

Python 程式設計入門 (三)

流程控制與迴圈

余家瑞 林口長庚醫院 檢驗醫學科

學習目標

- 了解與使用流程控制：if-elif-else
- 了解與使用迴圈
 - for-in迴圈
 - while迴圈

目的和原則

- 流程控制：讓電腦根據不同條件做事
- 迴圈：讓電腦重複做事
- 書寫通則
 - 冒號
 - 縮排 (Python的重要書寫原則)

Boolean 布林型別：代表真假的型態

整數	int
浮點數	float
字串	str
串列	list
序對	tuple
集合	set
字典	dict
布林	bool

只有兩個值：True & False

利用bool()將所有資料轉成布林型態

False	True
0	其他數值
""	其他字串
()	其他序對
[]	其他串列
set()	其他集合
{ }	其他字典
假的條件	真的條件

Comparison operator 比較運算子

大於	>
小於	<
大於等於	>=
小於等於	<=
是否等於	==
是否不等於	!=

「運算子」：和加減乘除一樣會得出一個結果
比較運算子只會得出True或False

Comparison operator 比較運算子

`score = 90`

將變數score設定為90

`score == 90`

score為一個已知數值

讓電腦判斷是否為90，回傳True或False

Logical operator 邏輯運算子

- `and`, `or` 用來陳述兩個以上的條件
- `not` 用來陳述相反的邏輯

流程控制與迴圈：書寫通則

- 利用Tab鍵或是四格空格縮排
- 必須要加冒號

if-elif-else 書寫法則

if 條件A:

程式碼_A

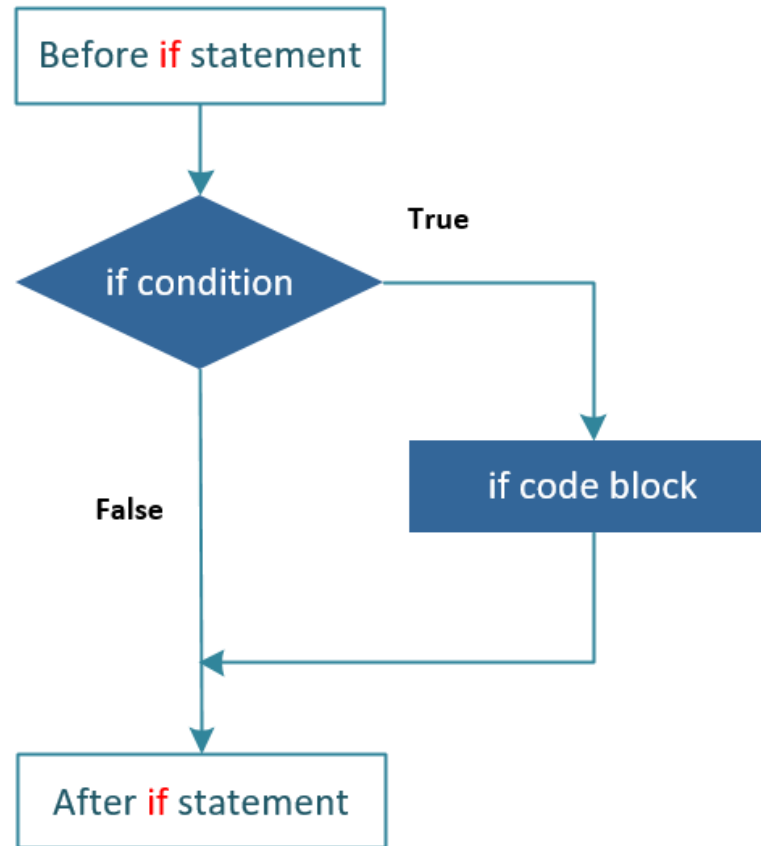
elif 條件_B:

