

# Python Programming

2023 Spring; week 13

Instructor: Cheng-Chun Chang (張正春)  
Department of Electrical Engineering

Textbook: Python程式設計:從入門到進階應用(第三版) 2020

## Ch4 選擇結構

# 流程圖圖示介紹

表 4-2 流程圖圖示介紹

流程圖圖示	意義
	程式的流程，表示程式的處理順序，表示循序結構。
	條件選擇，於菱形內寫入條件判斷，表示選擇結構。
	程式的敘述區塊，寫出所需完成的功能。
	程式的開始或結束。
	程式所需的輸入與輸出。
	流程圖的連接點，代表流程圖的進入點或離開點。當流程圖過長可以使用連接點將一個過長流程圖切割成多個流程圖組合起來。

## 4-1 單向選擇結構

---

- 單向選擇結構是最簡單的選擇結構，日常生活上經常用到，例如：「若週末天氣好的話，我們就去打球」。
- 單向選擇結構只做測試條件為真時，執行條件為真的動作，只有一個方向的選擇，因此稱做單向選擇結構。

## 選擇結構：單向選擇結構流程圖

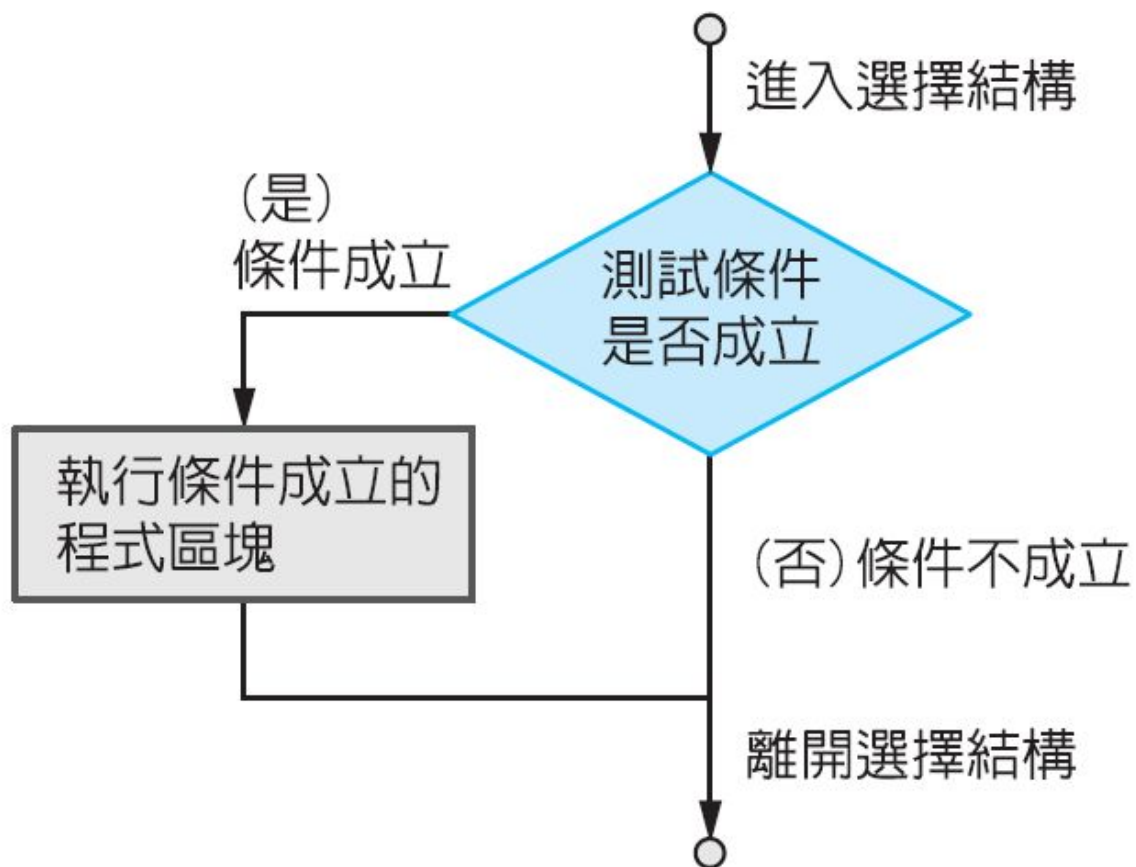


圖 4-1 單向選擇結構流程圖

## 4-1 單向選擇結構

---

表 4-1 單向選擇結構

單向選擇程式語法	程式範例
if 條件判斷： 條件成立的敘述	if score >= 60: print(" 很好，請繼續保持下去 ")
說明	
若變數 score 大於等於 60，則顯示「很好，請繼續保持下去」。	

---

# Programming codes I



## 範例4-1-1 判斷及格 ( ch4\4-1-1- 判斷及格.py)

---

- 寫一個程式判斷所輸入成績是否及格，成績及格則顯示「很好，請繼續保持下去」。



## 範例4-1-1 判斷及格 ( ch4\4-1-1- 判斷及格.py)

---

### □ 程式碼

01	score = int(input(' 請輸入一個成績？ '))
02	if score >= 60:
03	print(' 很好，請繼續保持下去')

