Python

技術者們 實踐!

第 1 章
資料型別、變數及運算

1-0 資料的種類：型別

- 資料的種類稱為資料型別 (Data Type), Python 的資料型別, 比較簡單的有整數、浮點數、字串...等, 較為複雜的則有串列、字典、集合...等

一步一腳印

```
In [1]: 12+34
Out[1]: 46

In [2]: '12'+'34'
Out[2]: '1234'

In [3]: 12+'34'
Traceback (most recent call last):

  File "<ipython-input-3-cfd00a66fe73>", line 1, in <module>

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

整數相加時，會進行數值相加

字串是一串文、數字或符號，前後用 ' 號標起來，例如 'strihg_123'

字串相加時，會進行字串串接而非數值相加

整數和字串，Python 不知道要怎麼加 !!!

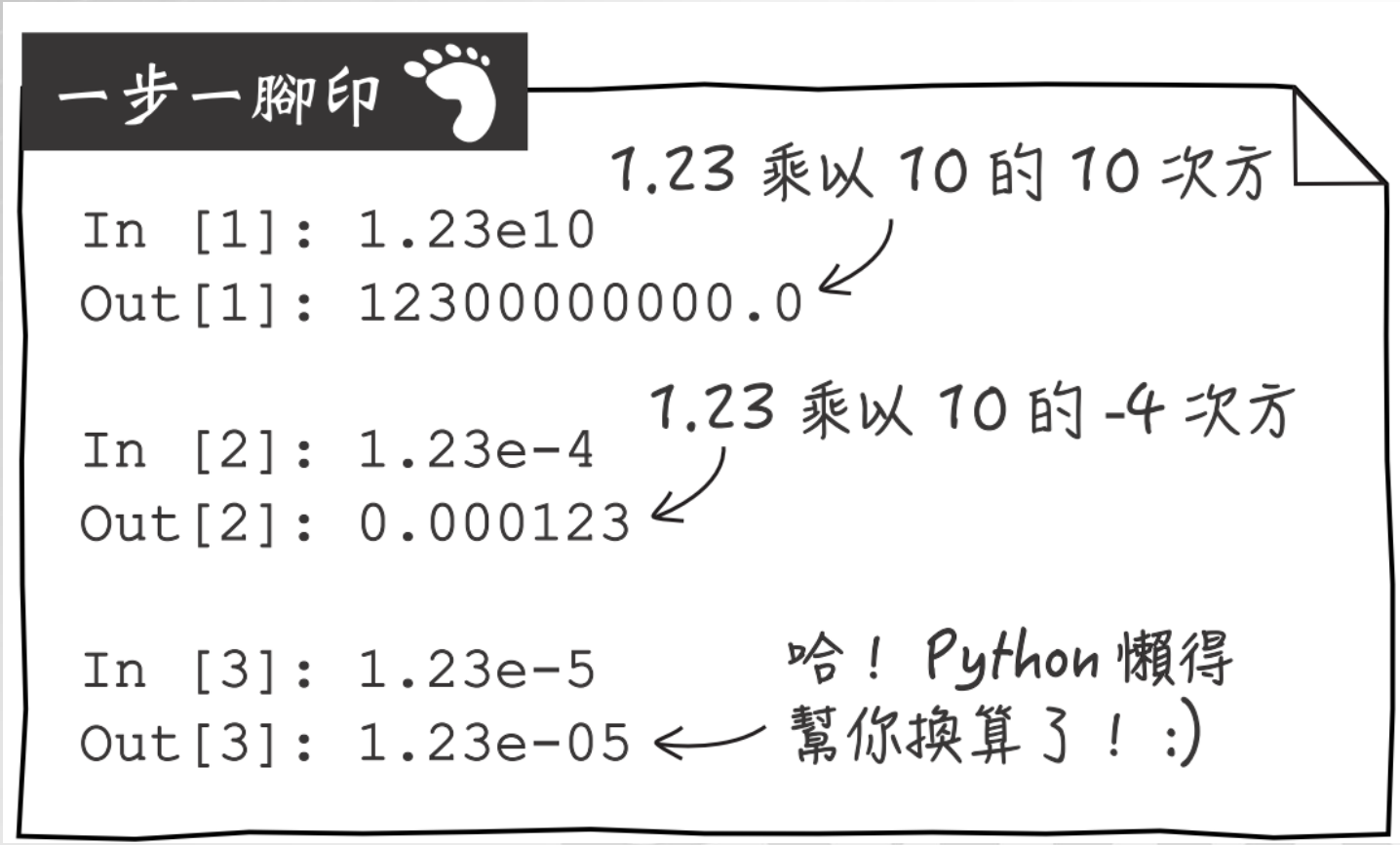
在 line 1, 12+'34' 這個地方發生

TypeError 型別錯誤，不支援 int 和 str 型別相加

最簡單的 4 種資料型別

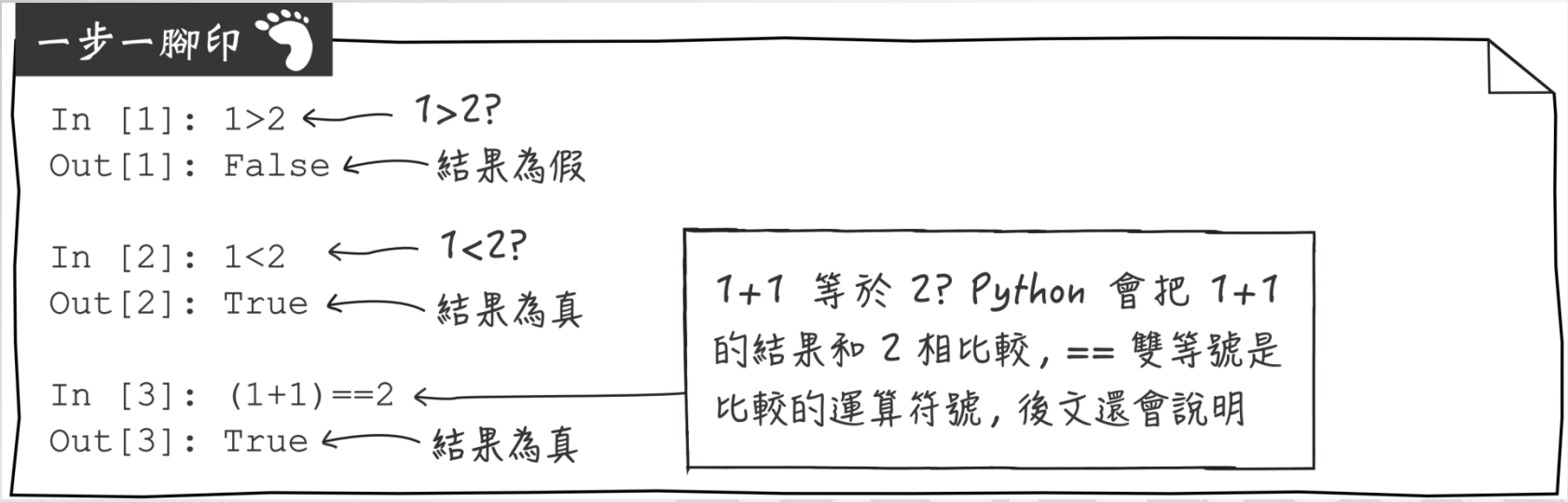
- 整數(int)
- 浮點數(float)
- 布林(bool)
- 字串(str)

整數 int 與浮點數 float



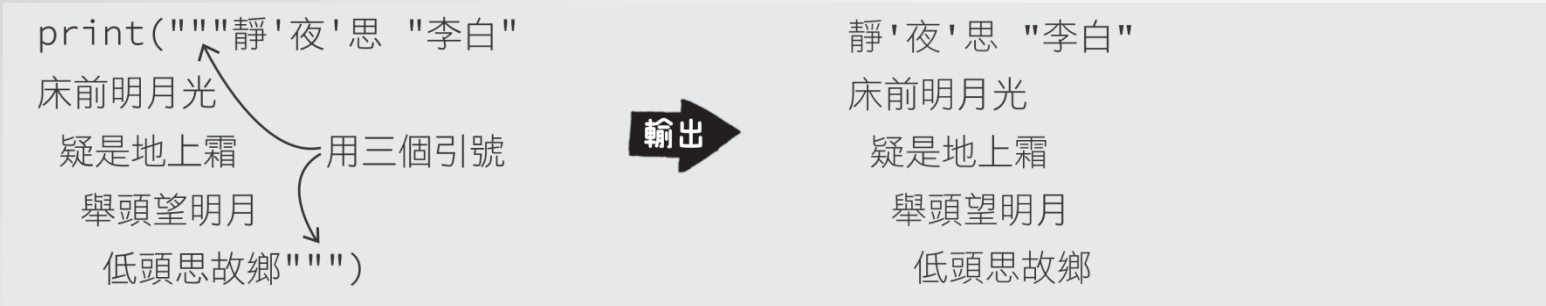
布林 bool 型別

- 布林 (bool) 只有兩個值，Python 內建為 True (真) 和 False (假)



字串 str 型別

- 字串 (str) 比較特別, 是由一連串的字元所組成
- 字串前後必須用單引號 ' 或雙引 " 號括起來
- 若是用 3 個引號 (""" 或 ''') 括住, 則字串中就可以含有單、雙引號



型別轉換

