

3-3 變數與常數

- 變數（variable），顧名思義就是它會隨著程式的執行，其值會有所改變。
- 一般我們會從題目中，訂定代表這些事項的變數名稱。
- 取變數名稱盡量要和代表的事項相貼近，如要算兩數的總和與平均分數，取total變數名稱比取t來得好，取平均分數的變數名稱為average，比取a來得佳

Lab 3

■ 請撰寫一個程式，程式執行時，首先要求輸入姓氏，顯示【請輸入您的姓氏：】，接著要求輸入名字，顯示【請輸入您的名字：】；呼叫 print 函數，分別產生以下的輸出

- a. 在同一行輸出您的姓氏和名字。（例如：張無忌）
- b. 在同一行輸出您的姓氏和名字，中間以逗號分隔。（例如：張,無忌）
- c. 將姓氏和名字分別輸出於兩行。 例如：

張

無忌

*請使用您自己的名字

3-5 我的第一個程式

- 請輸入三個數值，然後計算其總和與平均數，並加以輸出。

範例程式：p3-16.py

- `num1, num2, num3 = eval(input('Enter three numbers separated by commas: '))`
- `total = num1 + num2 + num3`
- `average = total / 3`
- `print('total = %d, and average = %f' %(total, average))`

輸出結果

- Enter three numbers separated by commas: 100, 150, 170
- total = 420, and average = 140.000000

-
- 可以使用格式指定器以指定印出小數點後兩位數的浮點數。更改一下print 函式即可，如下所示：

- `print('total = %d, and average = %.2f' %(total, average))`

- 輸出結果

- Enter three numbers separated by commas: 100, 150, 170
 - total = 420, and average = 140.00

範例程式：p3-17-1.py

- 題目：輸入一圓的半徑，然後計算其圓的面積與周長。

```
radius = eval(input('Enter the radius: '))  
area = radius * radius * 3.14159  
perimeter = 2 * 3.14159 * radius  
print('Area = %.2f' %(area))  
print('Perimeter = %.2f'%(perimeter))
```

輸出結果

■ Enter the radius: 5

■ Area = 78.54

■ Perimeter = 31.42

範例程式：p3-17-2.py

- 若將3.14159 以math 模組下的pi 取代會更好，如以下程式所示：

```
import math
```

```
radius = eval(input('Enter the radius: '))
```

```
area = radius * radius * math.pi
```

```
perimeter = 2 * math.pi * radius
```

```
print('Area = %.2f'%(area))
```

-
- 輸出結果相同。如果沒有設定格式指定器`%.2f`的話，可能會不一樣，因為`math.pi` 會比較精確。
 - 有關在`math` 模組下還有許多好用的函式。

3-6 註解敘述

- 程式的好壞在於是否每一個人都看得懂你寫的程式。
- 可利用註解敘述（comment statements）來協助。
Python 的註解敘述有二種：一為以# 開頭，在同一行後面的文字即為註解敘述，如以下敘述：

```
# This program display learning Python now!, and Python is fun.
```

-
- 另一方式是以 `"""` 開頭，並以 `"""` 結尾，之間的文字即為註解敘述。

```
"""
```

```
This program display learning Python now!,  
and Python is fun.
```

```
"""
```

-
- # 開頭的註解敘述，一般用於功能只限於此行。而 " 則用於多行。
 - 註解敘述很重要，其實不一定要每個地方都加上註解，在重要的區段加上註解就可以了。
 - 至少在程式開頭說明此程式的用意。

Lab 3-2

- 試撰寫一程式，要求使用者輸入一矩形的寬和高，並加以計算此矩形之面積和周長。最後請印出此矩形的寬、高、面積和周長。

範例輸入樣本：

Enter height: 23.5

Enter width: 19

範例輸出樣本：

Height = 23.50

Width = 19.00

Perimeter = 85.0

Area = 446.50

exercise3-3.py

5 選擇

- 5-1 單向的選擇敘述：if敘述
- 5-2 關係運算子
- 5-3 雙向的選擇敘述：if...else 敘述
- 5-4 多向的選擇敘述：if...elif...else 敘述
- 5-5 邏輯運算子
- 5-6 運算子優先順序與結合性