程式碼_B

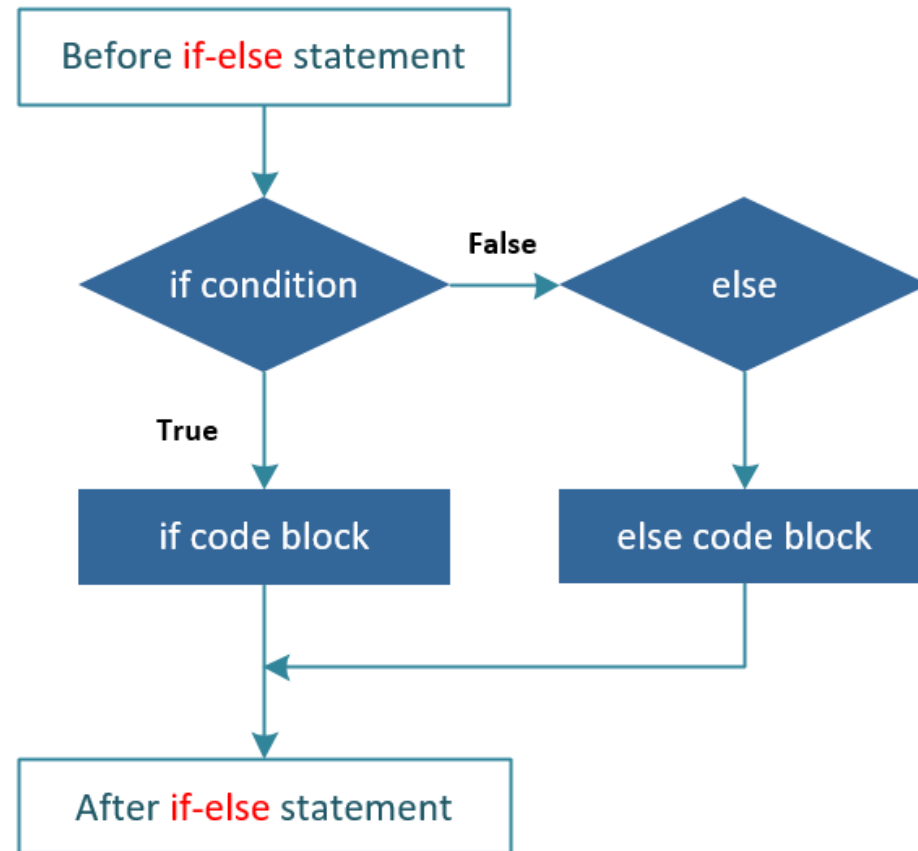
else:

