
Python 基本運算

臺北科技大學資訊工程系

資訊系統開發流程

- 需求/問題的取得/理解 – 需求規格書
 - 文字描述、圖形介面呈現
- 系統分析 – 系統規格書
 - 測試案例設計、測試劇本/使用情境設計
- 系統設計
 - 資料結構設計
 - 程式流程設計/問題處理邏輯設計
 - 三種結構 - 循序、條件選擇、反覆迴圈
- 程式碼撰寫
 - 程式語法正確性 (筆試/上機測驗)
 - 程式邏輯正確性 (筆試/上機測驗)
- 系統驗證
 - 靜態分析 – 程式碼檢視 – 人工追蹤程式運作 (筆試)
 - 動態測試 – 測試案例、測試程式執行 (上機測驗)

Python 簡介

□ Python

- 容易撰寫，適合初學者
- 套件功能多，可以做資料分析、機器學習、網頁爬蟲

□ Python 基本語法

○ 控制指令

- 結尾要使用冒號:
- 冒號所屬指令區塊，需縮排四個空白鍵
- 區塊區隔不須大括號

○ 其他指令

- 結尾，空白不須分號

標記(token)

□ Python 直譯器

- Python interpreter
- 利用 token 解譯程式，

□ token 包含：

- 關鍵字 (keyword)
- 定數 (literals)
- 運算子 (operator)
- 識別字 (identifier)

關鍵字(keyword)

□ 關鍵字

○ 具有語法功能的保留字 (reserved word)

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

定數(literals)

❑ 定數：固定數值

- 整數定數 (integer literal) 2025
- 浮點定數 (floating-point literal) 1.023
- 字串定數 (string literal) 'NTUT'
- 複數定數 (imaginary literal) 3 + 2i
- 字節定數 (bytes literal) b'Caf\xc3\xa9'

`\x hh`

十六進位標記法的 ASCII 字元

```
# Byte String Example  
byte_string = b'Caf\xc3\xa9'
```

```
print("Original Byte String:", byte_string)  
print("Type of Byte String:", type(byte_string))
```



執行結果？

```
Original Byte String: b'Caf\xc3\xa9'  
Type of Byte String: <class 'bytes'>
```

<https://homepage.cs.uri.edu/faculty/wolfe/book/Readings/R02%20Ascii/completeASCII.htm>

<https://www.geeksforgeeks.org/effect-of-b-character-in-front-of-a-string-literal-in-python/>

運算子

□ 運算子優先順序

運算子	描述
**	指數
~X	位元反轉
+X, -X	正負號
*, /, //, %	乘法、除法、 整除 、取餘數
+, -	加法與減法
<<, >>	移位
&	位元 and
^	位元 xor
	位元 or
<, <=, >, >=, !=, ==	比較
is, is not	同一性測試
in, not in	成員測試
not x	邏輯 not
and	邏輯 and
or	邏輯 or
lambda	無名稱函式

識別字(identifier)

❑ 識別字

- 包括：變數 (variable)、函數 (function)、類別 (class) 等
- 自行命名：最好使用有意義複合字命名，例如 computeCost...

❑ 特殊識別字

- 開頭和結尾，都加兩個底線, ex: __init__

Python，你到底是在__底線__什麼啦！

<https://aji.tw/2017/06/python%E4%BD%A0%E5%88%B0%E5%BA%95%E6%98%AF%E5%9C%A8%E5%BA%95%E7%B7%9A%E4%BB%80%E9%BA%BC%E5%95%A6/>

識別字(identifier)

□ 命名規則

- 不可使用 關鍵字

- 長度沒有限制

- Unicode 編碼：

- 第一字元 [_, A~Z, a~z, 任何語言字元, 包括中文]

- 後面字元 [_, A~Z, a~z, 0~9, 任何語言字元, 包括中文]

- 大小寫不同

- ASCII 編碼

- 第一字元 [_, A~Z, a~z]

- 後面字元 [_, A~Z, a~z, 0~9]

- 大小寫不同

資料型別(Data Type / Type)

□ 基本資料型別

- 整數(int): 1, 10, 20
- 浮點數(float): 有小數點的數字, 3.14159
- 字串(String): 一串字元或文字, 'Hello', '小明'
- 複數 complex: $2 + 3i$
- 字節(byte): 二進位資料, `b'Caf\xc3\xa9'`
- 布林(Boolean): `True` (真), `False` (假)

