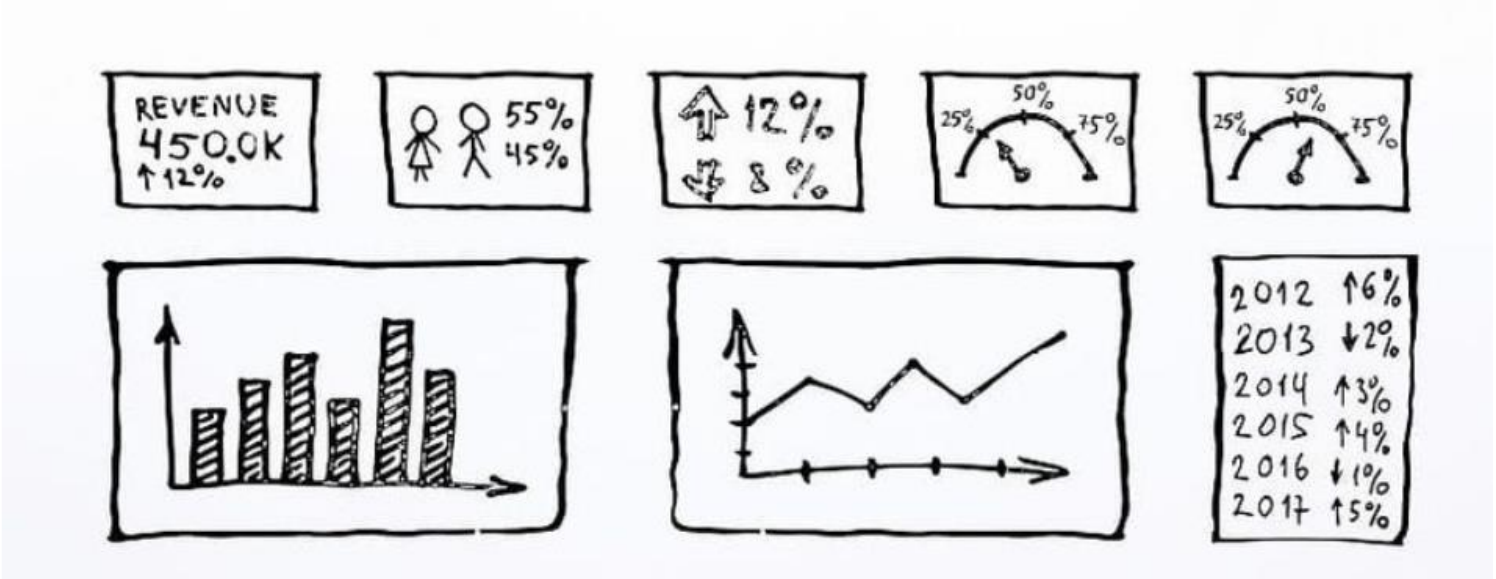
1. 右方表格請用python撰寫多個Series建立Data Frame物件並輸出csv檔。

| | A | B | C |
|---|---|---|---|
| 1 | | 科目 | 分數 |
| 2 | 陳小明 | 程式語言 | 82 |
| 3 | 李小洋 | 品質管理 | 77 |
| 4 | 張小東 | 工程數學 | 65 |
| 5 | 吳小花 | 社會統計 | 91 |

2. 將習題1.匯入後於表格第四欄加入日期，從20230801開始。

# 什麼是資料視覺化？

資料視覺化是將數據以圖表、圖形、地圖等視覺元素的形式呈現，以便更容易理解、分析和傳達數據的過程。通過資料視覺化，數據可以以直觀、可視化的方式展示，揭示出數據之間的模式、趨勢和關聯。



# 資料視覺化的好處

透過數據視覺化，我們能夠更快速地抓住數據的重點，並做出更有洞察力的決策。資料視覺化不僅有助於更好地理解數據，還能以更具說服力的方式將數據結果傳達給他人。它能將複雜的數據故事清晰、生動地呈現，並在不同受眾之間建立共識和理解。它在各個領域廣泛應用，包括商業、市場營銷、科學研究和報告呈現等。透過資料視覺化，我們能夠更深入地理解數據，做出更明智的決策，並最大化數據的價值。