## 範例4-1-1 判斷及格 流程圖

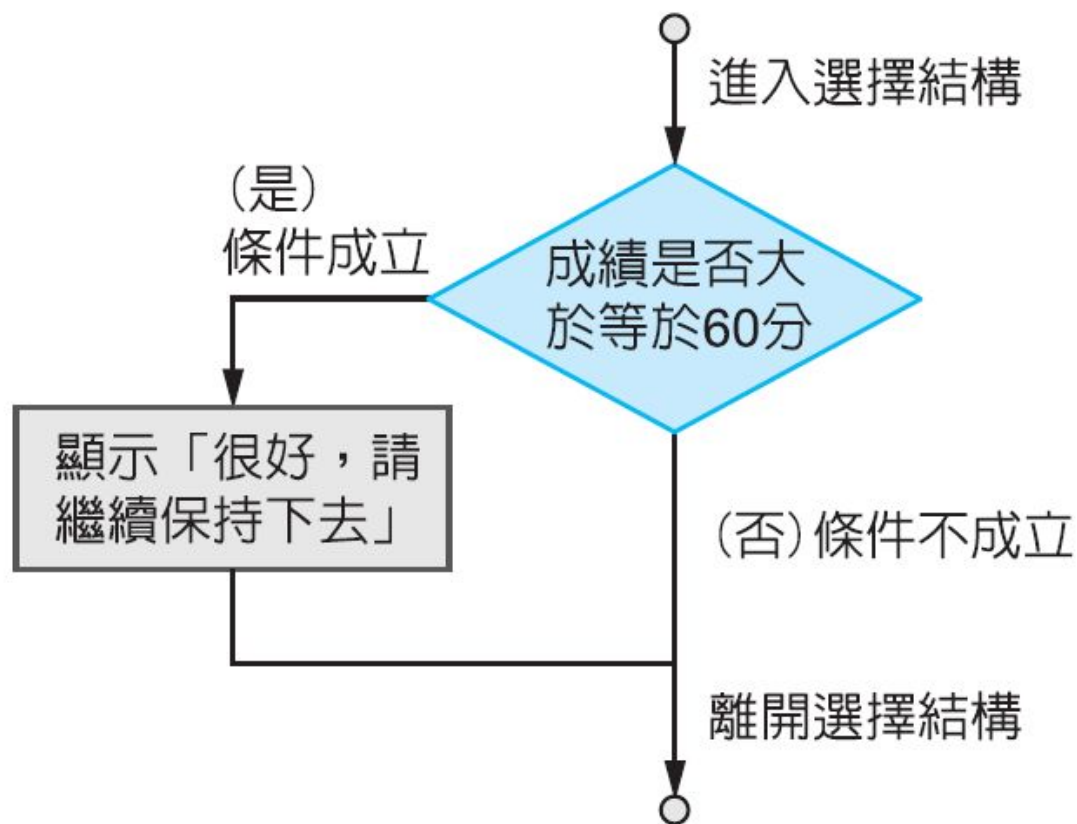


圖 4-2 判斷及格流程圖

## 4-1 單向選擇結構

---

- Python 程式撰寫的注意事項，有以下幾點：
  - 1.條件成立要做的動作有哪些，可能不只一行程式，那到底有幾行，Python 程式碼以**縮行**表示執行的範圍。
  - 2.條件成立要做什麼，需要執行的每一行程式碼都要縮相同的空白鍵(space) 個數，通常使用4 個空白鍵
  - 所以「if 條件判斷 :」後面接的條件成立的動作，**每一行都需要以4 個空白鍵開頭，表示執行的範圍。**
  - 3.Tab 也可用於表示縮行，但是**空白鍵與Tab 鍵不要混用**，整個程式只能從頭到尾選擇其中一種進行縮行。

## 4-2 雙向選擇結構

---

- 雙向選擇結構比起單向選擇結構更複雜一些，日常生活上屬於雙向選擇的對話，例如：「若週末天氣好的話，我們就出去打球，否則去看電影」。
- 雙向選擇結構為當測試條件為真時，執行測試條件為真的動作；否則做測試條件為假的動作。有兩個方向的選擇，因此稱做雙向選擇結構。

