

Chapter 3

變數與運算子

變數

程式語言中使用各種資料型態，都必須要跟系統取得記憶體空間

Python 裡定義變數就是跟系統要記憶體空間來存放

A = 5

B = "Python"

C = True

D = 3

A = A+D

變數

在 Python 裡面的所有東西都被稱為物件 (Object)

不同的變數可以代表同一個物件

而變數名稱可以使用中文命名
(一般還是用英文 + 數字的組合)

```
咖稱 = "貴賓狗"  
亮亮 = "雪納瑞"  
古力 = "吉娃娃"
```

```
print(咖稱)  
print(亮亮)  
print(古力)
```

C:\ PAUSE

貴賓狗
雪納瑞
吉娃娃

請按任意鍵繼續 . . .

變數命名規則

變數第一字不能是數字

5 = 500 (X)

_5 = 500 (O)

大小寫有區分

abc = 500

Abc = 500

aBC = 500 ... 以上 3 個變數是不一樣的

變數命名規則

不能使用關鍵字 (保留字) 來命名

Python 保留字						
false	class	finally	is	none	continue	for
lambda	true	def	from	nonlocal	and	del
global	not	as	elif	if	or	assert
else	import	pass	break	except	in	raise
return	try	while	with	yield	print	input

變數命名規則

對於變數的命名建議用小寫就好，也可以加底線

Ex. taiwan_no1

也可用一種叫做「小寫駝峰型」的方式命名

Ex. allenIverson



From wiki



From Bomb01.com

Assign(指派) 運算子 : “ = ”

什麼是 assign operator “ = ”

For example

a = 3

程式會先執行 “ = ” 右邊 , 也就是建立出一個整數 3
後

再把整數 3 assign 給 a 這個變數

Assign(指派) 運算子 : “ = ”

變數又可以分成全域變數和區域變數

全域變數就是整隻程式都可用，變數只要 `assign` 後，用到該變數名稱的程式碼便都會指向同一個物件

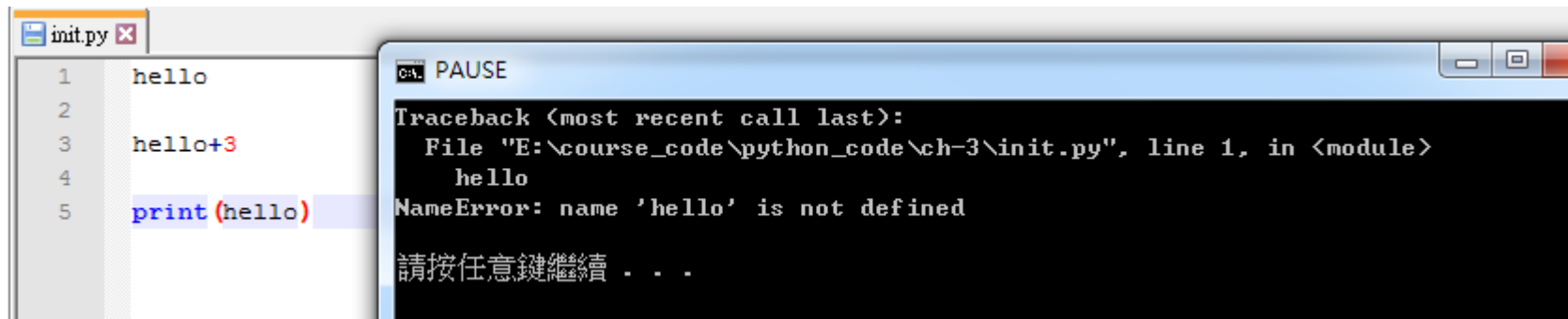
區域變數則大多是出現在迴圈與函數裡面，如果一個變數首次被 `assign` 在被迴圈或函數的程式區塊裡，此變數只能生存在該區域

Assign(指派) 運算子 : “ = ”

Python 屬於直譯式語言

使用變數時並不需要事先宣告，但變數第一次出現時須給予一個初始值

初始值時的可以是數值、字串、 set 、 list 等等



The screenshot shows a Python IDE with a file named `init.py`. The code in the editor is as follows:

```
1 hello
2
3 hello+3
4
5 print(hello)
```

Line 5 is highlighted. To the right, a console window titled "PAUSE" displays a `NameError` traceback:

