

Python60 系列 成績曲線



</PLUS>

推動大學(程式設計)教學
MOE/Ministry of National Education

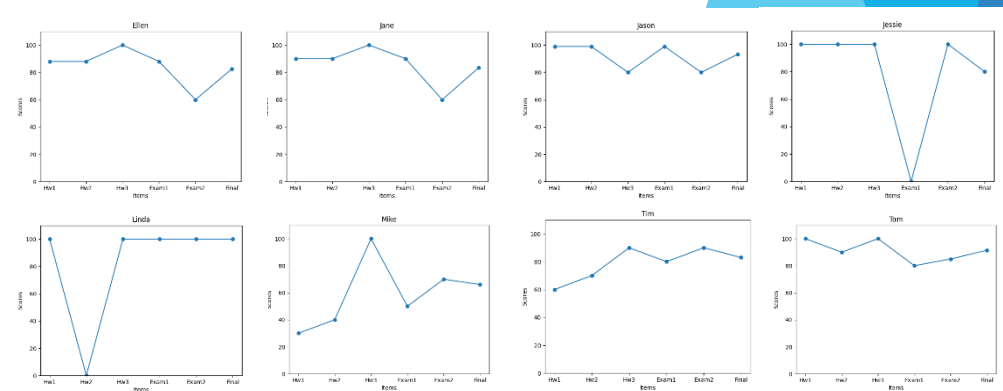
推動大學程式設計教學計畫。分項六：資料分析領域與學習評量推動團隊 (德明財經科大資科系江政杰老師主編)

成績曲線：課程簡述

▶ 問題情境：

- 教學過程中，教師經常透過測驗來讓師生雙方了解目前的學習狀況。為了了解學生全學期的成績趨勢，我們希望撰寫一個程式能批次繪製每位學生的成績曲線圖。
- 資料檔為 CSV 格式，每位同學有三次作業成績，兩次考試成績，以及總平均。

	A	B	C	D	E	F	G
1	Name	Hw1	Hw2	Hw3	Exam1	Exam2	Final
2	Tom	100	90	100	80	85	91.5
3	Jane	90	90	100	90	60	83.5
4	Mike	30	40	100	50	70	66
5	Tim	60	70	90	80	90	83
6	Jason	99	99	80	99	80	93.3
7	Ellen	88	88	100	88	60	82.6
8	Jessie	100	100	100	0	100	80
9	Linda	100	0	100	100	100	100



- ▶ 本課程將撰寫一個 24 行的程式，實際課程時間 20-30 分鐘。

成績曲線：學習內容

▶ 程式設計概念

- 套件與模組：安裝 matplotlib 套件
- 繪製圖表（折線圖）
- 列表與迴圈

▶ 資料分析概念

- 曲線圖表資料呈現

成績曲線

▶ 完整程式展示

```
import matplotlib.pyplot as plt

data = []
with open('學生成績.csv') as infile:
    text = infile.read().split()
    captions = text[0].split(',')
    for e in text[1:]:
        stu = e.split(',')
        scores = [float(sc) for sc in stu[1:]]
        data.append([stu[0]] + scores)

for stu in data:
    name = stu[0]
    scores = stu[1:]
    plt.clf()
    plt.plot(scores, marker='o')
    plt.title(name)
    plt.xticks(range(len(scores)), captions[1:])
    plt.xlabel('Items')
    plt.ylabel('Scores')
    plt.ylim(0, 110)
    plt.tight_layout()
    plt.savefig(name+'.png')
    #plt.show()
```

成績曲線

▶ 完整程式展示

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

匯入套件

```
data = []  
with open('學生成績.csv') as infile:  
    text = infile.read().split()  
    captions = text[0].split(',')
```

```
for e in text[1:]:  
    stu = e.split(',')  
    scores = [float(sc) for sc in stu[1:]]  
    data.append([stu[0]] + scores)
```

將成績資料取出
並存放入列表內

```
for stu in data:  
    name = stu[0]  
    scores = stu[1:]
```

```
plt.clf()  
plt.plot(scores, marker='o')  
plt.title(name)  
plt.xticks(range(len(scores)), captions[1:])  
plt.xlabel('Items')  
plt.ylabel('Scores')  
plt.ylim(0, 110)  
plt.tight_layout()  
plt.savefig(name+'.png')  
#plt.show()
```