## 選擇結構：雙向選擇結構 流程圖

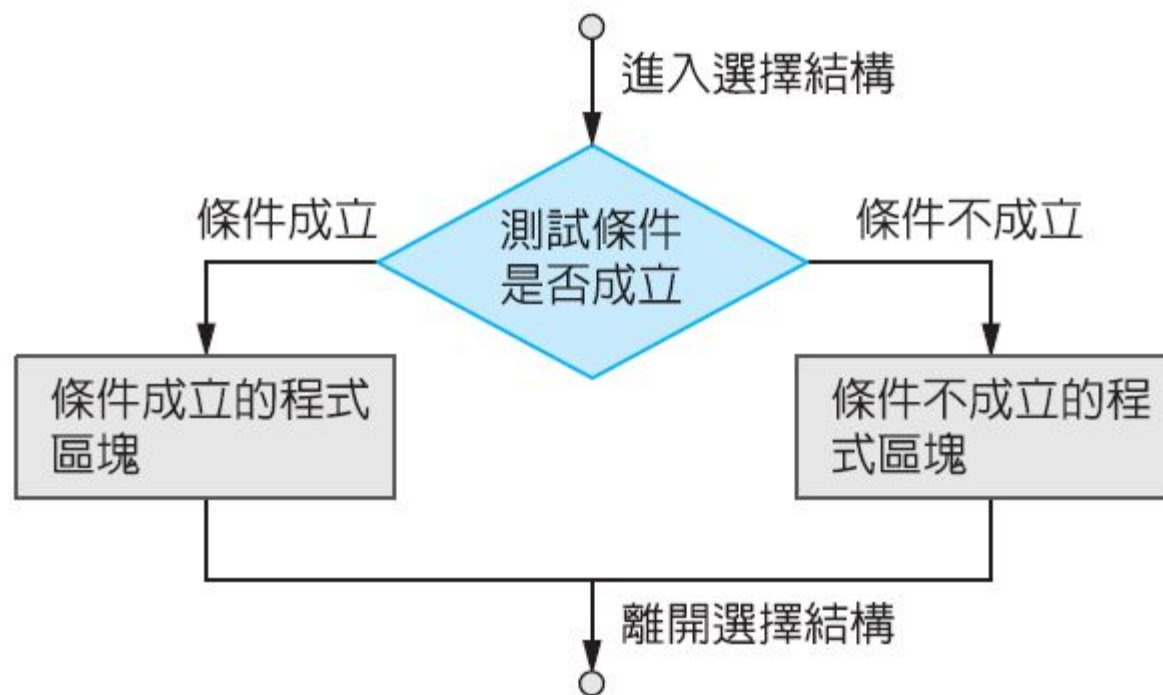


圖 4-3 雙向選擇結構流程圖

## 4-2 雙向選擇結構

表 4-2 雙向選擇結構

雙向選擇程式語法	程式範例 ( 滿 2000 打九折 )
if 條件判斷： 條件成立的敘述 else: 條件不成立的敘述	if cost >= 2000: print(cost*0.9) else: print(cost)
說明	
若 cost 大於等於 2000，則顯示為 cost 的值打九折，否則顯示 cost 的值	

---

# Programming codes 2-1



## 範例4-2-1 滿2000 打九折 (ch4\4-2-1- 滿2000 打九折.py)

---

- 採買物品時，有時會遇到店家為了刺激消費，會使用滿額折扣，如：滿2000 打九折，未滿2000 則不打折，請寫一個程式幫助店家計算顧客所需付出的金額。



## 範例4-2-1 滿2000 打九折 (ch4\4-2-1- 滿2000 打九折.py)

---

### □ 程式碼

01	<code>cost = int(input(' 請輸入購買金額？ '))</code>
02	<code>if cost &gt;= 2000:</code>
03	<code>    print(cost * 0.9)</code>
04	<code>else:</code>
05	<code>    print(cost)</code>

