

6_1 串列

處理少量的資料，可以個別設定一些單一變數，但是如果資料龐大，例如，若有資料^{3,8,4,7,2,9} 要求極大、極小、平均等運算，若還是使用6 個變數儲存，程式將會很冗長，本章將會介紹另一種資料結構『串列』，然後配合上一章的for 或while 迴圈，就可以以非常精簡的程式，處理以上大批資料。其次，串列還有維度之分，分別是一維、二維、三維等等，就要看您資料運算的方式，若只有同批資料的關係，例如，上例就是一維資料串列就可以，請看6_2節。但是，在一個班級學生裡，每個人都有國、英、數、自然、社會等5 科成績，我們不僅要求縱向的各科平均，還要求橫向的每人平均，那就需要使用二維串列，請看6_3節。（補充說明：C/C++、Java等連續資料的集合稱為Array翻譯為陣列，Python則稱為List，翻譯為串列。）

範例6_2a 設有資料如下

77, 66, 99, 44, 55

請寫一個程式，可以計算平均、最高分及最低分。

演算法則

1.此題我們在第四章已經介紹，我們使用5個單一變數儲存以上資料，但是程式有點冗長。本例為了能使用迴圈來簡化程式，資料結構採用一維串列儲存以上資料。

```
a=[77,66,99,44,55]
```

2.有了串列，就就有串列索引，有了串列索引，就可使用迴圈存取資料，參考程式如下：

```
a=[77,66,99,44,55] #以串列儲存資料
```

```
#先指派第0筆資料為和、極大與極小值
```

```
sum=a[0] #和
```

```
max=a[0] #極大值
```

```
min=a[0] #極小值
#使用迴圈逐一從第1筆資料開始處理每一筆資料
for i in range(1,5):
    sum=sum+a[i] #累加每一筆資料
    if a[i]> max: #若該筆資料大於極大值
        max=a[i] #極大值以該筆資料取代
    if a[i]<min: #若該筆資料小於極大值
        min=a[i] #極小值以該筆資料取代
print("Sum=%d" % sum) #輸出和
print("Avg=%d" % (sum/5)) #輸出平均， (sum/5)括號不可省
print("Max=%d" % max) #輸出極大值
print("Min=%d" % min) #輸出極小值
```

3.使用 Python 內儲函式。

```
a=[77, 66, 99, 44, 55]
sum=sum(a) #求和
max=max(a) #求極大值
min=min(a) #求極小值
print("Sum=%d" % sum)
print("Avg=%d" % (sum/5))#(sum/5)括號不可省
print("Max=%d" % max)
print("Min=%d" % min)
```

自我練習

1、設有資料如下：

-1, 0, 3, 4, -5, 8

(1)統計正數、0、負數的個數。

(2)請問您如何儲存以上資料？

2、假設有資料如下：

3,6,2,1,2,3,5,6,1

(1) 計算並輸出共有多少個3。(2)計算1~6各幾個。(3)哪一個數字出現最多次

3. 假設有資料如下：

11,22,33,44,55,66,77,88,99,12,23,25,31,32,33,42,72,82,99

統計與輸出0~9,10~19,20~29,30~39,40~49,50~59,60~69,70~79,80~89,90~100 等區間的人數。

範例6_2c英文打字練習。假設有單字如下：

look、at、one、two、tree、and、day、book、from、go

請寫一個程式，可以出現10題英文單字，讓使用者可鍵入單字，並評判正確與錯誤題數和使用時間。

程式列印

1、將單字用串列儲存如下：

```
import time #使用 time.time()
a=['look','at','one','two','tree','and','day','book','from','go']
n=10
right=0
wrong=0
t1=int(time.time())#取得系統時間的秒數
#使用迴圈逐一輸出每一題單字
for i in range(n):
    print(a[i],':')#每次取一個輸出
    b=input()#由使用者鍵入此單字
    if b==a[i]:#比較是否相同
        right=right+1
```

```

else:
    wrong=wrong+1

t2=int(time.time())

t=t2-t1

print('right=%d'% right)

print('wrong=%d'% wrong)

print('total time=%d second'% t)

```

6_3 二維串列

前面範例 6-2c 的單字

look、at、one、two、tree、and、day、book、from、go

此稱為一維資料，我們使用一維串列儲存如下：

```
a=['look','at','one','two','tree','and','day','book','from','go']
```