程式碼_C

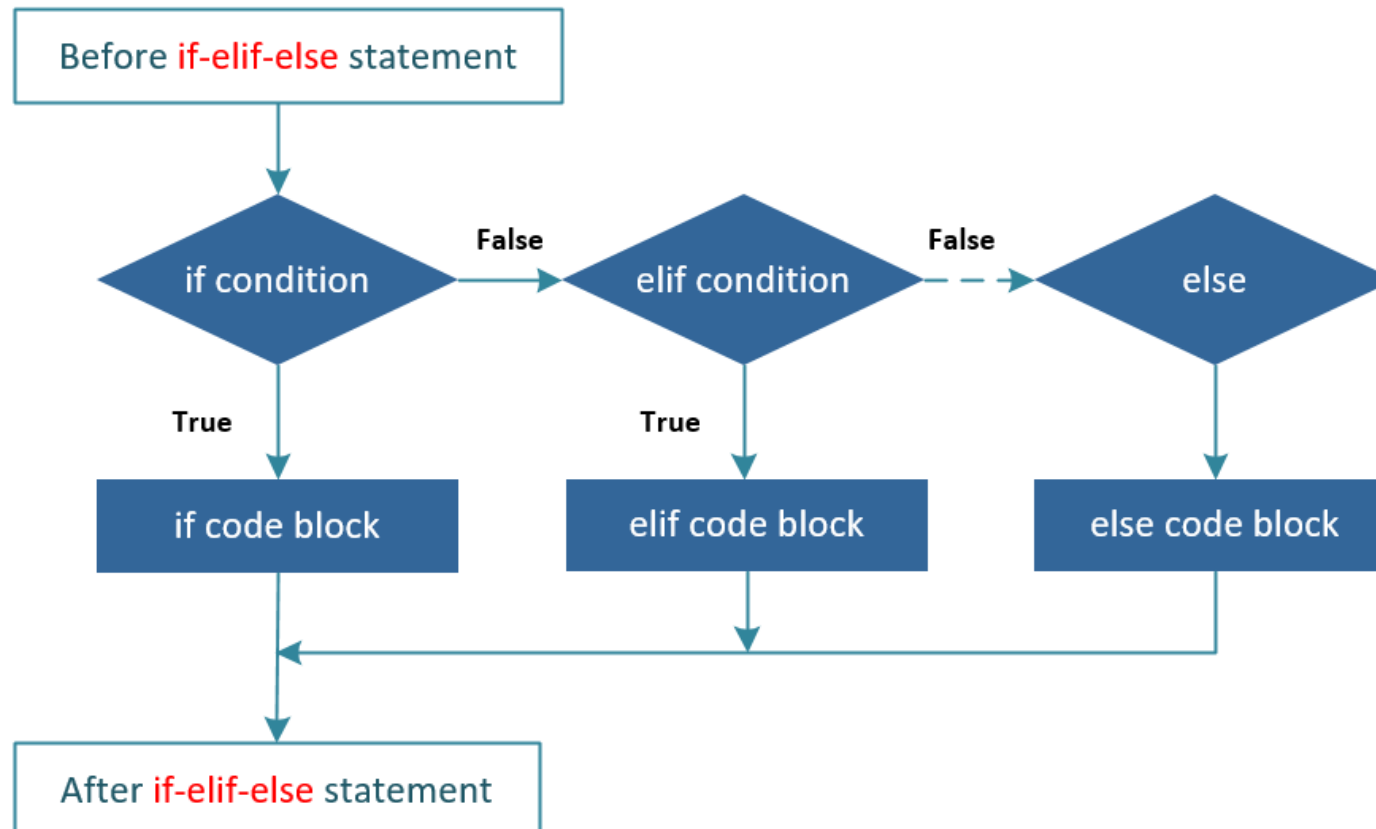
if



if-else

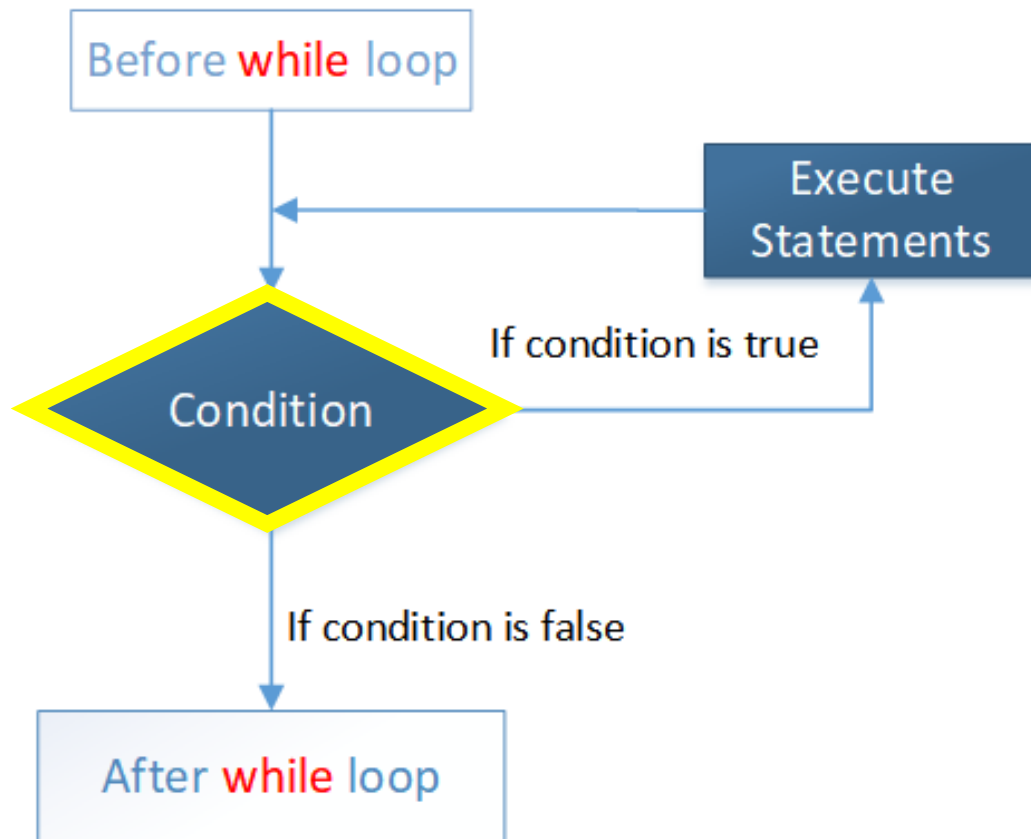