## 範例4-2-1 滿2000 打九折 流程圖

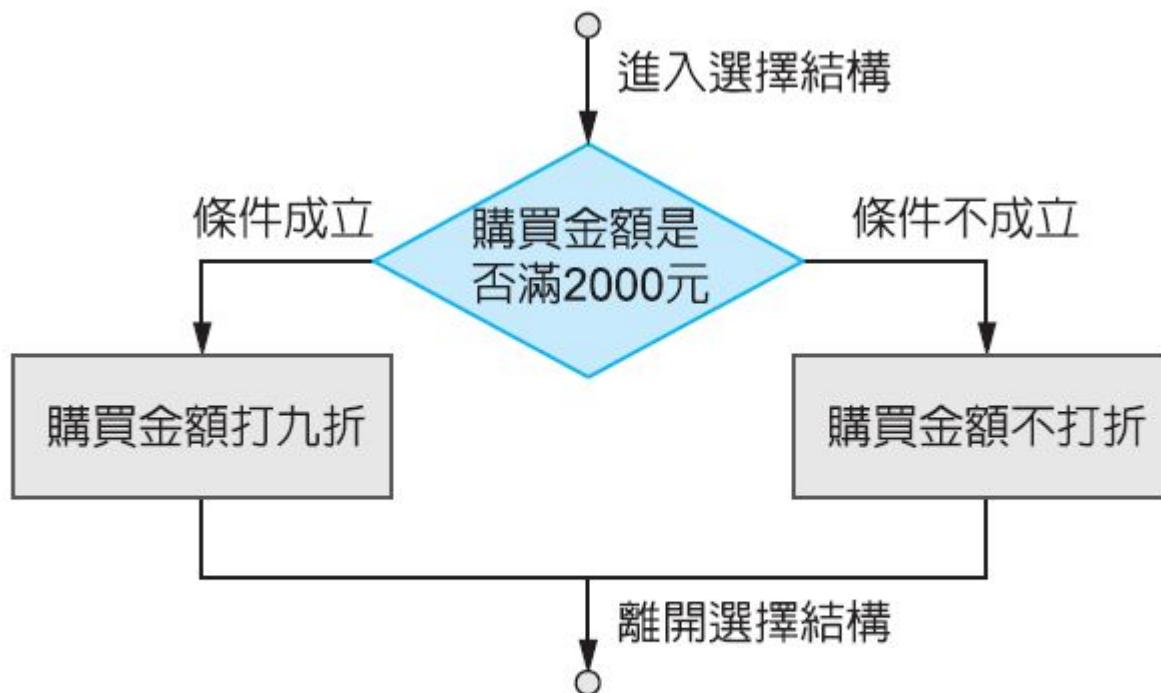


圖 4-4 滿 2000 打九折流程圖

---

# Programming codes 2-2



## 範例4-2-3 三角形判斷 (ch4\4-2-3- 三角形判斷.py)

---

- 設計一個程式允許輸入三角形三邊長，分別為a、b 與 c，根據三角形中任兩邊相加要大於第三邊，判斷是否為三角形。

## 範例4-2-3 三角形判斷 (ch4\4-2-3- 三角形判斷.py)

---

### □ 程式碼

01	<code>a = int(input(' 請輸入三角形邊長a 長度為？ '))</code>
02	<code>b = int(input(' 請輸入三角形邊長b 長度為？ '))</code>
03	<code>c = int(input(' 請輸入三角形邊長c 長度為？ '))</code>
04	<code>if (a&lt;b+c)and(b&lt;a+c)and(c&lt;a+b):</code>
05	<code>    print(' 可構成三角形')</code>
06	<code>else:</code>
07	<code>    print(' 無法構成三角形')</code>

## 範例4-2-3 三角形判斷 流程圖

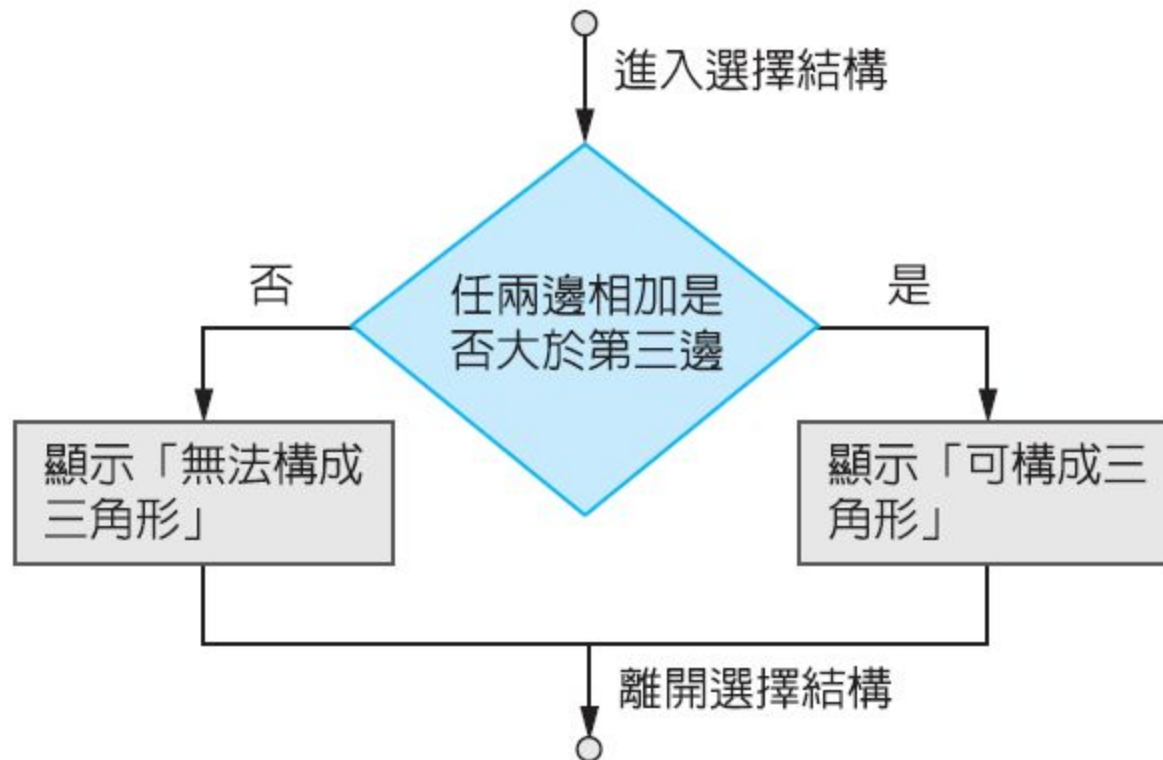


圖 4-6 三角形判斷流程圖

## 4-3 多向選擇結構

---

- 除了單向選擇與雙向選擇外，更廣義的選擇結構是**多向選擇**
- 意即選擇結構中還可以加入選擇結構，單向選擇與雙向選擇為多向選擇結構的特例，多向選擇結構讓程式有無限可能執行的路徑與狀態。

