

第六章、元組、串列、字典

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，必須使用 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，在記憶體中占用連續的記憶體位址。好像一列台車，每個車斗都有標記連續的編號，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，則台車上的每個車斗我們稱之為「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。



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，只要改變串列中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

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, ...]

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。

串列常見的函式與方法

函式	使用說明(<code>list1 = [10, 20, 30, 40, 50]</code>)
<code>len()</code>	計算串列長度，可取得串列元素的數目。 例： <code>num = len(list1) #num ← 5</code>
<code>sum()</code>	進行串列中所有元素的加總。 例： <code>total = sum(list1) #total ← 150</code>
<code>max()</code>	取得串列中元素的最大值。 例： <code>big = max(list1) #big ← 50</code>

min()	取得串列中元素的最小值。 例：small = min(list1) #small ← 10
-------	--

方法	使用說明(list1 = [10, 20, 30, 40, 50])
append(value)	新增串列元素，將 value 附加到串列最後一個元素的後面。 例：list1.append(66) # list1 = [10, 20, 30, 40, 50, 66]
Insert(index, value)	在串列標註 index 處插入一個值為 value 的元素。 例：list.insert(2, 77) #list1 = [10, 20, 77, 30, 40, 50]
pop()	刪除串列中最後一個元素。 例：list1.pop()
sort()	串列的元素，由小到大排列。
reverse()	串列的元素，可以按反方向重新排列元素的順序。

運算子	使用說明(list1 = [10, 20, 30, 40])
in	判斷指定的資料是否存在於串列中。 例：tf = 40 in list1 #tf ← True 例：tf = 95 in list1 #tf ← False
=	將串列的記憶體位置指定給另一個變數，此時兩者占用同一個串列內容。 例：data1 = list1 #data1 和 list1 都是[10, 20, 30, 40] 此時當 data1 中的元素改變，list1 也會跟著改變。 例：data1[2] = 33 #data1 和 list1 都是[10, 20, 33, 40]
+	連結兩個串列元素。 例：list2 = [66, 77, 88] list3 = list1 + list2 #list3 = [10, 20, 30, 40, 66, 77, 88]
*	複製串列元素。 例：list4 = 2 * list1 #list4 = [10, 20, 30, 40, 10, 20, 30, 40] list5 = list * 3 #list5 = [10, 20, 30, 40, 10, 20, 30, 40, 10, 20, 30, 40]

元組型別

元組(tuple)簡單來說就是由多個元素所組成的群組，元組的成員可以由不同型別的資料來組成。元組宣告後元素個數及元素值都不可以更改，其序列是固定的。因此希望能保護資料時，就可以採用元組行別。宣告元組時以逗號區隔，可以用「()」括住數列亦可省略，語法範例如下：

元組 = 元素 A, 元素 B, 元素 C,

元組 = (元素 A, 元素 B, 元素 C,))

字典型別

字典(dict)的每個元素是用冒號「:」分隔，冒號之前是鍵(Key)，冒號之後是值(Value)，換言之是

由「鍵:值」組成一個元素。元素間用逗號「,」分隔,所有元素被「{}」大括號包圍,即完成字典型別。字典顧名思義,就像我們使用的字典,可以根據英文單字查詢到中文解釋。例如 book 對應到「書本」,如果用字典語法可以寫成{'book': '書本'}。真實的字典是依規則排序以方便查詢,但是 python 中字典的元素是無序列儲存的,其語法如下:

字典 = {鍵 1: 值 1, 鍵 2: 值 2, 鍵 3: 值 3,}

1. 鍵必須是唯一而且不可變的,而鍵可以使用的型別為數值、字串或元組。
2. 值可以是任何資料型別,除了常用型別外,串列、元組、字典...等也可以。