這樣可以用陣列索引與迴圈存取每個單字。現在若有單字如下：希望能出現中文，讓使用者輸入英文，並檢查正確與否？

| 編號 | 英文 | 詞性 | 中文 |
|----|-----------|-----|-----|
| 1 | delighted | adj | 高興的 |
| 2 | gear | n | 服裝 |
| 3 | behave | v | 表現 |

以上每一個單字都還有詞性與中文註解，此稱為二維資料，此時就需要使用二維陣列儲存以上二

維資料。本例可宣告二維串列如下：

```

a=[["delighted", "adj", "高興的"],
    ["gear", "n", "服裝"],
    ["behave", "v", "表現"]]

```

則以上單字、詞性、中文的索引編號(列,行)如下：

| 英文 | 詞性 | 中文 |
|-----------|-----|-----|
| delighted | adj | 高興的 |

| | | |
|--------|-------|-------|
| (0,0) | (0,1) | (0,2) |
| gear | n | 服裝 |
| (1,0) | (1,1) | (1,2) |
| behave | v | 表現 |
| (2,0) | (2,1) | (2,2) |

以上縱向稱為列索引，橫向稱為行索引，有了索引，往後就可以使用變數名稱與索引存取以上資料。例如：

```
print(a[0][0]) #_____
```

```
print(a[2][1]) #_____
```

以上即為本節所要介紹的二維串列。

二維串列的宣告

二維串列的宣告常用語法如下：

串列名稱=[[初值 for i in range (行數)]for j in range (列數)]

例如：以下程式可快速初始化一個三行四列的二維串列，其初值為 0。

```
a=[[0 for i in range (3)]for j in range (4)]
```

```
b=[[0]*3 for i in range(4)]
```

```
print(a) #_____
```

```
print(b) #_____
```

二維串列初始化

單一變數、一維串列都可宣告變數同時給予初值，二維串列也可以，以下程式可初始化一個二維串列，且預設初值。

```
c=[[1,2,3],[4,5,6]]
```

```
print(c)
```

以上二維串列的索引如下表：

| | | |
|-----------|-----------|-----------|
| c[0][0]=1 | c[0][1]=2 | c[0][2]=3 |
| c[1][0]=4 | c[1][1]=5 | c[1][2]=6 |

請鍵入以下程式，寫出執行結果。

```
c=[[1,2,3],[4,5,6]]
```

```
print(c)
```

```
c[0][1]=12
```

```
d=c[0][1]
```

```
print(d)
```

```
print(c)
```

範例6-3a 假設有單字如下：請寫一個程式，程式會連續3次出現中文，讓使用者輸入英文，並統

計答對與答錯題數及答題花費時間。

| 編號 | 英文 | 詞性 | 中文 |
|----|-----------|-----|-----|
| 1 | delighted | adj | 高興的 |
| 2 | gear | n | 服裝 |
| 3 | behave | v | 表現 |

輸出結果

```
total time=23 second
```

運算思維

1.將以上資料以二維串列儲存，本例使用 a 串列。

2.使用串列，就有索引，可使用迴圈操作以上串列。

```
import time #使用 time() 函式
```

```
a=[["delighted", "adj", "高興的"],
```

```
    ["gear", "n", "服裝"],
```

```
    ["behave", "v", "表現"]]
```

```
n=3 #題數
```

```
right=0 #答對題數
```

```
wrong=0 #答錯題數
```

```
t1=int(time.time())#取得系統時間的秒數
```

```

#使用迴圈逐一輸出每一題的中文與詞性

for i in range(n):

    print(a[i][2],a[i][1],':',end='')#輸出每一題的中文、詞性

    b=input()#等待使用者輸入英文

    if b==a[i][0]:#將使用者輸入與原題目比對

        right=right+1

    else:

        wrong=wrong+1

t2=int(time.time())#取得系統時間的秒數

t=t2-t1

print('right=%d'% right)

print('wrong=%d'% wrong)

print('total time=%d second'% t)

```

補充說明

1. `t2=int(time.time())`#可取得系統時間，是系統函式，請看7-2節，單位是秒數，待程式結束時，再取系統時間，兩者相減，可以計算使用者使用的秒數。

