Python Basis

郭忠義

jykuo@ntut.edu.tw

臺北科技大學資訊工程系

Python簡介

- 特性
  - 容易撰寫，適合初學者
  - 功能強大
  - 跨平台
- 功能
  - 資料分析
  - 科學計算
  - 機器學習
  - 網頁設計
- 基本語法
  - 結尾不須分號，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(不須大括號)

標記

- 直譯器利用標記 (token) 解析程式的功能標記有
  - 關鍵字 (keyword)
  - 識別字 (identifier)
  - 字面常數 (literal)
  - 運算子 (operator)

關鍵字

- 關鍵字為具有語法功能的保留字 (reserved word) ，Python 的關鍵字

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	

識別字

- 識別字為寫程式時依需求自行定義的名稱，包括變數 (variable) 、函數 (function) 、類別 (class) 等
- 除關鍵字外， Python 可用任何 Unicode 編碼的字元當作識別字，包括中文。

字面常數

- 字面常數是字面上意義，1234代表整數數值一千兩百三十四，是直接寫進 Python 程式原始碼的數值
  - 字串字面常數 (string literal)　'abcd'
  - 字節字面常數 (bytes literal)
  - 整數字面常數 (integer literal)　1234
  - 浮點數字面常數 (floating-point literal)　1.023
  - 複數字面常數 (imaginary literal)　3 + 2i

運算子

- 運算子有優先順序

運算子	描述
**	指數
~x	按位翻轉
+x,-x	正負號
*,/,%	乘法、除法與取餘
+,-	加法與減法
<<,>>	移位
&;	按位與
^	按位異或
|	按位或
<,<=,>,>=,!=,==	比較
is,is not	同一性測試
in,not in	成員測試
not x	布林“非”
and	布林“與”
or	布林“或”
lambda	Lambda表示式

基本資料型別

- 整數(int)
- 浮點數(float):有小數點的數字
- 複數 complex
- 字串(String):一串文字內容
- 字節(byte): 二進位資料
- 序列(Tuple):又稱固定列表，一種有順序、不可變動的資料集合
- 串列(List):又稱可變列表，一種有順序、可變動資料集合
- 集合(Set):無順序的資料集合
- 字典(dictionary):以Key-Value Pair形式存入集合
- 布林(Boolean):表達True(正確、真)與False(錯誤、假)

資料型別-整數與浮點數

- 數學運算子 (operator)運算
  - +, -, *, /, ** 次方, //　整除, % 餘數

```
print(5 + 9)
print(10 - 7)
print(4 - 10)
print(5 + 3 - 17)
print(6 * 7)
print(6 ** 2)
print(9/5)
print(9//5)
print(9%5)
print(9/0)
```

⇨ 執行結果?

資料型別-複數

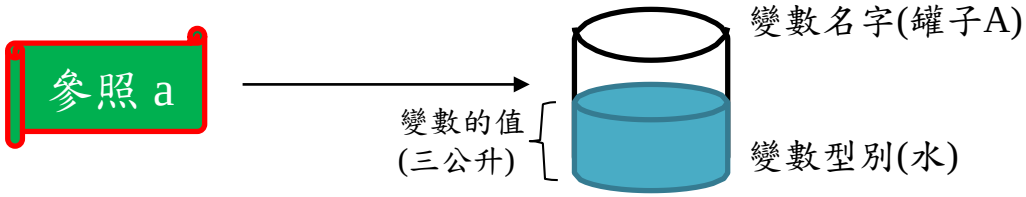
- 數學運算子 (operator)運算
  - +, -, *, /, ** 次方, // 整除, % 餘數

```
a = 1
b = 2
c = 3.6
d = 3.6
print(int(a + c), int(a +d))
print(float(a + b))
print(complex(a+b))
print(complex(a+b) + complex(2,3))
```

```
4 4
3.0
(3+0j)
(5+3j)
```

變數(Variable)與物件(Object)

- 變數特性 (不可與關鍵字/保留字相同)
  - 名稱自訂，參照/指向一個資料(值)物件
  - 資料(值)物件
    - 在電腦記憶體 (RAM)中某個位址，好比容器有水三公升