```
True + 1      # True 先自動轉為整數再做運算   輸出  2
1 + 2.1       # 1 先自動轉換為浮點數再做運算  輸出  3.1
```

一步一腳印

```
In [1]: 1+int('2') ← 將 '2' 轉為整數 2
Out[1]: 3

In [2]: 1+float('2') ← 將 '2' 轉為浮點數 2.0, 運算結果是浮點數 3.0
Out[2]: 3.0

In [3]: str(1.0)+'2' ← 將 1.0 轉為字串 "1.0", 結果是字串 '1.02'
Out[3]: '1.02'

In [4]: 2+int(True) ← 將 bool 型別 True 轉為整數 1, 結果是 3
Out[4]: 3

In [5]: 2+int('123') ← 將字串 '123' 轉為整數 123, 結果是 125
Out[5]: 125

In [6]: 2+int('hi guys')   ← 糟糕～爆掉了！int() 沒那麼厲害！它只能把外觀看起來是數字的字串（如：電話號碼）轉成整數
Traceback (most recent call last):

  File "<ipython-input-6-966474e0ae47>", line 1, in <module>
    2+int('hi guys')

ValueError: invalid literal for int() with base 10: 'hi guys'
                                                   'hi guys' 沒辦法轉啦!
```

自行用 IPython 試試看囉！