選擇敘述

- 選擇敘述表示在某一條件下，做某些事情。
- 由於Python 是一種程序性的語言（procedure language），表示上一行敘述執行後接下一行敘述，亦即一行接一行地執行。
- 但有時有些敘述不必執行想跳過去的話，此時就要靠選擇敘述來完成。利用當某一條件為真時，處理其對應要處理的敘述。

範例

- 1. 若中大樂透，則去買一台瑪莎拉蒂休旅車。
- 2. 若今天下雨，則帶傘，否則不帶傘。
- 3. 若您考了雅思（IELTS），成績是8，則可申請劍橋大學，否則申請其他學校。
- 4. 開車到某一十字路口，若紅燈，則停；若綠燈，則行。
- 5. 今天晚上要去戲院看電影或去錢櫃唱歌。
-

5-1 單向的選擇敘述：if 敘述

■ 單向的選擇敘述有如道路的單行道，如圖5-1所示。



● 圖5-1 單向選擇敘述示意圖

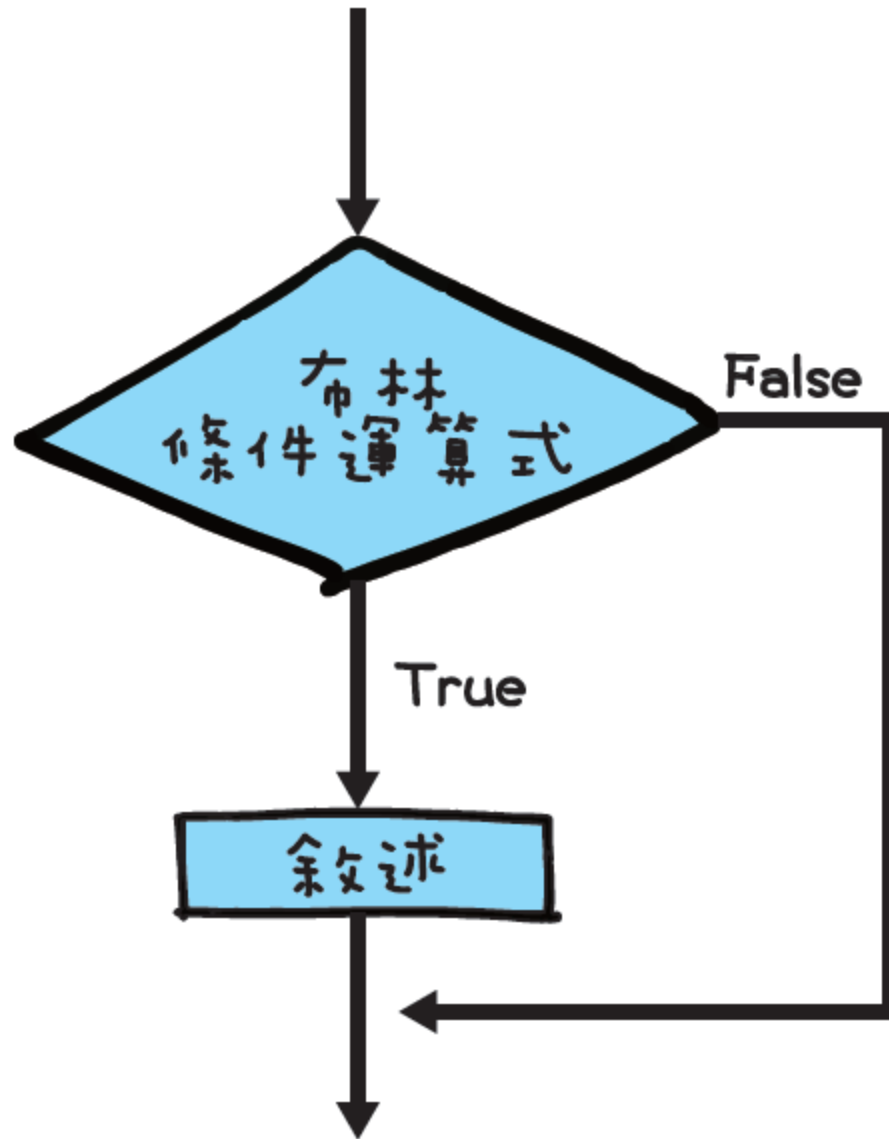
單向if敘述的語法

- if boolean-expression:

