

# 程式的流程控制-迴圈

# 學習目標

- 迴圈敘述 while 的用法
- 迴圈敘述 for 與 range() 的用法
- break、continue 與 else 的使用

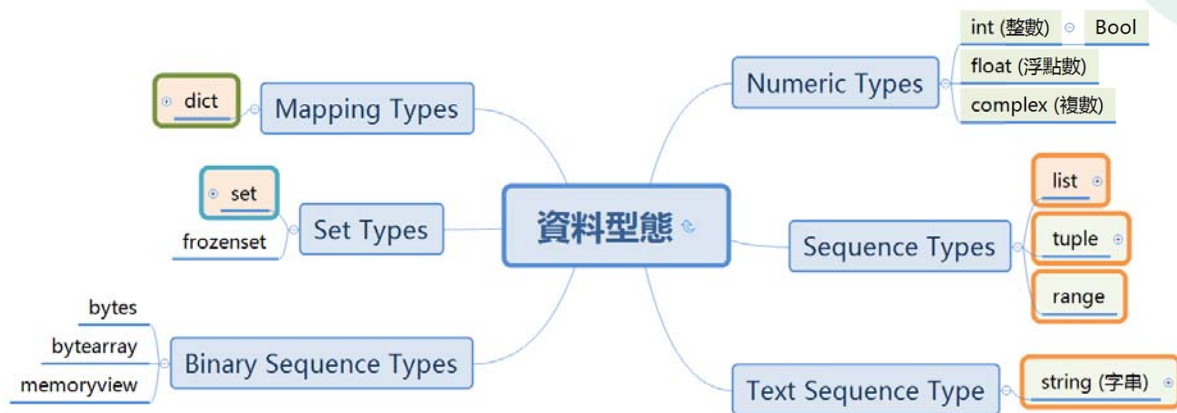


# 流程控制迴圈

## ■ 使用迴圈控制

□ while, for

## ■ range() 函數的用法



# 迴圈

## ■ while loop

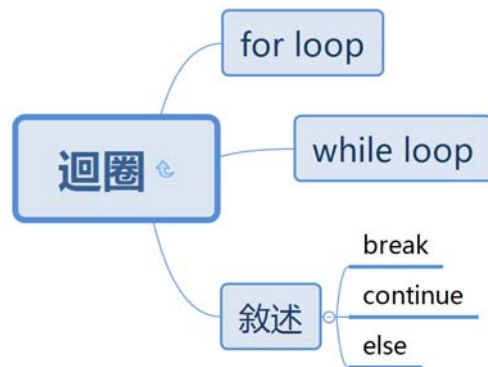
- 如果 while 的測試條件為 True，則執行迴圈敘述，直到 while 的測試條件為 False，則結束迴圈

## ■ for loop

- 依序處理一份有順序可迭代(iterable) 的資料，直到所有資料處完畢為止

## ■ Nested loops

- 迴圈內可以再包含迴圈，形成巢狀迴圈



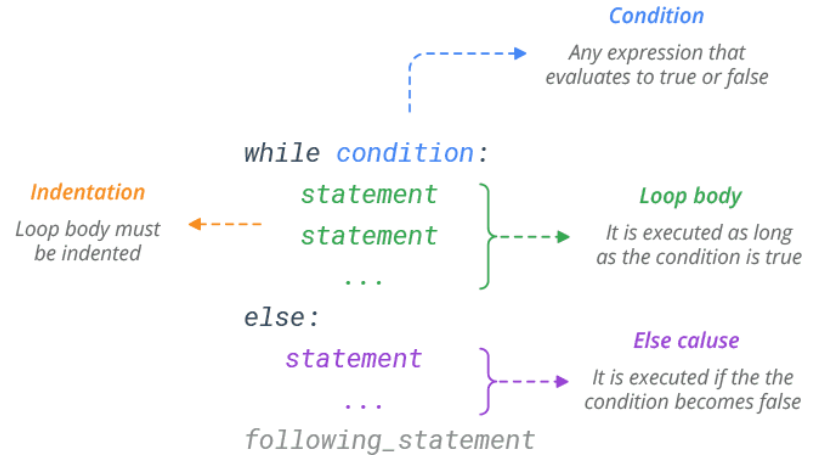
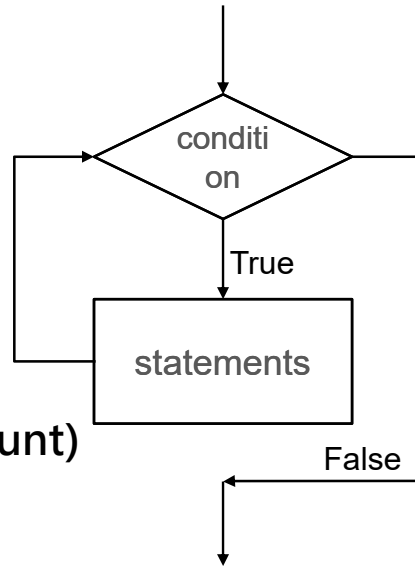
# while 迴圈