資料型別(Data Type / Type)

□ 基本資料型別

- 序列(Tuple): 又稱固定列表，有順序、不可變動的資料集，
(**'Tom'**, 170, 64)
- 集合(Set): 無順序的資料集，
{**'Tom'**, 170, 64}
- 字典(dictionary): 以 Key-Value 組成的資料集
{**'name': 'Tom'**, **'height': 170**, **'weight': 64**}

整數與浮點數運算

□ 運算子 (operator)，處理數學運算

- +
- -
- *
- /
- ** 次方
- // 整數除法
- % 求餘數
- `print()` 輸出螢幕

整數與浮點數運算

□ 運算子 (operator)，處理數學運算

-
-

01	print(5 + 9)		14
02	print(10 - 7)		3
03	print(4 - 10)		-6
04	print(5 + 3 - 17)		-9
05	print(6 * 7)		42
06	print(6 ** 2)		36
07	print(9 / 5)	# 除法，有小數	1.8
08	print(9 // 5)	# 除法，去掉小數	1
09	print(9 % 5)		4
10	print(9 / 0)		ZeroDivisionError:

print

- ❑ %d 10 進位 **整數** 輸出
- ❑ %f **浮點數** 10 進位 輸出
- ❑ %e, %E 浮點數 10 進位 **科學記號** 輸出
- ❑ %o 8 進位 **整數** 輸出
- ❑ %x, %X 16 進位 **整數** 輸出
- ❑ %c 編碼 **字元** 輸出
- ❑ %% 字串中顯示 **%**

- ❑ %s 使用 str() 輸出字串
- ❑ %r 使用 **repr()** 輸出字串，表達 **資料型別**，例如字串"
 - Python has two built-in functions for **turning an object into a string**:
 - **str()** vs. **repr()**.

1 99.32 Justin	給人看
"1 99.32 Justin"	除錯用

print

```
01 text = '%d %.2f %s' %(1, 99.321, 'Justin')
02 print(text) # print(str(text))
03 print('%r' %repr(text) )
04
05 print('%d %.2f %s' %(1, 99.3, 'Justin'))
06 print('=>%.2f' %(19.234) )
07 print('=>%6.5f' %19.234111 )
08 print('=>%.6e' %19.234111 )
09 print('=>%c %%' %65) # A 編碼65
```

```
1 99.32 Justin
"1 99.32 Justin"
```

```
1 99.30 Justin
=>19.23
=>19.23411
=>1.923411e+01
=>A %
```

字串

□ 字串：表示文字資料

- 以 單引號 ' 或 雙引號 " 包起來。
- 'hello' is the same as "hello"
- \n 表示換行
- \' 表示單引號

```
01 print('Hello')  
02 print('Hello' + ' World')  
03 print('Hello\n' + 'World')  
04 print('Hello\' ' + 'World')
```



執行結果？

```
Hello  
Hello World  
Hello  
World  
Hello' World
```


字 串

❑ Multiline Strings

```
a = """Lorem ipsum  
dolor sit amet,  
consectetur adipiscing  
elit,  
sed do eiusmod tempor  
incidunt  
ut labore et dolore  
magna aliqua."""
```

```
print(a)
```

❑ Multiline Strings

```
a = '''Lorem ipsum  
dolor sit amet,  
consectetur adipiscing  
elit,  
sed do eiusmod tempor  
incidunt  
ut labore et dolore  
magna aliqua.'''
```

```
print(a)
```

字串函式 - format

□ '字串'.format(參數)

- 字串內以 {} 代換 format 的參數 -- 字串資料
- 代換以 0, 1, 2... 位置識別，或者以符號識別

01	str='{0} is {1}~'.format('Justin', 20)	#第0個對應後面第0個參數
02	print(str)	# Justin is 20~
03	str='{name} is {age}~'.format(name = 'Justin', age = 20)	# 以符號對應
04	print(str)	# Justin is 20~
05	str='{name} is {age}~'.format(age = 20, name = 'Justin')	#符號順序無關
06	print(str)	# Justin is 20~
07	str='{0} is {age}~'.format('Justin', age = 20)	
08	print(str)	# Justin is 20~

運算表示式(Expressions)

❑ 變數(variable)

- 儲存資料(整數、浮點數、字串...)

