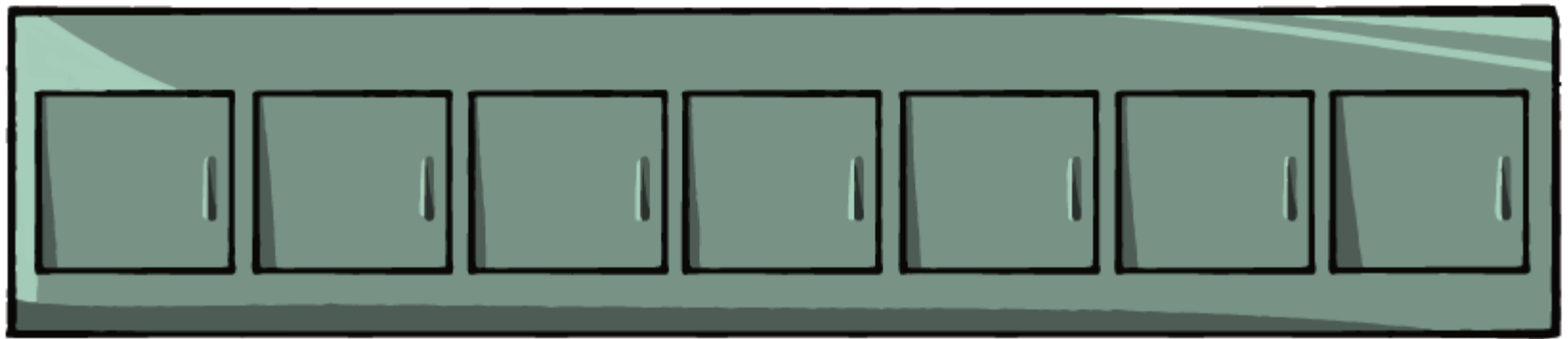


List

Python 的串列（list）相當於其他程式語言的陣列（array），這是一個用來儲存多個變數很好的方法。串列有一維串列、二維串列，以及多維串列。若串列沒有指明其維度時，則預設為一維串列（one dimension list）。



● 圖8-1 一維串列示意圖

8-1 建立串列

```
list1 = []
```

表示建立一空的串列list1。

```
list2 = [1, 2, 3, 4, 5]
```

表示建立一含有5個數值元素的串列list2。

```
list3 = ['Banana', 'Apple ', 'Orange ']
```

表示建立含有三個字串元素的串列list3。



8-2 計算串列的長度

■ 串列的長度其實就是串列中有多少個元素。若要計算某一串列的長度，可使用len函式，如：

```
➤>>> fruitsList= ['Banana', 'Apple', 'Grape', 'Pineapple',  
    'Orange']
```

```
➤>>> len(fruitsList)
```

```
➤5
```

■ 上述敘述表示有一fruitsList串列，利用len函式計算此串列的長度為5，計表示它有5個元素。

8-3 如何存取串列的元素

fruitList(0)	fruitList(1)	fruitList(2)	fruitList(3)	fruitList(4)
'Banana'	'Apple'	'Grape'	'Pineapple'	'Orange'

● 圖8-2 fruitsList串列有5個元素，範圍從0至4

■ 下列的程式碼將fruitsList的元素一一加以印出：

- fruitsList = ['Banana', 'Apple', 'Grape', 'Pineapple', 'Orange']
- for i in range(len(fruitsList)):
- print(fruitsList[i])

p8-5.py
p8-6.py

輸出結果

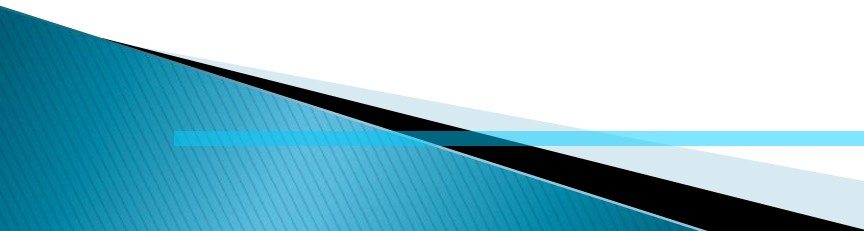
Banana

Apple

Grape

Pineapple

Orange



■ 可以連元素的名稱變數也一齊印出，如以下程式：

- `fruitsList = ['Banana', 'Apple', 'Grape', 'Pineapple', 'Orange']`
- `for i in range(len(fruitsList)):`
- `print('fruitsList[%d]=%s'%(i, fruitsList[i]))`

