


程式設計概論
Programming 101
一程式的流程控制與邏輯判斷語法

授課老師：邱淑怡

DATE: 10/11/2023










Outline

- 流程圖
- 程式結構有三種
- 單向判斷式結構
- 雙向判斷式結構
- 巢狀判斷式結構
- 多向判斷式結構

流程圖的功用

- 以圖形化的呈現方式來摘要程式流程。
- 可在撰寫程式碼前，做為事先規劃的工具。
- 可概覽程式流程的整體結構，做為與他人溝通程式流程的輔助工具。

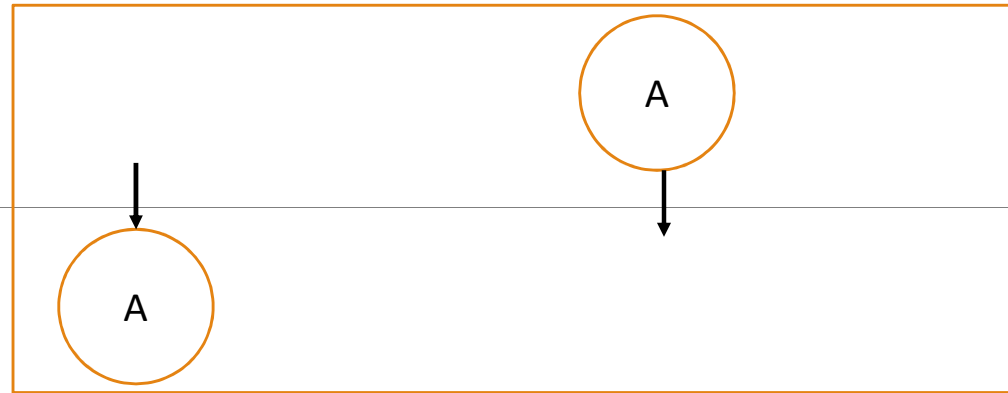
流程圖符號

名稱	符號圖形	意義
開始或結束符號		表示程式流程的開始或結束
流程符號		表示程式流程的進行方向
程序處理符號		表示要進行處理的工作
輸入或輸出符號		表示資料輸入或結果輸出
決策判斷符號		表示根據條件式來判斷程式接下來的進行方向
同頁連接點		連接同一頁的流程
跨頁連接點		跨頁連接流程圖