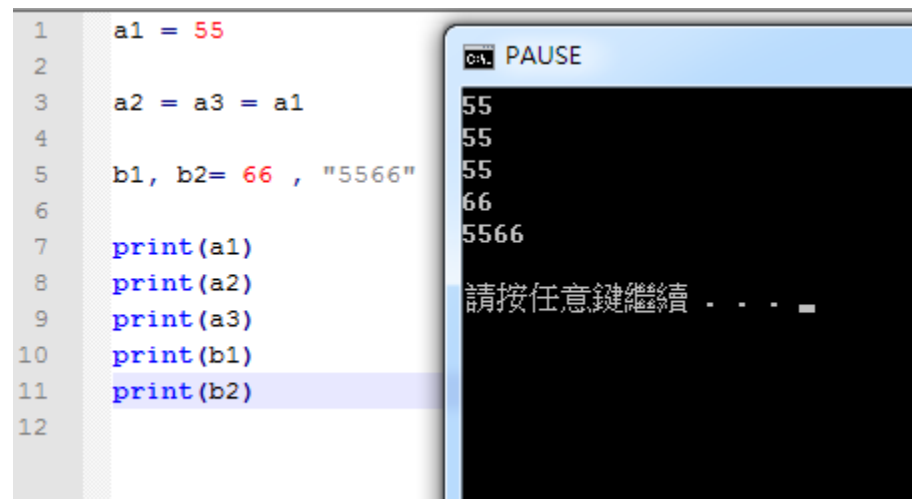
```
Traceback (most recent call last):
  File "E:\course_code\python_code\ch-3\init.py", line 1, in <module>
    hello
NameError: name 'hello' is not defined

請按任意鍵繼續 . . .
```

Assign(指派) 運算子 : “ = ”

Python 的 assign 跟別人不同

不同型態的變數可以藉「,」隔開，也可以偷懶在同一行中全部 assign



The image shows a Python code editor on the left and a terminal window on the right. The code in the editor is as follows:

```
1 a1 = 55
2
3 a2 = a3 = a1
4
5 b1, b2 = 66, "5566"
6
7 print(a1)
8 print(a2)
9 print(a3)
10 print(b1)
11 print(b2)
12
```

The terminal window on the right shows the output of the code, with a 'PAUSE' button at the top. The output is:

```
55
55
55
66
5566
請按任意鍵繼續 . . .
```

Assign(指派) 運算子 : “ = ”

垃圾回收機制 (Garbage collection)

```
a = [55, 66]
```

```
a = [77, 88]
```

```
print(a)
```

執行完 `a = [77, 88]` 後 , `[55, 66]` 會被當成垃圾回收機制的候選名單
主要用來清出更多的記憶體空間

運算式 運算子 運算元

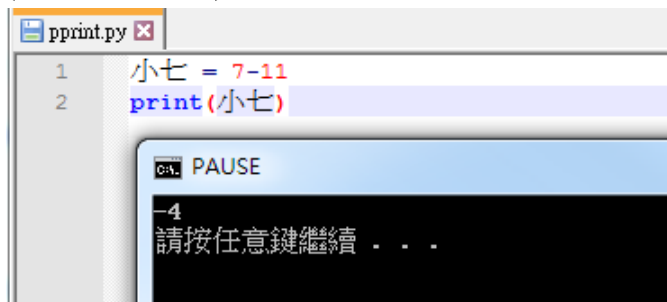
大部分的程式碼都是由判斷式及運算式組成的, Ex. $7-11 = -4$

運算式是由運算子及運算元所組成

運算子是運算的種類 (如 $+$, $-$, $*$, $/$, $=$ 等)

運算元是被用來運算的資料 (如 7, 11, 等)

變數也可以當運算元 (如 小七)



```
pprint.py x
1 小七 = 7-11
2  print(小七)

PAUSE
-4
請按任意鍵繼續 . . .
```

運算式 運算子 運算元

運算子可以分成：

算術運算子 (+ - * /)

邏輯運算子 (布林運算子)

比較運算子 (and, or, not)

指派運算子 (+= , -=, *=, =)

其他運算子 (:, ., (), [])

運算式 運算子 運算元

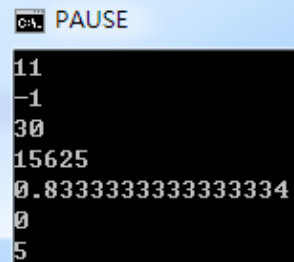
算術運算子：

運算子	function
+	加法
-	減法
*	乘法
**	指數
/	除法
//	除法 (商會取整數)
%	取餘數用 (Mod)

