

選擇結構程式設計

學習綱要

- ≡ 4-1 結構化程式設計架構
- ≡ 4-2 單一選擇結構敘述與撰寫
- ≡ 4-3 多重選擇結構敘述與撰寫
- ≡ 4-4 巢狀選擇結構敘述與撰寫
- ≡ 4-5 選擇結構程式實作演練
- ≡ 4-6 本章內容摘要

學習表現

- ≡ 1. 具備撰寫程式語言基本能力，運用資訊科技方法解決問題。
- ≡ 2. 具備程式設計之邏輯思考能力，展現系統思考、分析與探索之素養。
- ≡ 3. 具備實作程式設計能力，展現程式語言跨域應用之軟實力。

4-1 結構化程式設計架構

任何程式皆可由循序結構、選擇結構與重複結構等三種結構組合起來，且避免使用 Goto 指令，此稱為「結構化程式設計架構」。以上三種結構分別說明如下：

⚙️ 循序結構

程式執行順序與程式出現順序完全相同者，稱為循序結構。例如，本書目前為止的所有程式都稱為循序結構。

⚙️ 選擇結構

程式執行順序可以某些條件而改變者，此稱為選擇結構，例如，肚子餓了就執行「吃飯」，否則執行「繼續工作」等。Python 的選擇結構有「if ~else」、「if ~elif~else」此為本單元將介紹的內容。

⚙️ 重複結構

我們人類很多事都具有重複性。例如，重複做 10 個題目，重複連續上七節課等，程式設計是用來解決人類問題的工具，當然也有此重複結構。Python 的重複結構有 for 與 while 等指令，兩者的主要差別主要是——若程式設計階段就知道執行次數，可使用 for，請看 5-2 節；若於程式設計階段不知道執行次數，要等到程式執行階段，才能依照執行結果判斷何時結束迴圈，此時可用 while，請看 5-4 節。

4-2 單一選擇結構敘述與撰寫

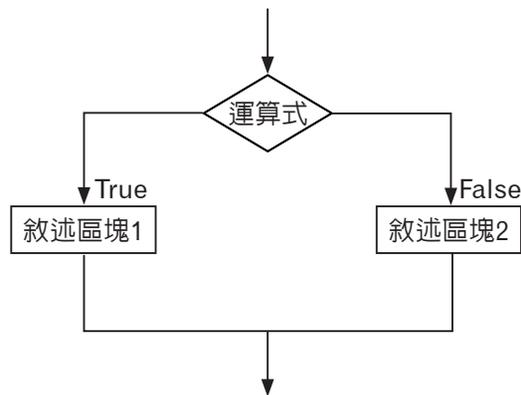
if~else ~

生活上常會面對『假如~否則~』，Python 的指令是『if~else~』，其語法如下：

```
if 運算式:  
    敘述區塊1;
```

```
[else :
    敘述區塊2 ;]
```

請特別留意冒號『:』與『程式縮排』，因為冒號『:』是 Python 特有；『縮排』在其他語言是美觀性質，但 Python 是語法也就是規定，因為縮排才是屬於 if else 要執行的敘述子區塊（若使用 Spyder 編寫程式，Spyder 會幫您補上冒號與縮排，若編譯器沒有幫您自動縮排，請按一下鍵盤的「Tab」鍵，而不是刻意按四個空白鍵。其次，縮排的空格數沒有強制性，但是在同一檔案中有相同空格數的強制性。），以上程式其流程圖如圖 4-1：



★ 圖4-1 if ~ else~流程圖

例如：

```
a=66
if a>=60:
    b="Pass"
else:
    b="Fail"
print(b)
```

以上程式一定要縮排。因為，程式縮排在其它語言是美觀問題，但 Python 是語法，所以沒有縮排就不行。例如，以下程式就不行。

```
a=66
if a>=60:
b="Pass"
```

```
else:
    b="Fail"
    print(b)
```

但若是僅執行一個敘述，這樣也行，但是視覺效果很差。

```
a=55
if a>=60:b="Pass"
else:b="Fail"
print(b)
```

請留意，以下兩個程式的 `print(b)`，因為縮排不同，歸屬就不同，執行結果也不相同。

```
a=66
if a>=60:
    b="Pass"
else:
    b="Fail"
print(b)
```

```
a=66
if a>=60:
    b="Pass"
else:
    b="Fail"
print(b)
```

C/C++ 用大括號『`{}`』來表示程式區塊的範圍，Python 直接用縮排表示，那就更簡潔。其次，也可以將否則的部分放在前面當預設值，程式如下：

```
a=46;b="Fail"
if a>=60:
    b="Pass"
print(b)
```

或

```
a=66;b="Fail"
if a>=60:b="Pass"
print(b)
```

`if` 內可否放 `if`，當然可以，此稱為巢狀 `if`，請看 4-4 節。

⚙️ 冒號(:)