## 選擇結構：多向選擇結構流程圖

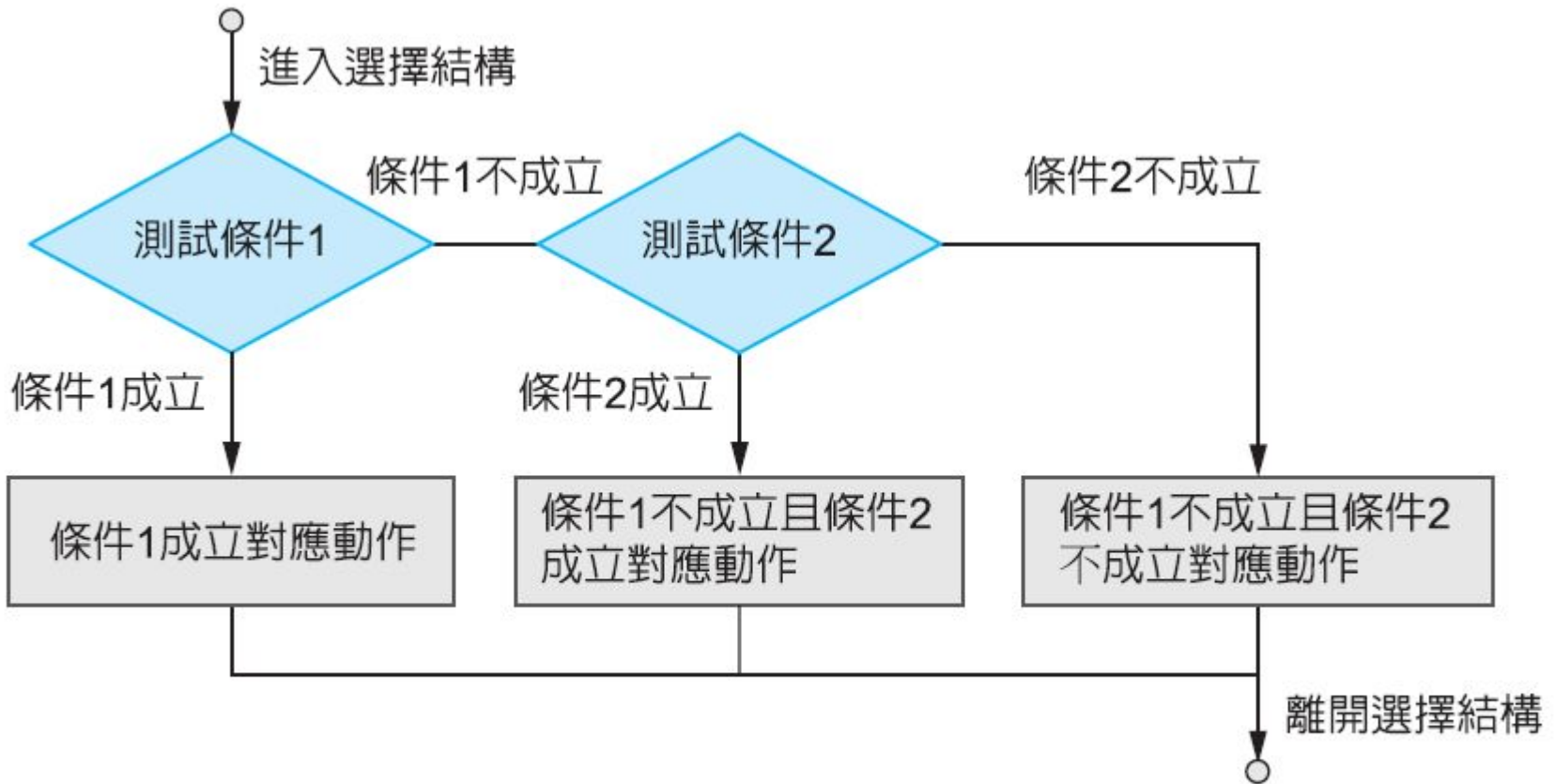


圖 4-7 多向選擇結構流程圖



## 4-3 多向選擇結構

表 4-3 多向選擇結構

多向選擇結構語法	程式範例（分數與評語）
<pre>if 條件判斷 1:     條件判斷 1 成立的敘述 elif 條件判斷 2:     條件判斷 2 成立的敘述 else:     條件判斷 2 不成立的敘述</pre>	<pre>if score &gt;= 80:     print(' 非常好 ') elif score &gt;= 60:     print(' 不錯喔 ') else:     print(' 要加油 ')</pre>

---

# Programming codes 3



## 範例4-3-1 分數與評語 (ch4\4-3-1- 分數與評語.py)

- 寫一個程式若成績大於等於80 分，評語為「非常好」，否則若成績大於等於60 分，評語為「不錯喔」，否則評語為「要加油」，將以上敘述表示為表格，如下。

表 4-4 分數與評語

成績	評語
成績 $\geq 80$	非常好
$80 > \text{成績} \geq 60$	不錯喔
成績 $< 60$	要加油

## 範例4-3-1 分數與評語 (ch4\4-3-1- 分數與評語.py)

### □ 程式

```
01 score = int(input(' 請輸入一個成績？ '))
02 if score >= 80:
03     print('非常好')
04 elif score >= 60:
05     print(' 不錯喔')
06 else:
07     print(' 要加油')
```