設定圖表的
相關資訊

Python 套件安裝

- ▶ Python 提供眾多套件讓寫程式更輕鬆
 - 每個套件就是一個函式庫，請想像成一個工具箱。
 - 套件內包含許多函式，可直接呼叫需要的函式進行各種功能，我們就不需要自行撰寫這些複雜的程式。
- ▶ 套件必須先安裝過才能使用
 - Python 內建許多套件，所以很多函式我們可以直接使用。
 - 這個程式範例使用的繪圖套件 `matplotlib`，並不是內建套件中，所以我們需要先安裝。

Python 套件安裝

- ▶ 安裝 Python 套件時，一般會使用 pip 這個工具。但是根據使用的電腦系統，會有不同的操作方式。
- ▶ 本教材中以 Windows 作業系統環境為例說明。

Python 套件安裝步驟

▶ 啟動「命令提示字元」

- 點選畫面左下角 Windows 圖示 ，找到並開啟「命令提示字元」這個程式。根據版本不同可能在不同位置，請在「Windows 系統」或「Windows 附屬應用程式」內尋找。

▶ 請在「命令提示字元」內輸入

```
pip install matplotlib
```

- 系統會自動下載並安裝套件。
- 請記得要連上網路。

命令提示字元

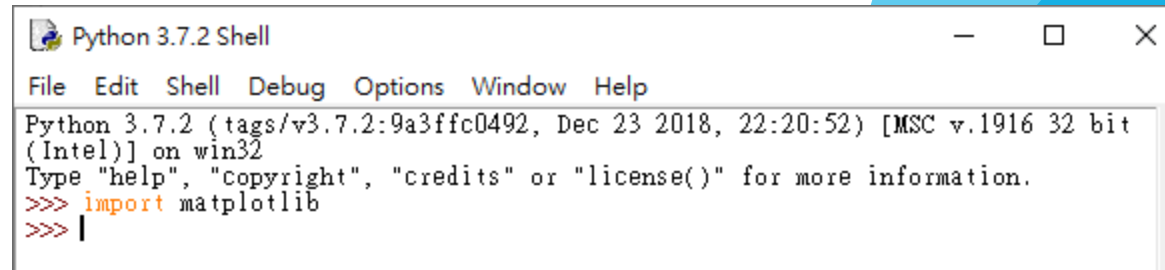
```
Microsoft Windows [版本 10.0.19043.1165]  
(c) Microsoft Corporation. 著作權所有，並保留一切權利。  
C:\Users\user>
```

命令提示字元

```
C:\Users\user>pip install matplotlib
```


Python 套件安裝步驟

- ▶ 執行 `pip install matplotlib` 後，畫面會出現成功或失敗訊息
 - 常見失敗原因是沒有連上網路、Python 版本太舊等。
 - 如果是 Python 版本太舊，可能需要另外安裝 pip 工具。
- ▶ 簡單測試套件安裝是否成功
 - 直接執行程式碼 `import matplotlib`，沒有錯誤訊息就是安裝成功。



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 22:20:52) [MSC v.1916 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import matplotlib
>>> |
```

程式碼解說：匯入套件

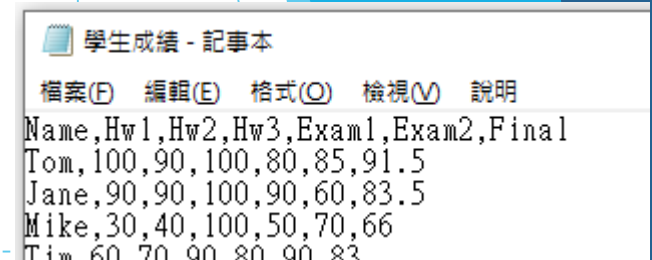
01 `import matplotlib.pyplot as plt`

- ▶ 第 01 行匯入 `matplotlib` 套件
- ▶ 由於 `matplotlib` 套件龐大，內含多個模組，本單元我們只需要使用其中的 `pyplot` 模組，所以匯入 `matplotlib.pyplot`。
- ▶ 程式中使用套件內的函式時，皆需要指明函式來源，為了簡單稱呼，用 `as plt` 將此模組另取別名為 `plt`，也就是後續程式碼中，我們可以使用 `plt` 代表 `matplotlib.pyplot`。
 - 如果不這樣作的話，後面的程式碼必須寫成 `matplotlib.pyplot.show()` 這麼長，但簡化後只需寫 `plt.show()` 就可以了。

程式碼解說：讀取與儲存資料

	A	B	C	D	E	F	G
1	Name	Hw1	Hw2	Hw3	Exam1	Exam2	Final
2	Tom	100	90	100	80	85	91.5
3	Jane	90	90	100	90	60	83.5
4	Mike	30	40	100	50	70	66
5	Tim	60	70	90	80	90	83

