

串列

- 何謂串列
- 一維串列
- 使用迴圈操作串列
- 串列常用的函式與方法
- 串列的排序
- 二維串列
- 檢測試題解析

5.1 何謂串列

當一個資料常值要儲存於記憶體時，便需要指定一個變數來存放。但若有 100 個相似的資料要儲存時，那可是要使用 100 個變數來存放，這 100 個變數放在程式敘述中，要維護時是相當麻煩的。若這些變數的用途有關聯，這時候我們可以使用「串列」(List)來取代變數。串列又稱為「清單」，在其它程式語言被稱為「陣列」(Array)。

串列是儲存資料的容器，是一組經過編號的變數，在記憶體中佔用連續的記憶體位址。好像一列台車，每個車斗都標記有連續編號，若要從某個車斗取得或放置物品，只要告知車斗的編號，很快地就能找到該車斗。若將一列台車比喻為一個串列，則台車上的一串串車斗我們稱之為「元素」(element)或稱為「串列項目」，台車編號就是元素的註標值。欲存取串列中某個記憶體的資料內容時，只要告知註標值就會由存放在記憶體中的串列起始位址往下數，即可存取該記憶體的內容。

譬如：下圖是名稱為 `score` 的串列，含有 `score[0]~score[49]` 共 50 個串列元素，存放學生成績的整數資料。所有串列元素一起存放在連續的記憶體空間，若要存取串列中的資料，只要告知是第幾個資料(即註標值)，便會自動計算該資料的實際位置。



至於串列中是第幾個元素的表示方式是在該串列名稱後面的中括號內填入該變數所對應的數字編號稱為「註標」(index)或稱「索引值」。譬如 `ary[3]` 表示串列名稱為 `ary` 串列中的第四個元素(註標由 0 開始)。所以設計程式時，若碰到需要連續輸入、處理或輸出相關資料，則使用串列取代變數，只要改變中串列括號內的數字編號即可，如此可減輕對變數逐一命名的困擾，在程式中串列元素配合迴圈使用可縮短程式的長度，使得程式變得精簡且可讀性高。

若串列中只有一組註標，稱為一維串列；有兩組註標，稱為二維串列；有三組註標，稱為三維串列；以此類推…。

5.2 一維串列

5.2.1 一維串列的建立

一維串列的建立方式是以 `[]` 運算子來存放元素資料，元素間再用逗號分隔，語法為：

```
串列名稱 = [元素 1, 元素 2, 元素 3, ...]
```

【例】 建立串列名稱 `score`，其元素內容分別為 72, 98, 86, 76 整數資料。

```
score = [72, 98, 86, 76]
```




說明

1. 串列中各個元素資料的型別可以相同，也可以不相同。

例：`score = [72, 98, 86, 76]`

元素皆為整數資料型別

例：`animal = ['cat', 'dog', 'monkey', 'tiger']`

元素皆為字串資料型別

例：`data = ['Mary', 45, True]`

元素存放的資料型別不相同

2. 串列建立時，也可以不含元素資料。

例：`list1 = []` # 建立的 `list1` 串列是空串列

5.2.2 串列的讀取與存放

利用 `[]` 運算子填入註標，便可對串列存取其對應的元素內容；若是使用 `[start:end]`，便是存取註標 `start` 起至註標 `end-1` 的串列元素。要注意的是，串列中第一個元素的註標是 0，第二個元素的註標是 1，以此類推…。

一. 讀取串列元素

讀取串列元素的語法為：

語法 1：串列名稱[註標]

語法 2：串列名稱[start:end:step]

1. **語法 1**：可以讀取註標值所對應的串列元素，註標不能超過範圍，否則程式執行時會產生錯誤。
2. **語法 2**：讀取 `start` 起到 `end-1` 註標之間的串列元素，`step` 為間隔值。若 `start` 省略，預設值為 0；若 `end` 省略，則預設值為串列長度；若 `step` 省略，則預設值為 1。