# 使用Matplotlib繪製折線圖(1/3)

1. 透過終端機安裝Matplotlib。

Pip install matplotlib

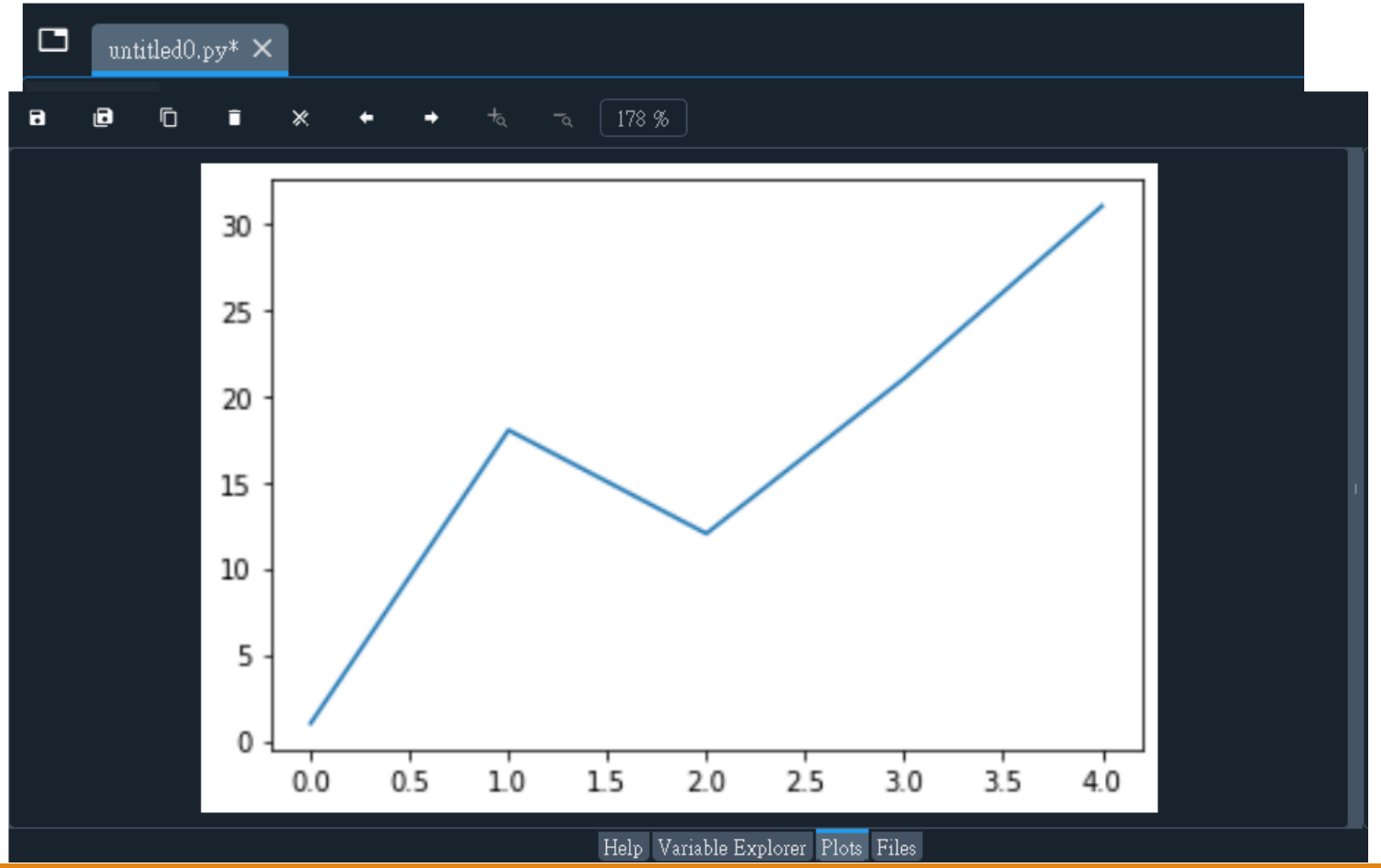
2. 安裝完畢後在python程式要先匯入Matplotlib模組

import matplotlib.pyplot as plt

# 使用Matplotlib繪製折線圖(2/3)

1. 單一數列繪製折線圖：

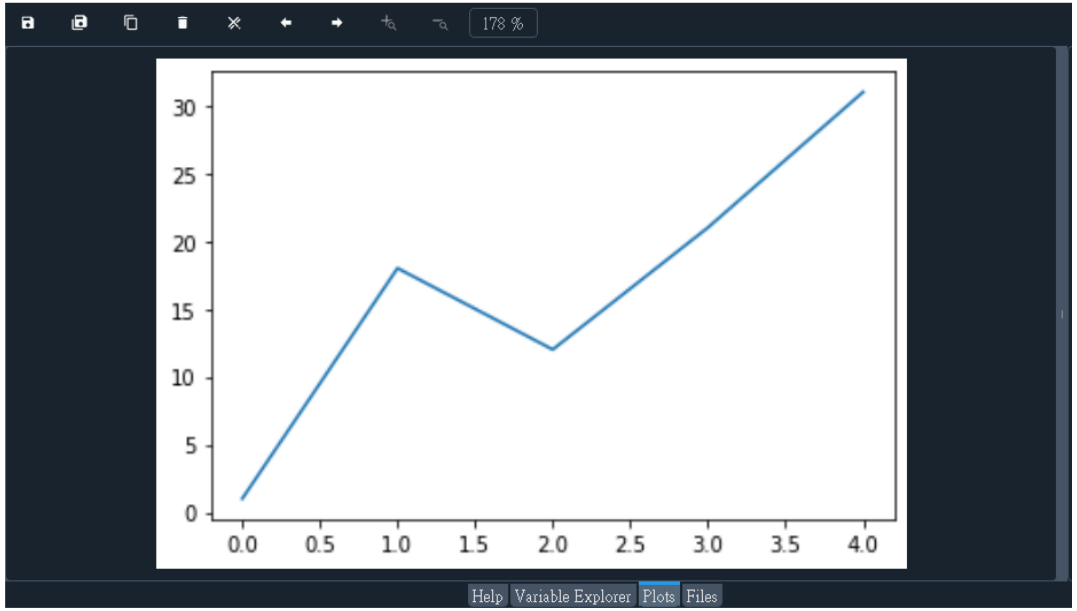
```
import matplotlib.pyplot as plt

data = [1, 18, 12, 21, 31]

plt.plot(data)

plt.show( )
```



# 使用Matplotlib繪製折線圖(3/3)

untitled0.py*