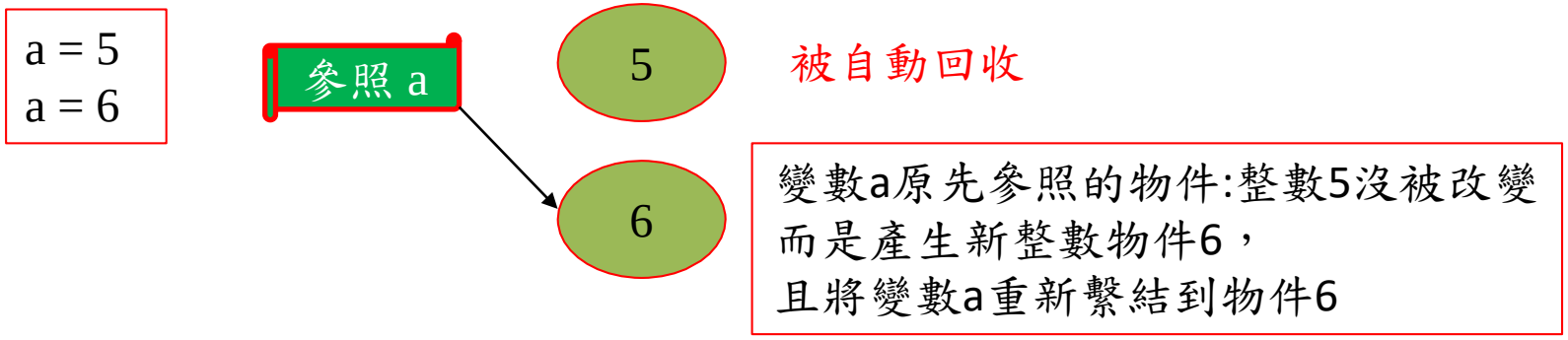
- Python 中所有資料 (data) 都是物件。物件有:
  - id()　　編號
  - type()　　資料型別
  - value　　值

```
a = 3
print(id(a))
print(type(a))
print(a)
```

```
140725290111872
<class 'int'>
3
```

變數(Variable)與物件(Object)

- 不可變的 (immutable)物件資料
  - 序對 (tuple) 、整數、浮點數、字串是不可變的
- 可變的 (mutable)物件資料
  - 串列 (list) 或字典 (dictionary) 是可變的。
- 物件不再使用時，直譯器會自動垃圾收集 (garbage collection) ，釋放記憶體空間。

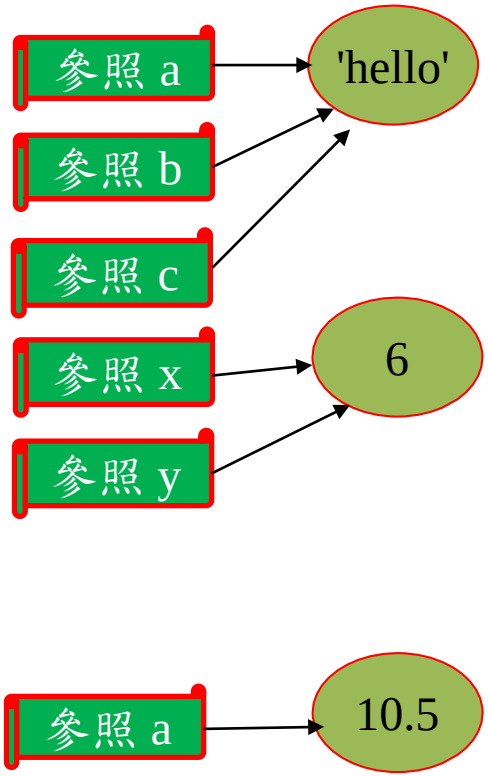


不可變物件

- 物件資料不可變的 (immutable)
  - 變數指定時會對應相同的 id

```
a = 'hello'
b = a
c = 'hello'
print (a is b)
print (a is c)
print(id(a))
print(id(b))
print(id(c))
a = 10.5
x = 6
y = 6
print(x is y)
print(x == y)
```

```
True
True
2042424477096
2042424477096
2042424477096
True
True
```