```
03 data = []
04 with open('學生成績.csv') as infile: #開啟學生成績檔案
05     text = infile.read().split()      #讀取學生成績資料
06     captions = text[0].split(',')     #讀取各項成績名稱
07     for e in text[1:]:
08         stu = e.split(',')
09         scores = [float(sc) for sc in stu[1:]]
10         data.append([stu[0]] + scores)
```

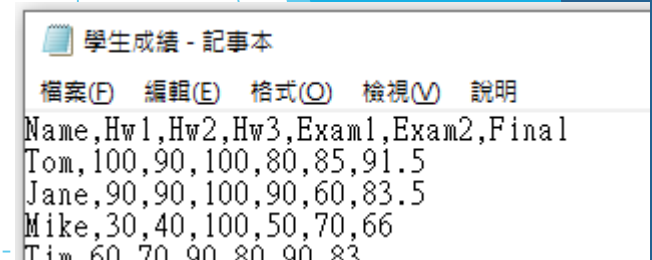


- ▶ 第 03 行宣告空列表 `data`，用以存放成績資料，以便後續繪圖使用。
- ▶ 第 04 行以 `open()` 函式開啟「學生成績.csv」檔案，並以變數 `infile` 代表此檔案。
- ▶ 第 05 行讀取出成績資料後，使用函式 `split()` 切割每一行資料成字串列表。
 - CSV 檔案是一種純文字檔案，資料項目之間以一個逗號間隔。
 - `text` 是一個字串列表，以本單元資料檔為例，其內容是 `['Name,Hw1,Hw2,Hw3,Exam1,Exam2,Final', 'Tom,100,90,100,80,85,91.5',]`。
- ▶ 第 06 行取出第一列資料 (`text[0]`)，用逗點來切割出每一個欄位。
 - 第一列資料是標題，`text[0]` 是一個字串，內容是 `'Name,Hw1,Hw2,Hw3,Exam1,Exam2,Final'`，而 `captions` 是一個字串列表，內容是 `['Name','Hw1','Hw2','Hw3','Exam1','Exam2','Final']`。

程式碼解說：讀取與儲存資料

	A	B	C	D	E	F	G
1	Name	Hw1	Hw2	Hw3	Exam1	Exam2	Final
2	Tom	100	90	100	80	85	91.5
3	Jane	90	90	100	90	60	83.5
4	Mike	30	40	100	50	70	66
5	Tim	60	70	90	80	90	83

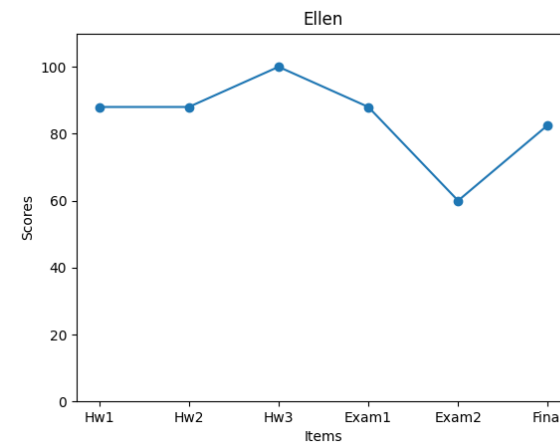
```
03 data = []
04 with open('學生成績.csv') as infile: #開啟學生成績檔案
05     text = infile.read().split()      #讀取學生成績資料
06     captions = text[0].split(',')    #讀取各項成績名稱
07     for e in text[1:]:
08         stu = e.split(',')
09         scores = [float(sc) for sc in stu[1:]]
10         data.append([stu[0]] + scores)
```



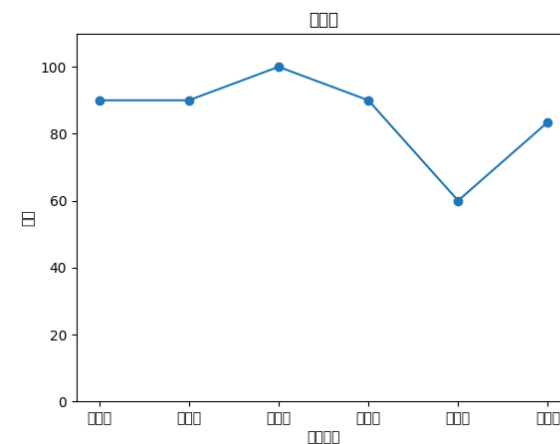
- ▶ 第 07 行以一個 for ... in ... 句型取出字串列表 text 的切片 text[1:]。
 - text[1:] 代表取出列表中第二個資料開始的所有資料，本例中即檔案第二列開始的內容。
- ▶ 第 08 行對從 text[1:] 取出的每一行字串作 split(',')，令 stu 代表之。
 - 本例中第一次執行到第 08 行時，stu 的內容為 ['Tom', '100', '90', '100', '80', '85', '91.5']，第二次執行到第 08 行時，stu 的內容為 ['Jane', '90', '90', '100', '90', '60', '83.5']，以此類推。
- ▶ 第 09 行以列表生成式從 stu[1:] 取出數字字串，轉換成為 float，建立一個浮點數列表 scores。
 - 若 stu 的內容為 ['Tom', '100', '90', '100', '80', '85', '91.5']，此處建立的 scores 為 [100,90,100,80,85,91.5]。
 - 對列表生成式不熟悉的同學，請參看本系列課程的單元 4「成績計算」。
- ▶ 第 10 行將每一筆成績資料放入列表 data 內，data 會是由列表組成的列表。
 - 每次放入 data 中的資料是 stu[0] 和 scores 合併成的列表，如 ['Tom', 100,90,100,80,85,91.5]。

程式概念：繪製曲線圖表

- ▶ 套件 `matplotlib` 提供強大的繪製圖表功能
 - 可根據資料特性，選擇不同的圖表類型。
 - 本課程練習繪製出右圖的成績折線圖。



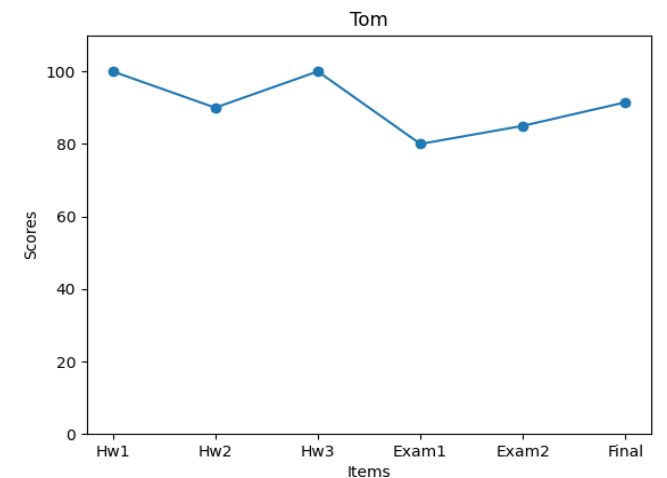
- ▶ 套件 `matplotlib` 的預設設定下無法顯示中文。
 - 要顯示中文圖表，需要另外安裝中文字型。
 - 本單元資料以英文呈現，省略中文的問題。



程式碼解說：繪製曲線圖表