## ■ 語法

■ while expression:  
statement(s)

## ■ 範例

```
count = 0
while count < 9:
    print('count is:', count)
    count += 1
print('Finished!')
```



■ 如果一開始判斷結果就為假，迴圈不會執行

■ while loop可以不進入迴圈，即先判斷是否需進入迴圈，如過不進入迴圈，則可以跳過迴圈

# for 迴圈

## ■ 語法

□ `for item_var in sequence:`  
    `statements(s)`

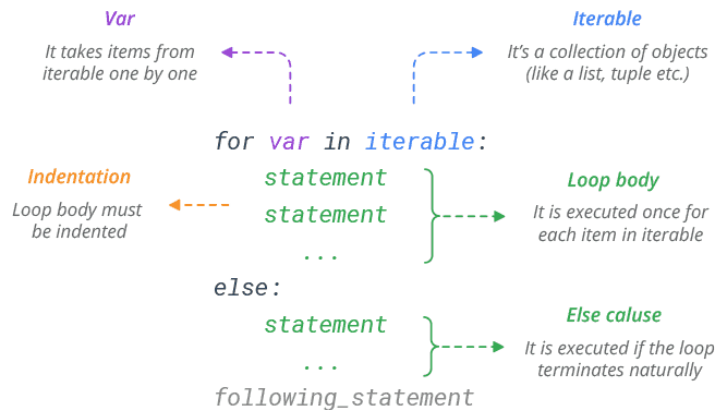
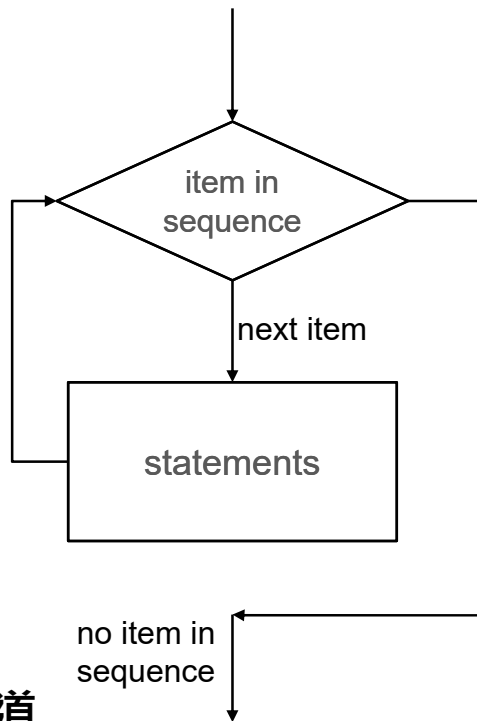
□ `sequence` 可以是任何可迭代的物件，例如字串(string)、元組(tuple) 或串列(list)

## ■ 範例

□ `for var in [1, 2, 3, 4, 5]:`  
    `print(var)`

■ `for loop`也可以不進入迴圈

■ `for loop`一開始，就已經知道未中斷的情況下將執行的次數



# Nested loop 敘述 -1

## ■ 語法

- for item\_var in sequence:  
    for item\_var in sequence:  
        statements(s)  
    statements(s)

- while expression:  
    while expression:  
        statement(s)  
    statement(s)