- statement(s)

- 上述的boolean-expression一般稱為布林條件運算式，此運算式會得到真（True）或假（False）兩種值。

-
- statement(s)為Python合法的敘述。
 - 必須以if關鍵字為基準點向右縮排至少一個空格，接著每一敘述都要相同的空格。
 - 注意，boolean-expression後面有冒號(:)，將要執行的敘述加以隔開。



● 圖5-2 單向選擇敘述流程圖

-
- 程序運算被標示在各方塊內，而箭頭連結代表控制流程順序。
 - 菱形方塊用來標示布林條件運算式，矩形方塊則用來代表敘述。

5-2 關係運算子

 表5-1 關係運算子

運算子	名稱	範例假設(score是85)	結果
<	小於	score < 80	False
<=	小於或等於	score <= 80	False
>	大於	score > 80	True
>=	大於或等於	score >= 80	True
==	等於	score == 80	False
!=	不等於	score != 80	True

範例程式：p5-5.py

- `X = 65`
- `if X > 60:`
- `X = X + 5`
- `print('X = %d'%(X))`

輸出結果

■ $X = 70$

■ 其中選擇敘述表示若x大於60時，則將x加5。

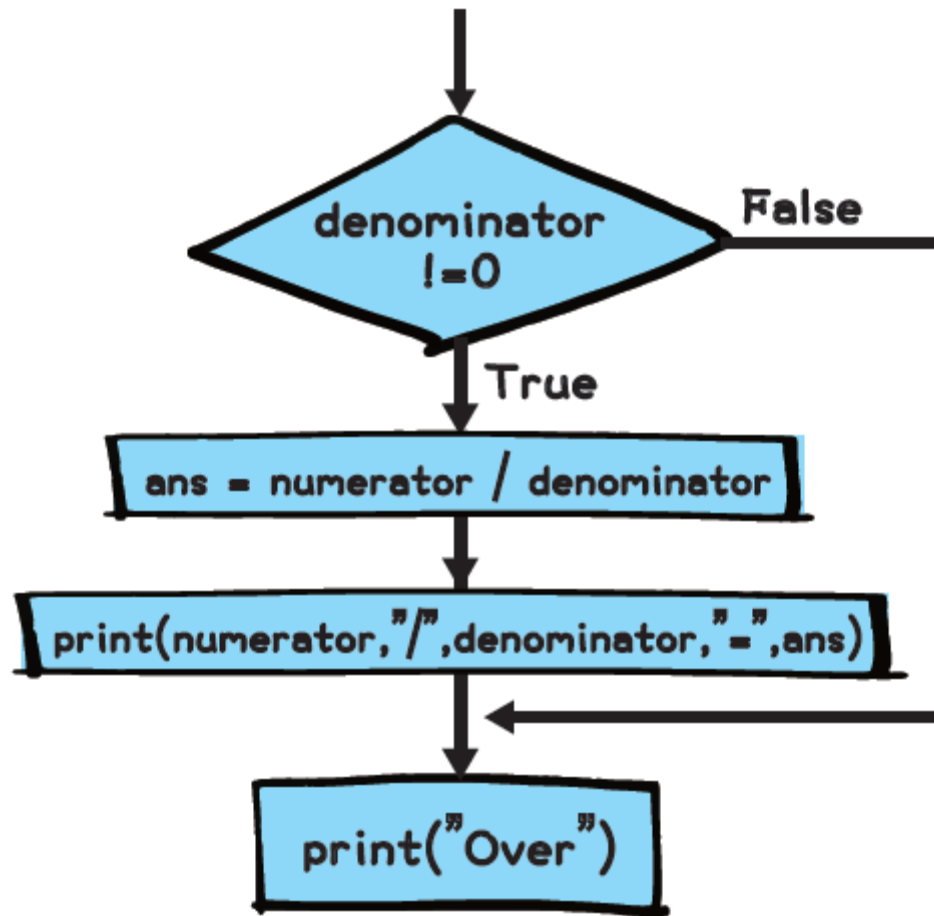
■ 需注意的是，關係運算子是用兩個等號（`==`）來表示，而不是單一等號（`=`）。

