程式設計概論
Programming 101

—程式變數型態與基本運算

授課老師：邱淑怡

Date: 3/11/2021

政大資訊科學系
Department of Computer Science, National Chengchi University

大綱

- Python 程式碼撰寫風格
- 資料型別
- 程式的變數
- 變數如何運用
- 如何輸入、輸出(印出)
  - print():印出字串、數值計算後的結果
  - input(): 接收使用者的輸入
  - int(): 將文字轉成數字
  - eval():

Python 程式碼撰寫風格

- 程式(program)：由一行一行的敘述(statement)所組成
- 敘述：由關鍵字、特殊字元或識別字所組成
  - 關鍵字(keyword):Python所定義特定意義或用途，ex: if, for,...
  - 特殊字元(special character): 小括號、單(雙)引號、井字(#)符號
  - 識別字(identifier): 程式人員自行定義的新字做為變數、函式的名稱
    - 字母大小寫意義不一樣
    - 利用空白 讓程式容易閱讀；但須避免無意義的空白
    - 變數名稱建議是有意義的名稱
- 敘述可包含函式(function)、流程控制(flow control)、類別(class)，可執行的單元

Python 程式碼撰寫風格(cont.)

- 縮排
  - 每個縮排層級使用 4個空白，或用[tab]鍵，但不能混合空白和**[tab]**鍵
  - Python使用縮排來劃分程式的執行區塊，程式不能隨意縮排
- 註釋(comment): # 標示單行註釋；''' 或"""標示多行註釋
  - 建議程式中多些註釋說明
- 建議一行一個敘述

變數型別(data type)

- 數值型別(numeric type)
  - int, float, bool, ~~complex~~
- 文字序列型別(text sequence type)
  - str
- 序列型別(sequence type)
  - list, tuple, range
- 集合型別(set type)
  - set
- 對映型別(mapping type)
  - dict

資料型態(data type)

- int: 表示整數
  - print(3+6)
- float: 表示浮點數
  - 如: 1.234567E+3 表示 $1.234567*10^3$ (=1234.567)
  - print(1.234567E+3)
- bool: 表示布林(Boolean): True (T) / False (F)
  - Ex: 1>2
- str: 字串(string)由一連串字元組成
  - 單引號、雙引號、三個單引號、三個雙引號
  - print('Python'); print("Python code")
  - 單引號和雙引號不能混合使用

注意大小寫

不建議使用

變數 (variable)命名

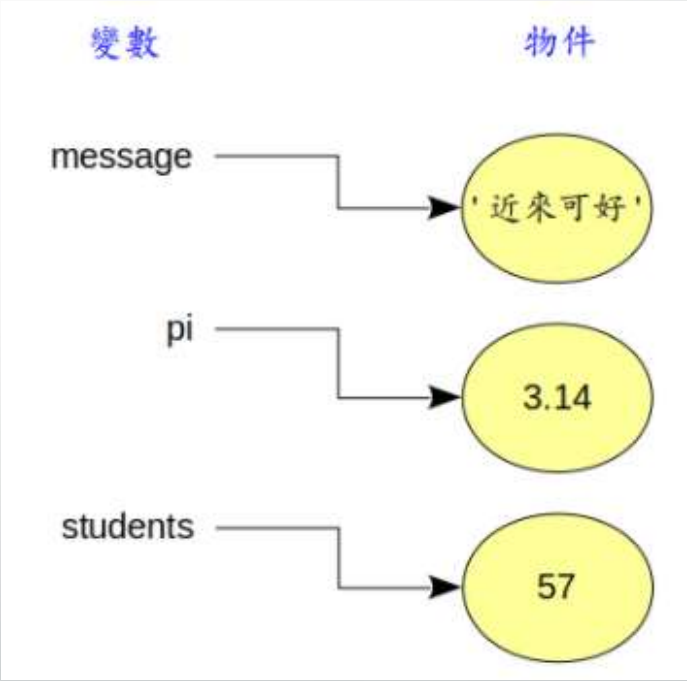
- 變數參照可改變的值，這個值儲存在記憶體
- 變數命名
  1. 以英文字母或底線(_)開頭，之後可以接字母、數字或底線
  2. Python 的關鍵字(keyword)及內建函數(built-in function)不能當變數名稱
  3. **字母大小寫有別(case-sensitive)**
  4. 變數名稱可以為中文 (Python 3)，不建議使用
  5. 建議變數名稱需有意義的名稱

