PREFACE

序言

由於大數據分析與人工智慧應用的興起，使得 Python 程式語言大受歡迎，不外乎它有很多以 Python 開發的套件可以加以使用。同時此程式語言也很容易入門，用它來撰寫程式以訓練邏輯思維的最佳選擇。

本書是筆者多年來在多所大學教授 Python 程式語言的筆記，將教學心得與學生所遇到的問題融入其中，希望它是教學與自習的最佳範本。本書共分 11 章（第 0 章～第 10 章），依照大部份在學習程式語言的主題順序加以編著而成，期盼你閱讀本書能收到事半功倍的效果。

內文淺顯易懂，並搭配範例程式加以解說，也在小節中有練習題，讓你動手做做看，以測試你對這一小節的了解程度，同時也在章末有代表的習題，測試你對本章是否有全盤的了解。最重要的是，不管練習題或是習題皆提供參考解答。不過在此，我有一小小的要求，不要先看參考解答，應該先自已做看看再來對解答。以此本為教科書的教授們，也會給予補充資料，以利於您出作業或考題。

本書僅涉及 Python 程式語言的主題，沒有論及 Python 的應用，如大數據分析或是人工智慧的機器學習和深度學習，因為它還需要一些套件才能發揮其作用，礙於篇輻有限，因此，將以另一本書來論及這些主題。不要好高騖遠，一步一步來吧，當你的根基隱固後，接下來的主題就可以迎刃而解，這好比要打好太極拳，得先練好蹲馬步，再配合拳架，這才會到位，否則你不像是在打太極拳，而是做太極操。

一本好書的定義是什麼？我的定義是，凡是你在書中找到一個你不會的就是好書，希望這本書能找到許多原來你不會的或不懂的地方，而成為很好或極好的書。若對本書有任何建議，請不吝來信批評指教。

mjtsai168@gmail.com

CHAPTER

3

選擇敘述

程式的運行基本上是一行接一行的執行，但有時我們會選擇哪些敘述要做、哪些不要做，而不是全部買單，這有如日常生活中的現象，會選擇做什麼而不做什麼。例如，在經濟許可下，你會出國讀研究所。條件是經濟許可，才會執行出國讀書，若此條件不成立，那就在國內讀研究所。

選擇敘述(selection statement)也是如此，判斷某條件若成立，則做什麼，不成立則做另一件事。其實人生都充滿了選擇不是嗎？你可以隨便舉一些例子。在 Python 程式中要判斷條件式是否成立(或為真)，則需要藉助前一章的關係運算子和邏輯運算子。

Python 的選擇敘述共有三類，分別是 if，if...else，以及 if...elif...else。我們以下將一一說明之。

3-1 if 敘述

if 選擇敘述只在乎條件為真時，要執行什麼而已，至於條件為假時，是不予以理會的。

3-1-1 if 敘述的語法

if 敘述的語法如下：

```
if condition:
    statement(s)
```

此語法對應的流程圖如圖 3.1 所示：

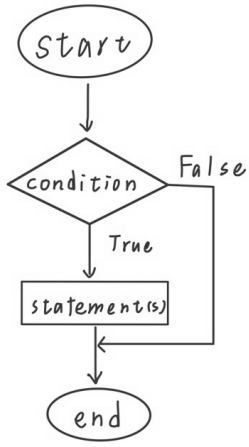

圖 3.1 if 敘述對應的流程圖

表示若 condition 的條件式是真，則執行 statement(s)。若為假，則什麼都沒做。其中 statement(s)要內縮，它可以是一條敘述或是多條敘述。若是多條敘述，則不僅要內縮，而且也要對齊。要內縮幾格沒有硬性規定，但一般常規是 4 個空格。

範例程式：if-1.py

```
01  ans = input('Is Python fun (y or n)? ')
02
03  if ans == 'y':
04      print('Yes, I want to learn Python now')
05  print('over')
```

```
Is Python fun (y or n)? y
Yes, I want to learn Python now
over
```

```
Is Python fun (y or n)? n
over
```