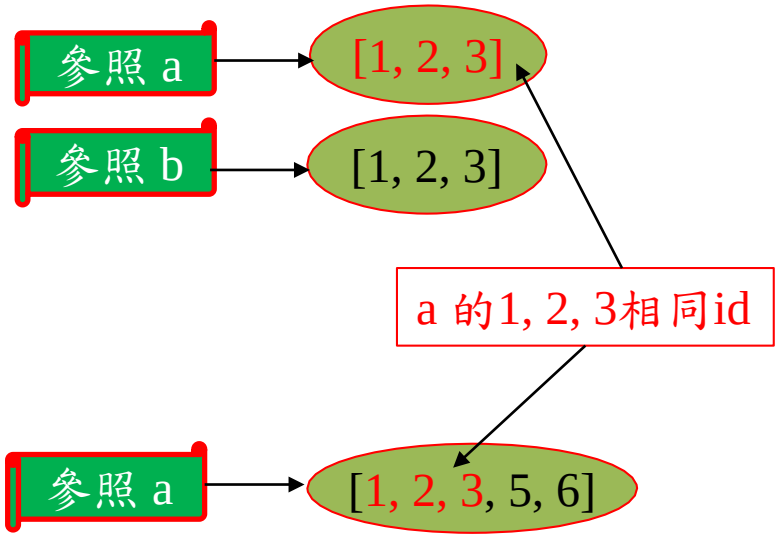


可變物件

- 物件資料可變 (mutable)
  - 變數指定時會產生新的 id
  - 擴充時可變物件變數 a 產生新的id
  - 擴充時原本物件資料 a[0], a[1], a[2] id相同

```
a = [1, 2, 3]
b = [1, 2, 3]
print(id(a))
print(id(b))
print(a is b)
print(id(a[0]))
print(id(a[1]))
a = a + [5, 6]
print(id(a))
print(a)
print(id(a[0]))
print(id(a[1]))
```

```
2042419984392
2042424366664
False
140725290111808
140725290111840
2042424396680
[1, 2, 3, 5, 6]
140725290111808
140725290111840
```



表示式(Expressions)

- 表示式

```
3 + 5
3 + (5 * 4)
3 ** 2
‘Hello’ + ‘World’
num + 3
```

變數(Variable)和表示式(Expressions)

- 賦值運算
  - 將右邊運算結果，用 = 賦予(存到/指派)左邊一個變數
  - 左邊變數之後可使用其值，其值也可改變。

a = 4 + 3

跟數學"相等"意義完全不同

```
a = 4 + 3
b = a * 4.5
c = (a+b)/2.5
a = "Hello World"
```

```
a = 10
b = a + 5
a = 10.1
print(a, b)
```

執行結果?

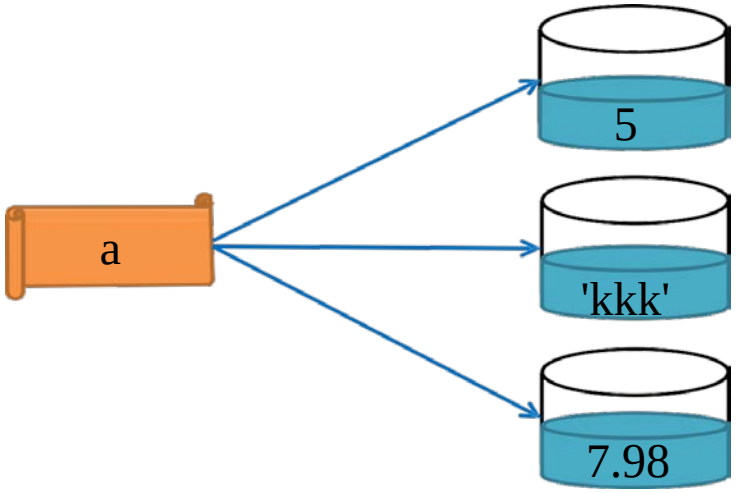
變數(Variable)和表示式(Expressions)