if-elif-else

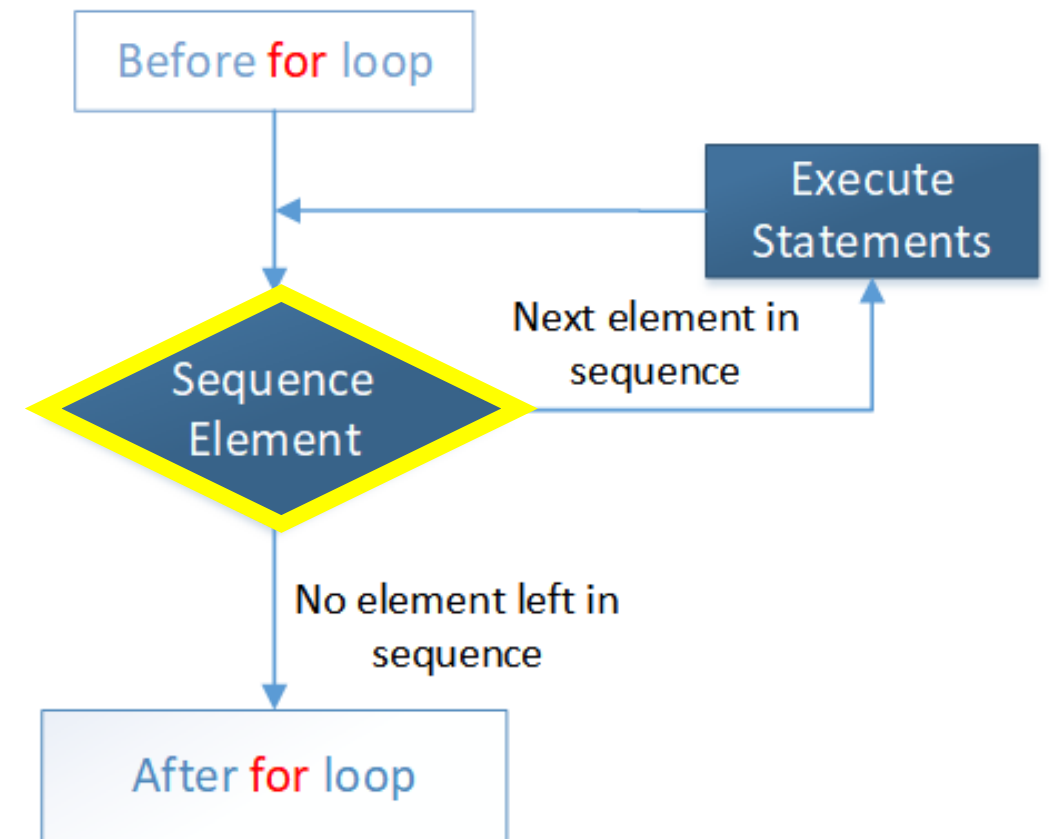


Loop 迴圈

while



for ... in ...