■ while 敘述可以包含 for 敘述，for 敘述也可包含 while 敘述，視實際狀況搭配使用

# Nested loop 敘述 -2

## ■ 範例

```
□ for i in range(1,10):  
    for j in range(1,10):  
        x = i * j  
        print(x, end=' ')  
    print()
```

```
□ 1 2 3 4 5 6 7 8 9  
  2 4 6 8 10 12 14 16 18  
  ...  
  8 16 24 32 40 48 56 64 72  
  9 18 27 36 45 54 63 72 81
```



# 內建函數 range()

■ Python 內建的 range() 函數可以產生一個數列

■ 語法

□ range(start, stop[, step])

□ start 引數預設為 0，step 引數預設為 1

■ 範例

□ >>> list(range(5))  
[0, 1, 2, 3, 4]

□ >>> list(range(1,5))  
[1, 2, 3, 4]

□ >>> list(range(1,5,2))  
[1, 3]

```
>>> for i in range(10): print(i,end="")
```

```
0123456789
```

```
>>> for i in range(5,10): print(i,end="")
```

```
56789
```

```
>>> for i in range(5,10,2): print(i,end="")
```

```
579
```

# 流程控制迴圈(Demo)

## ■ Python 程式的迴圈流程控制

- while、for 敘述

- range() 的用法

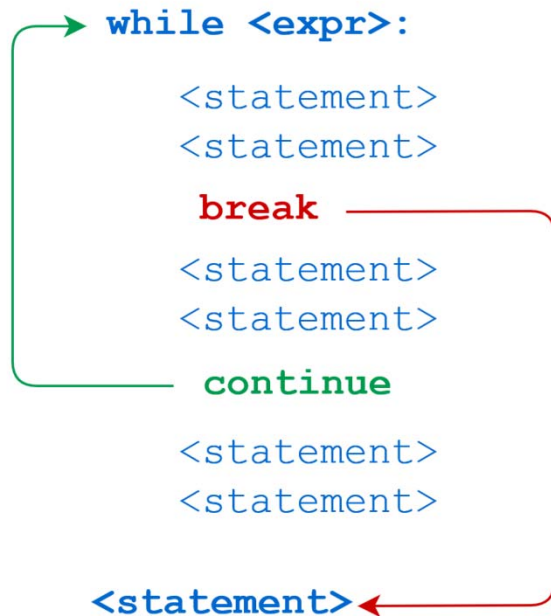


# 迴圈控制

## ■ 使用迴圈控制

□ break, continue

□ else

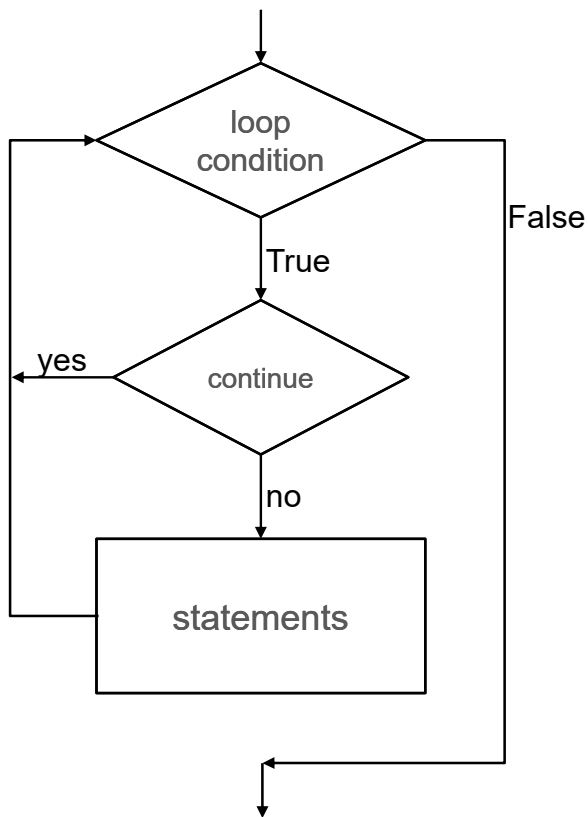
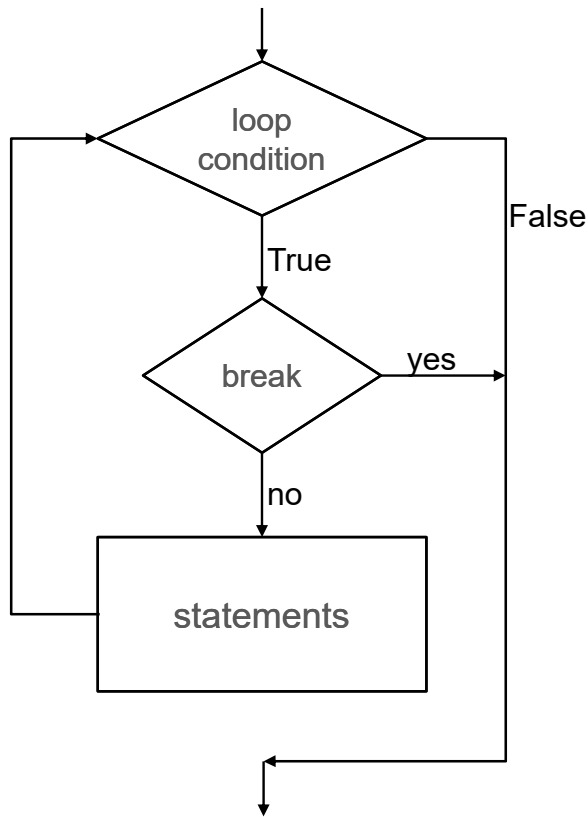