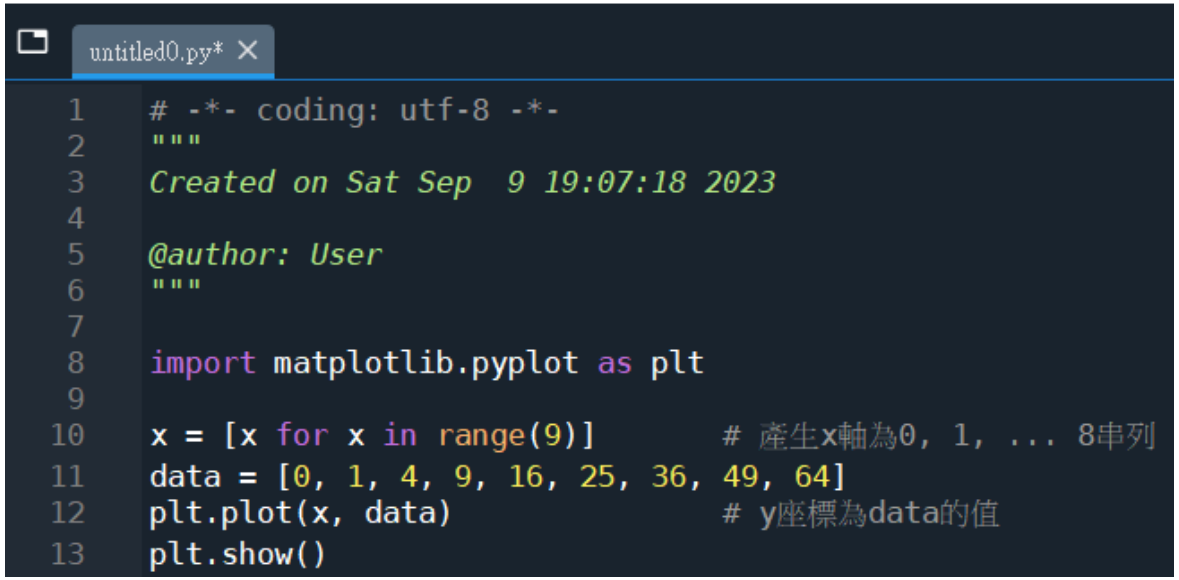
```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Sat Sep  9 19:07:18 2023
4
5  @author: User
6  """
7
8  import matplotlib.pyplot as plt     # 匯入matplotlib模組
9
10 data = [1, 18, 12, 21, 31]         # y軸的數據 (x軸預設為0,1,2,3,4)
11 plt.plot(data)                     # 繪製圖表
12 plt.show()                         # 圖表輸出
13
```



