

重複主題（一）：資料容器

by 田弘華 **Hung-Hua Tien**

1. 串列

1-1 資料結構

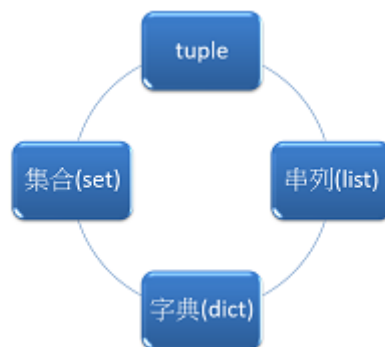
從純量（**scalars**）邁向結構(**collections**)

(1)Python 的資料單位稱作純量（**scalar**）。

純量扮演的角色就像是樂高、積木、模型或拼圖等遊戲中的基本單元零件，透過結合基本資料單位，堆疊出更進階的 **Python** 應用。常見的有數字（整數、浮點數）、文字（字串）與布林。

(2)Python 的資料容器有字串（**str**）+ **list** / **tuple** / **dict** / **set**

Python的資料容器可以讓我們將多個資料（複合資料）指派給一個物件。



～資料容器的特性整理

- 元組(**tuple**)：資料放置於小括弧（**()**）內，資料有順序性，是不可變物件。
- 串列(**list**)：資料放置在中括弧 [**]** 內，資料有順序性，是可變物件。
- 字典(**dict**)：是**dictionary**的縮寫，資料放置在大括弧**{}**內，是「鍵(**key**)」與「值(**value**)」對應的物件，是可變物件。
- 集合(**set**)：類似數學裡的集合概念，資料放置在大括弧**{}**內，是可變物件，資料沒有順序性，資料也不會重複出現。

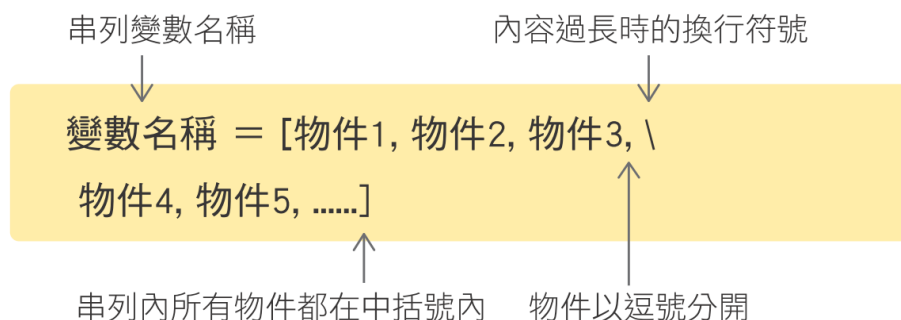
1-2. 資料容器-串列**list**

串列的建立

Python中最有用的資料型態之一是串列。誠如其名，串列是名單、清單或物件的集合。在程式碼中，創立串列，需指派一個名稱並將它分配給一個我們希望保存的物件集合。這個物件集合保存在中括號(Brackets)之中，它長的像這樣[]，並用逗號分隔物件。

~list 是 Python 常用的資料結構，可以容納不同類型的純量與不同長度的資料結構。

- 建立串列的時候
 - 用中括號 [] 將希望儲存的資訊包括起來。
 - 使用list()函數 list() 來創建。



▲ 圖 4-2.2 串列的定義

```
#1. 建立串列的方法
## **方法（一）：使用中括號 [] 創建**
shoplist = ['牛奶', '蛋', '咖啡豆', '西瓜', '鳳梨'] #文字
print(shoplist)

listm = [1, 2.0, 'Python'] #串列可以包含各種資料型別的元素
print(listm)

## 方法（二）：使用內建函數 list() 變更。
### (1) 字串變串列
list1 = list('python')
print(list1)
### (2) 元組變串列
tuple2 = ('a', 'b', 1, 2)
list2 = list(tuple2)
print(list2)
```

串列項目

當創立一個串列時，我們不僅保存物件的集合，更是保存它們的排序。排序很重要，這會影響更改、存取串列物件的方式，以及如何比較物件和串列。串列必須具有相同的物件和相同的排序才能算是真正的相等。