以上是執行兩次的結果。程式當你輸入是 y 時，將執行其條件式為 True 所對應的敘述，若輸入是 n，則不會執行任何敘述，直接印出 over。

若條件式為 True，要執行多個敘述時，則要將這些敘述內縮，而且對齊，請看以下的範例程式。

範例程式：if-2.py

```
01  ans = input('Is Python fun (y or n)? ')
02
03  if ans == 'y':
04      print('Yes, I want to learn Python now')
05      print('and you?')
06  print('over')
```

```
Is Python fun (y or n)? y
Yes, I want to learn Python now
and you?
over
```

```
Is Python fun (y or n)? n
over
```

先印出提示訊息詢問使用者，當使用者輸入是 y 時，則會執行對應內縮的兩個敘述。若使用者輸入是 n 時，則直接跳到印出 over。

上述程式要注意的事項：

1. if 敘述最後要冒號(:)
2. 若要比較左右是否相等時，需使用關係運算子 ==，而不是指定運算子 =。
3. Python 是以內縮與對齊的方式來執行該做的事項。

3-1-2 兩數對調

傳統上，要處理兩數對調，需要藉助一變數，而不可以執行以下的動作：

```
a = 100
b = 200
a = b
b = a
print('a =', a)
print('b =', b)
```

```
a = 200
b = 200
```

最後的結果 a 和 b 將都是 200。以下的程式是提示使用輸入兩個數值，若第一個大於第二個數值時，則加以對調。它需藉助另一暫時的變數，如 temp，程式如下所示：

範例程式：swap-1.py

```
01  a = eval(input('Enter a number: '))
02  b = eval(input('Enter the other number: '))
03  print('a = %d, b = %d'%(a, b))
04
05  #swap two variables when a > b
06  if a > b:
07      temp = a
08      a = b
09      b = temp
10  print('a = %d, b = %d'%(a, b))
```

```
Enter a number: 200

Enter the other number: 100
a = 200, b = 100
a = 100, b = 200
```

要特別注意的是，當 a > b 條件為真時，若執行多個敘述，則這些敘述不僅要內縮，而且也要對齊。

而 Python 將上述對調的程式簡化，以下一敘述就可完成兩數對調。

```
a, b = b, a
```

其實我們可以一次輸入兩個資料。以下的程式將予以驗證之。

範例程式：swap-2.py

```
01  a, b = eval(input('Enter two numbers: '))
02  print('a = %d, b = %d'%(a, b))
03
04  #swap two variables
05  a, b = b, a
06  print('a = %d, b = %d'%(a, b))
```

```
Enter two numbers: 200, 100
a = 200, b = 100
a = 100, b = 200
```

要注意的是，一次輸入兩個變數值時，之間要以逗號隔開，二個以上的變數值亦相同。

練習題

1. 請輸入一整數，判斷它是否大於 0，若是，則輸出此數大於 0。
2. 試問下一程式的輸出結果：

```
a = 168
b = 158
print('a = %d, b = %d'%(a, b))

if a > b:
    a = b
    b = a
print('a = %d, b = %d'%(a, b))
```

3. 小明撰寫了判斷某數是否為偶數的程式，請你幫忙除錯一下，感謝。

```
a = eval(input('Enter a number: '))
if a / 2 = 0:
print(a, '是偶數')
```

參考解答

1.
```
a = eval(input('Enter a number: '))
if a > 0:
    print('%d is greater than 0'%(a))
print('Over')
```

```
Enter a number: 100
100 is greater than 0
Over
```

```
Enter a number: -100
Over
```

以上是執行兩次的結果。

2.
```
a = 168, b = 158
a = 158, b = 158
```

說明：此程式的動作將會導致 a 的值和 b 的值是相同的。

3.
```
a = eval(input('Enter a number: '))
if a % 2 == 0:
    print(a, '是偶數')
```