冒號 (`:`) 也是 Python 特有的符號，往後的 `for`、`while`、函式等，都需要

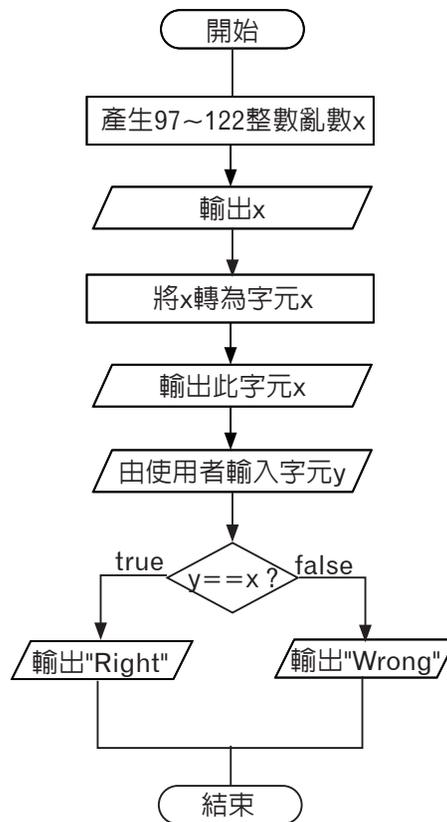
這個符號。

範例 4-2a

請寫一個程式，可以產生一個小寫字元，由使用者輸入此小寫字元，電腦評判是否正確。

👉 運算思維

1. 本例以流程圖分析如下：



2. 由以上流程圖分析，我們發現 if~ else~ 剛好可解決此問題。
3. 電腦產生字元、使用者如何輸入字元，前面已經說明，再套上 if~else~ 語法，所以全部程式如下：

```

import random
x=random.randint(97,122) #產生97~122的整數
print(x)#輸出此數字
  
```

```
x=chr(x)#轉為此數字所代表字元
print(x)
y=input('input the char:')
if y==x:
    print("Right")
else:
    print("Wrong")
```

👉 執行結果

```
119
w
input the char:w
Right
```

👉 自我練習

1. 請寫一程式，電腦自動產生 1~6 的整數亂數，並判斷是奇數或是偶數。
2. 直線。直線標準式為 $ax+by+c=0$ ，請寫一程式，可以指派一直線係數 a,b,c 。其次，可再輸入任一點座標，並判斷所輸入點是否在直線上。例如，指派 a,b,c 分別是 1,2,-4，那方程式就是 $x+2y-4=0$ ，其次，輸入點若是 (2,1)，那就在直線上，若是 (1,2)，那就不在直線上。
3. 請寫一程式，可以自動產生一個小寫字元，請判斷其是否為母音。說明，字元 a,e,i,o,u 稱為母音，其餘為子音。
4. 心算練習。請寫一個程式，可分別產生 2 個 1 位數整數亂數，由使用者輸入相加結果，電腦評判是否正確。
5. 音感練習。請寫一個程式，由電腦產生 1 ~ 8 的亂數，依照亂數發出 Do、Re、Mi 到高音 Do 的音，並由使用者輸入 1~8，電腦評判是否正確。

4-3 多重選擇結構敘述與撰寫

match case

大部分程式語言都有 switch case 解決多重選擇的決策問題，早期 Python 沒有 switch case 指令，但到了 Python 3.10 版本時，也順應潮流，使用 match case 解決多重選擇分歧問題，match case 簡易語法如下：

```
a=數值、字元或字串
match a:
    case 數值1、字元1或字串1:
        敘述區塊1
    case 數值2、字元2或字串2:
        敘述區塊2

    case ...
    case _: #以上沒有符合的條件時，自動執行此敘述區塊
        敘述區塊
```

例如，以下程式，可將 1,2,3,4 轉為春、夏、秋、冬。

```
a=1
match a:
    case 1:
        b='春'
    case 2:
        b='夏'
    case 3:
        b='秋'
    case 4:
        b='冬'
    case _:
        b='輸入錯誤'
print(b)
```

請留意以上程式僅能在 Python 3.10 以後版本執行，且同 if 指令，請特別留意縮排與冒號問題。若沒有一個條件符合，就有可能空手而回，因為 b 沒有宣告而輸出，導致程式錯誤，如以下程式。

```

a=3
match a:
    case 1:
        b='春'
    case 2:
        b='夏'
print(b)

```

應該在 `match` 前面先宣告變數「`a=3;b=' 輸入錯誤 '`」，或在 `match` 裡面加入以下程式：

```

case_ :
    b=' 輸入錯誤'

```

範例 4-3a

請寫一個程式，完成以下要求：

1. 輸入一個 0~100 的分數。
2. 當分數大於 90 分時，輸出 A。
3. 當分數介於 80~89 時，輸出 B。
4. 當分數介於 70~79 時，輸出 C。
5. 當分數介於 0~69 時，輸出 D。

👉 運算思維

1. 此一問題可先將分數除以 10 得到一個整數商，此整數商的結果僅剩 0~10 的 11 個整數，所以每個分數同時有 11 個選擇，此即為多重選擇問題，可以適用 `match case` 解決此問題。
2. 當不同的選擇條件，有相同的結果，可將這些選擇以「|」共用結果，例如，9 與 10 都是得到「A」，可以「9|10」表示，所以全部程式如下：

```

a=int(input("input a grade: "))
a=a//10; b=' 輸入錯誤'
match a:
    case 9|10:
        b='A' #大於等於90分為A

```

```

case 8:
    b='B' #大於等於80分且小於90分爲B
case 7: #大於等於70分且小於80分爲C
    b='C'
case 6|5|4|3|2|1|0:
    b='D'
case _:
    b='輸入錯誤'
print(b)

```

👉 執行結果

```

input a grade: 98
A

```

👉 自我練習

1. 請寫一程式，將所輸入的 0、1、2…6，轉爲 '星期日'、'星期一'…'星期六' 等字串。
2. 請寫一個程式，可以產生一個 0 到 25 的亂數，且依以下分數顯示燈號

```

21~25：五個燈。
16~20：四個燈。
11~15：三個燈。
6~10：兩個燈。
1~5：一個燈。
0：零個燈。