迴圈書寫法則

while

for ... in ...

while 條件:
程式碼

for 子物件 **in** 可迭代對象:
程式碼

能用其中一種迴圈表示的程式，
原則上都能用另一種表示！

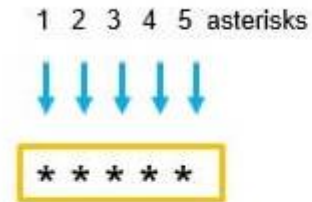
什麼是可迭代對象 (iterable) ?

整數	int
浮點數	float
字串	str
串列	list
序對	tuple
集合	set
字典	dict
「範圍」	range

Nested loop 巢狀迴圈

- 迴圈裡面還有迴圈。
- 創造出二維(或以上)的結構。

Step 1: Create a row of 5 asterisks with the help of the "for" loop.



Step 2: Display this row 3 times with the help of another "for" loop.



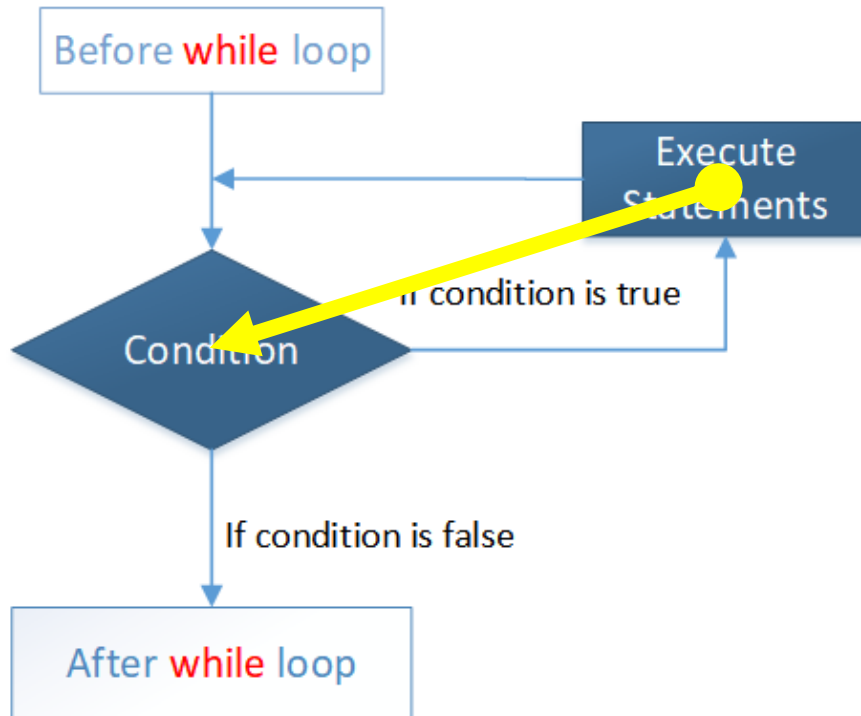
Infinite loop 無限迴圈

- 在python中的while loop可能出現
- 如果while後面的條件總是為真，迴圈會無止盡地跑下去！
- 避免無限迴圈的方法
 - 加入改變條件的程式碼
 - 利用break強制跳出迴圈