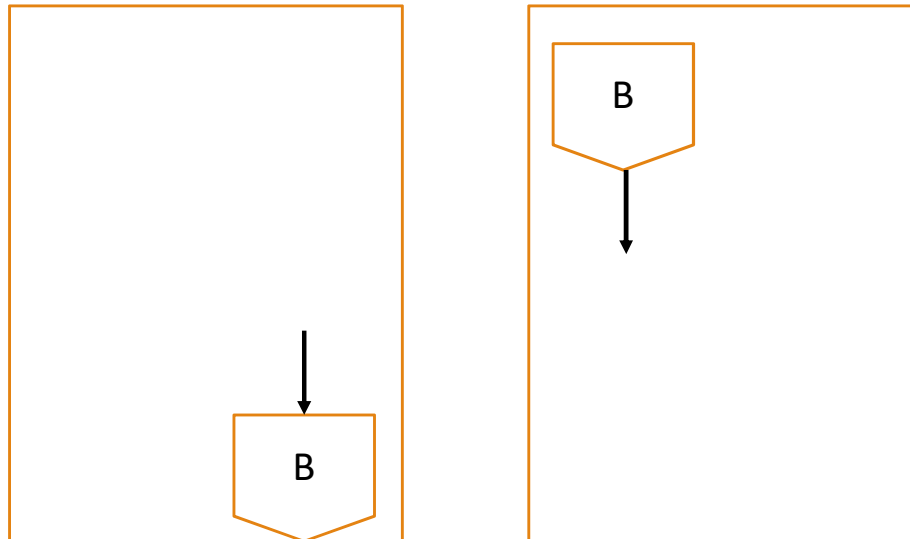


連接點

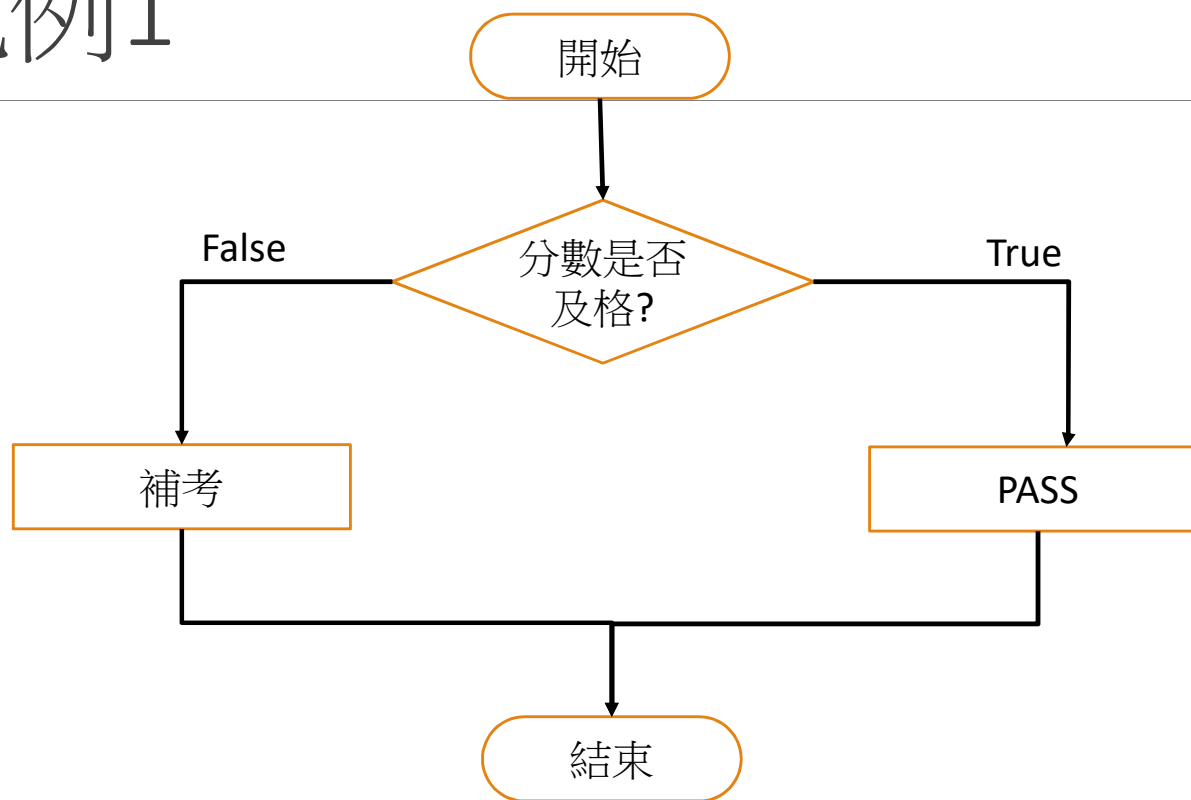
同頁連接點



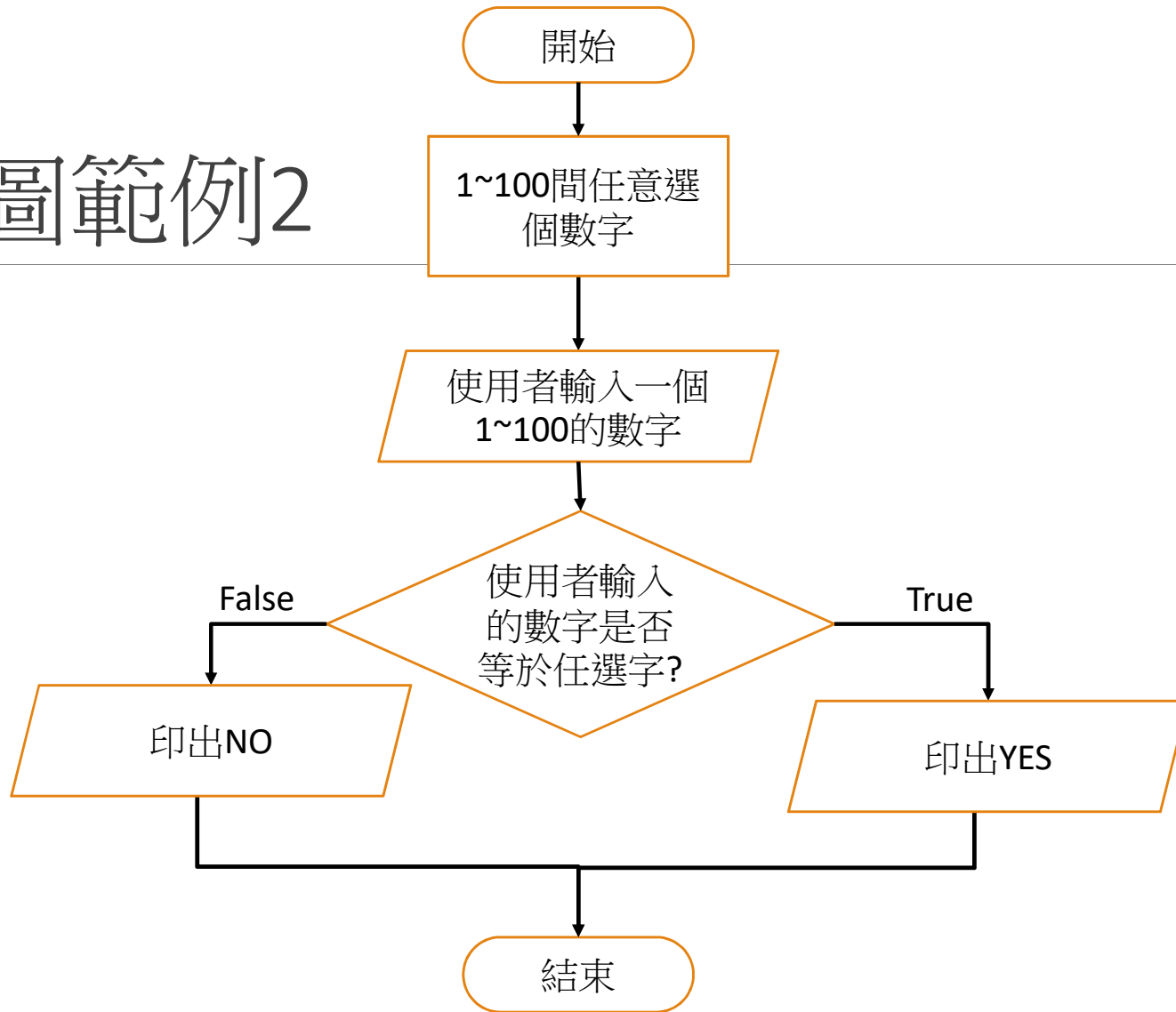
跨頁連接點



流程圖範例1



流程圖範例2

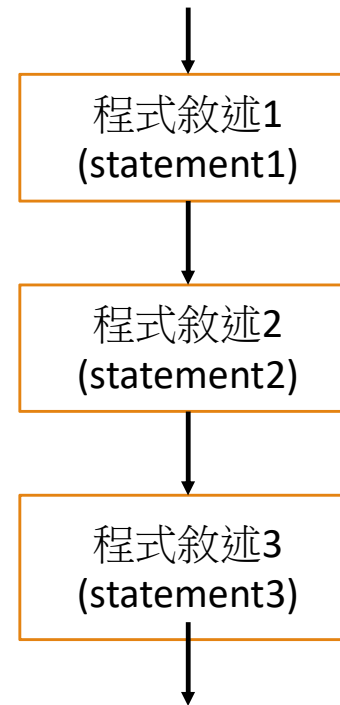


三種基本程式結構

1. 循序結構
2. 選擇結構
3. 重複結構(下周詳述)

1. 循序結構語法

- `print('Hello World!')`
- `a= 128//7`
- `print(a+3)`
- `print('Hello'+ 'Everyone')`