```

範例4-3b

猜拳遊戲。請寫一個程式，可以由人和電腦猜拳，並評定勝負。

👉 運算思維

1. 這一任務就是寫人工智慧程式的入門了，寫程式前先想一下您和一個不懂猜拳的人猜拳，那您們如何猜拳與評判勝負？首先，我們先規定有三種拳，分別是『剪刀、石頭與布』，每人每次僅能出一種拳法，且定義『剪刀贏布，石頭贏剪刀、布贏石頭，兩者相同則平手等』。『人工智慧』就是要把這些規定以程式語言表示，並由以上規定來評判勝負。

- 資料的數位化與變數的設計。我們人類猜拳是直接用手勢表示『剪刀、石頭與布』，但是如何讓電腦瞭解您的拳法呢，可以將以上『剪刀、石頭與布』以『1,2,3』表示，此即為資料的數位化。因為用『1,2,3』表示，只要 1byte 就可以，若您用『剪刀、石頭與布』表示，當然也可以，但一個中文字就佔用 2byte，且往後的資料處理也比較複雜。
- 電腦同時有三種拳法，所以適用 match，人也同時有三種拳法，也適用 match，所以共有 9 種情況。
- 您要讓電腦也能思考，也就是將以上九種情況的結果，以程式語言先規定好，所以程式如下：

```
import random
a=int(input("input 1,2,3:"))#記得要用int()轉型
b=random.randint(1,3)#產生1~3的整數亂數
astr='' #a代表的拳法
bstr='' #b代表的拳法
r='' #猜拳結果
match a:#people
    case 1:
        astr='剪刀'
        match b: #computer
            case 1:
                bstr='剪刀'
                r='平手'
            case 2:
                bstr='石頭'
                r='computer win'
            case 3:
                bstr='布'
                r='people win'
    case 2:
        astr='石頭'
        match b:
            case 1:
                bstr='剪刀'
                r='people win'
            case 2:
                bstr='石頭'
```

```

        r='平手'
    case 3:
        bstr='布'
        r='computer win'
case 3:
    astr='布'
    match b:
        case 1:
            bstr='剪刀'
            r='computer win'
        case 2:
            bstr='石頭'
            r='people win'
        case 3:
            bstr='布'
            r='平手'
print("您出 %s,電腦出 %s,結果是 %s" % (astr,bstr,r))

```

👉 執行結果

```

input 1,2,3:2
您出 石頭,電腦出 剪刀,結果是 people win

```

👉 補充說明

這一題輸入資料時，若忘了將資料轉為數值，如以下程式，因為沒有錯誤訊息，但就是沒結果。

```
a=input("input 1,2,3:")#請留意a是字串型態
```

👉 自我練習

1. 請寫一個程式，讓三個人同時猜拳。本例由電腦產生兩個亂數，由您和電腦猜，且評判輸贏。(提示：兩個人共 9 種情況，用兩層決策，三個人共 27 種情況，可用 3 層決策)

4-4 巢狀選擇結構敘述與撰寫

選擇結構內還有選擇結構稱為巢狀選擇結構，請看以下範例說明。

範例4-4a

同範例 4-3a，但改用 if else 巢狀迴圈重做。

👉 執行結果

```
input a grade: 88
the grade is B
```

👉 運算思維

1. 前面我們已經用 match case 完成此問題，但也可以使用巢狀迴圈，程式如下：

```
a=int(input("input a grade: "))#請留意input傳回字串型態
if(a>=90) : #大於等於90分爲A
    r='A'
else : #迴圈內允許還有迴圈
    if(a>=80):#大於等於80分且小於90分爲B
        r='B'
    else :
        if(a>=70):
            r='C'
        else :
            r='D'
print("the grade is %c" % r)
```

2. 以上逐漸縮排，有時太多的縮排，程式容易產生斷行，不易閱讀，所以也可以使用 if~elif~ 代替，程式如下：

```
a=int(input("input a grade: "))#請留意input傳回字串型態
if(a>=90) : #大於等於90分爲A
    r='A'
elif(a>=80):#大於等於80分且小於90分爲B
    r='B'
elif(a>=70):
```

```
    r='C'
else :
    r='D'
print("the grade is %c" % r)
```

3. 以下程式就不行。因為 if-elif-else 語法在進行條件比對時是由上往下逐一比對，只要一有條件滿足就進入該對應區塊執行，執行完畢後不會對於下方的剩餘條件繼續比對，所以比對條件的順序需要特別注意，這是許多學生在學習 if-elif-else 語法時常犯的錯誤。

```
a=int(input("input a grade: "))#請留意input傳回字串型態
if(a>=70) : #大於等於70分爲C
    r='C'
elif(a>=80):#大於等於80分爲B
    r='B'
elif(a>=90):
    r='A'
else :
    r='D'
print("the grade is %c" % r)
```

4. 考試成績的分布，通常中間分數的人較多，此時可以使用二分法調整執行順序，程式如下，這樣不管分數為何，至多比較 2 次，所以可以減少比較次數，提升程式執行效率，程式如下：

```
a=70
if a>=80:
    if a>=90:
        b='A'
    else:
        b='B'
else:
    if a>=70:
        b='C'
    else:
        b='D'
print(b)
```