```
bool(0)      # 0 是空的                    輸出 False
bool(1.1)    # 1.1 不是空的                輸出 True
bool(None)   # None 是空的                 輸出 False
bool("")     # 空字串是空的                輸出 False
bool(" ")    # 內含空白字元，不是空的      輸出 True
bool("a")    # 不是空的                    輸出 True
```

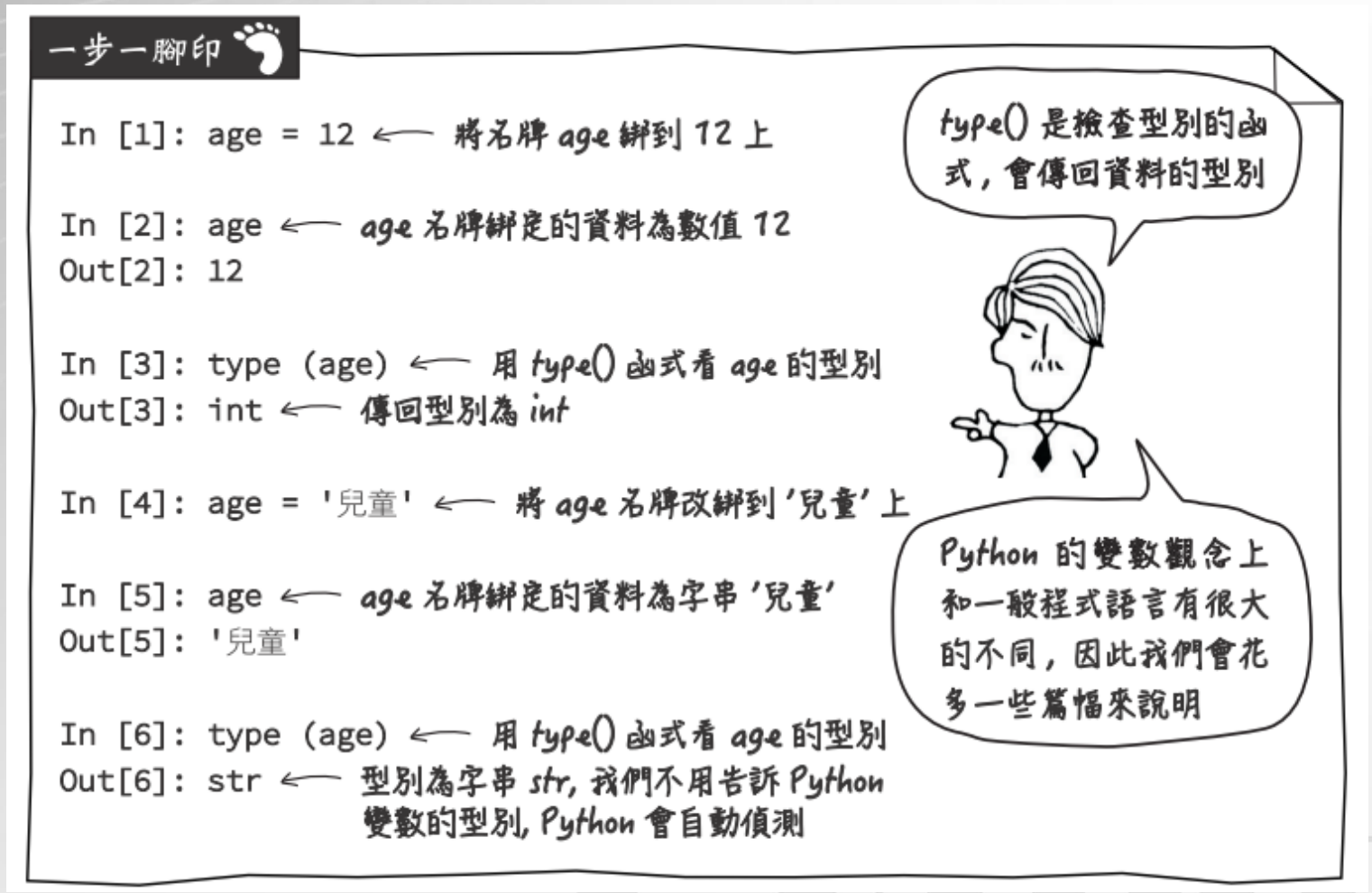
1-1 資料的名牌：變數

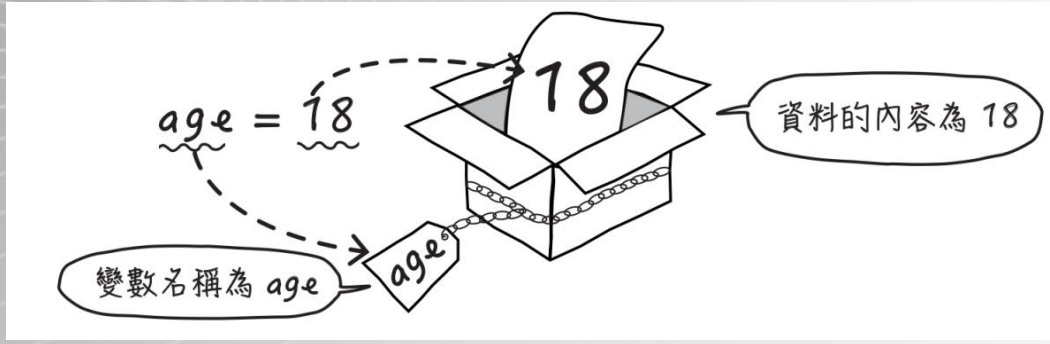
- 變數 (Variable) 可以讓我們重複使用資料

一步一腳印