❑ 運算表示式

- 定數與定數運算

- 變數與定數運算

- 變數與變數運算

01	$3 + 5$
02	$3 + (5 * 4)$
03	$3 ** 2$
04	<code>'Hello' + 'World'</code>
05	$num + 3$
06	$num + x$

變數(Variable)賦值

□ 賦值運算 =

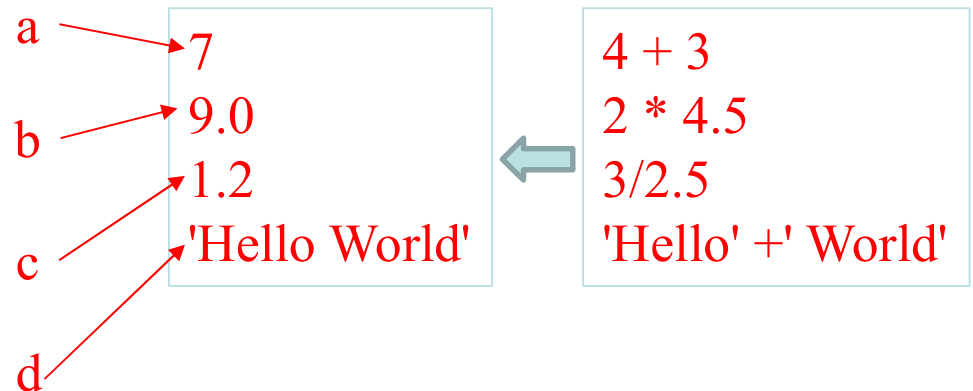
- 將右邊運算式的運算結果，用 = 賦予(存到/指派)左邊的變數

a = 4 + 3
a ← 4 + 3

跟數學"相等"意義完全不同

- 左邊變數的值，之後可以使用。

```
01 a = 4 + 3
02 b = 2 * 4.5
03 c = (1+2) / 2.5
04 d = 'Hello' + ' World'
05 print(a, b, c, d)
```



變數(Variable)與Expressions

□ 變數資料型別

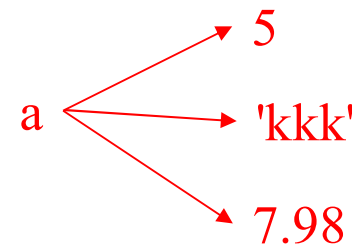
- 透過變數的值，可看出變數的資料型別

01	s_id = 70	# s_id 的資料型別是整數
02	score = 86.78	# score 的資料型別是浮點數
03	name = 'Tom'	# name 的資料型別是字串

- 變數可以變更資料型別

- 變數 **a** 代表一個參照 (Reference)

01	a = 5	# a 的資料型別是整數
02	a = 'kkk'	# a 的資料型別變成字串
03	a = 7.98	# a 的資料型別變成浮點數



01	a = 4 + 3
02	b = a * 4.5
03	c = (a + b) / 2.5
04	a = "Hello World"
05	print(a , b, c)



執行結果?

Hello World
31.5
15.4

變數複合運算

□ 變數複合運算

○ +=, -=, *=, /=

```
a = 10  
a += b
```

```
a = 10  
a += b
```

○ 多變數一次指派值

```
x, y = a, b
```

```
x = a  
y = b
```

○ 變數內容交換

```
x, y = y, x
```

```
tmp = x  
x = y  
y = tmp
```

串列 list 存取

□ 串列設定

- 串列變數 = [串列內容1, 串列內容2, ...]
- 串列內容可以是不同資料型別的值、或變數

□ 串列存取

- 串列變數[索引值]
- 索引值從0開始編號

□ 串列列印

- print(串列變數[索引值])

串列 list 存取

01	student = ['name', 901551, 96] # 'name' 是字串定數	
02	print(student)	['name', 901551, 96]
03	print('name=', student[2], 'score=', student[2])	name = 96 score = 96
04	student[2] = 100	
05	print(student)	['name', 901551, 100]
06	name = 'John' # name 是字串變數	
07	score = 80	
08	student = [name, score]	
09	print(student)	['John', 80]
10	print(len(student))	2