👉 自我練習

1. 以範例 4-4a 為例，請將「`a=int(input("input a grade: "))`」改為「`a=input("input a grade: ")`」，並觀察執行結果。
2. 請寫一程式，電腦自動產生 -5~5 的整數亂數，並判斷是正數、0 或負數。
3. 某一貨品定價 100 元，若購買 10 件 (含) 以上打 9 折，若購買 30 ~ 99 件則打 8 折，若購買 100 件以上則打 7 折，試寫一程式可以輸入購買件數而得總價。測試資料如下：

輸入	輸出
5	500
10	900 (100*10*0.9)
30	2400 (100*30*0.8)
100	7000 (100*100*0.7)

4. 請寫一個程式，可以產生一個 0 到 25 的亂數，且依以下分數顯示燈號
 - 21 ~ 25：五個燈。
 - 16 ~ 20：四個燈。
 - 11 ~ 15：三個燈。
 - 6 ~ 10：兩個燈。
 - 1 ~ 5：一個燈。
 - 0：零個燈。
5. 同範例 4-4a，但程式一執行，要求先輸入密碼，密碼若為『abcd』，那才可執行本程式。
6. 同範例 4-4a，但程式一執行，要求先輸入密碼，密碼若為『aa11』、『bb22』、『cc33』或『abcd』，才可執行本程式。

範例 4-4b

請寫一個程式，可以判斷所輸入座標的所在象限。爲了簡化問題，本例先不考慮 $x = 0$ 或 $y = 0$ 的座標軸與原點，而是留到本範例自我練習，再補齊程式。

👉 運算思維

當 $x > 0$ 時，還要繼續以 y 判斷在第 1，或第 4 象限，此即爲選擇內還要選擇，所以可套用巢狀選擇，程式如下：

```
x,y=eval(input("input x,y:")) #數字請用逗號『,』隔開，例如輸入 3,4
if x>0 :
    if y>0:
        b="I"
    else:
        b="IV"
else:
    if y>0:
        b="II"
    else:
        b="III"
print(b)
```

👉 執行結果

```
input x,y:3,-2
IV
```

👉 補充說明

1. 請留意 `eval()` 傳回數值型態。
2. 此題目有人會寫成

```
x,y=eval(input("input x,y:")) #數字請用逗號『,』隔開，例如 3,4
if (x>0 and y>0) :
    b="I"
if (x<0 and y>0) :
    b="II";
```

```

if (x<0 and y<0) :
    b="III"
if (x>0 and y<0 ):
    b="IV"
print(b)

```

這樣雖然沒有錯，但是執行效率非常差，因為電腦要不斷的比較。

自我練習

1. 同上範例 4-4b，但增加先判斷是否在原點或 x、y 軸上。測試資料如下

輸入(x,y)	輸出
(0,0)	原點
(0,3)	y 軸
(3,4)	I

4-5 選擇結構程式實作演練

極大與極小

極大與極小是日常資料處理最常見的問題，以下我們先介紹 3 筆資料的極大值的求法，4~5 筆資料的極大或極小請自行擴充。6 筆以上資料，程式就冗長了，資料就需要以串列儲存，並以迴圈撰寫，待學完串列就能理解迴圈與串列的特異功能。

範例 4-5a

假如有 3 筆資料如下：

3,8,2,

請寫一程式，找出極大值。

設計步驟

1. 資料的數位化與變數設計。本例以單一變數儲存如下：

```
a=3;b=8;c=2
```

2. 寫出演算法則。

(1) 設定極大值 (max) 為第一數。

```
max=a
```

(2) 當第二數 (b) 大於極大值時，極大值即以 b 取代。

```
if (b>max):  
    max=b
```

(3) 當第三數 (c) 大於極大值時，極大值即以 c 取代。

```
if (c>max):  
    max=c
```

(4) 輸出極大值。

```
print(max)
```

程式列印

```
a=4;b=5;c=1  
max=a  
if b>max:  
    max=b  
if c>max:  
    max=c  
print(max)
```

執行結果

5

補充說明

1. 以上是求極大值的演算法，但 Python 已經有 max() 函式，所以本例也可以寫成如下：

```
a=4;b=5;c=1  
max=max(a,b,c)  
print(max)
```

自我練習

1. 假如有 4 筆資料如下：
4,8,2,5
如何找出極大值。
2. 有一批資料，含有 3 個人的人名與成績，請寫一程式，可以找出最低分的人名與成績。例如：

```
aa, 60  
bb, 30  
cc, 80
```

3. 請寫一程式，可以指派 5 個數值，請去掉最大值與最小值，再求其平均。

排序

將資料由小而大、或由大而小排列稱為排序。

氣泡排序法

為了充分理解氣泡排序的原理，我們先逐一探討 2 筆、3 筆、4 筆、5 筆資料的排序，這樣循序漸進，就能找出規律性，等待學習迴圈與串列才能很快寫出程式。首先，我們先探討 2 筆資料，若有 2 筆資料如下：(備註：所有的排序程式，測試資料一定要用最遭的情況，也就是要與排序後完全相反，這樣才能觀察排序是否完成。)

```
5 4
```

我們只要比較 1 次就好。若第 1 筆大於第 2 筆，則 2 者交換，排序完成。程式如下：

```
a=5;b=4  
if a> b:  
    a,b=b,a  
print(a,b)
```

以上程式，比較過程如下表：

階次	5	4	比較方式	比較次數
1	4	5	(5,4)	1

若有 3 筆資料如下

5 4 3

則需要兩個階次，程式如下：

```
a=5;b=4;c=3
#階次1，比較2次
if a> b:
    a,b=b,a
if b>c:
    b,c=c,b
print(a,b,c)#c 此時最大，不用再比較
#階次2，比較1次
if a> b:
    a,b=b,a
print(a,b,c)
```