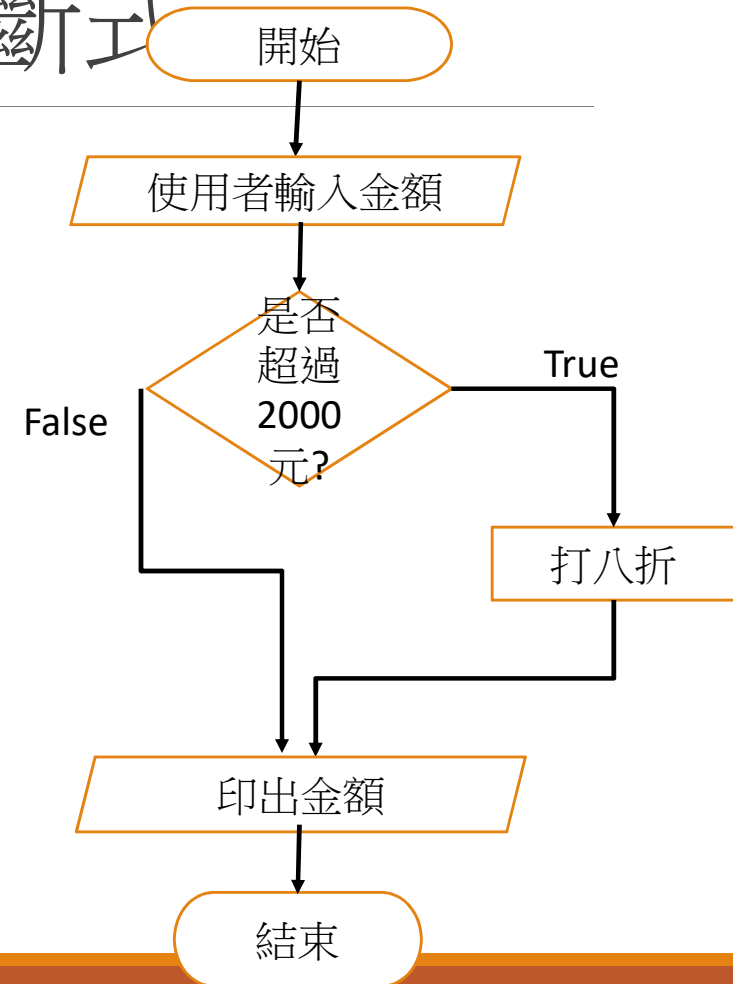
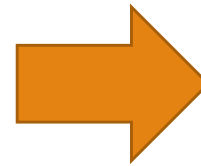