# x 軸與 y 軸顯示之設定(1/3)

★將x軸設定為 0 ~ 8：

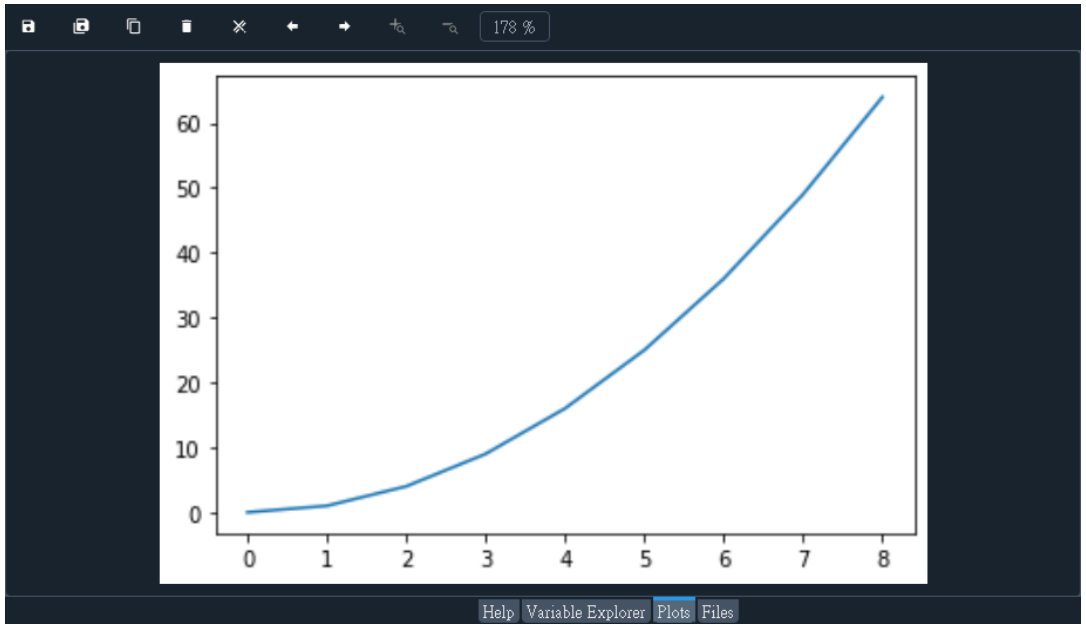
```
import matplotlib.pyplot as plt
x = [x for x in range(9)]  # 產生x軸為0, 1, ... 8串列
data = [0, 1, 4, 9, 16, 25, 36, 49, 64]
plt.plot(x, data)              # y座標為data的值
plt.show()
```



# $x$ 軸與 $y$ 軸顯示之設定(2/3)

untitled0.py*

```
1   # -*- coding: utf-8 -*-
2   """
3   Created on Sat Sep  9 19:07:18 2023
4
5   @author: User
6   """
7
8   import matplotlib.pyplot as plt
9
10  x = [x for x in range(9)]          # 產生x軸為0, 1, ... 8串列
11  data = [0, 1, 4, 9, 16, 25, 36, 49, 64]
12  plt.plot(x, data)                  # y座標為data的值
13  plt.show()
```