以上程式，比較過程如下表：

階次	5	4	3	比較方式	比較次數
1	4	3	5	(5,4)(5,3)	2
2	3	4		(4,3)	1

若有 4 筆資料，共需要 3 個階次，排序方法如下：

```
a=5;b=4;c=3;d=2
#階次1，比較3次
if a> b:
    a,b=b,a
if b>c:
    b,c=c,b
if c>d:
    c,d=d,c
print(a,b,c,d)#d 已經最大，不用再比較
```

```

#階次2，比較2次
if a > b:
    a,b=b,a
if b > c:
    b,c=c,b
print(a,b,c,d) #c 已經次大，不用再比較
#階次3，比較1次
if a > b:
    a,b=b,a
print(a,b,c,d) #排序完成

```

以上程式，比較過程如下表：

階次	5	4	3	2	比較方式	比較次數
1	4	3	2	5	(5,4)(5,3)(5,2)	3
2	3	2	4		(4,3)(4,2)	2
3	2	3			(3,2)	1

若有 5 筆資料，共需要 4 個階次，排序方法如下：

```

a=5;b=4;c=3;d=2;e=1
#階次1，比較4次
if a > b:
    a,b=b,a
if b > c:
    b,c=c,b
if c > d:
    c,d=d,c
if d > e:
    d,e=e,d
print(a,b,c,d,e) #e 已經最大，不用再比較
#階次2，比較3次
if a > b:
    a,b=b,a
if b > c:
    b,c=c,b
if c > d:
    c,d=d,c
print(a,b,c,d,e) #d 已經次大，不用再比較
#階次3，比較2次

```

```

if a > b:
    a, b = b, a
if b > c:
    b, c = c, b
print(a, b, c, d, e)
#階次4，比較1次
if a > b:
    a, b = b, a
print(a, b, c, d, e)

```

以上程式，比較過程如下表：

階次	5	4	3	2	1	比較方式	比較次數
1	4	3	2	1	5	(5,4)(5,3)(5,2)(5,1)	4
2	3	2	1	4		(4,3)(4,2)	3
3	2	1	3			(3,2)	2
4	1	2				(2,1)	1

以上 5 筆資料，程式就很冗長了，那如果 100 筆、1000 筆呢，程式是不是更冗長？所幸，答案是否定的，因為後續的迴圈與串列就可解決此問題。

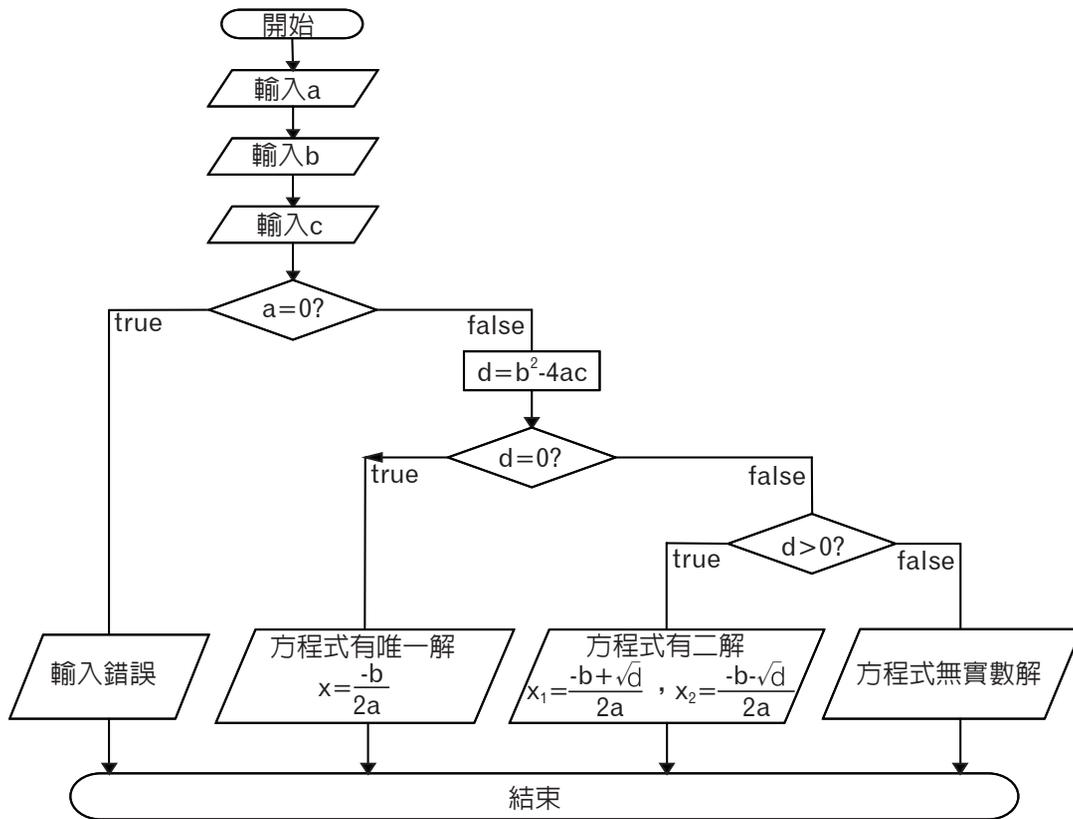
⚙️ 一元二次方程式

前面我們直接假設所輸入的一元二次方程式有解，但並不是隨便給三個係數，一個方程式就通通有解，本例則要加上判斷了。解一元二次方程式的演算法如下：

1. 設有一元二次方程式如下：

$$ax^2 + bx + c = 0$$

2. 若 $a=0$ 則輸出“輸入錯誤”。
3. 令 $d=b^2 - 4ac$ 。
4. 若 $d=0$ ，則方程式有唯一解 $x = \frac{-b}{2a}$ ；否則，若 $d>0$ ，則方程式有二解 $x_1 = \frac{-b+\sqrt{d}}{2a}$ ， $x_2 = \frac{-b-\sqrt{d}}{2a}$ ；否則，無實數解。以上演算分析，以流程圖說明如下：