自我練習

- 1、同範例 6-3a，但增加當使用者答題後，馬上告知正確與否；若是錯誤，也告知正確答案。
- 2、同範例 6-3a，但增加為 100 個單字，每次測驗依亂數出現 10 個單字，同次測驗中所出現單字不能重複。
- 3、同範例 6-3a，但增加為 100 個單字，每次測驗 3 分鐘，題目依照亂數出現，同次測驗中所出現單字不能重複，統計使用者可答對題數。
- 4、同範例 6-3a，但增加為 100 個單字，題目依照亂數出現，同次測驗中所出現單字不能重複，使用者答錯了才停止，統計使用者可連續答對題數。
- 5、請任選一個學科，如英文、國文等，製作 10 題填充題，寫一個程式讓使用者回答此 10 題填充題。

範例6_3b 選擇題測驗。假設英文單字的選擇題選項與答案資料如下：

| 編號 | 單字 | 類別 | 答案 | 選項1 | 選項2 | 選項3 | 選項4 |
|----|-----------|-----|----|---------|---------|---------|---------|
| 1 | delighted | adj | 2 | 高興 的 | 悲傷 的 | 生氣 的 | 憤怒 的 |
| 2 | gear | n | 3 | 背包 | 褲子 | 服裝 | 鞋子 |
| 3 | behave | v | 2 | 生氣 | 表現 | 難過 | 看到 |

請寫一程式，可以讓使用者使用選擇題測驗以上3個單字，使用者每回答1題、馬上回應正確或錯誤，結束測驗並統計答對與答錯題數及答題花費時間。

執行結果

```
The wrong number is :1
total time=12 second
```

運算思維

1. 將題目用串列儲存，如以下 a 串列，每筆資料第三個是答案，如以下第 1 題答案是 2。
2. 有了串列，就有索引編號，就可使用迴圈操作串列，程式如下：

```
import time
```

```
a=[["book","n","2","學校","書","老師","校長"],
```

```
    ["delighted","adj","1","高興的","悲傷的","生氣的","憤怒的"],
```

```
    ["gear","n","3","背包","褲子","服裝","鞋子"]];
```

```
n=3 #題數
```

```
c=0 #答對題數
```

```
d=0 #答錯題數
```

```
t1=int(time.time())
```

```
for i in range(1,n+1):
```

```
    print("%d . %s,%s : " % (i,a[i][0],a[i][1]),end=")#輸出題號,英文單字,詞
```

性

```
    for j in range(1,4+1):
```

```
        print("(%d) %s : " % (j,a[i][j+2]),end=")#輸出四個選項
```

```
    b=input()#等待使用者輸入答案
```



```

if (b==a[i][2]): #比對
    c=c+1 #正確題數加1
    print("Right")
else:
    print("Wrong")
    d=d+1 #錯誤題數加1

t2=int(time.time())

t=t2-t1

print("The Right number is :%d"%c)
print("The wrong number is :%d"%d)
print('total time=%d second'% t)

```

範例 6_3a 假設有兩個矩陣 A、B 如下：

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & 3 \end{pmatrix}$$

請求其相加的結果如下：

$$\begin{pmatrix} 2 & 4 & 3 \\ 6 & 6 & 9 \end{pmatrix}$$

請問要使用何種資料結構較省事？

運算思維

1. 本題因為是二維資料，若用二維串列儲存，就可以使用雙迴圈處理縱向與橫向二維資料。

#逐一初始化每一串列

```
a=[[1,2,3],[4,5,6]]
```

```
b=[[1,2,0],[2,1,3]]
```

```
c=[[0,0,0],[0,0,0]]
```

#兩個二維串列 a 與 b，每一行、列資料逐一相加，放到 c 串列

```
for i in range(0,2): #逐一處理每一行
```

```

    for j in range(0,3): #逐一處理每一列
        c[i][j]=a[i][j]+b[i][j]
print(a) #使用函式輸出結果
print(b)
print(c)
#自行使用雙迴圈輸出結果
for i in range(0,2):
    for j in range(0,3):
        print(c[i][j],end=" ")
    print() #跳列

```

自我練習

1. 假設有矩陣如下：

$$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 0 \end{pmatrix}$$

- (1) 請問以何種資料結構儲存。
- (2) 請檢查幾個 0。
- (3) 請統計所有元素和。
- (4) 再將每一元素乘以 2 後輸出。