■ 單一等號用在指定的意思。

■ 含有關係運算子的運算式，運算後的結果為一個布林值（**Boolean value**）True或False。

範例程式：p5-8.py

- `numerator = eval(input("Please input分子: "))`
- `denominator = eval(input("Please input分母: "))`
- `if denominator != 0 :`
- `ans = numerator / denominator #1`
- `print(numerator, "/", denominator, "=", ans) #2`
- `print("Over") #3`



● 圖5-4 上一範例程式流程圖

輸出結果 1

Please input numerator: 125

Please input denominatr: 25

$125 / 25 = 5.0$

Over

輸出結果 2

Please input numerator: 125

Please input denominatr: 0

Over

-
- 請debug 以下的程式。程式的大意是，當輸入的數值是5 的倍數時，則顯示它是5 的倍數。

範例程式：p5-9.py

```
■ number = eval(input("Please input a number: "))  
■ if number % 5 == 0:  
    print(number, "is 5's multiple")  
■ print("Over")
```

一個等號（=）是表示指定的意思

範例程式：p5-10.py

```
■ number = eval(input("Please input a number: "))  
■ if number % 5 == 0:  
■     print(number, "is 5's multiple")  
■ print("Over")
```

將if 敘述後面的 = 改為 ==

解說

- 一個等號（=）是表示指定的意思，而兩個等號（==）表示判斷兩個值是否相等，千萬不可以搞混，Python 在條件運算式中是不允許這樣寫的。

5-3 雙向的選擇敘述：if...else 敘述



● 圖5-5 雙向選擇敘述示意圖

-
- 將兩數相除時，若分母不為0，則計算其商，否則顯示分母不可為0的訊息，則如下敘述所示：

範例程式：p5-11.py

```
■ numerator = eval(input("Please input 分子: "))
■ denominator = eval(input("Please input 分母: "))
■ if denominator != 0:
■     ans = numerator / denominator
■     print(numerator, "/", denominator, "=", ans)
■ else:
■     print("Denominator is invalid")
■ Print("Over")
```