```
12 for stu in data:
13     name = stu[0]
14     scores = stu[1:]
15     plt.clf() #設定全新圖表
16     plt.plot(scores, marker='o')
17     plt.title(name)
18     plt.xticks(range(len(scores)), captions[1:])
19     plt.xlabel('Items')
20     plt.ylabel('Scores')
21     plt.ylim(0, 110)
22     plt.tight_layout()
23     plt.savefig(name+'.png') #將圖表存成png檔
24     #plt.show() #顯示圖表，但目前以註解方式省略執行
```

	A	B	C	D	E	F	G
1	Name	Hw1	Hw2	Hw3	Exam1	Exam2	Final
2	Tom	100	90	100	80	85	91.5
3	Jane	90	90	100	90	60	83.5
4	Mike	30	40	100	50	70	66
5	Tim	60	70	90	80	90	83

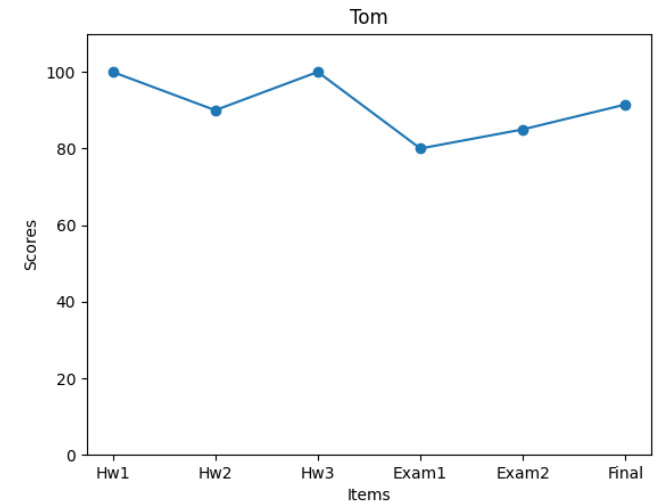


- ▶ 前面的程式碼已從成績資料檔中讀入所有學生成績並儲存為列表 `data`，接下來我們將基於 `data` 內容繪製每位學生的成績折線圖。
 - 以本單元資料檔為例，`data` 內容為 `[['Tom',100,90,100,80,85,91.5], ['Jane',90,90,100,90,60,83.5], ...]`。
- ▶ 第 12 行以 `for ... in ...` 從 `data` 中逐一取出資料，以 `stu` 代表，第 13 行取出 `stu` 第一項，這是學生姓名，第 14 行取出 `stu` 第二項以後所有資料項，包含六個成績資料。
- ▶ 第 15-24 行使用 `plt` 模組中的函式來設定、繪製與儲存圖表。

程式碼解說：繪製曲線圖表