p8-7.py





輸出結果



```
fruitsList[0]=Banana
```

```
fruitsList[1]=Apple
```

```
fruitsList[2]=Grape
```

```
fruitsList[3]=Pineapple
```

```
fruitsList[4]=Orange
```

-
- 注意，存取超過串列範圍的元素是初學者常犯的錯誤，為了避免此種錯誤，必須確認使用的索引不可超過 `len(fruitsList)-1`。

- Python 允許負的索引值參考其串列相對的位置。
- 而其實際的位置是將串列的長度與負的索引相加。沿用上述的fruitsList串列，如以下範例所示：

➤ >>> fruitsList[-1]

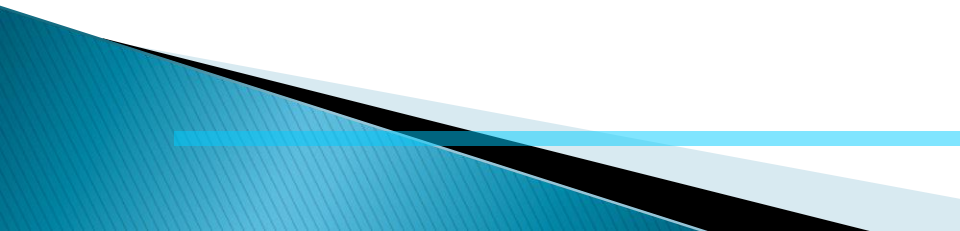
➤ 'Orange'

➤ >>> fruitsList[-2]

➤ 'Pineapple '

p8-8.py

-
- 第1行的`fruitsList1[-1]`與`furitsList[-1 + len(fruitList)]`相同，亦即`fruitsList[4]`，表示最後的元素。
 - 在第3行，`furitsList1[-2]`與`fruitsList[-2 + len(furitsList)]`相同，表示串列的倒數第二個元素。

-
- 若只要串列的部份連續元素呢？例如我只要第1個到第3個元素即可，那該如何表示呢？
 - Python 允許妳選擇特定索引的元素，此時就要使用 `[start : end]` 分割運算子（slicing operator）。
 - `[start : end]` 回傳從 `start` 到 `end - 1` 的子串列。
- 

■ 沿用上述的fruitsList串列，如以下範例程式：

➤ >>> print(fruitsList[1:3])

➤ ['Apple', 'Grape']

■ 表示印出fruitsList串列索引1到2的元素。



-
- 起始的索引和結束的索引是可以省略的。此程式的起始索引是0，結束索引是最後索引，例如：

- >>> print(fruitsList[:3])

- ['Banana', 'Apple ', 'Grape']

- 表示從索引0到2。

➤ >>> print(fruitsList[1:])

➤ ['Apple ', 'Grape', 'Pineapple ', 'Orange']

■ 程式中的 `list1[:3]` 和 `list1[0:3]` 是一樣的，而 `list1[1:]` 和 `list1[1:len(list1)]` 是相同的。

■ 也可以在分割運算子的部份使用負的索引。例如：

➤ `>>> print(fruitsList[1:-1])`

➤ `['Apple', 'Grape', 'Pineapple']`

p8-11.py



8-4

利用 `append` 和 `insert` 方法加入一元素於串列

➤ `>>> fruitsList.append('Kiwi')`

➤ `>>> fruitsList`

➤ `['Banana', 'Apple', 'Grape', 'Pineapple', 'Orange', 'Kiwi']`

- 第一行表示將 'Kiwi' 加入到 `fruitsList` 串列的尾端。在 IDLE 的模式下可以直接以 `fruitsList` 列印此串列的所有元素。

p8-12.py

■ 除此之外，也可以使用 `insert(i, x)`，將 `x` 元素加入於串列的 `i` 位置，如下程式所示：

➤ `>>> fruitsList.insert(0, 'Guava')`

➤ `>>> fruitsList['Guava', 'Banana', 'Apple', 'Grape', 'Pineapple', 'Orange', 'Kiwi']`