- 變數資料型別
  - 變數值同時可看出變數型別

```
Var = 70 (整數)
Var = 6.78 (浮點數)
Var = "alcom lab" (字串)
```

  - 變數可以變更資料型別
    - 變數 a 代表一個參照 (Reference)

```
a = 5
a = 'kkk'
a = 7.98
```



變數

- 變數複合運算
  - +=, -=, *=, /=
  - 多變數一次指派
  - 變數內容對調

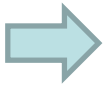
```
a = 10
a+=b
x, y = a, b

x, y = y, x
```

```
a = 10
a+=b
x = a
y = b
tmp = x
x = y
y = tmp
```

變數

- 變數命名規則
  - 允許 a-z A-z 0-9 和 _ , 但是開頭必須是 [_a-zA-Z] 其中一個
  - 變數區分大小寫
  - 變數長度沒有限制
  - 特殊的變數, 開頭和結尾都加兩個下底線, ex: __init__

```
_a = 10
print(a)
Bb_ = _a + 5
2c = 3 + Bb
a = bb + 5
```



執行結果?

資料型別-字串

- 字串以單引號'或雙引號"包起來 ，表示文字資料。
- 字串可存於變數

```
x = 'Hello world'
print(x)
```

- "\" 可轉字串某些字元含義，最常見如: \n \t

```
x = 'Hello \n world'
print(x)
```

- 字串表示單引號，須以 \ 處理，避免被誤認為字串結束。

```
x = 'I\'m Lucas'
print(x)
```

使用者輸入

- input('提示字')
  - 以字串型別從鍵盤輸入資料
  - 等待輸入時，顯示提示字
- int() 轉換成整數

```
def inOps():
    numX = "123"
    numY = "456"
    num = int(numX) + int(numY)
    print(num)
    name = input('name: ')
    numX = input('First Number: ')
    numY = input('Second Number: ')
    num = int(numX) + int(numY)
    print(name, num)
```

```
579

name: John

First Number: 80

Second Number: 90
John 170
```

Exercise

- 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。

print

- %d　10 進位整數輸出
- %f　浮點數10進位輸出
- %e, %E 將浮點數以10進位科學記號輸出
- %o　以8進 位整數方式輸出
- %x, %X　　將整 數以16進位方式輸出
- %s　使用str()將字串輸出
- %c　以字元方式輸出
- %r　使用repr()輸出字串
  - 表達資料型別，例如字串''
- %%在字串 中顯示%

```
>>> text = '%d %.2f %s' % (1, 99.321, 'Justin')
>>> print(text)
1 99.32 Justin
>>> print('%d %.2f %s' % (1, 99.3, 'Justin'))
1 99.30 Justin
>>> 'example:%.2f' % (19.234)
'example:19.23'
>>> "example:%6.5f" % 19.234111
'example: 19.23'
```

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+math.sqrt(b*b-4*a*c))

- Input
  - 1
  - -2
  - 1

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

```
import math
x=math.sqrt(34)
print(x)
```

- Output
  - 1.0
  - 1.0

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

資料型別-字串操作

- 使用+運算子結合字串
- 使用*複製字串
- 使用[]擷取字元

```
def strOp():
    print("Hello" + " World")
    name = "小明"*2
    print(name)
    Hi = "Hi~ 您好!"
    print(Hi[1])
    print(Hi[4])
```

```
Hello World
小明小明
i
您
```

資料型別-字串操作

- 擷取字串[start:end:step]，
  - start~(end-1), each step
  - start~(前一個), each step
  - Start, end, -1 表示倒數過來第一個

```
def strGet():
    words = "This is a book, that is a cat~"
    print(words)
    print(words[::])
    print(words[5:])
    print(words[-4::])
    print(words[-14:20:1])
    print(words[-6:5:-1])
    print(words[-4:-10:-2])
```

```
This is a book, that is a cat~
This is a book, that is a cat~
is a book, that is a cat~
cat~
that
a si taht ,koob a s
cas
```