變數的第一個字元不能是數字

Source: http://yltang.net/tutorial/python/3/

變數(variable)

- 變數：代表一個值的名稱
- 指派指令 (Assignment instruction) 可以建立新的變數，並設定其值
- 例子
  - message = '近來可好?'
  - pi = 3.14
  - students = 57



http://yltang.net/tutorial/python/3/

關鍵字(keyword)及內建函數(built-in function)

keyword

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

built-in function

abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	

務必注意字母大小寫

Source: http://yltang.net/tutorial/python/3/

python module docs

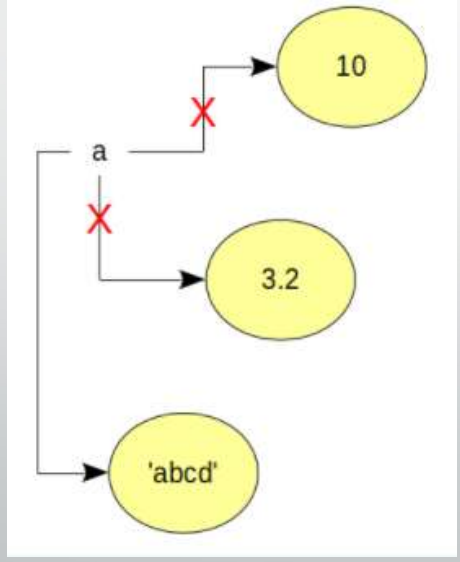
Built-in Modules

_abc	_imp	_stat	errno
_ast	_io	_statistics	faulthandler
_bisect	_json	_string	gc
_blake2	_locale	_struct	itertools
_codecs	_lsprof	_symtable	marshal
_codecs_cn	_md5	_thread	math
_codecs_hk	_multibytecodec	_tracemalloc	mmap
_codecs_iso2022	_opcode	_warnings	msvcrt
_codecs_jp	_operator	_weakref	nt
_codecs_kr	_pickle	_winapi	parser
_codecs_tw	_random	_xxsubinterpreters	sys
_collections	_sha1	array	time
_contextvars	_sha256	atexit	winreg
_csv	_sha3	audioop	xxsubtype
_datetime	_sha512	binascii	zlib
_functools	_signal	builtins	
_heapq	_sre	cmath	

可查看python module docs

Python變數特性

- 利用指派指令直接創造新變數
- 動態資料型別(dynamic typing)，只要指派指令表示式的資料型態改變，變數的資料型態就改變了
- Ex:
  - a=10
  - a=3.2
  - a='abcd'
  - print(a)



Python特殊用法:
指派鏈 (chained assignments)
- X=Y=Z=100
- X, Y, Z=100, 3.14159, "Hello"
- print(X,Y,Z)

http://yltang.net/tutorial/python/3/

Numeric: 整數和浮點數

- 運算子(operator):運算符號
- 運算元(operand): 運算子所運算的值
- **運算順序(由高至低):**
  1. 括號內的表示式 **(最高)**
  2. 次方
  3. 乘、除(/,//)及餘數(%)
  4. 加減 **(最低)**
  5. **優先權相同的，由左至右進行運算**

實例練習

- 基本題
  - a=3
  - print(a+7)
  - print(a**2)
  - print(100/3)
  - print(100//3)
  - print(100%3)
- 進階題
  - X1= 2*(3+4)
  - X2= 2**3+4
  - X 3= 3**2*4
  - x 4= 2*3/4
  - x5 = 3**2**4
  - print(x1,x2,x3,x4,x5)