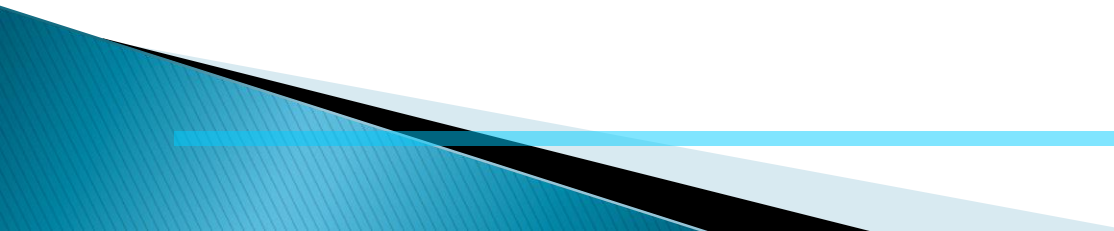
p8-13.py

■若要檢視某一元素x在串列的索引位置，則可利用 `index(x)` 方法，如下程式所示：

➤`>>> fruitsList.index('Orange')`

➤5

■承上述的fruitsList串列，得知 'Orange' 元素在串列中的索引位置是5。



8-5

利用 pop 和 remove 方法刪除串列的元素

- 從串列中刪除一元素也有兩種，其一為 `pop(i)`，表示將 `i` 索引的元素值刪除之，如下程式所示：

➤>>> fruitsList

➤['Guava', 'Banana', 'Apple', 'Grape', 'Pineapple', 'Orange',
'Kiwi']

➤>>> fruitsList.pop(2)

➤'Apple'

➤>>> fruitsList

➤['Guava', 'Banana', 'Grape', 'Pineapple', 'Orange', 'Kiwi']

■ 第三行敘述表示將fruitsList 串列中索引2 的元素
'Apple' 刪除。

p8-15.py



-
- `pop(i)` 方法中的 `i`，也可以是負的
 - 其實 `pop(-1)` 相當於 `pop(-1+len(fruitsList))`，亦即是刪除 `pop(-1+5)` 的元素。
 - 另一個方法是利用 `remove(x)`，直接刪除串列中第一個出現 `x` 的元素。

p8-16.py
p8-16-1.py

8-6 排序：由大至小或由小至大

■ Python 利用 sort 方法，如下程式所示：

```
➤ >>> numbersList = [1, 3, 4, 2, 6, 5]
```

```
➤ >>> numbersList.sort()
```

```
➤ >>> numbersList
```

```
➤ [1, 2, 3, 4, 5, 6]
```

■ 這是由小至大，若要由大至小，只要再利用 reverse() 方法就可以，因為 reverse() 方法是將串列的元素反轉過來

p8-17.py

■ `reverse()` 方法不會排序，只是我們利用它的特性，將 `sort()` 方法排序後的由小至大資料，反轉為由大至小資料。

■若要計算串列中某一元素的個數，則可利用count方法完成。

➤>>> numbersList.append(3)

➤>>> numbersList

➤[6, 5, 4, 3, 2, 1, 3]

➤>>> numbersList.count(3)

➤2



8-7

其他有用的函式：`max()`、`min()`、`sum()`

 表8-1

函式	功能
<code>max(s)</code>	回傳s串列的最大的元素值
<code>min(s)</code>	回傳s串列的最小的元素值
<code>sum(s)</code>	回傳s串列的所有元素值之總和

■ >>> numbersList

■ [6, 5, 4, 3, 2, 1, 3, 3]

■ >>> max(numbersList)

■ 6

■ >>> min(numbersList)