串列 list 排序

□ **sorted**依數值或字母由小到大排序

```
01 names = ['Tom', 'John', 'Mary']
02 scores = [85, 90, 70]
03
04
05
06
07
```

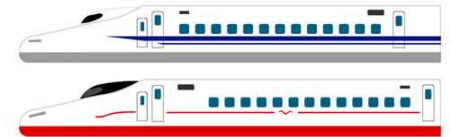
names

[0]	[1]	[2]
'Tom'	'John'	'Mary'

scores

[0]	[1]	[2]
85	90	70

排序前



串列 list 排序

□ **sorted** 依數值或字母由小到大排序

```
01 names = ['Tom', 'John', 'Mary']
02 scores = [85, 90, 70]
03 names = sorted(names)
04 scores = sorted(scores)
05
06
07
```

names	[0]	[1]	[2]
	'Tom'	'John'	'Mary'

scores	[0]	[1]	[2]
	85	90	70

排序前
排序後

names	[0]	[1]	[2]
	'John'	'Mary'	'Tom'

scores	[0]	[1]	[2]
	70	85	90

names



scores



串列 list 排序

□ **sorted**依數值或字母由小到大排序

```
01 names = ['Tom', 'John', 'Mary']
02 scores = [85, 90, 70]
03 names = sorted(names)
04 scores = sorted(scores)
05 students = [names, scores]
06
07
```

names	[0]	[1]	[2]
	'Tom'	'John'	'Mary'
scores	[0]	[1]	[2]
	85	90	70

排序前
排序後

names	[0]	[1]	[2]
	'John'	'Mary'	'Tom'
scores	[0]	[1]	[2]
	70	85	90

students[0] names



students[1] scores



串列 list 排序

□ **sorted**依數值或字母由小到大排序

```
01 names = ['Tom', 'John', 'Mary']
02 scores = [85, 90, 70]
03 names = sorted(names)
04 scores = sorted(scores)
05 students = [names, scores]
06 print(students[0][0], students[0][1])
07 print(students[1][0], students[1][1])
```

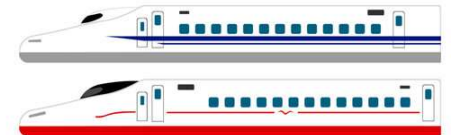
John Mary
70 85

students 索引值

[0]	[0] [0]	[0] [1]	[0] [2]	'John'	'Mary'	'Tom'
[1]	[1] [0]	[1] [1]	[1] [2]	70	85	90

students[0] names

students[1] scores



names	[0]	[1]	[2]
	'Tom'	'John'	'Mary'

scores	[0]	[1]	[2]
	85	90	70

排序前
排序後

names	[0]	[1]	[2]
	'John'	'Mary'	'Tom'

scores	[0]	[1]	[2]
	70	85	90

串列 list 排序



- 串列元素是一個串列，根據元素的第一個元素排序

```
01 nums = [[202, 101], [321, 123], [9, 8]]  
02 nums = sorted(nums)  
03 print(nums)
```

```
[[9, 8], [202, 101], [321, 123]]
```



使用者輸入

□ input('提示字')

- 從鍵盤輸入，資料型別是「字串」
- 等待輸入時，顯示「提示字」

□ int()，將整數字串，轉換成整數資料型別

```
01 numX = '123'  
02 numY = '456'  
03 num = int(numX) + int(numY)  
04 print(num)  
05 name = input('name: ')  
06 numX = input('First Number: ')  
07 numY = input('Second Number: ')  
08 num = int(numX) + int(numY)  
09 print(name, num)  
10 numbers = [numX, numY]  
11 print(name, numbers)
```

```
579  
name: John  
First Number: 80  
Second Number: 90  
  
John 170  
  
John ['80', '90']
```