# x 軸與 y 軸顯示之設定(3/3)

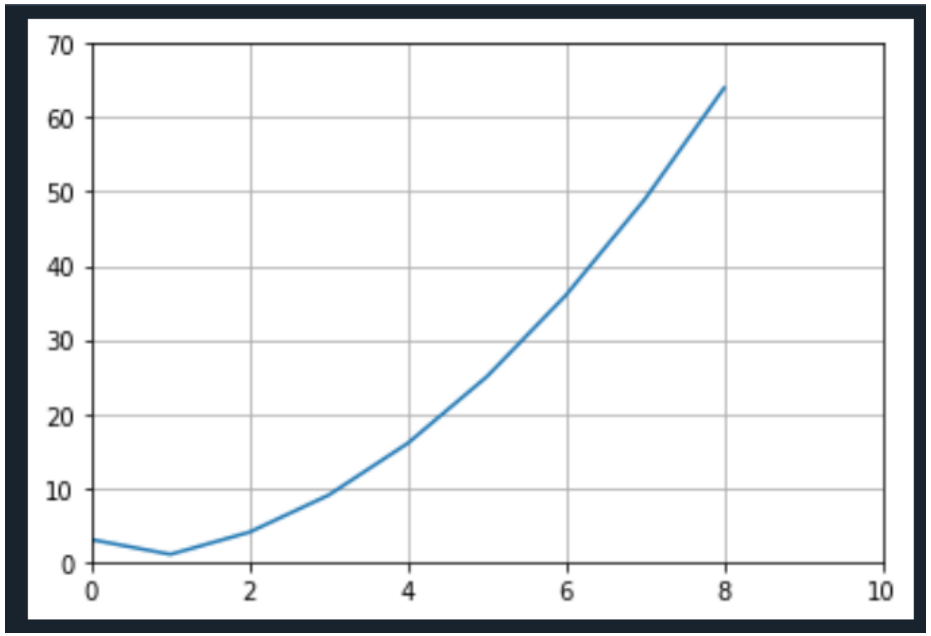
設定 x 軸及 y 軸的函數語法

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

若要增加格線，使用函數 grid( )

untitled0.py*

```
1   # -*- coding: utf-8 -*-
2   """
3   Created on Sat Sep  9 19:07:18 2023
4
5   @author: User
6   """
7
8   import matplotlib.pyplot as plt
9
10  data = [3, 1, 4, 9, 16, 25, 36, 49, 64]
11  plt.plot(data)                  # y座標為data的值
12  plt.axis([0,10,0,70])
13  plt.grid()
```



# 標題、*x* 軸及 *y* 軸的名稱(1/2)

1. 標題、*x* 軸及 *y* 軸的名稱：

title(‘標題名稱’)

xlabel(‘x軸名稱’)

ylabel(‘y軸名稱’)

2. 線條寬度：

預設線條寬度是1，若要變粗可以加上 lw

# 標題、x 軸及 y 軸的名稱(2/2)

untitled0.py*

```
# -*- coding: utf-8 -*-
"""
Created on Sat Sep  9 19:07:18 2023

@author: User
"""

import matplotlib.pyplot as plt

data = [0, 32, 35, 41, 33, 56, 63, 39, 45, 58 , 67, 50, 95]
plt.plot(data, lw=8)                    # y座標為data的值
plt.axis([1,12,0,100])
plt.title("Salary")
plt.xlabel("Month")
plt.ylabel("dollar")
plt.grid()
```

# 習題二

下表為某部們每個月的員工人數

| 一月 | 二月 | 三月 | 四月 | 五月 | 六月 |
|---|---|---|---|---|---|
| 118 | 131 | 138 | 145 | 157 | 175 |

| 七月 | 八月 | 九月 | 十月 | 十一月 | 十二月 |
|---|---|---|---|---|---|
| 142 | 136 | 129 | 125 | 116 | 107 |

請將 $x$ 軸數值代表月份，$y$ 軸數值代表員工數繪製折線圖。