### 執行結果

輸入成績「60」，結果顯示在螢幕。

請輸入一個成績？ 60  
不錯喔

## 範例4-3-1 分數與評語 流程圖

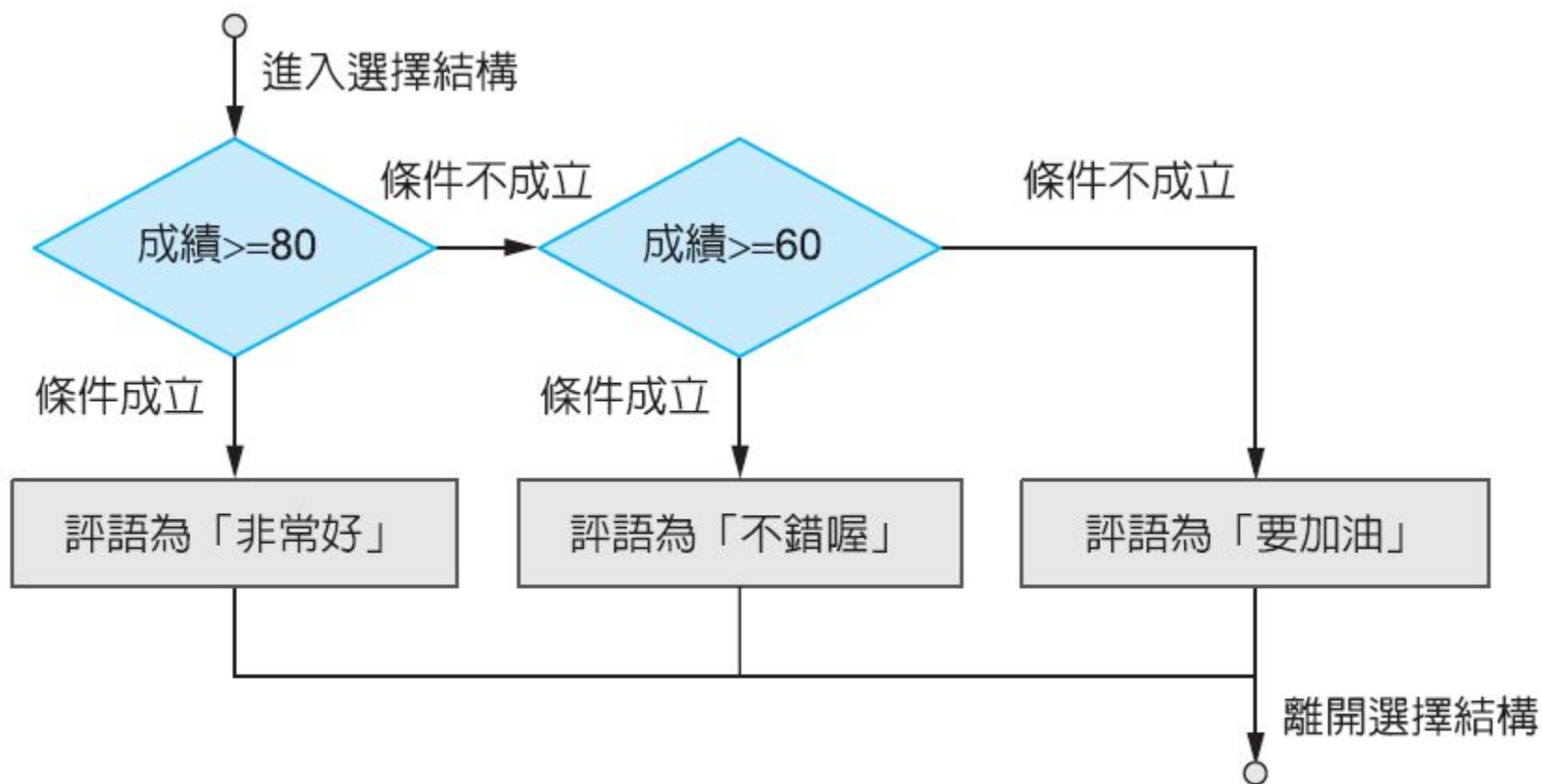


圖 4-8 分數與評語流程圖

---

# Programming codes 4



## 4-4 條件判斷與運算子「in」 (ch4\ 條件判斷與運算子「in」.py)

---

- 可以使用條件判斷與「in」測試資料容器tuple、串列、字典與集合是否包含某個元素。

## 4-4-1 判斷tuple 是否包含某個元素

使用條件判斷與「[in](#)」進行判斷，程式如下。

程式	執行結果
<pre>a = (1,2,3,4) if 1 in a:     print(' 數字 1 在 tuple a 中 ') else:     print(' 數字 1 不在 tuple a 中 ')</pre>	數字 1 在 tuple a 中



## 4-4-2 判斷串列是否包含某個元素

使用條件判斷與「in」進行判斷，程式如下。

程式	執行結果
<pre>a = list('abcdefghijklmnopqrstuvwxyz') if 'q' in a:     print('q 在串列 a 中 ') else:     print('q 不在串列 a 中 ')</pre>	q 在串列 a 中