運算式 運算子 運算元

算術運算子：

```
1  x=5
2  y=6
3
4  c= x+y
5  print(c)
6  d= x-y
7  print(d)
8  e= x*y
9  print(e)
10 f= x ** y
11 print(f)
12 g= x/y
13 print(g)
14 h= x//y
15 print(h)
16 i= x%y
17 print(i)
18
19
20
```



```
Ctrl PAUSE
11
-1
30
15625
0.8333333333333334
0
5
```

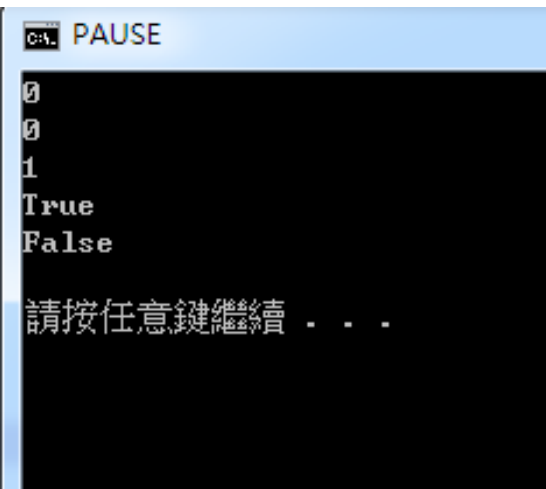
請按任意鍵繼續 . . .

運算式 運算子 運算元

邏輯運算子：

運算子	function
and	若 x 為假則回傳 x ；若 x 為真則回傳 y
or	若 x 為假則回傳 y ；若 x 為真則回傳 x
not	若 x 為假則回傳 True ；若 x 為真則回傳 False

```
1  x = 0
2  y = 1
3
4  print ( x and y)
5  print ( y and x)
6
7  print ( x or y)
8
9  print ( not x)
10 print ( not y)
11
```



```
PAUSE
0
0
1
True
False
請按任意鍵繼續 . . .
```


運算式 運算子 運算元

邏輯運算子：

條件式 運算子	x, y 皆真	x 假 , y 真	x 真 , y 假	x, y 皆假
x and y	y	x	y	x
x or y	x	y	x	Y
not x	False	True	False	True
not y	False	False	True	True

運算式 運算子 運算元

比較運算子：

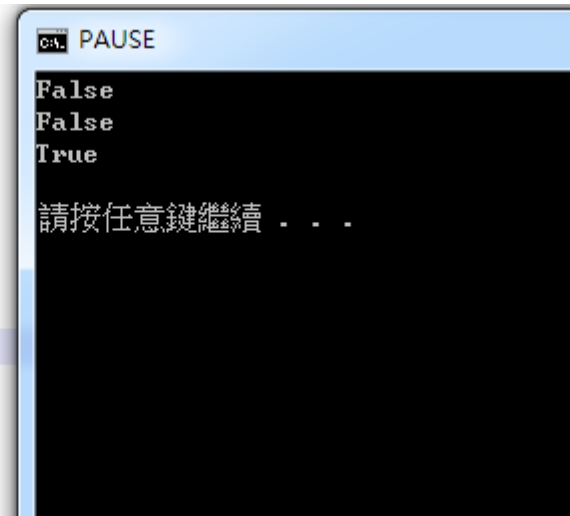
又稱為關係運算子，將兩個運算元的值拿來比較已得知兩者的關係
經過比較運算子運算後所得出的結果則是布林值

運算子	function
==	二個運算元是否相等
<(=)	二個運算元是否小於 (等於)
>(=)	二個運算元是否大於 (等於)
!=	二個運算元是否不相等
is	可判斷兩個元素是否為相同物件
in	判斷一個元素是否為一集合的元素

運算式 運算子 運算元

比較運算子：

```
1  x = 5
2  y = 6
3
4  c = (x == y)
5  print(c)
6
7  d = (x >= y)
8  print(d)
9
10 e = (x <= 5 <= y)
11 print(e)
```

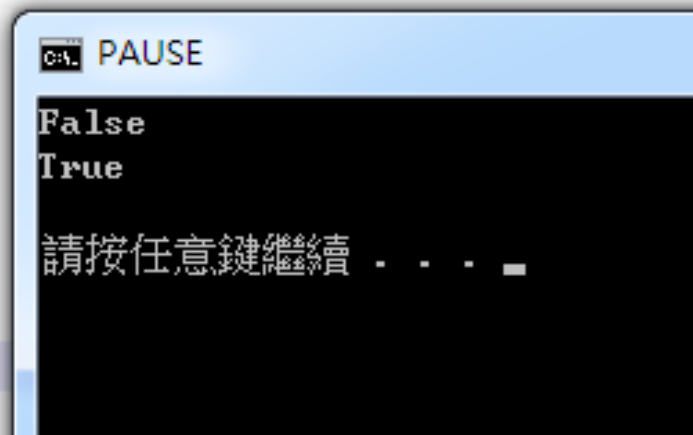


```
PAUSE
False
False
True
請按任意鍵繼續 . . .
```