**範例 4-5b**

請設計一個程式，可以解一元二次方程式 $ax^2 + bx + c = 0$ 。

程式列印

```

a,b,c=eval(input("input a,b,c:"))
if(a==0):
    print("input error")# 若a=0，則列印錯誤訊息
else:
    d=b**2-4*a*c# 計算d值 */
    if(d==0):
        print("only one answer,x= %d" % (-b/(2*a)))
    elif d>0 :
        d=d**(1/2)
        x1=(-b+d)/(2*a)
        x2=(-b-d)/(2*a)
        print("two answer,x1= %f,x2=%f"%(x1,x2))
    else:
        print("no real answer")
  
```

👉 執行結果

```
input a,b,c:1,-5,6
two answer,x1= 3.000000,x2=2.000000
```

👉 自我練習

1. 解二元一次方程式。解二元一次方程式的演算法如下：

(1) 設二元一次方程式如下：

$$a_1x + b_1y = c_1$$

$$a_2x + b_2y = c_2$$

(2) 令 $d = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}$ (表示 $d = a_1b_2 - a_2b_1$)

(3) 假如 $\frac{a_1}{a_2} = \frac{b_1}{b_2} = \frac{c_1}{c_2}$ ，則方程式無限多解，且程式結束。(代表兩重疊直線)

(4) 否則，假如 $\frac{a_1}{a_2} = \frac{b_1}{b_2} \neq \frac{c_1}{c_2}$ ，則方程式無解，且程式結束。(代表兩平行直線)

$$(5) x = \frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{d} = (c_1b_2 - c_2b_1)/d$$

$$(6) y = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{d} = (a_1c_2 - a_2c_1)/d$$

2. 請設計一個程式，可以解二元一次方程式。

範例4-5c

三角形判斷 (APCS105 第二梯次試題)

若已知三角形三邊長，判斷是否構成三角形、評定三角形種類與計算三角形面積的演算法如下：

(1) 輸入三角形的三邊長 a 、 b 、 c 。

(2) 任兩邊之和要大於第三邊，才能繼續以下計算，否則輸出『無法構成三角形』，且程式結束。

(3) 若 c 是最長邊，假如 $a^2 + b^2 > c^2$ 則為銳角三角形；否則，假如 $a^2 + b^2 = c^2$ ，則為直角三角形；否則此三角形為鈍角三角形。

(4) 令 $d = \frac{1}{2}(a + b + c)$

(5) 三角形面積 = $\sqrt{d(d-a)(d-b)(d-c)}$

請輸入三角形三邊長，首先判斷是否構成三角形、其次判別三角形的種類，最後計算其面積。

👉 程式列印

任兩邊之和要大於第三邊的程式是「if a+b>c and b+c>a and c+a>b」，本例就將最大邊找出來，只要最小兩邊之和大於第三邊，這樣就一石二鳥，可以方便判斷是否形成三角形，也能判斷三角形種類。

```
a,b,c=eval(input("input a,b,c:"))
#以爲氣泡排序法將a,b,c三個數字排序
if a>b:
    a,b=b,a
if b>c:
    b,c=c,b
#c已經是最大
if (a+b>c) :
    c2=c**2
    b2=b**2
    a2=a**2
    if ((a2+b2)>c2):
        t="銳角三角形"
    elif ((a2+b2)==c2):
        t="直角三角形"
    else:
        t="鈍角三角形"
    d=(a+b+c)/2
    area=(d*(d-a)*(d-b)*(d-c))**(1/2)#留意括號的對稱
    print(t)
    print(area)
else:
    print("無法構成三角形")
```

👉 執行結果

```
input a,b,c:3,4,5
直角三角形
6.0
```

※ 範例4-5d

示範實作電子琴。

👉 程式列印

1. 前面已經介紹電腦如何發音，本例希望按鍵盤「1」發「Do」，按「2」發「Re」，按「3」發「Mi」，以上即是選擇結構的應用，程式如下：
2. thinter 可實作視窗介面，不是本書介紹範圍，請在此先體驗就好。

```
from tkinter import * #載入視窗模組
import ctypes #載入發音模組
p = ctypes.windll.kernel32
def aa(e):
    k=e.char
    tk.title(k)
    if k=='1':
        p.Beep(523,200) #發Do音
    elif k=='2':
        p.Beep(587,200)
    elif k=='3':
        p.Beep(659,200)
    elif k=='4':
        p.Beep(698,200)
    elif k=='5':
        p.Beep(784,200)
    elif k=='6':
        p.Beep(880,200)
    elif k=='7':
        p.Beep(988,200)
    elif k=='8':
        p.Beep(1046,200)
tk=Tk() #產生視窗
tk.geometry("300x100+50+70") #視窗大小
```

```

enyl=Entry(tk)#產生輸入盒
enyl.bind('<Key>',aa)#綁定aa函式
enyl.pack()#版面配置
tk.mainloop()