## 4-4-3 判斷字典是否包含某個元素

使用條件判斷與「[in](#)」進行判斷，程式如下。

程式	執行結果
<pre>lang1={' 早安 ':'Good Morning',' 謝謝 ':'Thank You'} if ' 謝謝 ' in lang1:     print(' 謝謝的英文爲 ', lang1[' 謝謝 ']) else:     print(' 查不到謝謝的英文 ')</pre>	謝謝的英文爲 Thank You

## 4-4-4 判斷集合是否包含某個元素

使用條件判斷與「`in`」進行判斷，程式如下。

程式	執行結果
<pre>a = set('tiger') if 't' in a:     print('t 在集合 a 內 ') else:     print('t 不在集合 a 內 ')</pre>	t 在集合 a 內

---

# □ Exercise



# 學號判別

## (學號判別.py)

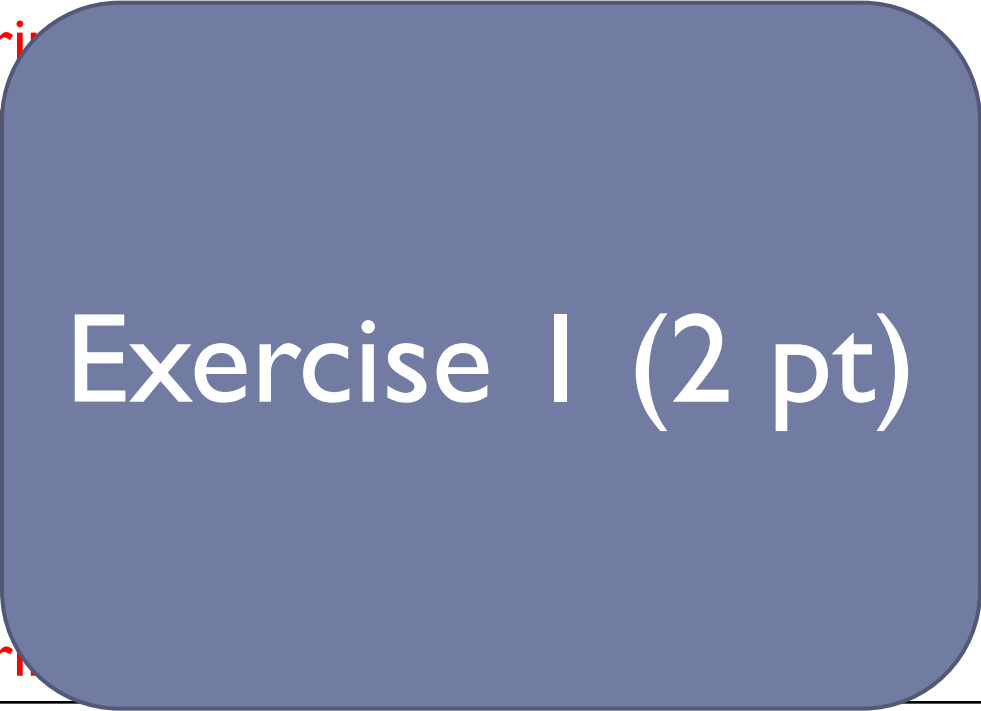
---

- 設計一個以自己學號為判斷標準的程式, 使用者輸入學號後會判斷, 該學號持有者與自己的關係。

<b><u>都不一樣</u></b>	<b><u>只有前3碼一樣</u></b>	<b><u>只有前5碼一樣</u></b>	<b><u>只有前7碼一樣</u></b>	<b><u>都一樣</u></b>
非同年級生	同年級	同年級 同科系	同年級 同科系 同班	本人

# 學號判別 (學號判別.py)

```
01 w = int(input(' 請輸入學號？ '))
02 if int(w) == 107650014: ☐更改為您的學號 (英文letter改數字)
03     print('學號正確')
04 elif int(w) == 107650015:
05     print('學號正確')
06 elif int(w) == 107650016:
07     print('學號正確')
08 elif int(w) == 107650017:
09     print('學號正確')
10 else:
11     print('學號錯誤')
```



# 測試用結果

## □ 1. 自己的學號

```
✓ w = int(input(' 請輸入學  
... 本人
```

## □ 2. 自己的學號+1

```
✓ w = int(input(' 請輸入學  
.. 同年級 同科系 同班
```

## □ 3. 自己的學號前三碼改成101

```
✓ w = int(input(' 請輸入學  
非同年級生
```

## 範例4-3-3 BMI 計算 (ch4\4-3-3-BMI 計算.py)

- 輸入自己的學號，並判斷長度是否為 9 碼，輸入錯誤顯示「學號格式錯誤」並程式終止。
- BMI 常用來判斷肥胖程度，BMI 等於體重(KG)除以身高(M)的平方，會有一個分類標準，假設BMI與肥胖分級如下。請寫一個程式讓使用者輸入身高與體重，顯示BMI值與肥胖程度。

表 4-7 BMI 計算

BMI 值	肥胖分級
$BMI < 18$	體重過輕
$18 \leq BMI < 24$	體重正常
$24 \leq BMI < 27$	體重過重
$27 \leq BMI$	體重肥胖