```
12 for stu in data:
13     name = stu[0]
14     scores = stu[1:]
15     plt.clf() #設定全新圖表
16     plt.plot(scores, marker='o')
17     plt.title(name)
18     plt.xticks(range(len(scores)), captions[1:])
19     plt.xlabel('Items')
20     plt.ylabel('Scores')
21     plt.ylim(0, 110)
22     plt.tight_layout()
23     plt.savefig(name+'.png') #將圖表存成png檔
24     #plt.show() #顯示圖表，但目前以註解方式省略執行
```

	A	B	C	D	E	F	G
1	Name	Hw1	Hw2	Hw3	Exam1	Exam2	Final
2	Tom	100	90	100	80	85	91.5
3	Jane	90	90	100	90	60	83.5
4	Mike	30	40	100	50	70	66
5	Tim	60	70	90	80	90	83

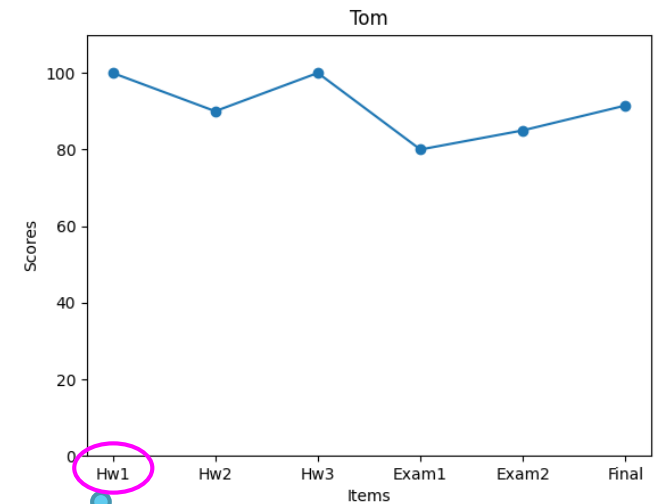


- ▶ 第 15 行呼叫 `plt` 模組的 `clf()` 函式設定全新圖表。
 - 如果沒有這一行，學生的成績曲線會不斷疊加在同一張圖表。
- ▶ 第 16 行呼叫 `plt` 模組的 `plot()` 函式繪製折線圖。
 - `plot()` 函式如果只傳入一個列表，會以此列表作為 `y` 軸的值，如果傳入兩個列表，會以第一個列表作為 `x` 軸的值，第二個作為 `y` 軸的值。
 - `plot()` 函式可設定 `marker` 參數，代表點的圖例（可嘗試其它圖例如 `'x'`，`'-'`，`'v'`，`'+'` 等）。另外還可設定線條顏色 `color`（`'red'`，`'black'`，`'green'`，`'blue'`，`'pink'` 等）及線型寬度 `linewidth`。

程式碼解說：繪製曲線圖表