[例] 建立串列 `lst1`，並進行相關的串列元素讀取操作。(檔名：`list1.py`)

```
01 lst1 = [10, 20, 30, 40, 50, 60, 70, 80, 90]
02 print(lst1)                # 印出 [10, 20, 30, 40, 50, 60, 70, 80, 90]
03 print(lst1[0])             # 印出 10
04 print(lst1[3])             # 印出 40
05 print(lst1[2:6])           # 印出 [30, 40, 50, 60]
06 print(lst1[:6])            # 印出 [10, 20, 30, 40, 50, 60]
07 print(lst1[0:9])           # 印出 [10, 20, 30, 40, 50, 60, 70, 80, 90]
08 print(lst1[1:9:2])         # 印出 [20, 40, 60, 80]
09 print(lst1[1::2])          # 印出 [20, 40, 60, 80]
10 print(lst1[-1])            # 印出 90
11 print(lst1[-9])            # 印出 10
12 print(lst1[-6:-3])         # 印出 [40, 50, 60]
13 print(lst1[: -2])          # 印出 [10, 20, 30, 40, 50, 60, 70]
14 print(lst1[10])            # 錯誤，註標超過範圍
```



說明

1. 第 2 行：印出 `lst1` 串列所有元素內容。
2. 第 3~4 行：串列中的第一個元素註標是 0，第二個元素的註標是 1，以此類推…。
3. 第 5~9 行：使用 `[start:end:step]`，便是存取註標 `start` 起，到註標 `end-1` 的串列元素，`step` 為間隔值。
4. 第 10~13 行：註標可以為負值，串列倒數第一個元素為 -1，串列倒數第二個元素為 -2，以此類推…。`lst1[:-2]` 與 `lst1[0:-2]` 相同。
5. 第 14 行：註標不能超過範圍，否則程式執行時會產生錯誤。

二. 存放串列元素

串列在建立時已存放有元素的初值，但如同變數一樣，所存放的元素值可以再改變內容，其改變元素內容的語法為：

```
串列名稱 [註標] = 資料
```

[例] 建立串列 `lst2`，並進行相關的串列元素存放操作。(檔名：`list2.py`)

```
01 lst2 = ['one', 'two', 'three', 'four', 'five', 'six']
02 lst2[4] = '五'      # lst2[4]的內容由'five'更改為'五'
03 print(lst2[4])     # 印出 五
04 print(lst2)        # 印出 ['one', 'two', 'three', 'four', '五', 'six']
```

5.3 使用迴圈操作串列

5.3.1 使用 `for ... range()` 迴圈

`for ... range()` 迴圈可以用 `range()` 函式的遞增或遞減方式指定串列的註標，進而循序讀取串列的元素內容。其中 `range()` 函式的最大值便是串列的長度，而串列的長度可以由 `len()` 函式取得。

[例] 建立串列 `lst3`，並使用 `len()` 函式取得 `lst3` 串列的長度。(檔名：`list3.py`)

```
01 lst3 = [23, 45, 67, 89, 100]
02 x = len(lst3)      # x ← 5
03 print(x)          # 印出 5
```

串列的長度取得後，可以設定給 `range()` 函式範圍的最大值。接著 `for ... range()` 迴圈便可以來控制串列的註標，循序讀取指定範圍的串列元素。

[例] 建立串列 `lst4`，使用 `for ... range()` 迴圈循序讀取指定範圍的串列元素。
(檔名：`list4.py`)

```
01 lst4 = [10, 21, 32, 43, 54, 65, 76, 87, 98]
02 for i in range(len(lst4)):
03     print(lst4[i], end = ' ')    # 印出 10 21 32 43 54 65 76 87 98
04 print()
05 for j in range(1, 4):
06     print(lst4[j], end = ' ')    # 印出 21 32 43
07 print()
```

```
08 for k in range(5, len(lst4)):  
09     print(lst4[k], end = ' ')    # 印出 65 76 87 98  
10 print()  
11 for m in range(len(lst4)-1, 5-1, -1):  
12     print(lst4[m], end = ' ')    # 印出 98 87 76 65  
13 print()  
14 for g in range(0, len(lst4), 2):  
15     print(lst4[g], end = ' ')    # 印出 10 32 54 76 98
```