輸出結果

Please input numerator: 125

Please input denominatr: 25

$125 / 25 = 5.0$

Over

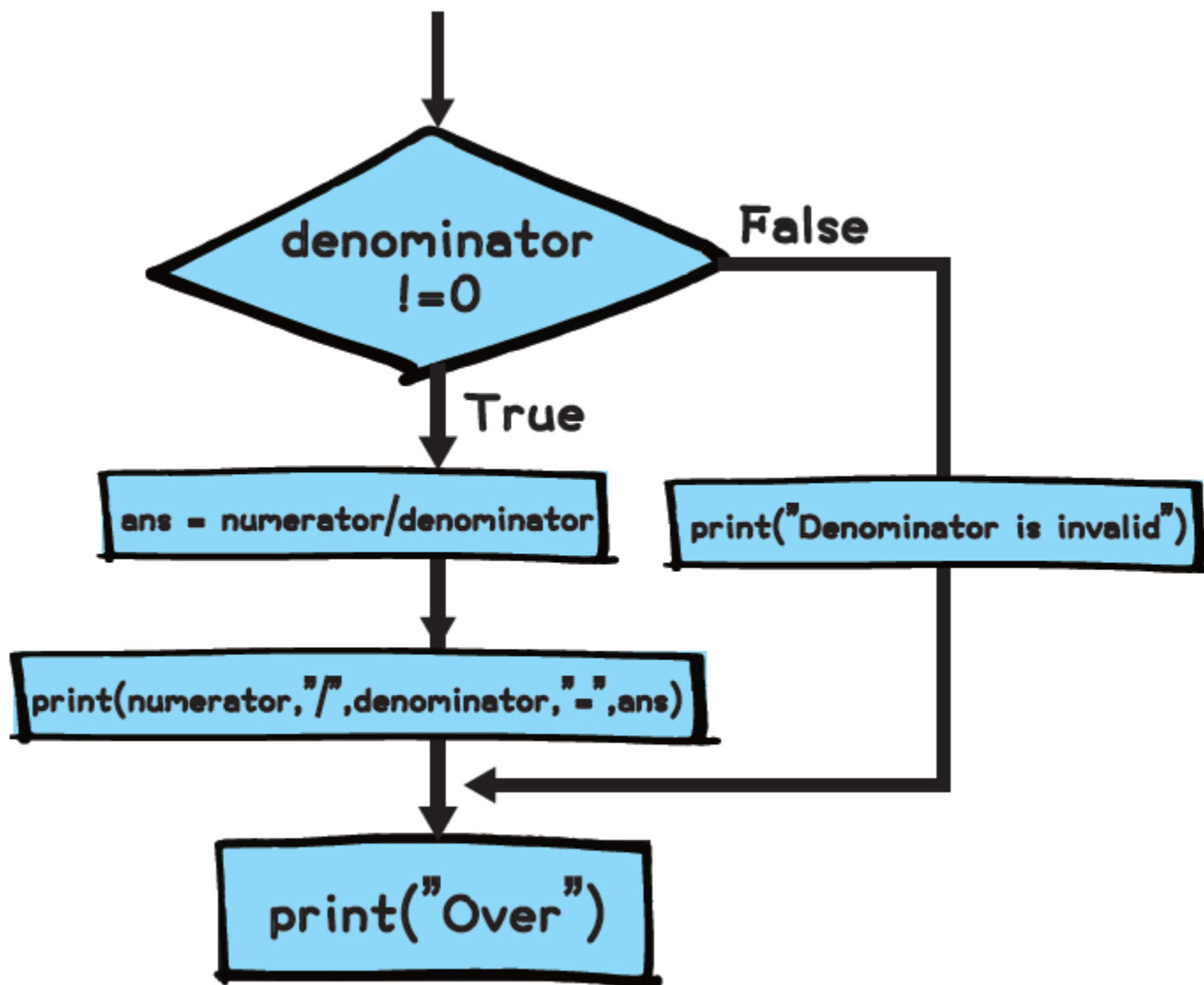
輸出結果

Please input numerator: 125

Please input denominatr: 0

Denominator is invalid

Over



● 圖5-6 雙向選擇敘述流程圖

-
- 請debug 以下的程式。程式的大意是，判斷輸入的數值是偶數或是奇數。

範例程式：p5-13-1.py

debug

```
■ num = eval(input("Please input a number: "))  
■ if num % 2 == 0:  
■     print(num, "is an even number.")  
■ else  
■     print(num, "is a odd number.")  
■ print("Over")
```

else 後面要加冒號，並且也要內縮喔！

範例程式： p5-13-2.py

```
■ num = eval(input("Please input a number: "))  
■ if num % 2 == 0:  
■     print(num, "is an even number.")  
■ else:  
■     print(num, "is a odd number.")  
■ print("Over")
```


輸出結果

Please input a number: 88

88 is an even number

Over

輸出結果

Please input a number: 89

89 is a odd number

Over

5-4 多向的選擇敘述：if…elif…else 敘述



● 圖5-7 多向選擇敘述示意圖

-
- 若有多個方案要選擇的話，則可使用多個if...else或if...elif...else敘述。
 - 例如輸入一數字，判斷它大於0、或小於0、或等於0。

範例程式：p5-15.py

```
■ number = eval(input("Please input a number: "))  
■ if number > 0:  
■     print(number, "is greater than 0")  
■ else:  
■     if number < 0:  
■         print(number, "is less than 0")  
■     else:  
■         print(number, "is equal to 0")  
■ print("Over")
```