實例練習_執行結果

- 基本題
  - a=3
  - print(a+7)
  - print(a**2)
  - print(100/3)
  - print(100//3)
  - print(100%3)
- 進階題
  - X1= 2*(3+4)
  - X2= 2**3+4
  - X 3= 3**2*4
  - x 4= 2*3/4
  - print(x1,x2,x3,x4)

- 基本題_執行結果
  - a=3
  - # result: 10
  - #square of a, result: 9
  - # result: 33.3333336
  - # result: 33
  - # result: 1
- 進階題_執行結果
  - # print(x1) result: 14
  - # print(x2) result: 12
  - # print(x3) result: 36
  - # print(x4) result: 1.5

Boolean value(布林值)

- 為了解決條件判斷的問題
  - 布林(boolean)型態為: True/ False
  - 運算子(Relational operator)
    - x == y (x等於y)
    - x != y (x不等於y)
    - x > y (x大於y)
    - x >= y (x大於等於y)
    - x < y (x小於y)
    - x <= y (x小於等於y)

比較運算子

'=' 表示指派指令(右邊值給左邊的變數)
'==' 表示判斷兩邊是否相等或一致

Example code

- 5==5
- 5 !=2+4
- a=8
- a>=7+3
- a < 7+3

多項條件的判斷

依據這些進行組合

- Logical operator: and, or, not

真值表

A	B	A and B	A or B
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

A, B 分別代表一個判斷式

A	not A
True	False
False	True

Boolean 練習

- 請用print()函數將結果(True/False)呈現出來
  - x = 7
  - y=8
  - 0<x and x<10
  - 0<x <10
  - not(y<6)
  - x%2==0 or y%3==0
  - not (x%2==0 and y%3==0)

```
x = 7
y=8
print(0<x and x<10)
print(0<x <10)
print(not(y<6))
print(x%2==0 or y%3==0)
print(not (x%2==0 and y%3==0))
```

Boolean value 優先層次

優先層次 (由高至低)	分類	Operators
8	括號	( )
7	次方	**
6	乘除法	*, /, //(取商數), %(取餘數)
5	加減法	+, -
4	關係	==, !=, <, >, <=, >=
3	邏輯	not
2	邏輯	and
1	邏輯	or

- Example code
  - x=3
  - y=10
    - print(10<x*5 and y-6<2)
      - # 1. compute x*5, y-6
      - # 2. relation 10<15, 4<2
      - # 3. logical True and False
      - # Finally, result: False

字串(string)

- 由一連串的字元組成，且由兩個單引號或兩個雙引號所包含
- 字串相加
- K倍字串
- 有趣的三個單引號、三個雙引號

- Example code
  - print(“Hello, world”)
  - name = “Ben"

    print(“Hello, “ + name + “!!!”)

    # result: Hello, Ben!!!
  - text=“ok”

    print(3*text) #result: okokok

字串(string)操作

- a=‘Hello’
- b=‘Python’
- 「+」字串相加:　c=a+b
  - print('我今年’ + ’20 ’ + ‘歲’)
- 「 * 」重複: a*2
- 「 [], [:]」取出部分字串，取部分內容這個動作叫 slicing(切片)
- 「 in 」成員:如果一個字符存在給定的字符串中，則返回true
  - print('boy’ in var1)
- 「 not in 」: 與上面in類似

字串的索引及片段運算子

- 索引運算子([])可取得字串中的字元
- x1='Python程式設計'