```

執行結果



自我練習

1. 同範例 4-5d，但改為以 match 重作。

範例4-5e

實例探討。郵局信件郵資計算。

說明

中華郵政普通信函郵資依照信件重量費用如圖 4-1。

 A screenshot of a web browser displaying the Chinese Post website. The browser's address bar shows 'post.gov.tw/post/internet/Postal/index.jsp?ID=2050103'. The page title is '國內郵件資費表 | 國內函件資費表'. On the left, there is an image of various mail items with the text '郵件資費表'. On the right, there is a table with the following data:

重量 (公克)	不逾 20	21- 50	51- 250	250- 500	500- 1000	1001- 2000
郵資	8	16	40	72	112	160

★ 圖4-1 中華郵政普通信件資費表

請寫一個程式，可以輸入重量，而得到資費。

👉 程式列印

此為巢狀選擇問題，程式如下：

```
w=12
m=0
if w<=20:
    m=8
elif w<=50:
    m=16
elif w<=250:
    m=40
elif w<=500:
    m=72
elif w<=1000:
    m=112
elif w<=2000:
    m=160
else :
    m="超重"
print(m)
```

👉 執行結果

8

範例4-5f

勞動基準法。

勞動基準法第 38 條內文如下：

第 38 條：勞工在同一雇主或事業單位，繼續工作滿一定期間者，應依下列規定給予特別休假：

- 一、六個月以上一年未滿者，三日。
- 二、一年以上二年未滿者，七日。
- 三、二年以上三年未滿者，十日。
- 四、三年以上五年未滿者，每年十四日。
- 五、五年以上十年未滿者，每年十五日。
- 六、十年以上者，每一年加給一日，加至三十日為止。

請寫一個程式，可以依照勞工任職年資，計算其特休天數。

👉 程式列印

此也是巢狀選擇經典問題，程式如下：

```
#測試資料，年資
#a=0.5 #b=0
#a=0.6 #b=3
#a=0.7 #b=3
#a=1 #b=7
#a=9.5 #b=15
#a=10.1 #b=16
a=11.2 #b=17
b=0 #特休天數
if a<0.6 :
    b=0
elif a<1:
    b=3
elif a<2:
    b=7
elif a<3:
    b=10
elif a<5:
    b=14
elif a<10:
    b=15
else :
    b=15+(a-9)//1 #取最小整數
    #每超過1年，年休增加1天
    if b>=30: #超過30天，以30天計算
        b=30
print(b)
```

👉 執行結果

4-6 本章內容摘要

1. Python 選擇結構有 if~else 、match case 、if~elif~else ，且請留意要有冒號與縮排問題。
2. if~else 用法如下：

```
a=66
if a>=60:
    b="Pass"
else:
    b="Fail"
print(b)
```

3. match case 用法如下：

```
a=1 ;b=''
match a:
    case 1:
        b='春'
    case 2:
        b='夏'
    case _:
        b='輸入錯誤'
print(b)
```

請留意以上程式僅能在 Python 3.10 以後的版本執行。

4. 巢狀的 if~else~if~else 可用 if~elif~else 代替，用法如下：

```
a=3
if a<0 :
    re='負數'
elif a==0:
    re='Zero'
else:
    re='正數'
print(b)
```

但請留意條件的擺放方式與順序，避免執行錯誤與浪費時間。

課後評量

一、填充題 (請寫出以下程式執行結果)

題號	題目	執行結果
1	<pre>a=3 if a%2==0: b='偶數' else: b='奇數' print(b)</pre>	
2	<pre>a,b,c=1,2,-4 x,y=2,1 if a*x+b*y+c==0: print('在直線上') else: print('不在直線上')</pre>	
3	<pre>x='a' x=input('input a char:') if x=='a' or x=='e' or x=='i' or x=='o' or x=='u': print('母音') else: print('子音')</pre>	
4	<pre>a=20 match a: case 0: b='0' case 1 2 3 4 5: b='1' case 6 7 8 9 10: b='2' case 11 12 13 14 15: b='3' case 16 17 18 19 20: b='4' case 21 22 23 24 25: b='5' print(b)</pre>	

5	<pre>x=20 print(x) if x>=21: print('five lights') else: if x>=16: print('four lights') else: if x>=11: print('three lights') else: if x>=6: print('two lights') else: if x>=1: print('one light') else: print('zero light')</pre>	
6	<pre>x=4 print(x) if x>=1: print('five lights') else: if x>=6: print('four lights') else: if x>=11: print('three lights') else: if x>=16: print('two lights') else: if x>=20: print('one light') else: print('zero light')</pre>	

7	<pre>a=75 if a>=80: if a>=90: b='A' else: b='B' else: if a>=70: b='C' else: b='D' print(b)</pre>	
8	<pre>a=6;b=8;c=4;d=2;e=9 max=min=a if b>max:max=b if c>max:max=c if d>max:max=d if e>max:max=e if b<min:min=b if c<min:min=c if d<min:min=d if e<min:min=e sum=a+b+c+d+e-max-min avg=sum/3 print(avg)</pre>	
9	<pre>a=5;b=4;c=3;d=2 #階次1 if a>b: a,b=b,a if b>c: b,c=c,b if c>d: c,d=d,c print(a,b,c,d) #階次2 if a>b:a,b=b,a if b>c:b,c=c,b print(a,b,c,d)</pre>	