輸出結果

Please input a number: 20

20 is greater than 0

Over

輸出結果

Please input a number: -10

-10 is less than 0

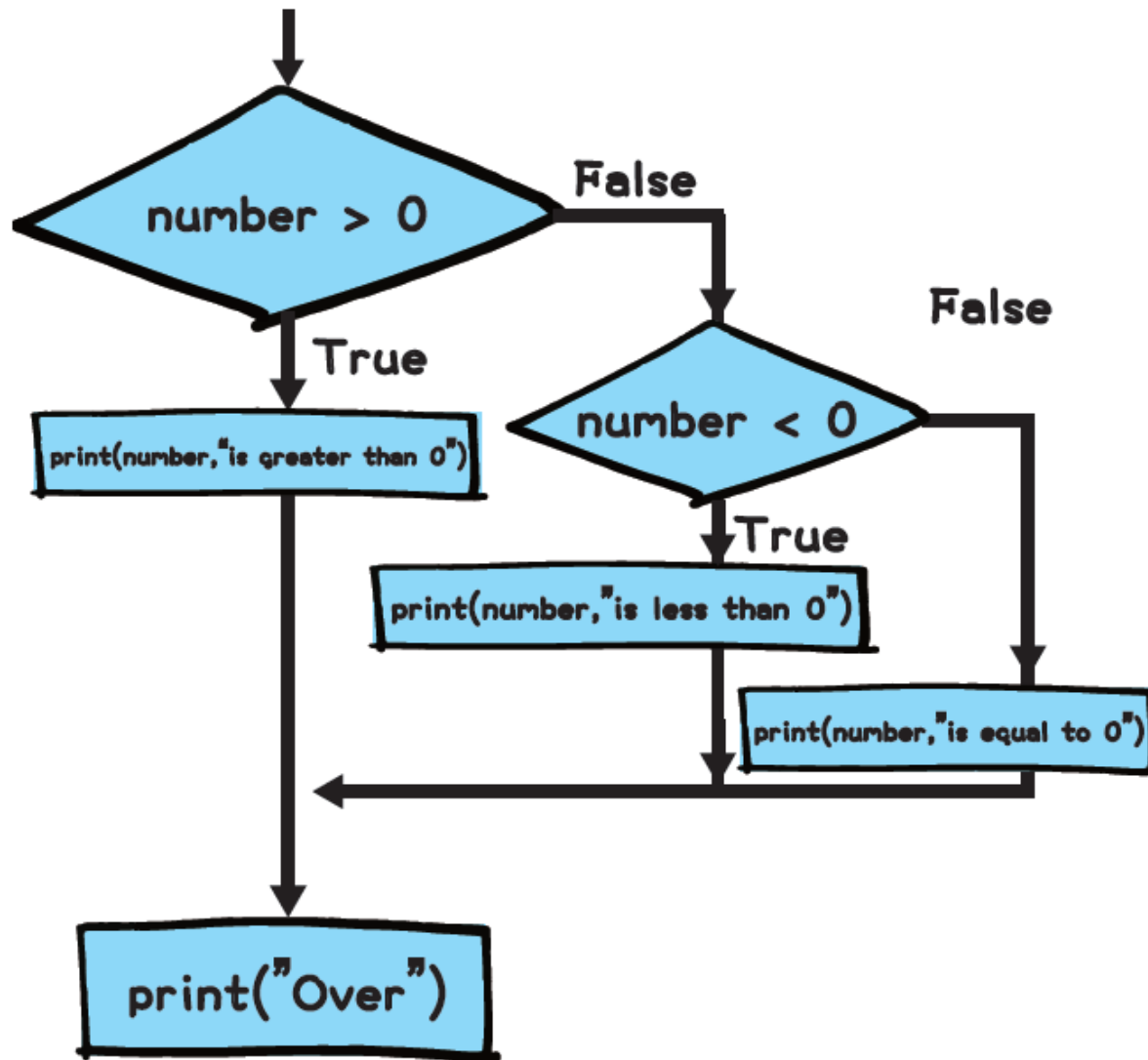
Over

輸出結果

Please input a number: 0

0 is equal to 0

Over



● 圖5-8 多向選擇敘述流程圖

範例程式：p5-17.py

```
■ number = eval(input("Please input a number: "))  
■ if number > 0:  
■     print(number, "is greater than 0")  
■ elif number < 0:  
■     print(number, "is less than 0")  
■ else:  
■     print(number, "is equal to 0")  
■ print("Over")
```