**說明**

1. 第 2~3 行：讀取 lst4[0]~lst4[8] 的元素內容，其中 len(lst4) = 9。
2. 第 5~6 行：讀取 lst4[1]~lst4[3] 的元素內容。
3. 第 8~9 行：讀取 lst4[5]~lst4[8] 的元素內容。
4. 第 11~12 行：讀取 lst4[8]、lst4[7]、lst4[6]、lst4[5] 的元素內容。
5. 第 14~15 行：讀取 lst4[0]、lst4[2]、lst4[4]、lst4[6]、lst4[8] 的元素內容。

5.3.2 使用 for ... in 串列迴圈

使用 for ... in 串列 迴圈可以讀取串列的元素內，其語法為：

```
for 變數 in 串列:
```

[例] 建立串列 season，並以使用 for ... in 迴圈讀取 season 串列的元素內容。
(檔名：list5.py)

```
01 season = ['spring', 'summer', 'autumn', 'winter']  
02 for item in season:  
03     print(item, end = ' ')    # 印出 spring summer autumn winter
```

**說明**

第 2~3 行：進入 for ... in 迴圈時，第一次 item 的值會是『spring』，第二次 item 的值會是『summer』，第三次 item 的值會是『autumn』，第四次 item 的值會是『winter』。當讀取到串列最後一個元素資料時後，程式流程便會離開迴圈，執行迴圈的下一個敘述。

5.3.3 串列生成器

建立串列時搭配 range() 函式，可以指定串列的元素內容，簡例如下：

[例] 建立串列 `arr`，並使其 `arr[0] = arr[1] = ... = arr[5] = 3`。

(檔名：`list6.py`)

```
01 arr = [3 for x in range(6)]
02 for item in arr:
03     print(item, end = ',')      # 印出 3,3,3,3,3,3,
```



說明

1. 第 1 行：若是要使 `arr` 串列的 `arr[0]`、`arr[2]~arr[9]` 之元素內容皆為 'abc'，則程式敘述為 `arr = ['abc' for x in range(10)]`。
2. 第 1 行：若是要使 `arr` 串列的 `arr[0]`、`arr[2]~arr[9]` 之元素內容皆為空字串，則程式敘述為 `arr = [" for x in range(10)]`。

[例] 建立串列 `lst`，並使其 `lst[0]=0`、`lst[1]=1`、`lst[2]=2`、`lst[3]=3`、`lst[4]=4`。

(檔名：`list7.py`)

```
01 lst = [y for y in range(5)]
02 for item in lst:
03     print(item, end = ',')      # 印出 0,1,2,3,4,
```



說明

第 1 行：若程式敘述改為 `lst = [(y+10) for y in range(5)]` 時，則 `lst[0]=10`、`lst[1]=11`、`lst[2]=12`、`lst[3]=13`、`lst[4]=14`。



簡例 (檔名：`max.py`)

使用串列儲存使用者連續輸入的 5 個整數，再從串列中找出最大數。



結果

請依序輸入 5 個整數...

輸入第 1 個元素內容：18

Enter↵

輸入第 2 個元素內容：-90

Enter↵

輸入第 3 個元素內容：6

Enter↵

輸入第 4 個元素內容：25

Enter↵

輸入第 5 個元素內容：-25

Enter↵

最大值為 25

[例] 用串列生成器建立二維串列 `arr`，使其內容為`[[0, 0], [0, 0], [0,0], [0, 0]]`。
(檔名：`list8.py`)

```
01 A = [0 for x in range(2)]
02 print(A)                # 印出 [0,0]
03 arr = [A for y in range(4)]
04 print(arr)              # 印出 [[0,0],[0,0],[0,0],[0,0]]
```