```
In [1]: age=18      ← 將資料 18 命名為 age

In [2]: age+1       ← 重複使用 age
Out[2]: 19

In [3]: pi=3.14159  ← 圓周率好長 ㄛ

In [4]: r=1.23456   ← 半徑量到小數點 5 位

In [5]: 2*pi*r      ← 用變數命名就不用每次 key 到手酸還 key 錯！
Out[5]: 7.7569627008

In [6]: pi*r*r      ← 算圓的面積
Out[6]: 4.788217935949825
```

- Python 的變數其實是名牌 (Name tag)





一步一腳印

```
In [1]: age = 12 ⟵ 將名牌 age 綁到 12 上

In [2]: age ⟵ age 名牌綁定的資料為數值 12
Out[2]: 12

In [3]: type (age) ⟵ 用 type() 函式看 age 的型別
Out[3]: int ⟵ 傳回型別為 int

In [4]: age = '兒童' ⟵ 將 age 名牌改綁到 '兒童' 上

In [5]: age ⟵ age 名牌綁定的資料為字串 '兒童'
Out[5]: '兒童'

In [6]: type (age) ⟵ 用 type() 函式看 age 的型別
Out[6]: str ⟵ 型別為字串 str, 我們不用告訴 Python
               變數的型別, Python 會自動偵測
```

- 那麼名牌可以不綁在資料上嗎？當然可以！此時名牌的值即為「無」

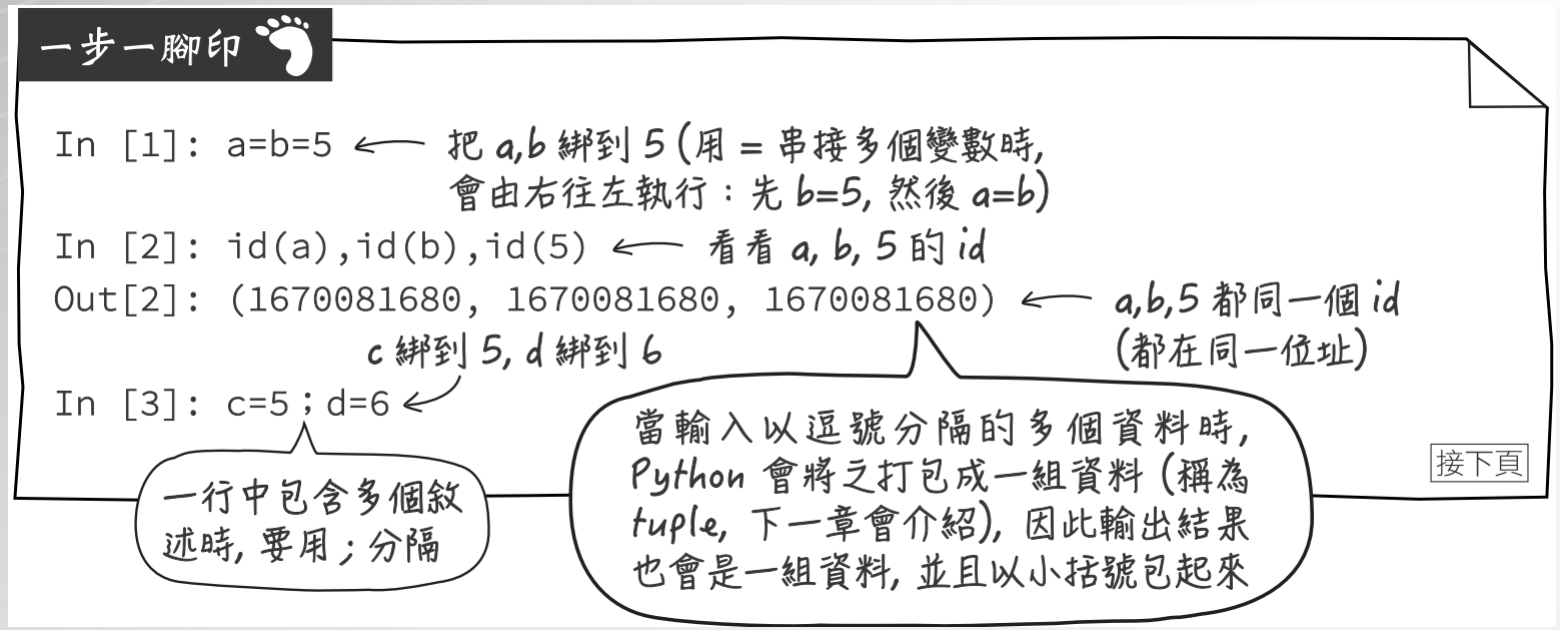
一步一腳印