x1[a:b]: 取出x1的索引值從a到(b-1)的內容

索引	0	1	2	3	4	5	6	7	8	9
內容	P	y	t	h	o	n	程	式	設	計
索引	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
print(x1[2:5])     # 'tho'
print(x1[3:7])     # 'hon程'
print(x1[6:-1])    # '程式設'
print(x1[:2])    # 印出索引0,1(不包含2)
print(x1[2:])    # 印出索引2...9(不包含10)
print(x1[9:10]) # 取出最後一個字元印出
print(x1[-1]) # 取出最後一個字元印出
print(len(x1)) # 印出該字串的長度
```

字串的索引及片段運算子_練習題

- S1="HappyNewYear"
- S2="happynewyear"
- s3="new"
- S4 = ' birthday'
- 問題1: s1的長度
- 問題2: s1和s2長度是否相等
- 問題3: s3是否存在於s1?
- 問題4: s1的第5~9個字元(yNewY)是什麼?
- 問題5: 取出 s2中的new字串?
- 問題6: 如何利用上述的四個字串得到'Happy birthday'

字串的索引及片段運算子_練習題參考程式

- S1="HappyNewYear"
- S2="happynewyear"
- s3="new"
- S4 = ' birthday'
- 問題1: s1的長度　# print(len(s1))
- 問題2: s1和s2是否相等　# print(len(s1)==len(s2))
- 問題3: s3是否存在於s1?　# print(s3 in s1)
- 問題4: s1的第5~9個字元是什麼?　#print(s1[4:9])
- 問題5: 取出 s2中的new字串?　# print(s2[5:8])
- 問題6: 如何利用上述的四個字串得到'Happy birthday' # print(S1[0:5],S4, sep=' ')

資料型態轉換

- 如何知道X變數的資料型態
  - print(type(X))
- int():傳入之參數轉為整數，若參數為浮點數則將小數捨去
- float():將傳入之參數轉為浮點數
- str():將傳入之參數轉為字串

如何接收使用者的輸入資訊?

使用 input() 函數

input() 函數

- 要接收使用者輸入的變數，則可使用 input 函式(先介紹一次讀取一個數字的方法)
- Example
  - a = input('Please enter 1st number: ')
  - b = input('Please enter 2nd number: ')
  - print(a + b)

結果是什麼?

說明:因為輸入(input)的內容被當成一般文字，而加號為文字的連接

解決方法
print(int(a) + int(b))

input() 函數(cont.)

- 要接收使用者輸入的變數，則可使用 input 函式(先介紹一次讀取一個數字的方法)

參考程式

```
a = input('Please enter 1st number: ')
b = input('Please enter 2nd number: ')
print(int(a)+ int(b))
```

=

```
a = int(input('Please enter 1st number: '))
b = int(input('Please enter 2nd number: '))
print(a + b)
```

input() + eval() 函數

- eval()函式字串轉換成數值資料型態

Example: 計算數學、國文、英文三科分數的總成績

```
score_math = eval(input("請輸入數學分數： "))
score_Eng = eval(input("請輸入英文分數： "))
score_ch = eval(input("請輸入國文分數： "))
total = score_math+score_Eng+score_ch
print("總成績為", total)
```

輸入的資料若無法被eval()函式轉換為數值，將發生錯誤並終止程式

input() + eval() 函數_練習題

- 基本題
  - 請撰寫一個Python程式，它可以讓使用者輸入梯形的上底、下底與高，計算梯形面積並印出結果
- 進階題
  - 請撰寫一個Python程式，它可以讓使用者輸入兩個點的座標，然後計算兩點的距離並印出結果

參考程式：

請撰寫一個Python程式，它可以讓使用者輸入兩個點的座標，然後計算兩點的距離並印出結果

兩種方式都做到相同結果

但int() 無法接受兩個變數的輸入

```
x1, y1 = eval(input('請輸入第一點的座標： '))
x2, y2 = eval(input('請輸入第二點的座標： '))
distance = ((x2-x1)**2+(y2-y1)**2)**0.5
print("兩點距離為 ", distance)
```

```
x1_1 = int(input("請輸入第一點的x座標： "))
y1_1 = int(input("請輸入第一點的y座標： "))
x2_1 = int(input("請輸入第二點的x座標： "))
y2_1 = int(input("請輸入第二點的y座標： "))
distance_1 = ((x2_1-x1_1)**2+(y2_1-y1_1)**2)**0.5
print("兩點距離為 ", distance_1)
```