Exercise

- 輸入兩個英文句子 A, B，
  - 將兩個英文句子A, B串聯成 C
  - 將C倒著輸出
  - C從第五個字到最後第三個字，每隔兩個字輸出

Input

This is a book
That is a cat

資料型別-字串操作

- len()取得長度
- .replace()替換
- .split()切割字串，回傳str list

```
def strOps():
    words = "小明今天去學校，小明遲到。"
    wordLength = len(words)
    print(wordLength)
    sentence = words.replace("小明","湯姆")
    print(words)
    print(sentence)
    wordsSplit = words.split("明")
    print(wordsSplit)
```

```
13
小明今天去學校，小明遲到。
湯姆今天去學校，湯姆遲到。
['小', '今天去學校，小', '遲到。']
```

Exercise

- 輸入兩個英文句子 A, B，兩個英文字 x, y
  - 將兩個英文句子A, B串聯成 C
  - 將C其中的 x 替換成 y，變成 D
  - 輸出C, D 長度的加總
  - 輸出C前三個字，每一個字重複輸出三次。

Input

This is a book
That is a cat
is
was

string與byte轉換

- 字元編碼
  - 有ASCII、Unicode(\uXXXX)、GB2312(資訊交換用漢字編碼字元集)、GBK(漢字內碼擴展規範)、GB 18030、Big5等，不同編碼間可以互相轉換。
  - UTF-8是Unicode的一種電腦儲存實現方式，即字元編碼格式。UTF-8一般用於網路傳輸。
- Python
  - 二進制資料由 bytes型別表示
  - 文字/字串使用Unicode，由 str型別表示
- str 跟 bytes 可互相轉換
  - str.encode() 預設編碼 utf-8

str.encode() ⇔ bytes.decode()
(Unicode) (utf-8)

string與byte轉換

```
a = bytes([1,2,3,4,5,6,7,8,9])
b = bytes('python', 'ascii')
print(type(a))    # <class 'bytes'>
print(type(b))    # <class 'bytes'>
print(a)          # b'\x01\x02\x03\x04\x05\x06\x07\x08\t'
print(b)          # b'python'
```

```
s = u'\u4eba\u751f\u82e6\u77ed\uff0cpy\u662f\u5cb8'
print(type(s))  # <class 'str'>
print(s)        #人生苦短，py是岸
s_utf8 = s.encode(encoding='utf-8')
print(s_utf8)
#b'\xe4\xba\xba\xe7\x94\x9f\xe8\x8b\xa6\xe7\x9f\xad\xef\xbc\x8cpy\xe
6\x98\xaf\xe5\xb2\xb8'
```

string與byte轉換

```
# bytes轉字符串方式一
b=b‘\xe9\x80\x86\xe7\x81\xab‘
string=str(b,'utf-8')
print(string)

# bytes轉字符串方式二
b=b'\xe9\x80\x86\xe7\x81\xab'
string=b.decode() # 第一參數預設utf8，第二參數預設strict
print(string)

# bytes轉字符串方式三
b=b'\xe9\x80\x86\xe7\x81haha\xab'
string=b.decode(‘utf-8‘,‘ignore‘) # 忽略非法字符，用strict會拋出異常
print(string)

# bytes轉字符串方式四
b=b'\xe9\x80\x86\xe7\x81haha\xab'
string=b.decode(‘utf-8‘,‘replace‘) # 用？取代非法字符
print(string)
```

string與byte轉換

```
# 字符串轉bytes方式一
str1='逆火'
b=bytes(str1, encoding='utf-8')
print(b)

# 字符串轉bytes方式二
b=str1.encode('utf-8')
print(b)
```

```
b'\xe9\x80\x86\xe7\x81\xab'
b'\xe9\x80\x86\xe7\x81\xab'
```