```
In [1]: age=None ⟵ 建立名牌 age, 但不綁到資料上

In [2]: print(age)
None

In [3]: age=18 ⟵ 將名牌 age 綁到 18 上

In [4]: print(age)
18

In [5]: age=None ⟵ 再將名牌 age 由 18 上拿走並閒置
```

接下頁

```
In [6]: print(age)
None

In [7]: del age ←— 將名牌 age 刪除

In [8]: print(age)
Traceback (most recent call last):

  File "<ipython-input-8-5f7a7c5b2c60>", line 1, in <module>
    print(age)

NameError: name 'age' is not defined ←— 會發生名稱未定義的錯誤
```

- 沒有綁名牌的資料, 就表示無法再次使用了 (不能指名存取), 因此 Python 的環保車會不定時將之回收, 以便記憶體空間再利用：
- 介紹一個 Python 內建函式 id(), 這個函式會傳回變數或資料的 id, 我們可以把 id 想成是資料在記憶體的位址。


一步一腳印
In [1]: a=b=5
把 a,b 綁到 5 (用 = 串接多個變數時,
會由右往左執行：先 b=5, 然後 a=b)
In [2]: id(a),id(b),id(5)
看看 a, b, 5 的 id
Out[2]: (1670081680, 1670081680, 1670081680)
a,b,5 都同一個 id
(都在同一位址)
c 綁到 5, d 綁到 6
In [3]: c=5 ; d=6
一行中包含多個敘述時, 要用 ; 分隔
當輸入以逗號分隔的多個資料時, Python 會將之打包成一組資料 (稱為 tuple, 下一章會介紹), 因此輸出結果也會是一組資料, 並且以小括號包起來
接下頁

```
In [4]: id(c),id(d)
Out[4]: (1670081680, 1670081712)

In [5]: c='abcd'

In [6]: type(c)
Out[6]: str

In [7]: id(c),id(a)
Out[7]: (2084249107792, 1670081680)
```

看看 c, d 的 id

c 的位址和 a,b,5 一樣, d 綁到 6 所以不一樣

把 c 綁到 'abcd' 字串資料上

c 的型別變成字串 str 了

c 和 a 的 id (位址) 本來一樣現在不一樣了

1-2 變數命名規則

- 名稱中只能包括：數字、大小寫英文字及底線 (也可以用中文或他國文字如日、韓文等, 一般不建議), 請注意！變數名當中大寫和小寫字母是不一樣的
- 名稱開頭的第 1 個字不可為數字。
- 名稱不可是 Python 的保留字 (如下表)

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

一步一腳印