■ 1

■ >>> sum(numbersList)

■ 27



■ 使用 + 運算子和extend 方法將兩個串列結合

➤ >>> list2 = [1, 3, 5, 7, 9]

➤ >>> list3 = [2, 4, 6, 8]

➤ >>> list5 = list2 + list3

➤ >>> list5

➤ [1, 3, 5, 7, 9, 2, 4, 6, 8]

8-8 判斷某一元素是否存在於串列中

■ 利用 **in** 或 **not in** 可判斷某一元素是否存在於串列中。

➤ >>> fruitsList

➤ ['Apple', 'Banana', 'Orange']

➤ >>> 'Apple' in fruitsList

➤ True

➤ >>> 'Guava' in fruitsList

➤ False

➤ >>> 'Guava' not in fruitsList

➤ True

8-9

利用for迴圈印出串列的每一元素

■ 下列的程式顯示目前在fruitsList串列的所有元素：

➤ for x in fruitsList:

➤ print(x)

➤ print()

輸出結果

Apple

Banana

Orange

-
- 追蹤在fruitsList的每一元素x，並將它顯示出來。上述的輸出結果，顯示目前的fruitsList串列有三個元素。
 - 由於沒有使用索引變數，所以要印出以下的結果：
 - fruitsList[0] = Apple
 - fruitsList[1] = Banana
 - fruitsList[2] = Orange

■ 需要另一變數加以輔助才可。如下程式所示：

➤ `fruitsList = ['Apple', 'Banana', 'Orange']`

➤ `i=0`

➤ `for x in fruitsList:`

➤ `print('fruitsList[%d] = %s'%(i, x))`

➤ `i += 1`

➤ `print()`

■ 上述程式以i來加以輔助索引變數。

p8-18.py

8-10 串列的比較

■ 用關係運算子（>、>=、<、<=、+=<，以及 !=）來比較串列的元素。

➤>>> list10 = ['Apple', 'Banana', 'Guava', 'Kiwi']

➤>>> list20 = ['Apple', 'Banana', 'Guava', 'Orange']

➤>>> list10 > list20

➤False

➤>>> list10 < list20

➤True

➤>>> list10 == list20

➤False

8-11 如何將串列的元素重排

■ 串列元素要加以隨機重排，可以利用random模組下的shuffle() 方法。

```
➤>>> import random
```

```
➤>>> cardsList = ['Ace', 'King', 'Queen', 'Jack', 10, 9, 8, 7, 6, 5, 4, 3,  
2]
```

```
➤>>> random.shuffle(cardsList)
```

```
➤>>> cardsList
```

```
➤['Queen', 9, 'Ace', 2, 7, 6, 8, 'King', 5, 3, 10, 'Jack', 4]
```

```
➤>>> random.shuffle(cardsList)
```

```
➤>>> cardsList
```

```
➤[7, 9, 5, 2, 3, 8, 4, 10, 'Queen', 6, 'Ace', 'Jack', 'King']
```

8-12

串列的函式、運算子與方法總整理

 表8-2

函式	說明
len(List)	計算List串列的長度。
max(List)	求出List串列的最大值。
min(List)	求出List串列的最小值。
sum(List)	加總List每一的元素值。

運算子	說明
[a]	存取串列索引值為a的元素。
[begin: end]	存取串列索引值為begin到end-1的元素。
in	判斷某一元素是否存在於串列中。
not in	判斷某一元素是否不存在於串列中。
+	串列的結合。
*	複製串列的元素。
<, <=, >, >=, ==, !=	比較兩個串列的大小。

方法	說明
index(x)	檢視x元素於串列的索引。
append(x)	將x附加於串列的尾端。
insert(i, x)	加入x 於串列的i索引。
pop(i)	刪除串列索引為i的元素。
remove(x)	刪除串列第一個出現x元素。
extend(List)	將List串列的元素附加於某一串列。
count(x)	串列中出現x元素的個數。
sort()	將串列的元素由小至大排序。
reverse()	將串列的元素加以反轉。
shuffle(List)	將List串列的元素重排。