10	<pre> a1,b1,c1=3,1,5 a2,b2,c2=1,-2,-3 if a1/a2==b1/b2==c1/c2: print('二直線重疊') elif a1/a2==b1/b2!=c1/c2: print('二直線平行') else: d=a1*b2-a2*b1 x=(c1*b2-c2*b1)/d y=(a1*c2-a2*c1)/d print(x,y) </pre>	
11	<pre> x=200 #度數 if x<100: print(x*3) elif x<300: print(100*3+(x-100)*5) else: print(100*3+200*5+(x-300)*6) </pre>	
12	<pre> a='1';b=0 if a==1: b=1 print(b) </pre>	
13	<pre> b=0 a=input("input 1:")#本例輸入1 if a==1: b=1 print(b) </pre>	
14	<pre> a='1';b=0 match a: case 1: b=1 case '1': b=2 case _: b=3 print(b) </pre>	
15	<pre> b=0;a=2 if a>0: b=1 if a==2: b=2 print(b) </pre>	

課後習題

1. 假設所得稅稅率累進法則如下：

(1) 淨所得 30 萬以下繳納 6%。

淨所得 30 ~ 80 萬之間，則前面 30 萬繳納 6%，超過 30 萬的部分繳納 13%。

(3) 淨所得在 80 ~ 200 萬之間繳納 21%。(前面 30 萬繳 6%，30 ~ 80 萬之間繳 13%)

(4) 淨所得超過 200 萬，超過的部分繳納 30%。

試寫一程式可以輸入淨所得，並計算應繳納稅額。測資如下：

淨所得	納稅額
20 萬	$20 \text{ 萬} * 6\% = 12000$
40 萬	$30 \text{ 萬} * 6\% + 10 \text{ 萬} * 13\% = 31000$
100 萬	$30 \text{ 萬} * 6\% + 50 \text{ 萬} * 13\% + 20 \text{ 萬} * 21\% = 125000$
220 萬	$30 \text{ 萬} * 6\% + 50 \text{ 萬} * 13\% + 120 \text{ 萬} * 21\% + 20 \text{ 萬} * 30\% = 395000$

2. 假設自來水費率如下：

100 度以下，每度 3 元。

100 ~ 300 度，超過 100 度的部分，每度 5 元。

300 度以上，超過 300 度的部分，每度 6 元。

根據以上條件，寫一程式，可輸入用水度數，得到水費。例如，測資如下表：

度數	水費
50	$50 * 3 = 150$
200	$100 * 3 + 100 * 5 = 800$
400	$100 * 3 + 200 * 5 + 100 * 6 = 1900$

3. 寫一程式輸入 x 值，並印出其所對應的值，其函數對應如下：

$$y = f(x) = \begin{cases} x+3 & x > 100 \\ x^2 & 50 \leq x \leq 100 \\ \sqrt{x} & 0 < x < 50 \\ 0 & x \leq 0 \end{cases}$$

4. 三點共線。請寫一個程式，任意指派 3 點座標，檢查是否共線，若未共線，輸出其面積。
提示：任意 3 點面積若為 0，則共線。例如：(1,1),(2,2),(3,3) 共線；(1,1),(2,2),(5,2) 不共線。
5. 判斷任意點 D 是否在三角形 ABC 內或外。已知 ABC 三點座標，請寫一程式，可以輸入 D 點座標，並判斷任一點 D 是否在三角形 ABC 內或外。
(提示：D 若在三角形內，則三角形 ABC 面積 = DAB + DBC + DAC 的面積)
例如：三角形三點分別是 (0,0),(3,0),(0,4)，則 (1,1) 在三角形內，(5,0) 則在三角形外。
6. 衛福部國民健康署，對於體重身高的比例訂有 BMI 標準如圖 4-2：請寫一程式完成此一指數計算，並給予建議。



成人肥胖定義	身體質量指數(BMI)(kg/m ²)	腰圍(cm)
體重過輕	BMI < 18.5	
健康體位	18.5 ≤ BMI < 24	
體位異常	過重：24 ≤ BMI < 27 輕度肥胖：27 ≤ BMI < 30 中度肥胖：30 ≤ BMI < 35 重度肥胖：BMI ≥ 35	男性：≥ 90 公分 女性：≥ 80 公分

$$BMI = \frac{\text{體重 (公斤)}}{\text{身高}^2 \text{ (公尺)}}$$

★ 圖4-2 BMI 定義與建議表

7. 台北市計程車費率計算如下：
 - (1) 計程運價：起程 1.25 公里 70 元，續程每 200 公尺 5 元（日間）。
 - (2) 延滯計時運價：車速每小時 5 公里以下，累計每 1 分鐘 20 秒 5 元（日間）。
 - (3) 夜間加成運價：自夜間 11 時起至翌晨 6 時止（遇跨夜間加成時段之情形，統一以「上車時間」為準），每趟次依日間運價加收 20 元（即起程運價 1.25 公里 90 元，續程及延滯計時運價與日間相同）。

請問如何規劃與設計此程式？

測試資料編號	搭乘時間(時:分)	行駛距離(公里)	延滯計時累計時間(分:秒)	費率
1	07:20	1	1:20	$70+5=75$
2	08:30	1.3	5:00	$70+5*1+5*3=90$
3	23:10	1.9	6:00	$70+20+5*4+5*4=130$
4	03:40	2	7:00	$70+20+5*4+5*5=135$

8. 營業稅的計算。營業稅若外加，則營業稅 = 定價 * 0.05，若內含，則營業稅 = 定價 / 21，且都四捨五入到整數。請寫一程式，程式一執行，先詢問是稅外加，還是內含，然後輸入定價，最後由電腦計算營業稅。