2. 選擇結構語法---單向判斷式

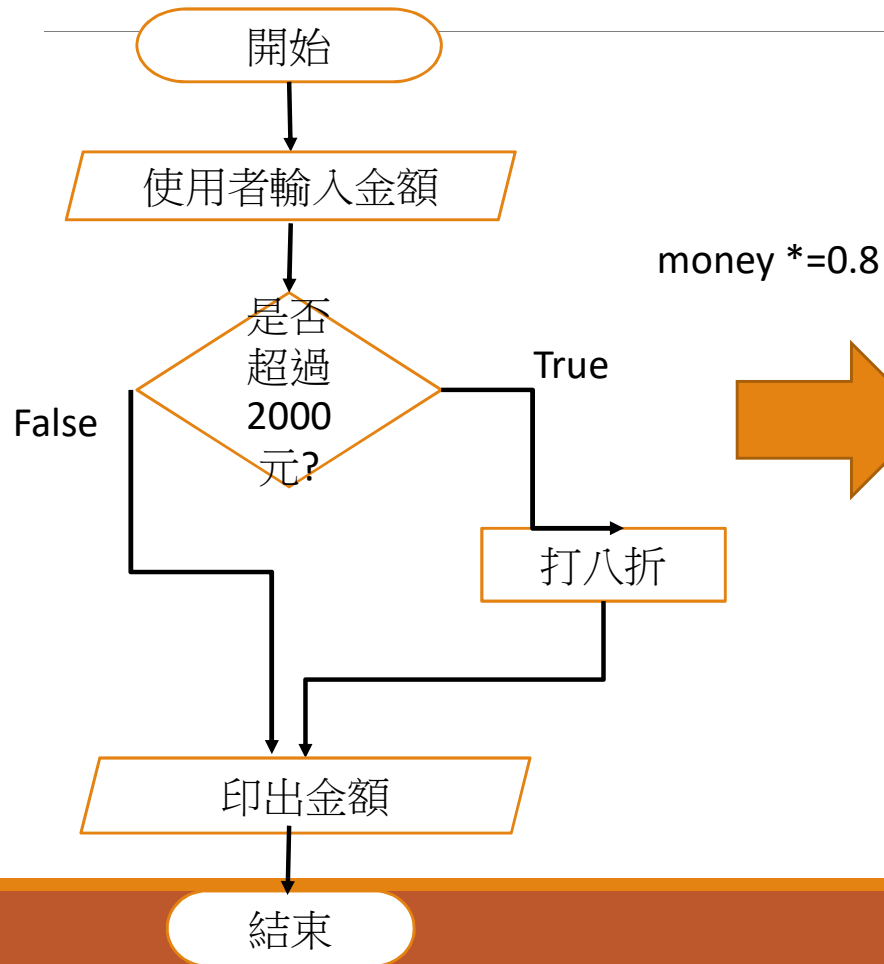
- if 條件運算式:
- 程式敘述一 #可以多筆敘述

題目：

讓使用者輸入金額，是否超過2000元，若超過就金額就打八折，最後印出實付金額。



單向判斷式(cont.)

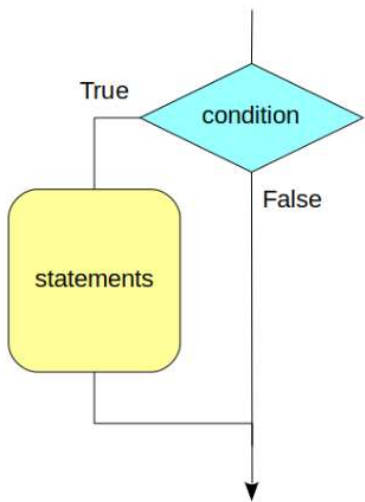


Python code:

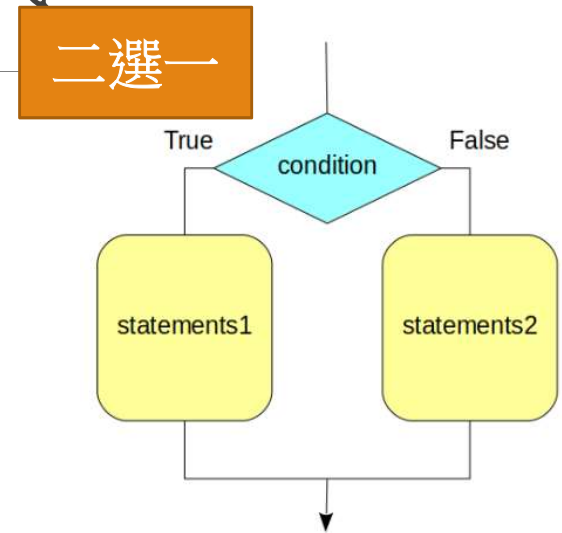
```
money = int(input('輸入金額: '))  
if money > 2000:  
    print('打八折')  
    # money=money*0.8  
print("實際付款{:8.2f}".format(money))
```