```
In [1]: print = 3 ⟵ 把 print 當變數名稱用

In [2]: print(1+2) ⟵ 哇！ print() 不能當函式用了！！！
Traceback (most recent call last):

  File "<ipython-input-2-3d60426f0d28>", line 1, in <module>
    print(1+2)

TypeError: 'int' object is not callable
```

正確的名稱	錯誤的名稱	錯誤原因
filename	9dogs	不可數字開頭
file_name	file-name	不可有符號 -
giveMe5	dog&cat	不可有符號 &
_8i	None	不可是保留字
姓名	print	不可是內建函式的名稱

- 變數名稱最好能夠自我詮釋 (Self-document)

風格範例	風格說明
filename	全部小寫
file_name	全部小寫但各單字以底線分隔
fileName	由第 2 個單字開始都首字大寫, 此方法又稱為**駝峰式大小寫** (Camel case)

1-3 運算式與算符

- 運算式就是「運算資料的式子」，例如「1+2」

指 (綁) 定算符 =

一步一腳印

```
In [1]: a=b=1
```
← 可以同時將多個名牌, 例如 a 和 b 綁在 1 上

```
In [2]: a=a+1
```
← 把 a 的內容 1 取出來加 1, 產生一個新資料, 再把 a 綁上去, 而不是直接把 1 改成 2

```
In [3]: a, b
Out[3]: (2, 1)
```
← 這裡有一個**重點**！就是現在 a 這個名牌已經綁到新資料 2 上面了, 至於名牌 b 仍然綁在資料 1 上面！

接下頁

```
In [6]: a,b,c=1,2,3
```
← 一次分別將 3 個名牌綁在 3 個資料上：a ▶ 1, b ▶ 2, c ▶ 3

```
In [7]: print(a,b,c)
1 2 3
```
← 顯示 a,b,c 的內容

用 = 做變數的初始化

一步一腳印

```
In [1]: a = 1          ⟵ a 有用 = 做初始化, 綁到整數 1 物件上

In [2]: a = a + 1      ⟵ 所以 a 可以開始使用, 沒問題

In [3]: b = b + 1      ⟵ b 沒有用 = 做初始化, 一使用就 NameError 了
Traceback (most recent call last):

  File "<ipython-input-3-5b4f6730ad03>", line 1, in <module>
    b = b + 1

NameError: name 'b' is not defined  ⟵ 'b' 這個變數名未定義 (not defined)
```

算數算符 + - * / 以及 // % **

一步一腳印

```
In [1]: 2*3.0
Out[1]: 6.0 ←— 整數與浮點數做算數運算時, 結果會是浮點數 (避免損失小數)
```

接下頁

```
In [2]: 4/2, 5/2
Out[2]: (2.0, 2.5) ← 使用除法時，無論是否整除結果都會是浮點數

In [3]: 5//2, 5%2
Out[3]: (2, 1) ← 用 // 及 % 計算整數除法的商及餘數，結果都是整數

In [4]: 4.4//2, 3.4%2
Out[4]: (2.0, 1.4) ← 浮點數除法的商及餘數，結果都是浮點數（商數的小數一定是 0）

In [5]: False+1, True/2 ← 還記得嗎？True 可以做為 1 而 False 可做為 0
Out[5]: (1, 0.5)

In [6]: 2%0, 2/0 ← 任何除法（/、//、%）都不可以除以 0，否則會出錯
Traceback (most recent call last):

  File "<ipython-input-6-46a8feee4f35>", line 1, in <module>
    2%0, 2/0
                ← 發生除數是 0 的錯誤
ZeroDivisionError: integer division or modulo by zero

In [7]: 4**0.5, 8**(1/3)
Out[7]: (2.0, 2.0) ← 次方若為小數，會變成開根號(如平方根、立方根等)
```

一步一腳印