## 範例4-3-3 BMI 計算 (ch4\4-3-3-BMI 計算.py)

```
01 student_id = input("請輸入學號: ")
02 if len(student_id) != 10:
03     print("學號錯誤")
04     exit()
05 weight = float(input("請輸入體重: "))
06 height = float(input("請輸入身高: "))
07 BMI = weight / (height ** 2)
08 print("BMI 指數: ", BMI)
09 if BMI < 18.5:
10     print("體重過輕")
11 elif BMI < 24:
12     print("體重適中")
13 elif BMI < 28:
14     print("體重過重")
15 else:
16     print("嚴重肥胖")
```

Exercise 2 (2 pt)

## 範例4-3-3 測試結果

### □ 1.輸入錯誤的學號

```
請輸入學號：100  
學號格式錯誤！
```

### □ 2.輸入“自己的學號”，體重 70 kg，身高 1.7 m

```
請輸入學號：110310001  
請輸入體重(kg)：70  
請輸入身高(m)：1.7  
BMI = 24.221453287197235  
體重過重
```

### □ 3.輸入“自己的學號”，體重 40 kg，身高 1.5 m

```
請輸入學號：110310001  
請輸入體重(kg)：40  
請輸入身高(m)：1.5  
BMI = 17.777777777777778  
體重過輕
```

## 範例4-3-3BMI 計算 流程圖

流程圖

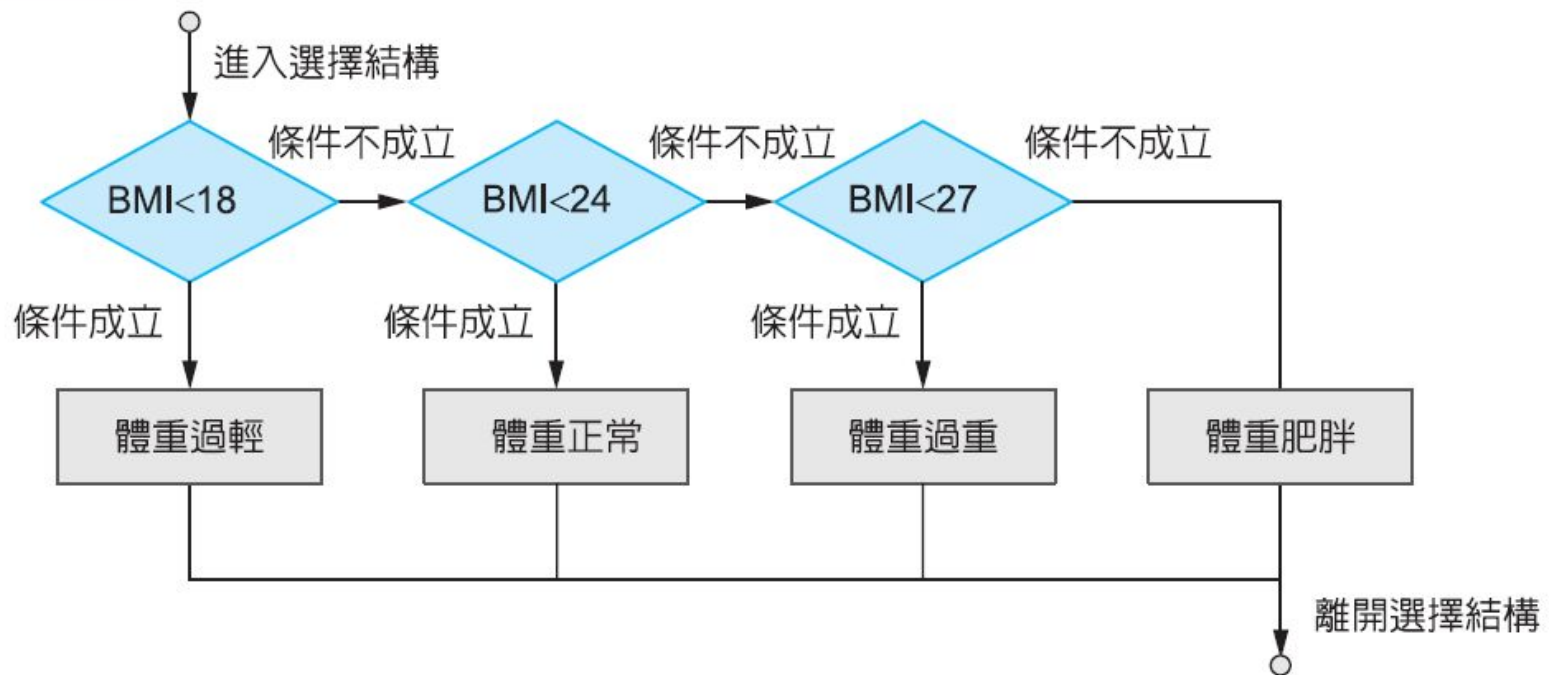


圖 4-10 BMI 計算流程圖

# To be continued.....

Instructor: Cheng-Chun Chang (張正春)  
Department of Electrical Engineering