說明

1. 第 1 行：使用串列生成器建立一維串列，並使元素內容為 `[0, 0]`。
2. 第 3 行：以已建立的一維串列 `A` 的元素註標做為 `arr` 串列的元素行註標，而 `0~3(4-1)` 為列註標。即使
`arr = [A[0], A[1], A[2], A[3]] = [[0, 0], [0, 0], [0, 0], [0, 0]]`。
3. 若 `A = [(2x+1) for x in range(2)]`; `arr = [A for y in range(4)]`，則
`A = [1, 3]`，`arr = [[1, 3], [1, 3], [1, 3], [1, 3]]`。
4. 若 `arr = [[0 for x in range(2)] for y in range(4)]`，則
`arr = [[0, 0], [0, 0], [0, 0], [0, 0]]`。



簡例 (檔名：`score.py`)

下表為四位考生參加資優評選的成績，請完成空白部分的數據。

編號	語文	數理	智力	總分
1	87	64	88	
2	93	72	86	
3	80	88	89	
4	79	91	90	
平均				



結果

編號	語文	數理	智力	總分
1	87	64	88	239
2	93	72	86	251
3	80	88	89	257
4	79	91	90	260
平均	84.8	78.8	88.2	

 程式碼

檔名：\ex05\score.py

```

01 no = [1,2,3,4]                # 編號
02 score = [[87,64,88],[93,72,86],[80,88,89],[79,91,90]]    # 成績
03 print('編號  語文  數理  智力  總分')
04 print('=====')
05 for i in range(len(no)):
06     print('%2d' %no[i], end = '    ')
07     hSum = 0
08     for j in range(len(score[i])):
09         print('%3d' %score[i][j], end = '    ')
10         hSum += score[i][j]
11     print('%3d' %hSum)
12
13 print('%s' %'平均', end = '    ')
14 for j in range(3):
15     vSum = 0
16     for i in range(len(no)):
17         vSum += score[i][j]
18     print('%4.1f' %(vSum/len(no)), end = '    ')

```

 說明

1. 第 1 行：使用一維串列 `no` 存放編號。
2. 第 2 行：使用二維串列 `score` 存放 語文、數理、智力 三科目成績。
3. 第 5~11 行：使用兩層 `for` 迴圈將 `score` 二維串列元素的內容顯示。內層迴圈用 `hSum` 變數累加每一個編號(水平列)各科的成績總分，並在內層迴圈結束後顯示出來。
4. 第 14~18 行：使用兩層 `for` 迴圈計算每一科(垂直行)的分數總和 `vSum`。在內層迴圈結束後，用 `vSum/len(no)` 計算該垂直行的分數平均並顯示出來。

5.7 檢測試題解析

 題目 (一)

(APCS 檢測試題)

輸入 5 個學生資料，資料有姓名、國文、英語成績，計算學生的平均分數後，學生的所有資料再依平均分數由高而低對應排列。

結果

```
輸入第 1 個學生姓名：王大維 Enter↓
輸入國文成績：92 Enter↓
輸入英語成績：72 Enter↓
輸入第 2 個學生姓名：李四端 Enter↓
輸入國文成績：78 Enter↓
輸入英語成績：45 Enter↓
輸入第 3 個學生姓名：林森春 Enter↓
輸入國文成績：97 Enter↓
輸入英語成績：84 Enter↓
輸入第 4 個學生姓名：陳明通 Enter↓
輸入國文成績：83 Enter↓
輸入英語成績：73 Enter↓
輸入第 5 個學生姓名：吳仁康 Enter↓
輸入國文成績：82 Enter↓
輸入英語成績：77 Enter↓
```

```
姓名 國文 英語 平均
```

```
=====
林森春 97 84 90.5
王大維 92 72 82.0
吳仁康 82 77 79.5
陳明通 83 73 78.0
李四端 78 45 61.5
```