```
12 for stu in data:
13     name = stu[0]
14     scores = stu[1:]
15     plt.clf() #設定全新圖表
16     plt.plot(scores, marker='o')
17     plt.title(name)
18     plt.xticks(range(len(scores)), captions[1:])
19     plt.xlabel('Items')
20     plt.ylabel('Scores')
21     plt.ylim(0, 110)
22     plt.tight_layout()
23     plt.savefig(name+'.png') #將圖表存成png檔
24     #plt.show() #顯示圖表，但目前以註解方式省略執行
```

	A	B	C	D	E	F	G
1	Name	Hw1	Hw2	Hw3	Exam1	Exam2	Final
2	Tom	100	90	100	80	85	91.5
3	Jane	90	90	100	90	60	83.5
4	Mike	30	40	100	50	70	66
5	Tim	60	70	90	80	90	83

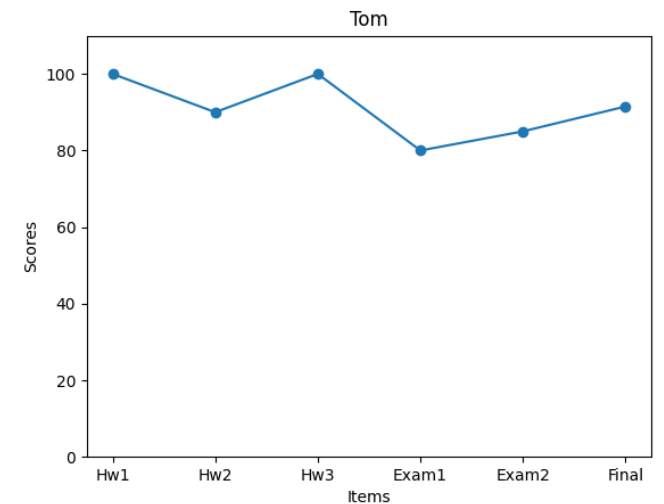


- ▶ 第 17 行呼叫函式 `title()` 設定圖表的標題，這裡使用學生的姓名。
- ▶ 第 18 行呼叫函式 `xticks()` 設定圖表在 x 軸的資料刻度與文字。
 - 還記得嗎？我們在程式第 06 行將資料檔的第一行存在 `captions` 中，其內容為 `['Name', 'Hw1', 'Hw2', 'Hw3', 'Exam1', 'Exam2', 'Final']`。
 - `xticks()` 傳入兩個列表（或類似列表），第一個代表 x 軸位置，第二個代表對應位置的文字。此處我們以 `range()` 產生類似 `[0, 1, 2, 3, 4, 5]` 列表的內容，而以 `captions[1:]`（內容為 `['Hw1', 'Hw2', 'Hw3', 'Exam1', 'Exam2', 'Final']`）作為 x 軸該位置顯示的文字。

程式碼解說：繪製曲線圖表