5-4-1 計算GPA

■美國大學四點制的算法，如表5-2所示：

表5-2 GPA對應表

A	80 ~ 100	4
B	70 ~ 79	3
C	60 ~ 69	2
D	50 ~ 59	1
E	49 以下	0

■ $GPA = (\text{學分} \times \text{點數}) \text{ 的總和} / \text{總學分}$ 。

範例程式：p5-18.py

```
■ score = eval(input("Please input your score: "))  
■ if score >= 80:  
■     print("Grade A")  
■ elif score >= 70:  
■     print("Grade B")  
■ elif score >= 60:  
■     print("Grade C")  
■ elif score >= 50:  
■     print("Grade D")  
■ else:  
■     print("Grade E")
```

輸出結果 1

Please input your score: 86

Grade A

輸出結果 2

Please input your score: 62

Grade C

5-4-2 計算BMI

- w = 體重，單位：公斤；
- h = 身高，單位：公尺；
- $BMI = w / h^2$ ，單位：公斤/平方公尺

 表5-3 BMI對應表

BMI	說明
低於18.5	體重不足
18.5-24.9	正常
25.0-29.9	過重
超過30.0	肥胖

計算BMI p5-21.py

- `weight = eval(input("Please input your weight(kilogram): "))`
- `height = eval(input("Please input your height(centimeter): "))`
- `heightInCentimeter = (height/100)`
- `bmi = weight / (heightInCentimeter * heightInCentimeter)`
- `print("Your BMI is", format(bmi, ".2f"))`

-
- `if bmi < 18.5:`
 - `print("Underweight")`
 - `elif bmi < 25:`
 - `print("Normal")`
 - `elif bmi < 30:`
 - `print("Overweight")`
 - `else:`
 - `print("Obese")` # obese肥胖的

輸出結果

Please input your weight(kilogram): 67

Please input your height(centimeter): 185

Your BMI is 19.58

Normal

5-5 邏輯運算子

 表5-4 邏輯運算子

運算子	說明
not	邏輯運算子（反）
and	邏輯運算子（且）
or	邏輯運算子（或）

 表5-5 not運算子的真值表

c	not c	範例（假設 score=80, gender=F）
True	False	not(score > 78) 是 False，因為 (score > 78) 是 True
False	True	not(gender == 'M') 是 True，因為 (gender == 'M') 是 False

 表5-6 and運算子的真值表

c1	c2	c1 and c2	範例 (假設 score = 85, behavior = 80)
False	False	False	
False	True	False	(score > 90) and (behavior >= 70) 是 False，因為 (score > 90) 是 False
True	False	False	
True	True	True	(score >= 85) and (behavior >= 80) 是 True，因為 (score >= 85) 與 (behavior >= 80) 皆為 True

 表5-7 or 運算子的真值表

c1	c2	c1 or c2	範例 (假設 score = 85, behavior = 80)
False	False	False	(score >= 90) or (behavior > 80) 是False，因為 (score >= 90) 與 (behavior > 80) 皆為 False
False	True	True	(score > 90) or (behavior >= 80) 是 True，因為 (behavior >= 80) 是 True
True	False	True	
True	True	True	

閏年

■ 判斷閏年的條件有二，分別為

➤ (1) 此年份能被400 整除

➤ (2) 可被4 整除而且不能被100 整除。

■ 這兩個條件只要一個條件成立即可。

p5-25.py

- `year = eval(input("Please input what year? "))`
- `if year % 400 == 0 or (year % 4 == 0 and year % 100 != 0):`
- `print(year, "is leap year")`
- `else:`
- `print(year, "is not leap year")`

練習：求兩個數的最大值


- 請撰寫一個程式，求兩個數的最大值
- 程式執行時，首先要求輸入第一個數值numA，顯示【請輸入數值numA：】，接著要求輸入第二個數值numB，顯示【請輸入數值numB：】
- 輸出%是比較大的數

cmp-num-1.py
cmp-num-2.py

5-6 運算子優先順序與結合性

- 運算子由上往下，優先順序往下遞減。
- 邏輯運算子的優先順序低於關係運算子，而關係運算子的優先順序又低於算術運算子。

 表5-8 運算子優先順序

運算優先順序	運算子
	+, - (一元運算子的正、負號)
	** (次方)
	not
	*, /, //, % (乘、除、整數相除、餘數)
	+, - (二元運算子的加與減)
	<, <=, >, >= (比較運算子)
	==, != (相等運算子)
	and
	or
	=, +=, -=, *=, /=, //=, %= (指定運算子)