程式碼

```
檔名：\ex05\test05_1.py
```

```
01 name = ['' for x in range(5)]
02 score = [[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0]]
03 # 輸入資料
04 for i in range(5):
05     print('輸入第 %d 個學生姓名：' %(i+1), end=' ')
06     name[i] = input()
07     print('輸入國文成績：', end=' ')
08     score[i][0] = eval(input())
09     print('輸入英語成績：', end=' ')
10     score[i][1] = eval(input())
11     score[i][2] = (score[i][0] + score[i][1]) / 2
12 # 氣泡排序法
13 for loop in range(1,5):
14     for index in range(0, (5-loop)):
```

```

15         if score[index][2] < score[index+1][2]:
16             Temp = score[index]
17             score[index] = score[index+1]
18             score[index+1] = Temp
19             nTemp = name[index]
20             name[index] = name[index+1]
21             name[index+1] = nTemp
22 # 表列
23 print()
24 print('姓名      國文   英語   平均')
25 print('=====')
26 for p in range(5):
27     print(name[p], end = '    ')
28     print(score[p][0], end = '    ')
29     print(score[p][1], end = '    ')
30     print(score[p][2])

```



說明

1. 第 1~2 行：建立 **name** 串列，用來存放使用者所輸入的姓名；建立 **score** 串列，用來存放使用者所輸入的國文、英語分數成績。
2. 第 4~11 行：依序輸入學生姓名及國文、英語分數成績。其中 **score[i][2]** 用來存放國文、英語兩科的平均分數。
3. 第 13~21 行：依平均分數的遞減排序，相對應的學生姓名串列 **name**、成績串列 **score** 的元素資料隨平均分數排列時同步交換元素內容。
4. 第 24~30 行：顯示兩串列遞減排序後的所有資料。



題目 (二)

(APCS 檢測試題)

使用串列使 4*3 矩陣順時針旋轉 90 度轉換成 3*4 的矩陣。



結果

矩陣旋轉前：

```

1 2 3
4 5 6
6 7 8
10 11 12

```

矩陣旋轉後：

```

10 6 4 1

```

11	7	5	2
12	8	6	3


程式碼

檔名：\ex05\test05_2.py

```

01 a1 = [[1,2,3],[4,5,6],[6,7,8],[10,11,12]]
02 a2 = [[0,0,0,0],[0,0,0,0],[0,0,0,0]]
03
04 print('矩陣旋轉前:')
05 for i in range(4):
06     for j in range(3):
07         print('%2d' %a1[i][j], end = ' ')
08     print()
09 # a1 矩陣順時針旋轉 90 度的轉換運算成 a2 矩陣
10 for k in range(3):
11     for h in range(4):
12         a2[k][h] = a1[4-h-1][k]
13
14 print()
15 print('矩陣旋轉後:')
16 for i in range(3):
17     for j in range(4):
18         print('%2d' %a2[i][j], end = ' ')
19     print()

```


說明

- 第 1~2 行：a1=[[1,2,3], [4,5,6], [7,8,9], [10,11,12]] 為旋轉前的 4*3 矩陣內容，而 a2 串列用來存放 a1 矩陣順時針旋轉 90 度轉換後的 3*4 矩陣內容。a2 串列的所有元素內容先預設為 0。
- 第 10~12 行：a1 矩陣順時針旋轉 90 度的轉換運算成 a2 矩陣。觀察 [0][0]⇒[0][3]、[0][1]⇒[1][3]、[0][2]⇒[2][3]、[1][0]⇒[0][2]...，會發現轉換後第一維註標值等於原第二維註標值，而第二維註標值會由 0~3。

a1 矩陣(4*3)

[0][0]	1	[0][1]	2	[0][2]	3
[1][0]	4	[1][1]	5	[1][2]	6
[2][0]	7	[2][1]	8	[2][2]	9
[3][0]	10	[3][1]	11	[3][2]	12



a2 矩陣(3*4)