Section 2

# break 與 continue 敘述 -1

- break 敘述類似 C 語言，可以跳開迴圈
- continue 敘述也是由 C 語言衍生而來，略過目前迴圈的敘述，進行下一個迴圈(如果條件為真)
- 通常 break 與 continue 會配合 if 判斷式使用
- 語法
  - while condition | for item\_var in sequence :  
    statement  
    ...  
    if condition:  
        break | continue  
    ...  
    statement

# break 與 continue 敘述 -2



```
while <expr>:  
    <statement>  
    <statement>  
    break  
    <statement>  
    <statement>  
    continue  
    <statement>  
    <statement>  
<statement>
```

# 搭配迴圈的 else 敘述 -1

- Python 支援迴圈敘述結合 else 敘述
- 如果 else 敘述配合 for 迴圈，只有 for 迴圈處理完所有的項目後，才會進入 else 區塊
- 如果 else 敘述配合 while 迴圈，則只有 while 判斷式結果為假時，才會執行 else 區塊
- 如果是用 break 敘述跳開迴圈則不會執行 else 區塊
- 只有 for/while 迴圈正常結束時，才會執行 else 的敘述

## 搭配迴圈的 else 敘述 -2

### ■ 範例

```
numbers = [31,53,75,85,27,93,63,19]
for num in numbers:
    if num%2 == 0:
        print('Data contain even numbers')
        break
else:
    print('Data have no even numbers')
```

# 本章重點精華回顧

- 迴圈敘述 while
- 迴圈敘述 for 與 range()
- 迴圈控制
  - break 、 continue 與 else





# Lab: 程式的流程控制迴圈

- Lab01: 使用while, for, range() 迴圈流程控制

# Lab01: 使用 while, for, range()

1. 啟動Python IDLE環境，做以下練習
2. 使用 “File/Open...” 開啟 “whilestmt1.py” 程式，了解while敘述的用法
3. 關閉whilestmt1.py程式視窗
4. 使用 “File/Open...” 開啟 “whilestmt2.py” 程式，了解while搭配break敘述的用法
5. 關閉whilestmt2.py程式視窗
6. 使用 “File/Open...” 開啟 “forstmt1.py” 程式，了解for敘述的用法
7. 關閉forstmt1.py程式視窗
8. 使用 “File/Open...” 開啟 “forstmt2.py” 程式，了解range() 函數的用法
9. 關閉forstmt2.py程式視窗
10. 使用 “File/Open...” 開啟 “continuestmt.py” 程式，了解continue敘述的用法
11. 關閉continuestmt.py程式視窗
12. 使用 “File/Open...” 開啟 “elsestmt.py” 程式，了解迴圈配合else敘述的用法
13. 關閉elsestmt.py程式視窗。