3-2 if…else 敘述

上一節 if 敘述只在乎當條件是真時所要處理的事項，條件是假時不予以理會。本節將討論當真和假時皆有處理的事項，此時就要使用 if…else，就是條件為真時，做什麼事，否則，做另一些事。講白話一點，若有兩個選項要單選時，則以 if…else 敘述表示之。

3-2-1 if…else 敘述的語法

if…else 敘述的其語法如下：

```
if condition:
    statementA
else:
    statementB
```

此語法對應的流程圖如圖 3.2 所示：

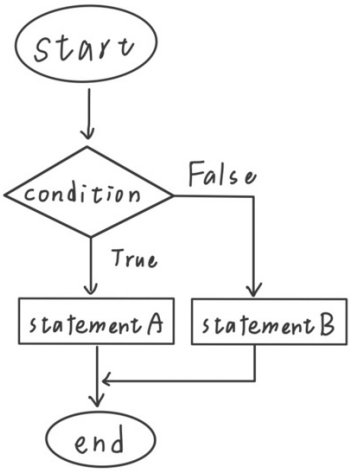

圖 3.2 if...else 語法對應的流程圖

若 condition 的條件式是真，則執行 statementA。若為假，則執行 statementB。statementA 和 statementB 皆要內縮，若是多條敘述不僅內縮，而且也要對齊。

我們再將 3-1 節的程式加以擴充，請看以下範例程式。

範例程式：if_else.py

```
01  ans = input('Is Python fun (y or n)? ')
02
03  if ans == 'y':
04      print('Yes, I want to learn Python now')
05  else:
06      print('No, I prefer to learn C')
07  print('over')
```

```
Is Python fun (y or n)? y
Yes, I want to learn Python now
over
```

```
Is Python fun (y or n)? n
No, I prefer to learn C
over
```

以上是執行兩次的結果。注意！else 後面也要冒號(:)，其要執行的敘述也要內縮。

有時常會使用 random 模組下的 randint(a, b)方法，產生[a, b]之間的數值。

```
import random
a = random.randint(1, 100)
print(a)
```

```
52
```

程式首先使用 import random 載入 random 模組。random.randint(1, 100)產生 1~100 之間的數值，不過你我產生的數值可能會不一樣。也可以呼叫此模組下的 seed(number)方法，以 number 當其種子，若你我使用相同的種子，則會產生相同的數值。

```
import random
random.seed(1)
a = random.randint(1, 100)
print(a)
```

```
18
```

此時皆會產生相同的數值 18。

3-2-2 除了 0 以外，其餘的數字皆為 True

在判斷真、假時，Python 將數字 0 視為 False，其它數字不管是正數或是負數皆為 True，以下我們以幾個範例來說明。

```
a = 0
if a:
    print('True')
else:
    print('False')
```

```
False
```

因為 a 等於 0，所以是假，因此印出 False。你可以利用邏輯運算子 not ，將真變假，假變真。

```
a = 0
if not a:
    print('True')
else:
    print('False')
```

```
True
```

原來 a 等於 0 是假，經由 not 運算後變為真。所以印出 True。

```
a = 1
if a:
    print('True')
else:
    print('False')
```

```
True
```

此程式 a 是 1，所以是真，因此印出 True。即使 a 是負數 -1 或是浮點數 1.2 也都是真，你可以自行執行看看。

3-2-3 if...else 另一種表達方式

在 if...else 的表示上，有一種方式更簡潔，其語法如下：

條件式真時的值 `if` **條件式** `else` **條件式為假時的值**

如計算某一整數的絕對值，一般我們會以 if...else 這樣撰寫：

```
num = eval(input('Enter an integer: '))
if num >= 0:
    abs = num
else:
    abs = -num
print('abs(%d) is %d'%(num, abs))
```

```
Enter an integer: -100
abs(-100) is 100
```

```
Enter an integer: 100
abs(100) is 100
```

```
Enter a number: -100
-100 is True
```