[0][0]	10	[0][1]	7	[0][2]	4	[0][3]	1
[1][0]	11	[1][1]	8	[1][2]	5	[1][3]	2
[2][0]	12	[2][1]	9	[2][2]	6	[2][3]	3

**題目 (三)**

(APCS 檢測試題)

下列程式片段中，若依序輸入整數 0、1、2、3、4、5、6、7、8、9，

```
01 arr = [0 for x in range(10)]
02 for i in range(10):
03     arr[(i+2) % 10] = eval(input())
```

則在 arr 串列的元素值依順序為何？

- (A) [0, 1, 2, 3, 4, 5, 6, 7, 8, 9] (B) [2, 0, 2, 0, 2, 0, 2, 0, 2, 0]
 (C) [9, 0, 1, 2, 3, 4, 5, 6, 7, 8] (D) [8, 9, 0, 1, 2, 3, 4, 5, 6, 7]

**說明**

1. 答案是 (D)，程式碼請參考 test05_3.py。
2. 所輸入的數值存入串列元素的註標順序為：arr[2]、arr[3]、arr[4] ... arr[9]、arr[0]、arr[1]。

**題目 (四)**

(APCS 檢測試題)

經過運算後，下列程式片段的輸出為何？

```
01 for i in range(100):
02     b[i] = i
03 a[0] = 0
04 for i in range(1,100):
05     a[i] = b[i] + a[i-1]
06 print(a[50] - a[30])
```

- (A) 1275 (B) 20 (C) 1000 (D) 810

**說明**

1. 答案是 (D)，程式碼請參考 test05_4.py。
2. $a[1] = b[1] + a[0] = 1 + 0 = 1$
 $a[2] = b[2] + a[1] = 2 + 1 = 3$
 $a[3] = b[3] + a[2] = 3 + 3 = 6$
 \vdots
 $a[30] = b[30] + a[29] = 30 + 435 = 465$
 \vdots
 $a[50] = b[50] + a[49] = 50 + 1225 = 1275$
 故 $a[50] - a[25] = 1275 - 465 = 810$

 題目 (十一)

(MAT 模擬試題)

一個名為 `emp` 的清單中有 150 個員工姓名，最後 10 名為臨時員工。若要顯示不含臨時員工的員工姓名，則這分割清單的表示法如何？

- (A) `emp[0:-9]` (B) `emp[1:-10]` (C) `emp[:-10]` (D) `emp[:-9]`



說明

1. 答案是 (C)。
2. `emp[:-10]` 與 `emp[0:-10]` 相同。

 題目 (十二)

(MAT 模擬試題)

一個名為 `colors` 的清單中有 256 個顏色值。若要分割這個清單，使顯示由清單的第 2 個顏色值開始，間隔 3 個顏色值後再顯示下一個顏色值，直到最後。則這分割清單的表示法如何？

- (A) `colors[2::4]` (B) `colors[1::4]` (C) `colors[::3]` (D) `colors[1::3]`



說明

1. 讀取串列元素的語法為：串列名稱[start:end:step]。題目是由第 2 個顏色值開始，所以 `start` 為 1(起始值為 0)。因為是讀取到最後，所以 `end` 可以省略。題目是讀取間隔 3 個顏色值後的第 4 個顏色值，所以 `step` 為 4。
2. 答案是 (B)。

5.8 習題

一. 選擇題

1. 建立串列，以哪種括號來存放元素？
① `{}` ② `[]` ③ `<>` ④ `()`
2. `arr = [17,8,-9,24,-3,20]`，請問 `arr[4]` 是多少？
① -3 ② 24 ③ -9 ④ 20
3. `word = ['甲', '乙', '丙', '丁', '甲', '甲', '乙', '辛']`，則 `word[-3]` 的內容是什麼？
① '丁' ② '甲' ③ '乙' ④ '丙'
4. `word = ['甲', '乙', '丙', '丁', '甲', '甲', '乙', '辛']; n = len(word)`，則 `n` 的值為何？
① 8 ② 9 ③ 7 ④ 10