

第五章、重複結構

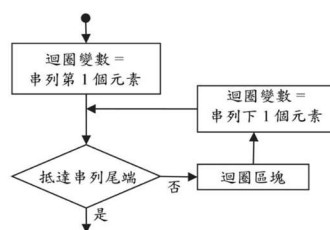
結構化的程式是以循序(sequence)、選擇(selection)及重複(repetition)三種基本邏輯架構所組成。上一章已經學習了「選擇結構」，本章將學習「重複結構」也稱為迴圈。程式流程如果執行到重複結構時，會一直重複迴圈敘述，直到某一特定條件不成立，或是一集合體中的所有元素都已巡訪過，程式流程才會脫離迴圈，繼續循序向下執行。換句話說，重複結構就是我們只需撰寫一次，就可以連續多次執行的程式碼區塊。

for 迴圈

for 迴圈通常配合串列來運作，迴圈執行時會依序取出串列元素，指定給迴圈變數，一直到串列內無元素為止。for 迴圈敘述要以「:」冒號為結尾，迴圈區塊要往後縮排。如果迴圈區塊只有一行程式碼時，迴圈區塊可以接續在冒號後面。語法如下：

for 迴圈變數 in 串列:

 迴圈區塊



練功坊

```
01 for x in 'Python':
02     print(x)
```

range()函式

range()是 Python 的內建函式，功能是產生一個整數串列，語法如下：

range(初值, 終值[, 間隔值])

1. 初值：是串列的初始值，為非必要參數。如果省略初值時，則預設串列初值
2. 終值：是串列的終止值，此為必要參數不可省略。
3. 間隔值：是串列的間隔值，該值可以為正值、負值或者省略。如果省略時，則預設間隔值為 1。

假如間隔值是負值，則串列是遞減序列。

練功坊

```
range(10)
>>> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
range(1, 7)
>>> 1, 2, 3, 4, 5, 6
range(2, 11, 2)
>>> 2, 4, 6, 8, 10
```

for 迴圈與 range() 函式

for 迴圈時常搭配 range() 函式來與用，由於 for 迴圈後方接的變數，因此也稱「變數迴圈」。又因為其使用的情境，往往是重複有規律的事情，用於計算確定的次數，因此又稱「計次迴圈」。for 迴圈與 range() 搭配使用的情況，變數就會依照 range() 函式所產生的陣列來改變。

練功坊

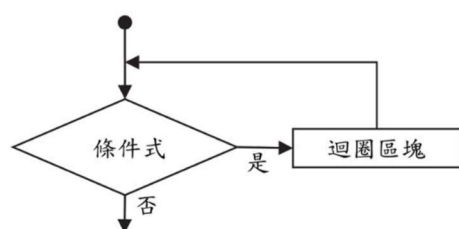
```
sum = 0
for i in range(1, 101):
    sum += i
print(sum)
```

while 迴圈

while 迴圈在迴圈開始前，會先檢查條件式是否成立，如果不成立，程式流程會略過迴圈區塊，向下繼續執行。反之，程式流程會在迴圈內循環，直到條件式的結果不成立，才會脫離迴圈。語法如下。

while (條件式):

迴圈區塊



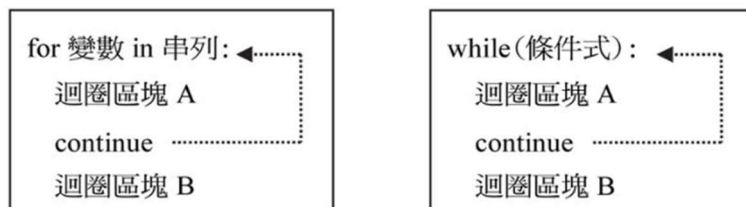
由於 while 迴圈後是接條件式，因此又可以稱為「條件迴圈」。跟 for 迴圈最大的不同是，本身無法宣告新的變數，變數必須在迴圈之前宣告。當條件成立時迴圈得以運作，因為是條件判斷，當條件不成立即跳出迴圈，因此 while 迴圈更加適合處理不確定次數的重複問題(例如：下雨天會持續撐傘直到沒有下雨)。請詳見以下語法差異：

for 迴圈	while 迴圈
<pre>sum = 0 for i in range(1, 101): sum += i print(sum)</pre>	<pre>i = 1 sum = 0 while i <= 100: sum += i i += 1 print(sum)</pre>

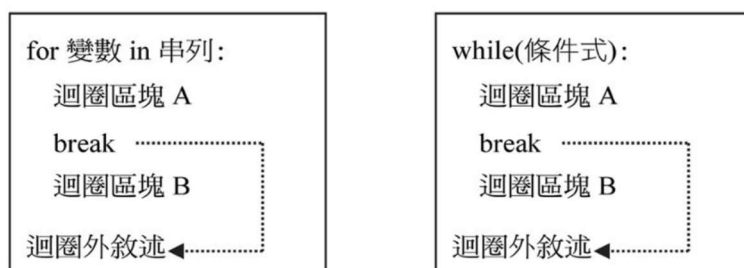
迴圈控制 continue 與 break

在某些情況下，我們可能有中斷迴圈或放棄其後敘述的需求，這時候就要使用 continue 或 break 敘述。因為這兩個敘述有強制性，換句話說，程式流程遇到這兩個敘述，流程會無條件轉移。所以在使用 continue 與 break 敘述時，必需配合選擇結構，以條件式來控制程式流向。

如果在迴圈區塊中，在某些條件下，要忽略其後的敘述，跳回迴圈開頭繼續執行，這時候就要使用 continue 敘述。continue 敘述的語法和流程如下：



在迴圈區塊中，當程式流程遇到 break 敘述時，會忽略其後的敘述，直接脫離迴圈，向下繼續執行。break 敘述的語法和流程如下：



巢狀迴圈、無窮迴圈

若迴圈內還有迴圈，一層一層由外而內即構成「巢狀迴圈」(Nested Loop)，也可以稱為「多重迴圈」。無論是 for 或 while 迴圈，或是同時使用都可以構成巢狀迴圈。撰寫巢狀迴圈程式時，要特別注意縮排程式才能正確執行。

假使迴圈的條件式永遠為 True(真)，會形成無窮迴圈，程式流程會周而復始的執行迴圈區塊。因此撰寫無窮迴圈內的迴圈區塊時，必需有改變條件的敘述，或者使用 break 作為脫離迴圈的出口。