2. 選擇結構語法 --- 雙向判斷式

if <condition(條件運算式)>:
statements



if <condition>:
statement1
else:
statement2

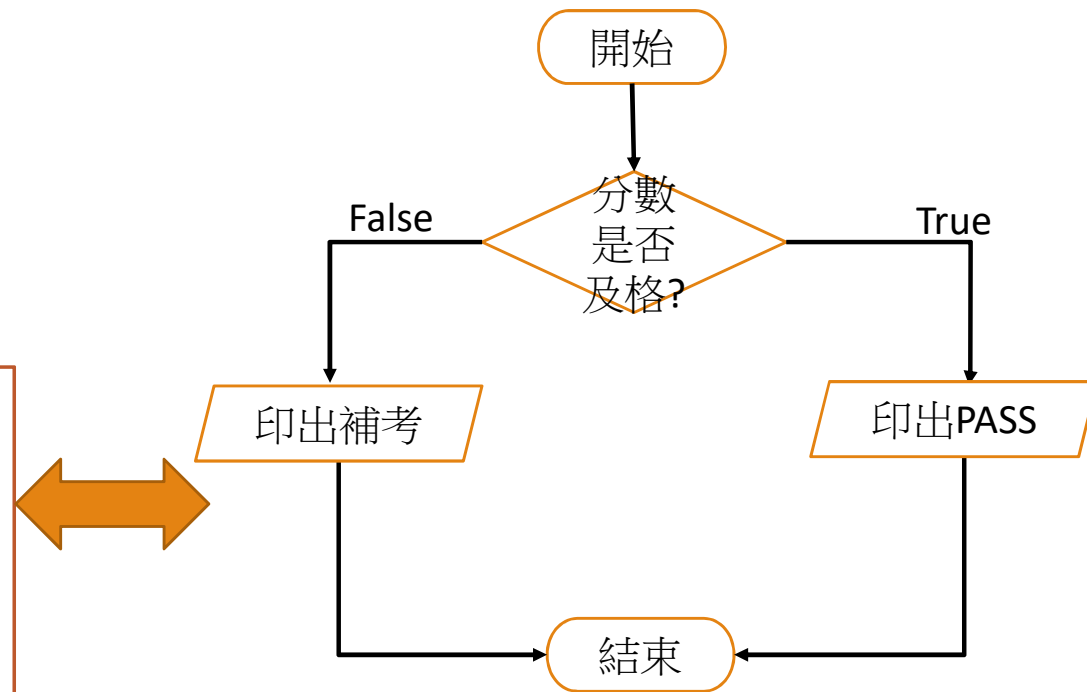


雙向判斷式

- if 條件運算式:
- 程式敘述一 #可以多筆敘述
- else:
- 程式敘述一 #可以多筆敘述

例子：

```
grade = eval(input('輸入分數: '))  
if grade >= 60:  
    print('PASS')  
else:  
    print("補考")
```



條件運算式(condition)說明

□ 「>=」，稱之為條件運算子，可以用來判斷左邊的運算結果，是否大於等於右邊的運算結果

□ 其他的條件運算子

- 等於：「==」
- 不等於：「!=」
- 大於：「>」
- 大於等於：「>=」
- 小於：「<」
- 小於等於：「<=」

```
grade = int(input('輸入分數: '))
if grade >= 60:
    print('Pass')
else:
    print("Fail")
```

if敘述句_練習題

- 請撰寫一個Python程式，它可以要求使用者輸入一個整數，然後檢查該整數若為偶數，就印出“它是偶數”並且顯示使用者所輸入的整數數值，否則就印出“它是奇數”並且顯示使用者所輸入的整數數值。

巢狀判斷式結構：if敘述句還可以有[if else]敘述句

```
if <condition1>:
```

```
    if <condition2>:
```

```
        statement1
```

```
    else:
```

```
        statement2
```

```
else:
```

```
    if <condition3>:
```

```
        statement3
```

```
    else:
```

```
        statement4
```

例子: 找出a, b, c 中最大的數

a, b, c=3, 5, 7

```
x=0
```

```
if a < b:
```

```
    if b < c:
```

```
        x = c
```

```
    else: # (b>=c)
```

```
        x = b
```

```
else: # (a>=b)
```

```
    if a < c:
```

```
        x = c
```

```
    else: # (a>=c)
```

```
        x = a
```

```
print(x)
```


多重條件運算式

□ 且：「and」

◦ 範例：「`a == b and 5 == c`」

□ 或：「or」

◦ 範例：「`x != y or z == 3`」

實例練習題

- 請撰寫一個Python程式，它可以要求使用者輸入數值，並且判斷是否能整除3而且能整除4，若能整除這兩個數就顯示“正確”；若不是則顯示“不正確”