```
12 for stu in data:
13     name = stu[0]
14     scores = stu[1:]
15     plt.clf() #設定全新圖表
16     plt.plot(scores, marker='o')
17     plt.title(name)
18     plt.xticks(range(len(scores)), captions[1:])
19     plt.xlabel('Items')
20     plt.ylabel('Scores')
21     plt.ylim(0, 110)
22     plt.tight_layout()
23     plt.savefig(name+'.png') #將圖表存成png檔
24     #plt.show() #顯示圖表，但目前以註解方式省略執行
```

	A	B	C	D	E	F	G
1	Name	Hw1	Hw2	Hw3	Exam1	Exam2	Final
2	Tom	100	90	100	80	85	91.5
3	Jane	90	90	100	90	60	83.5
4	Mike	30	40	100	50	70	66
5	Tim	60	70	90	80	90	83



- ▶ 第 19-20 行分別呼叫 `xlabel()` 和 `ylabel()` 函式設定 x 軸與 y 軸的標題。
- ▶ 第 21 行呼叫 `ylim()` 函式設定 y 軸的資料範圍，此處設定最小值 0，最大值 110。
- ▶ 第 22 行呼叫 `tight_layout()` 函式以將整個圖表內容限縮在圖片中。
 - 若省略這行，有可能有些圖表內容會超出圖片範圍。
- ▶ 第 23 行呼叫 `savefig()` 函式將圖表存檔，這裡以學生姓名（以 `name` 代表）為主檔名，副檔名為 `'.png'`。
- ▶ 第 24 行呼叫 `show()` 函式讓圖片顯示在畫面上。由於在迴圈內，因此這個指令會執行多次而顯示眾多圖片，因此先用註解「#」讓這行程式不會被執行。如果需要顯示圖片時可將註解拿掉。

成績曲線

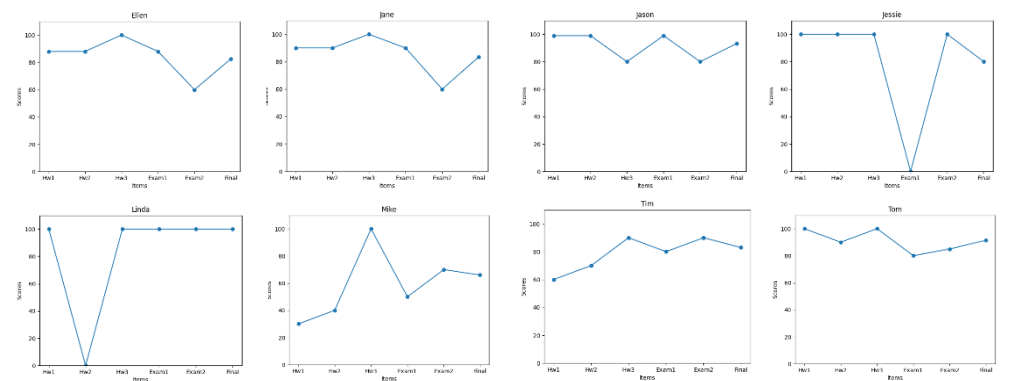
完整程式展示

```
import matplotlib.pyplot as plt

data = []
with open('學生成績.csv') as infile:
    text = infile.read().split()
    captions = text[0].split(',')
    for e in text[1:]:
        stu = e.split(',')
        scores = [float(sc) for sc in stu[1:]]
        data.append([stu[0]] + scores)

for stu in data:
    name = stu[0]
    scores = stu[1:]
    plt.clf()
    plt.plot(scores, marker='o')
    plt.title(name)
    plt.xticks(range(len(scores)), captions[1:])
    plt.xlabel('Items')
    plt.ylabel('Scores')
    plt.ylim(0, 110)
    plt.tight_layout()
    plt.savefig(name+'.png')
    #plt.show()
```

	A	B	C	D	E	F	G
1	Name	Hw1	Hw2	Hw3	Exam1	Exam2	Final
2	Tom	100	90	100	80	85	91.5
3	Jane	90	90	100	90	60	83.5
4	Mike	30	40	100	50	70	66
5	Tim	60	70	90	80	90	83
6	Jason	99	99	80	99	80	93.3
7	Ellen	88	88	100	88	60	82.6
8	Jessie	100	100	100	0	100	80
9	Linda	100	0	100	100	100	100



挑戰時刻

- ▶ 成績曲線能看出單一學生的成績變化，如果能再加上全班的平均數，就更能知道學生的相對程度。
- ▶ 下面的程式碼可接在程式範例第 10 行後，計算每個計分項目的全班平均分。

```
avg = []  
for i in range(1, len(data[0])):  
    all_scores = [data[stu][i] for stu in range(len(data))]  
    avg.append(sum(all_scores)/len(all_scores))
```

- ▶ 請修改程式以便在學生成績曲線圖中加上全班平均曲線。
 - 呼叫 `plot()` 函式時新增設定 `label` 參數。當有設定 `label` 參數時，之後需再呼叫 `legend()` 函式。

```
plt.plot(scores, marker='o', label='Your Score')  
plt.plot(avg, marker='x', label='Class Average')  
plt.legend()
```