以上是執行三次的結果。

3-3 if…elif…else 敘述

當有三個選項要做單選時，則需使用 if…elif…else 敘述。

3-3-1 if…elif…else 敘述的語法

if…elif…else 敘述的語法如下：

```
if conditionA:
    statementA
elif conditionB:
    statementB
else:
    statementC
```

此語法對應的流程圖如圖 3.3 所示：

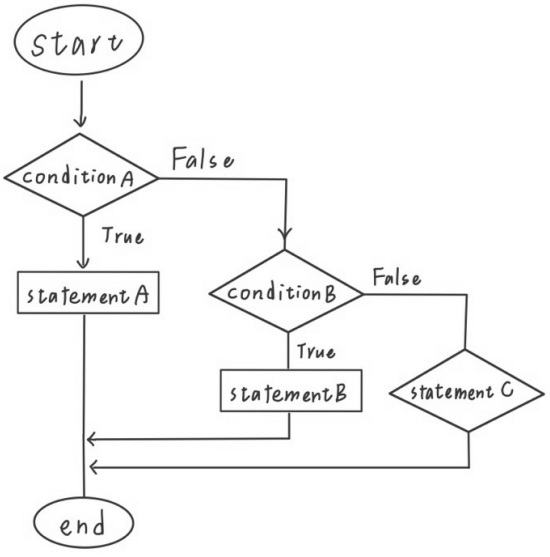

圖 3.3 if…elif…else 敘述對應的流程圖

圖 3.3 的執行步驟如下：

1. 若 conditionA 的條件式是真，則執行 statementA，並結束此選擇敘述。
2. 若 conditionA 的條件式為假，則再判斷 conditionB 的條件式是否為真，若為真，則執行 statementB，並結束此選擇敘述。
3. 若 conditionB 條件式為假，則執行 statementC。

在 if…elif…else 敘述中只會執行 statementA、statementB、statementC 其中一個敘述而已。請看以下範例程式。

範例程式：language.py

```
01  print('What do you prefer to learn Python, C or None?')
02  ans = input('p for Python, c for C, n for None: ')
03  if ans == 'p':
04      print('I prefer to learn Python')
05  elif ans == 'c':
06      print('I prefer to learn C')
07  else:
08      print('I do not like programming language')
```

```
What do you prefer to learn Python, C or None?
p for Python, c for C, n for None: p
I prefer to learn Python
```

此程式由使用者輸入一字串，因為有三種選擇分別是 Python、C 和 None，所以使用 if…elif…else。

若有更多的條件時，可依此類推，其敘述將為：

```
if...elif...(elif...)else
```

其中(elif…)可以多個，視條件的個數而定。

3-3-2 製作交談式人機介面的選單

有些問題可能要多個 elif 才能夠執行。例如，在陌生的地方，在紅綠燈前，要直走或左轉或右轉常常會傷腦筋。以下的程式將製作一交談式人機介面(interactive human-machine interface)的選單，提示使用者若輸入 1 時，則直走；若輸入 2 時，則右轉；若輸入 3 時，則左轉。

3-4 判斷是否為閏年

若要多個條件組合在一起才能判斷其真假時，則需利用第 2 章談到的 and(且)、or(或)、not(反)等邏輯運算子來結合完成，最終的結果不是真就是假。我們以一判斷某一年是否為閏年的範例程式來說明。

```
#add logical operator
#leap year or not
#version 1.0
year = eval(input('Enter a year: '))
if year % 400 == 0 or (year % 4 == 0 and year % 100 != 0):
    print('%d is leap year'%(year))
else:
    print('%d is not leap year'%(year))
```

```
Enter a year: 2020
2020 is leap year
```

```
Enter a year: 2022
2021 is not leap year
```

你也可以將條件式獨立出來，指定給一個變數，再以此變數當做條件式，判斷它是否為真(True)。

```
#version 2.0
year = eval(input('Enter a year: '))
conditions = year % 400 == 0 or (year % 4 == 0 and year % 100 != 0)
if conditions:
    print('%d is leap year'%(year))
else:
    print('%d is not leap year'%(year))
```

```
Enter a year: 2020
2020 is leap year
```

```
Enter a year: 2022
2021 is not leap year
```