串列是可變動的。一旦創立串列，就可以添加新物件、刪除現有物件，並移動物件。

我們可以把串列想像成一個倉庫，裡面有一系列的箱子和櫃子。電腦會把串列裡面的項目從0開始編號。第一個項目就是串列裡的第0個項目。如果要取得裡面第i個項目，可以用中括號包含索引值來處理，也就是，利用串列名稱[i]做索引，取得當中項目來存取或變更。如果是多個項目，則用串列名稱[i:j]做切片處理。

```
# 串列項目
## 索引
shoplist = ['牛奶', '蛋', '咖啡豆', '西瓜', '鳳梨']
print(shoplist[0]) # 取第0個項目

## 變更
shoplist = ['牛奶', '蛋', '咖啡豆', '西瓜', '鳳梨']
shoplist[1] = '皮蛋' # 變更第1個項目
print(shoplist)

## 切片
shoplist = ['牛奶', '蛋', '咖啡豆', '西瓜', '鳳梨']
print(shoplist[1:3]) # 取1,2,3三個項目
```

範例：在變數best_chicago_bulls中，我們用list串列儲存了6個資料內容，分別是三個文字記錄球季、隊名與總教練；一個長度為2的list記錄了勝場數與敗場數；一個長度為5的list記錄了先發的五位球員；一個布林記錄是否有獲得該球季的總冠軍。

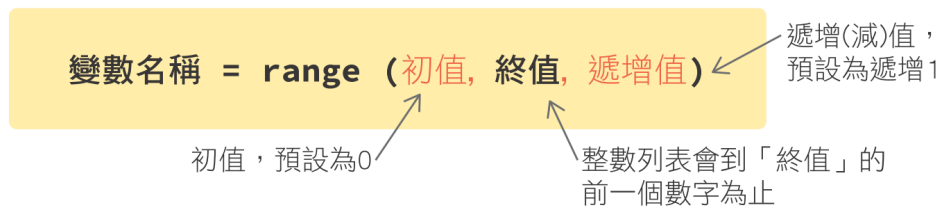
```
season = "1995-1996"
team = "Chicago Bulls"
coach = "Phil Jackson"
records = [72, 10]
starting_lineup = ["Ron Harper", "Michael Jordan", "Scottie
Pippen", "Dennis Rodman", "Luc Longley"]
won_championship = True

best_chicago_bulls = list((season, team, coach, records,
starting_lineup, won_championship))
print(best_chicago_bulls)
print(type(best_chicago_bulls))

# indexing 索引
print(best_chicago_bulls[0])
print(best_chicago_bulls[-1])
# slicing 切片
print(best_chicago_bulls[1:4])
print(best_chicago_bulls[::-1])
```

2. range()函數

range為python的內建函數，可以產生數字串列，給我們某一個範圍與特性的數字清單。



▲ 圖 4-3.3 range 語法說明

～(1)range 函數產生的序列為串列（List）結構。

- range函數三種表示方法
 - (1) range (start)
 - (2) range (start, stop)
 - (3) range (start, stop, step)

～(2)range函數的預設值

- 預設的初值是0、遞增值是1，結尾不含終值、是到終值的前一個數字。
 - 括號中若只有一個數值，是省略初值、遞增值、僅留終值，例如 range(10) 相當於range(0, 10)或range(0, 10, 1)。
 - 若省略遞增值，則變數每次自動+1；-1 則表示變數每次遞減1。
 - 變數等於終值時離開，所以執行的最後值是終值前的一個數字。
- 變數會依據初值到終值的變化，依序指定給左邊的變數名稱。
 - d[0] 代表變數 d 第 1 筆資料內容，因為電腦是從 0 開始編號。

例：range指令練習

```
a = range(5) # [0,1,2,3,4]
print(a)
print(a[0], a[1], a[2], a[3], a[4])

b = range(2, 6) # [2,3,4,5]
print(b[0], b[1], b[2], b[3])

c = range(2, 10, 3) # [2,5,8]
print(c[0], c[1], c[2])

d = range(10, 3, -2) # [10,8,6,4]
print(d[0], d[1], d[2])
```

程式碼說明

1. 第1行range(5) 相當於range(0, 5)，亦即最前面初值0是可以省略，可自行觀察第2行的執行結果，從字面來看是範圍0～5，但實際是0～4而已，要特別留意。
2. 第3行 print(a[0], a[1], a[2], a[3], a[4]) 會依序分別輸出 0 1 2 3 4，若第3行 不小心多a[5] 項目，則執行會出現「Index Error: range object index out of range超出索引範圍錯誤」。
3. 第4行range(2, 6) 相當於range(2, 6, 1)，亦即最後面遞增值1是可以省略，可觀察第5行的執行結果。
4. 第6行range(0, 6, 2) 表示遞增值2不可以省略，可觀察第7行執行結果。

5. 第8行`range(0, 6, -2)` 表示遞增值-2，相當於遞減2，可觀察第9行執行結果。