-
- 如果擁有相同優先順序的運算子一起使用，它們的結合性（associativity）將決定解析的順序。
 - 所有除了指定運算子之外的二元運算子，皆為**向左結合**（left associative）

$$\mathbf{a * b / c * d}$$

等同於

$$\mathbf{((a * b) / c) * d}$$

若有多個括號，則由內括號先執行。

5-7 範例集錦

■ 5-7-1 絕對值

➤ 請撰寫一程式，要求使用者輸入一數字並印出其絕對值。

p5-27.py

範例程式： p5-27.py 絕對值

```
■ number = eval(input("Please enter a number: "))  
■ if number >= 0:  
■     print(number)  
■ else:  
■     print(-number)
```

輸出結果

Please enter a number: -100

100

輸出結果

Please enter a number: 100

100

5-7-2 判斷及格成績

- 請撰寫一程式，要求使用者輸入一個分數並顯示you pass或you fail。
- (分數 ≥ 60 為you pass)

範例程式：p5-28.py

```
■ score = eval(input('Enter your score: '))  
■ if score >= 60:  
■     print('you pass')  
■ else:  
■     print ('you fail')
```

輸出結果 1

Enter your score: 90
you pass

輸出結果 2

Enter your score: 50
you fail

5-7-3 算術四則運算

- 請撰寫一程式，提示使用者輸入兩個數字以及一個符號（'+'、'-'、'*'或'/'）。接著對這兩個數字作所指定符號的算術運算並顯示結果。

範例程式：p5-29.py

- `number1= eval(input('Enter number1: '))`
- `number2= eval(input('Enter number2: '))`
- `operator = input('Enter an arithmetic operator (+, -, *, /): ')`
- `result = 0`
- `if operator == '+':`
 - `result = number1 + number2`
- `elif operator == '-':`
 - `result = number1 - number2`

-
- `elif operator == '*':`
 - `result = number1 * number2`
 - `elif operator == '/':`
 - `result = number1 / number2`
 - `print('% .4f %s % .4f = % .4f' % (number1, operator, number2, result))`

輸出結果

Enter number1): 88

Enter number2): 66

Enter an arithmetic operator (+, -, *, /): -

88.0000 - 66.0000 = 22.0000

輸出結果

Enter number1): 100

Enter number2): 200

Enter an arithmetic operator (+, -, *, /): +

100.0000 + 200.0000 = 300.0000

5-7-4 年齡劃分

- 請撰寫一程式，提示使用者輸入年齡。根據以下標準顯示對應的年齡劃分：

表5-9 年齡劃分

年齡	年齡劃分
0 ~ 6	童年
7 ~ 17	少年
18 ~ 40	青年
41 ~ 65	中年
≥ 66	老年

範例程式：p5-30.py

■ `age = eval(input('請輸入年齡: '))`

■ `state = "`

■ `if age < 6:`

■ `state = '童年'`

■ `elif age >= 7 and age <= 17:`

■ `state = '少年'`

■ `elif age >= 18 and age <= 40:`

■ `state = '青年'`


- `elif age >= 41 and age <= 65:`

- `state = '中年'`

- `elif age >= 66:`

- `state = '老年'`

- `print('您的年齡屬於%s' % state)`



輸出結果



請輸入年齡：18
您的年齡屬於青年



輸出結果



請輸入年齡：45
您的年齡屬於中年

Lab 5-1

1. 試撰寫一程式，提示使用者輸入一分數的分子與分母，若分母為0，則印出分母不可為0的錯誤訊息，否則印出此分數。
2. 假設您到一家行銷公司上班，薪水包括底薪和佣金。底薪是6,000，而其佣金率如下表：

試撰寫一程式，提示使用輸入此月的銷售金額，然後算出其薪水。

銷售金額	佣金率
0.01 ~ 5,000	10%
5,001 ~ 10,000	12%
10,001 ~ 15,000	14%
15,001以上	16%

practice5-5.py
practice5-6.py