若要讓程式更簡潔，可以將每一個條件獨立出來，所以共有三個條件，如下所示：

```
#version 3.0
year = eval(input('Enter a year: '))
cond1 = year % 400 == 0
cond2 = year % 4 == 0
cond3 = year % 100 != 0

if cond1 or (cond2 and cond3):
    print('%d is leap year'%(year))
else:
    print('%d is not leap year'%(year))
```

你覺得上述的程式是否較易閱讀呢？

練習題

1. 申請獎學金的條件有下列兩個條件：

   (1) 平均分數大於等於 85，而且體育成績大於 80，

   (2) 或是在 Python 競賽中得獎皆可參加。

   請撰寫一程式測試之。

參考解答

1.
```
score = eval(input('輸入平均分數: '))
sports = eval(input('輸入體育成績: '))
awards = input('輸入有無得到獎項(yes or no): ')
condition = (score >= 85 and sports > 80) or awards =='yes'
if condition:
    print('你可以申請獎學金')
else:
    print('你不可以申請獎學金')
```

```
輸入平均分數: 90

輸入體育成績: 90

輸入有無得到獎項(yes or no): no
你可以申請獎學金
```

3-5 巢狀 if

在 if 敘述中又有一 if 敘述，此稱為巢狀 if(nested if)。由於 Python 是以對齊來判斷是屬於哪一個敘述對應的執行動作，所以較易分辨，請看以下範例與說明。

```
num = eval(input('Enter an integer: '))
if num >= 60:
    if num >= 90:
        print('you are better')
else:
    print('fail')

print('over')
```

```
Enter an integer: 90
you are better
over
```

此程式的 else 是與第一個 if 對應，而不是第二個 if，因為 else 和第一個 if 對齊的關係。因此，輸入 90 時會印出以上的輸出結果，若是輸入低於 60，如 59，則輸出結果如下：

```
Enter an integer: 59
fail
over
```

下一程式的 else 與第二個 if 敘述對齊，所以與之對應。

```
num = eval(input('Enter an integer: '))
if num >= 60:
    if num >= 90:
        print('you are better')
    else:
        print('you are fine')

print('over')
```

```
Enter an integer: 78
you are fine
over
```

```
Enter an integer: 92
you are better
over
```

此程式只要輸入的值是介於 60 與 90 之間時，則如輸出結果所示，若輸入的值小於 60 時，如 59，則輸出結果如下：

```
Enter an integer: 59
over
```

第三個程式每一個 if 敘述皆有對應的 else 敘述，端看哪一個 else 敘述與哪一個 if 敘述對齊。

```
num = eval(input('Enter an integer: '))
if num >= 60:
    if num >= 90:
        print('you are better')
    else:
        print('you are fine')
else:
    print('fail')
print('over')
```

```
Enter an integer: 92
you are better
over
```

```
Enter an integer: 80
you are fine
over
```

```
Enter an integer: 50
fail
over
```

我們可以從輸出結果得到上述程式的對應關係。有底線的 if 與有底線的 else 對應，而外圍的 if 與最後一個 else 對應。

練習題

1. 若要輸出以下的結果：

```
Enter an integer: 88
you are fine
over
```

```
Enter an integer: 97
you are fine
over
```

```
Enter an integer: 50
over
```

試問以下的程式哪裡出錯了，請你加以 Debug。

```
num = eval(input('Enter an integer: '))
if num >= 60:
    if num >= 90:
        print('you are better')
else:
    print('fail')

print('over')
```

參考解答

1.
```
num = eval(input('Enter an integer: '))
if num >= 60:
    if num >= 90:
        print('you are better')
    else:
        print('you are fine')

print('over')
```