運算式 運算子 運算元

比較運算子：(is)

```
1 x = (1,2,3)
2 y = (1,2,3)
3
4 c = ( x is y)
5 print(c)
6
7 x = y
8 d = ( x is y)
9 print(d)
```



運算式 運算子 運算元

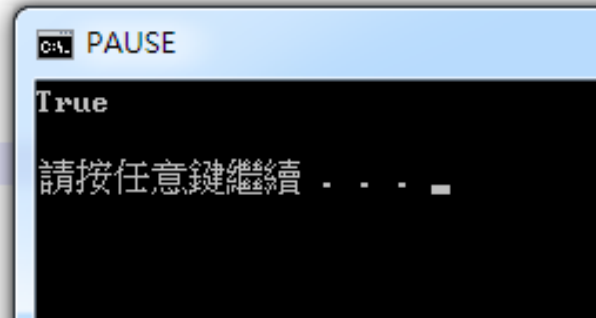
比較運算子：(is)

Special Case：

AI 跟 AI3 所指向的物件應該是不同的

Python 將簡單資料型態且其內容相同的不同變數指向同一個物件

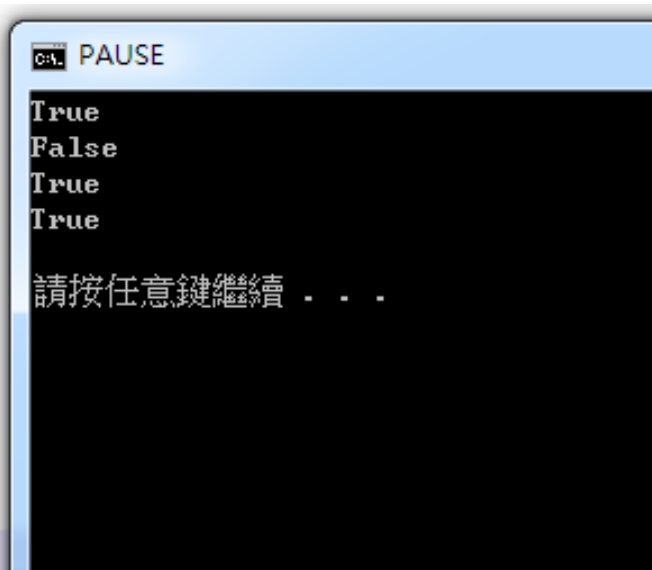
```
AI = "Allen Iverson"  
AI3 = "Allen Iverson"  
  
e = ( AI is AI3 )  
print(e)
```



運算式 運算子 運算元

比較運算子 : (in)

```
ai = "Allen Iverson"  
tmp1 = (1,2,3,4,5)  
tmp2 = ["5","6","7","8"]  
tmp3 = {'a','b','c'}  
  
f = ( "Iverson" in ai)  
print(f)  
g = ( 6 in tmp1 )  
print(g)  
h = ( "5" in tmp2 )  
print(h)  
i = ( 'a' in tmp3 )  
print(i)
```



```
Ctrl-C PAUSE  
True  
False  
True  
True  
請按任意鍵繼續 . . .
```

運算式 運算子 運算元

指派運算子：

將運算結果存在某個變數中，可以讓運算式變簡單

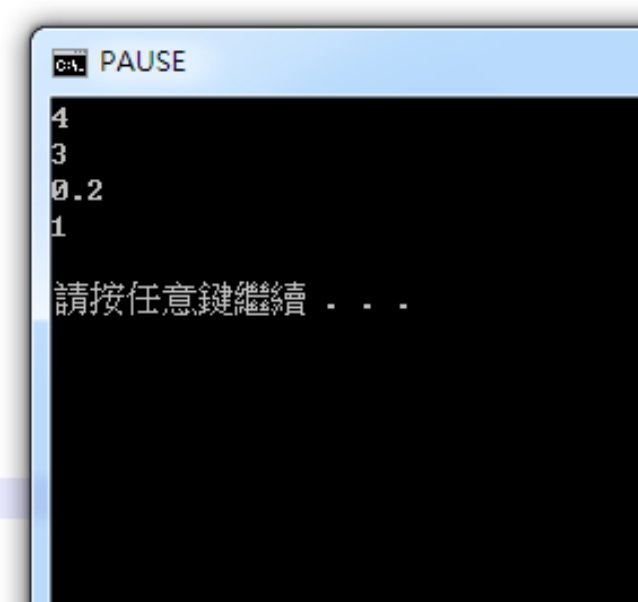
運算子	function
=	y 的值 assign 給 x
+ (-) =	x = x +(-) y
* (/) =	x = x *(/) y
%=	x = x % y