迴圈中止

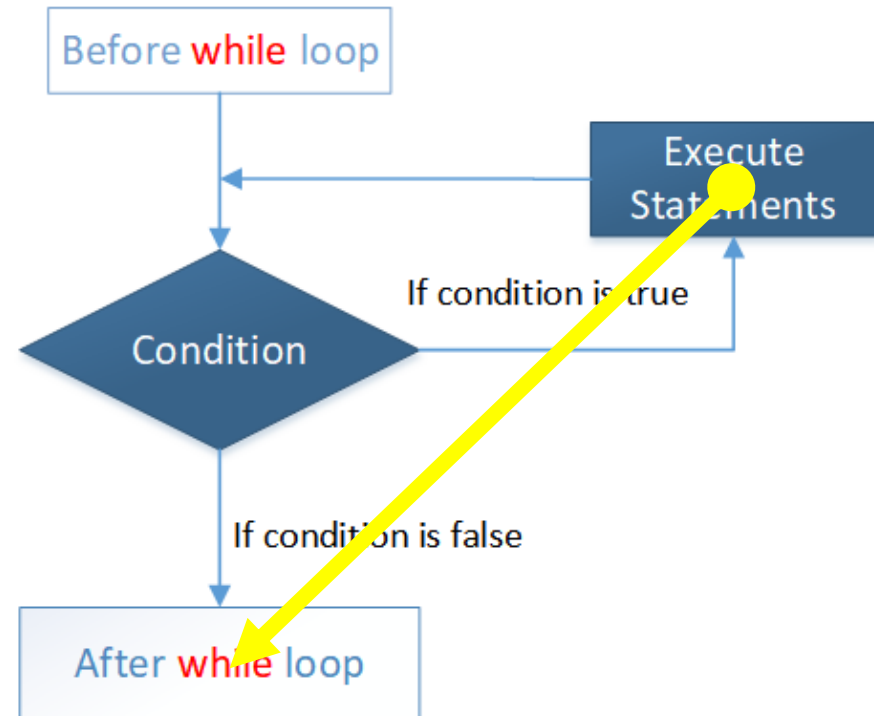
continue

跳過該子物件，迭代下個物件



break

跳出整個迴圈



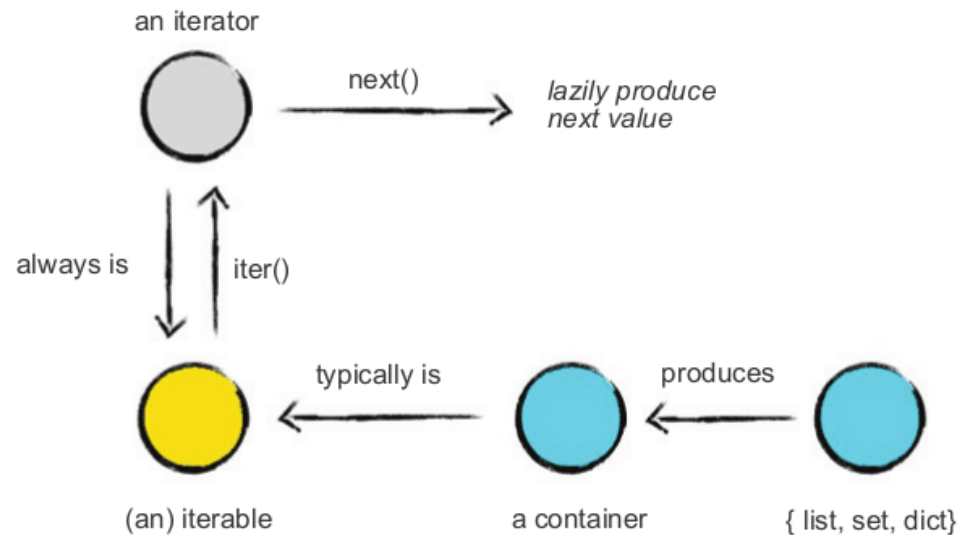
更多迭代器 (iterator)

- 除了常見資料型態，Python 的 for 迴圈還有各種不同的可迭代對象。
- 內建模組 itertools 有許多更多樣的迭代器。

<u>permutations()</u>	排列
<u>combinations()</u>	組合
<u>chain()</u>	鏈接二迭代對象
<u>product()</u>	組合二迭代對象

進階：iterable 和 iterator

- Iterator 和 iterable 略有不同。
- 能放入 for 迴圈者都是 iterable。在迴圈中，會用 iterable 產生對應的 iterator 進行操作。



你要知道的是：什麼物件可以使用 **for** 迴圈，和他們迭代的對象。

Python 程式設計入門 (三)

流程控制與迴圈

提問與回饋！