I/O

- print預設印一行換行，使用,在每次印出間以空格取代，更自由可使用library中的write函式。

```
import sys
file_in = open('db.txt','r')
file_out = open('copy.txt','w')
for line in file_in:
    for i in range(0,len(line)):
        if line[i]!="\n":
            sys.stdout.write(line[i]+',')
        else:
            sys.stdout.write(line[i])

file_out.write(line[i])
sys.stdout.write("\n")
file_in.close()
file_out.close()
```

```
# 造出 db.txt
1111
2222
ssss
wwww
5555

Output:
1,1,1,1,
2,2,2,2,
s,s,s,s,
w,w,w,w,
5,5,5,5,
copy.txt內容和db.txt一樣
```

import

- 用import直接匯入整個python函式庫中所有函式，或用from 函式庫import函式，插入特定函式
  - import x.py
    - 會執行 x.py 未空四格的指令
    - 一般函式庫不會有

```
#匯入sys函式庫所有函式，使用write函式前須加sys
import sys

#從time函式庫匯入time()函式，使用time()前不需加 time
from time import time
sys.stdout.write(str(time())+"\n")
```

Output:
1409796132.99 #當下的time

import

- module
  - 一個包含有Python的程式定義及敘述的檔案。
  - 檔案名稱是module名稱加上延伸檔名 .py 。
  - module裡存在module名字變數，如sys, time，當作全域變數(global variable)使用。
- 常見module
  - os模組：封裝不同作業系統的通用api，在不同作業系統下，可使用相同的函數呼叫
  - sys模組：作業系統的相關資訊與函數的模組
  - built-in模組：內建模組
  - time模組：時間相關模組
  - re模組：正則表達式

import

- 常見module
  - thread模組：執行緒模組
  - urllib模組：接受url相關呼叫的模組，接受編碼及解碼
  - urllib2模組：接受url相關呼叫的模組
  - socket模組：網路通訊模組
  - file模組：檔案使用模組

import

- __main__
  - A模組/檔案，直接執行時，__main__的值預設是【'__main__'】

```
# A.py
if __name__ == '__main__':
    print('Direct running')
else:
    print('import __name__= ',__name__)
```

Direct running

  - A模組/檔案，被import後，__main__的值是，檔案名稱
    - 執行 B.py

```
#B.py
import A
print('import A')
```

import __name__= A
import A

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

```
def format00():
    str='{0} is {1}!!'.format('Justin', 'caterpillar')              #第0個對應後面第0個參數
    print(str)                                                      #'Justin is caterpillar!!'
    str='{real} is {nick}!!'.format(real = 'Justin', nick = 'caterpillar')  #以符號對應
    print(str)                                                      #'Justin is caterpillar!!'
    str='{real} is {nick}!!'.format(nick = 'caterpillar', real = 'Justin')  #符號順序無關
    print(str)                                                      #'Justin is caterpillar!!'
    str='{0} is {nick}!!'.format('Justin', nick = 'caterpillar')
    print(str)                                                      #'Justin is caterpillar!!'
    str='{name} is {age} years old!'.format(name = 'Justin', age = 35)
    print(str)                                                      #'Justin is 35 years old!'
```

format

```
import math
import sys
def format01():
    str=math.pi
    print(str)                                                          #3.14159265359'
    str=format(math.pi, '.18f')                                        #PI 取小數點18位
    print(str)                                                          #3.141592653589793116
    str='PI = {0.pi}'.format(math)                                     #第0個對應後面第0個參數
    print(str)                                                          #'PI = 3.14159265359'
    str='My platform is {pc.platform}'.format(pc = sys)                #以符號 pc 對應 sys
    print(str)                                                          #'My platform is win32'
    str='My platform is {0.platform}. PI = {1.pi}.'.format(sys, math)
    print(str)
    #'My platform is win32. PI = 3.14159265359.'
    str='element of index 1 is {0[1]}'.format([20, 10, 5])             #第0個對應後面第0個參數
    print(str)                                                          #'element of index 1 is 10'
    str='My name is {person[name]}'.format(person = {'name' : 'Justin', 'age' : 35})
    print(str)                                                          #'My name is Justin'
```

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

Input
K
905067
100
100
100

Output
Name:K
Id:905067
Total:300
Average:100

Output
Name K, Id is 905067, Total score is 300, Average is100