比較差異:前提:讓使用者輸入數字

當使用者輸入為偶數時，再判斷是否能被 10 整除：

請分別印出“該數為偶數且被10整除”及“該數為偶數但不能被10整除”

判斷使用者輸入為偶數時，再判斷是否能被 10 整除

請分別印出“該數為偶數且被10整除”及“該數為奇數”

比較差異:前提:讓使用者輸入數字

當使用者輸入為偶數時，再判斷是否能被 10 整除：

請分別印出“該數為偶數且被10整除”及“該數為偶數但不能被10整除”

判斷使用者輸入為偶數時，再判斷是否能被 10 整除

請分別印出“該數為偶數且被10整除”及“該數為奇數”

```
if condition_1:  
    if condition_2:  
        doSomething()  
    else:  
        doOther()
```

```
if condition_1:  
    if condition_2:  
        doSomething()  
else:  
    doOther()
```

多向判斷式結構

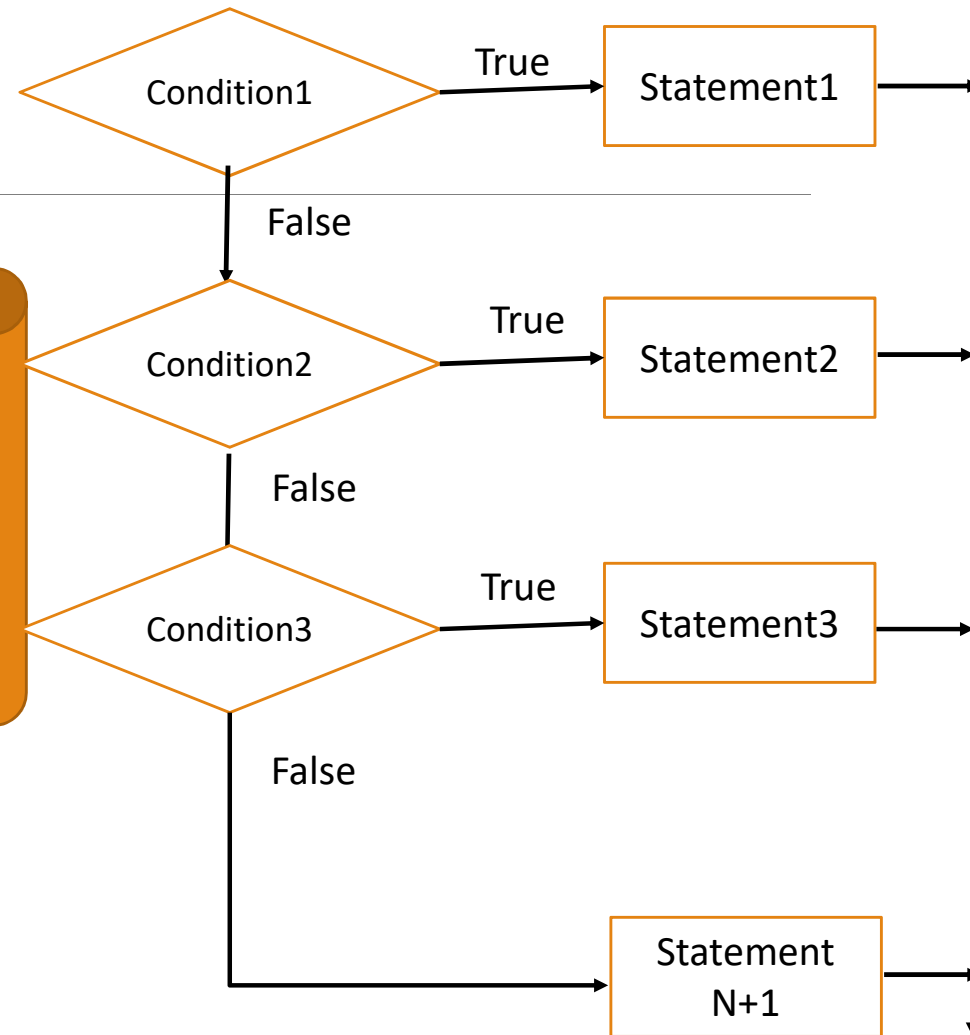
程式語法

```
if <condition1>:  
    statement1  
elif <condition2>:  
    statement2  
elif <condition3>:  
    statement3  
...  
else:  
    statement N+1
```