Exercise

<https://www.onlinegdb.com/>

□ 程式讀取兩個整數，將其四則運算印出

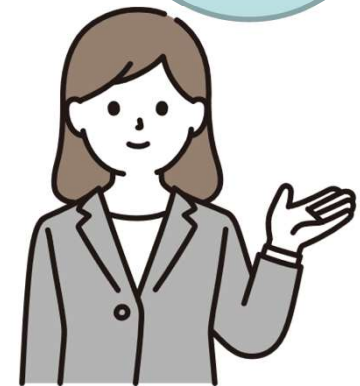
○ 其中除法印出整數、與小數

○ 提示: `input()` 回傳是 String (字串)，須轉型成 `int`。

```
01 m =
02 n =
03 r1 = m + n
04 r2 = m - n
05 r3 = m*n
06 r4 =
07 r5 =
08 print('add:', r1)
10 print('sub:', r2)
11 print('product:', r3)
12 print('divide int:', r4)
13 print('divide float :', r5)
```

```
m = int(input('m = '))
n = int(input('n = '))
r1 = m + n
r2 = m - n
r3 = m*n
r4 = int(input('r4 = '))
r5 = int(input('r5 = '))
print('add:', r1)
print('sub:', r2)
print('product:', r3)
print('divide int:', r4)
print('divide float :', r5)
```

解答



Exercise

- 一元二次方程式， $ax^2 + bx + c = 0$ ，輸入a, b, c, 求方程式的兩個實根。

○ $x1 = (-b + \text{math.sqrt}(b*b - 4*a*c)) / 2 / a$

○ $x2 = (-b - \text{math.sqrt}(b*b - 4*a*c)) / 2 / a$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

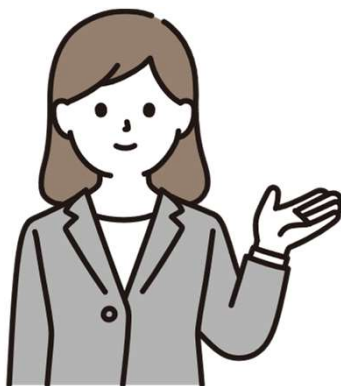
- Input

- 1
- -2
- 1

```
01 import math
02 x = math.sqrt(34)
03 print(x)
```

- Output

- 1.0
- 1.0



```
import math
```

```
a = int(input())
```

```
b = int(input())
```

```
c = int(input())
```

```
x1 = (-b + math.sqrt(b*b - 4*a*c)) / 2 / a
```

```
x2 = (-b - math.sqrt(b*b - 4*a*c)) / 2 / a
```

```
print(x1)
```

```
print(x2)
```

解答

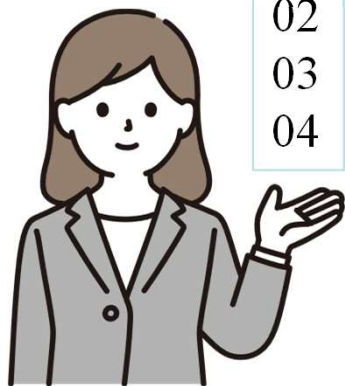
Exercise

- A、B、C三本書價格如下，
- 一顧客欲購買A: x 本、B: y 本、C: z 本（x、y、z 為使用者輸入），請計算需花費多少錢？

	定價
A	380
B	1200
C	180

```
01 A, B, C = 380, 1200, 180
02 x, y, z = int(input()), int(input()), int(input())
03 cost = A * x + B * y + C * z
04 print('cost=', cost)
```

解答



Exercise

- 某一學生修國文、計算機概論、計算機程式設計三科，使用者輸入名字、學號、三科成績。
- 計算學生總成績、平均。
 - 印出名字、學號、總成績、平均。
 - 使用 format 印出

```
01 student = ["", 0, 0, 0, 0] #初始化串列
02 student[0] = input()      # 輸出名字
03 student[1] = input()      # 輸入學號
04 student[2] = int(input())
05 student[3] = int(input())
06 student[4] = int(input())
07 Total = student[2] + student[3] + student[4]
08 Average = Total / 3.0
09 print('Total = ' + str(Total))
10 print('Average = ' + str(Average))
```

Input	Output
Tom	Name:Tom
11259001	Id:905067
90	Total:300
80	Average:90
100	



程式註解

- ❑ 單行註解為 #
- ❑ 多行註解用 """ 開頭與結尾。

```
"""  
Compute Age Equation  
As a Function  
"""  
def getAge(cell):  
    age = 3*cell*cell-6*cell+1  
    return age  
  
# Compute Total Age  
totalAge = getAge(100) + getAge(200) + getAge(300) + getAge(400)  
print(totalAge)
```

END