example
x = y
x += y, x -=y
x *= y, x /=y
x %= y

運算式 運算子 運算元

指派運算子：

```
1  #assign py
2
3  x = 5
4  y = 3
5
6  c = d = e = f = 1
7  c += y    # c = c+y
8  print (c)
9  d *= y    # d = d*y
10 print (d)
11 e /= x    # e = e/x
12 print (e)
13 f %= x    # f = f%x
14 print (f)
```



```
PAUSE
4
3
0.2
1
請按任意鍵繼續 . . .
```


運算式 運算子 運算元

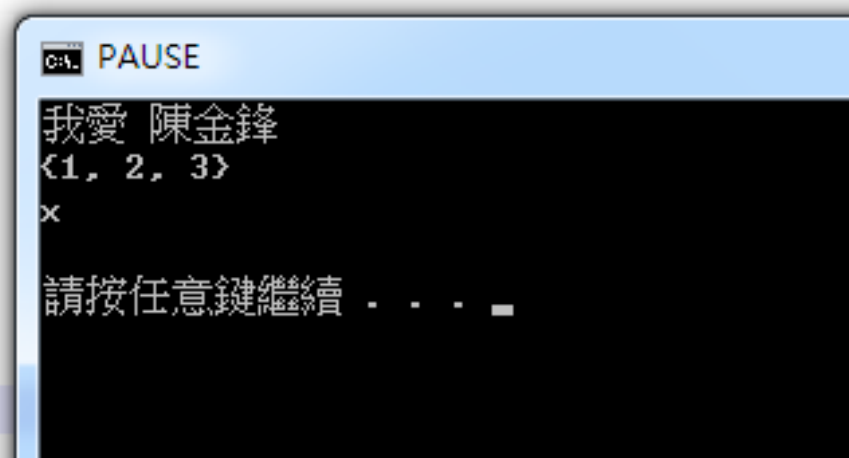
其他運算子：

運算子	function
,	分隔變數、資料集裡的元素等等
;	分隔運算式
.	存取類別、模組的方法或屬性
()	定義 tuple、函式 / 方法呼叫
[]	定義 list、序列形態的索引符號
{}	定義字典
:	控制條件後的分隔符號或辭典元素之配對

運算式 運算子 運算元

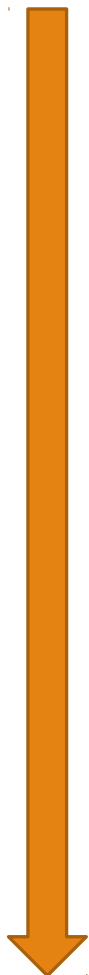
其他運算子：

```
1
2 x = (1, 2, 3)
3
4 CCF52 = "陳金鋒" ; Lin31 = "林智勝"
5
6 print("我愛", CCF52)
7
8 setsetset = {1, 2, 3}
9 print(setsetset)
10
11 dictdict = {1: "x", 2: "y"}
12 print(dictdict[1])
13
14
```



運算式 運算子 運算元

運算子優先順序：



運算子	說明
(...) 、 [...] 、 {...}	tuple 、 list 、 dict
a[i] 、 a[i:j] 、 a.b 、 a.b(...)	內含中小括號之呼叫
+k 、 -k 、 ~k	正負數及補數
a * b 、 a / b 、 a % b 、 a // b	乘法、除法、取餘數、整數除法
a + b 、 a - b	加法、減法
a << b 、 a >> b	位移運算
a & b	AND 位元運算
a ^ b	XOR 位元運算
a b	OR 位元運算

運算子	說明
<= 、 >= 、 == 、 <> 、 !	比較運算子
= 、 is 、 in	
not a	not 邏輯運算
a and b	and 邏輯運算
a or b	or 邏輯運算

隨堂練習

1. 查出今日新台幣換日元的匯率
2. 假設一台 PS4 是日幣 35000 元 (不含稅), 稅金是 13%
3. 阿宅準備了 10000 新台幣去換日元 (不考慮手續費)

請問他買的起嗎？

寫出一隻程式輸出 True / False !!!