elif = else if

1. elif 區塊可以沒有、一個或多個
2. else 區塊可以沒有或一個

流程圖結構



多向判斷式練習

多向判斷式練習題

- 請撰寫一個Python程式，它可以要求使用者輸入0-100的分數，然後依據90以上(含)、89-80、79-70、69-60、59以下(含)等級距，將該分數的劃分為A, B, C, D, E

多向判斷式練習題參考程式

□ 請撰寫一個Python程式，它可以要求使用者輸入0-100的資訊課分數，然後依據90以上(含)、89-80、79-70、69-60、59以下(含)等級距，將該分數的劃分為A, B, C, D, E

```
score= eval(input('輸入一個資訊課分數(0-100): '))
if score >=90:
    print('You got A')
elif score >= 80: # elif means score<90
    print('You got B')
elif score >= 70:
    print('You got C')
elif score >= 60:
    print('You got D')
else:
    print('You got E')
```


巢狀判斷式解這題問題

□請撰寫一個Python程式，它可以要求使用者輸入0-100的資訊課分數，然後依據90以上(含)、89-80、79-70、69-60、59以下(含)等級距，將該分數的劃分為A, B, C, D, E

```
score= eval(input("輸入一個資訊課分數(0-100): "))
if score >=90:
    print('You got A')
else:
    if score >= 80:
        print('You got B')
    else:
        if score >= 70:
            print('You got C')
        else:
            if score >= 60:
                print('You got D')
            else:
                print('You got E')
```

兩種方法比較

```
score= eval(input("輸入一個資訊課分數(0-100): "))
if score >=90:
    print('You got A')
elif score >= 80:
    print('You got B')
elif score >= 70:
    print('You got C')
elif score >= 60:
    print('You got D')
else:
    print('You got E')
```

```
score= eval(input("輸入一個資訊課分數(0-100): "))
if score >=90:
    print('You got A')
else:
    if score >= 80:
        print('You got B')
    else:
        if score >= 70:
            print('You got C')
        else:
            if score >= 60:
                print('You got D')
            else:
                print('You got E')
```

練習題

1. 計算BMI的程式

- 請撰寫一個Python程式，它可以要求使用者輸入身高(cm)和體重(kg)，然後計算BMI等於體重(公斤)/身高²(公尺²)，
 - 若低於18.5 (不含)，就印出“過輕”；
 - 若介於18.5~24 (不含)，就印出“正常”；
 - 若超過24 ~ 27 (不含)，就印出“過重”；
 - 若超過27以上，就印出“肥胖”

2. 判斷是否為閏年的程式

- 讓使用者輸入年後印出閏年或平年(閏年判斷方式是: 四年一閏，逢一百年不閏，但四百年有閏)
- if...elif...else
- if (...) or (... and ...) ... else

3. 猜數字的小程式(只須執行一次判斷)

- Q1：程式設計人員先給定一個兩位數字以內的正整數(真值)，接著讓使用者猜這個數字(“請使用者輸入一個兩位數字以內的正整數:”)，然後，程式進行比較大小，若猜測數字>真值，請印出“你猜的數字太大”；若猜測數字<真值，請印出“你猜的數字太小”；若剛好相等，請印出“恭喜你猜對了”
- Q2：請用random module下的randint() function
- Hint: random.randint(a,b) → Return a random integer N such that $a \leq N \leq b$.

4.顯示成績資料

□請用Python設計一個輸入成績的程式，讓使用者輸入學生姓名及成績後並判斷他的成績等第及GPA，印出如“林小明的Python成績為92.65分，他的成績等第是A，GPA=4”(成績分數顯示至小數點後兩位，GPA顯示至整數位)。

百分制分數區間	成績等第	換算GPA
80分以上	A	4.0
70~79分	B	3.0
60~69分	C	2.0
60分以下	F	0