3-6 if...else 與使用兩個 if 之差異

if...eles 當有一條件成立時，就印出其結果，同時此敘述就結束執行，但若以兩個 if 敘述執行的話，雖然也可以輸出同樣的結果，但較耗時，因為每一個 if 敘述皆會被執行一次，即使第一個 if 已得到答案了，下一個 if 還是需要再執行。在 if...else 或是 if...elif...(elif)else 敘述中只會有一個條件是成立的，當有條件式成立時，整個敘述將結束之。

下一範例是判斷你輸入的數值是偶數或是奇數：

```
n = eval(input('Enter an integer: '))
if n % 2 == 0:
    print(n, 'is even number.')
else:
    print(n, 'is odd number')
```

```
Enter an integer: 12
12 is even number.
```

```
Enter an integer: 13
13 is odd number.
```

若以兩個 if 來撰寫：

```
n = eval(input('Enter an integer: '))
if n % 2 == 0:
    print(n, 'is even number.')
if n % 2 != 0:
    print(n, 'is odd number')
```

```
Enter an integer: 12
12 is even number.
```

```
Enter an integer: 13
13 is odd number.
```

你會發現以 if...else 和兩個 if 來撰寫程式，其結果是一樣的。因為一數值不是偶數就是奇數，所以答案是單選的，因此使用 if...else 較佳，因為當你輸

入是 12 時，if...else 這敘述就結束了，但用兩個 if，即使第一個成立了，第二個 if 還是要執行判斷。

若是答案是複選時，則必須一一的以 if 敘述來完成，例如要判斷某一數值是 3 或是 5 的倍數，它有可能同時滿足 3 和 5 的倍數，因此不可以使用 if...else 來處理。

```
num = eval(input('Enter an integer: '))
if num % 3 == 0:
    print(num, '是 3 的倍數')
if num % 5 == 0:
    print(num, '是 5 的倍數')
if num % 3 != 0 and  num % 5 != 0:
    print(num, '不是 3 的倍數，也不是 5 的倍數')
print('over')
```

```
Enter an integer: 9
9 是 3 的倍數
over
```

```
Enter an integer: 30
30 是 3 的倍數
30 是 5 的倍數
over
```

```
Enter an integer: 13
13 不是 3 的倍數，也不是 5 的倍數
over
```

練習題

1. 若將上述的程式改為如下，試問執行時會有正確的結果嗎？有什麼問題存在？

```
num = eval(input('Enter an integer: '))
if num % 3 == 0:
    print(num, '是 3 的倍數')
elif num % 5 == 0:
    print(num, '是 5 的倍數')
else:
    print(num, '不是 3 的倍數，也不是 5 的倍數')
print('over')
```

習題

1. 下列程式有一些 bugs，請你加以 debug。

```
#1 輸入一整數，判斷它是否為偶數或是奇數
a = input('Enter an integer: ')
If a / 2 = 0
    print('%d is an even number:')
Else
    print('%d is an odd numbers.)

#2 輸入一整數，判斷此數大於 0、或等於 0、或小於 0
a = input(Enter an integer: )
If a >= 0
    print(a, 'is greater than 0.')
elseif a = 0:
    print(a, 'is equal to 0.')
Else
    print(a, 'is less than 0')
```

2. 試撰寫一程式，提示使用者輸入你的身高、體重，然後計算你的 BMI。其中輸入身高時以公分為單位，體重以公斤為單位。BMI = 體重/(身高的平方)。以下是 BMI 的對照表：

表 3-1　BMI 對照表

BMI	說明
BMI < 18.5	過輕
18.5 <= BMI < 24	正常、健康
24 <= BMI < 27	過重
27 <= BMI < 30	輕度肥胖
30 <= BMI < 35	中度肥胖
BMI >= 35	重度肥胖

3. 試撰寫一程式，讓對方在紙上寫出 1~100 之間的一個數字，然後設計一些數字集合表格讓對方答是或不是，最後輸出他所寫的數字。
4. 修改 3-7 節個案研究的練習題，讓它可以告訴你生日是何年、何月、何日。