```
In [1]:'Ab'+'12'+'3' ← 串接字串
'Ab123'

In [2]:'Ab'*3 ← 字串重複3次
'AbAbAb'
```


比較算符

- ① 大於(>)、大於等於(>=)
- ② 小於(<)、小於等於(<=)
- ③ 等於(==)、不等於(！=)

一步一腳印

```
In [1]: print(5 >= 5, 5 >= 6)
True False

In [2]: print(5 == 5, 5 != 6)
True True

In [3]: print(True ==1, False != 0)
True False
```

- 在比對字串時, 會由第 0 個字元開始往後比, 直到比出大小, 或是有字串先
- 結束為止 (先結束的比較小)

一步一腳印

```
In [1]: "abc" == "abc" ⟵ 完全相同
Out[1]: True

In [2]: "abc" > "Abc" ⟵ ASCII 碼小寫比大寫大
Out[2]: True

In [3]: "abc" < "b" ⟵ 第 0 個字 a 比 b 小
Out[3]: True

In [4]: "abc" < "abcd" ⟵ 前面相同, 較長的大
Out[4]: True

In [5]: "Eng" < "中文" ⟵ 中文一定大於英文
Out[5]: True
```

邏輯算符

- 如果運算元不是布林值呢？那麼就會檢
- 查資料是否為空的：若是空的就視為 False,
- 否則視為 True。空

運算式	運算結果
A **and** B	A 和 B **全部都為真**才是真, 否則為假
A **or** B	A 和 B **有一個為真**就是真, 否則為假
not A	A **為真則變假**, A **為假則變真**

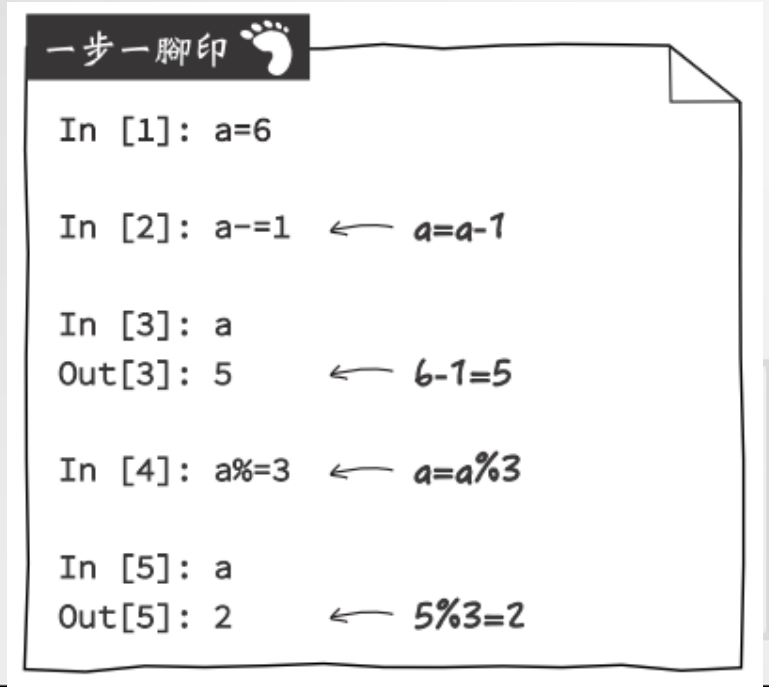
一步一腳印

```
In [1]: not 5 > 4 ←— 5>4 是 True, 所以 not Ture 就是 False
Out[1]: False

                  ↙ 5>4 和 4<3 是布林值 True 和 False, 結果是 . . . :
In [2]: (5 > 4 and 4 < 3
   ...: ) ←— 因為上一行右括號未輸入就按 Enter , Ipython 會等你輸入完成
Out[2]: False

In [3]: (5 > 4 or  4 < 3) ←— 5 >4 是 True, 4<3 是 False, 結果是 True
Out[3]: True
```

一步一腳印

```
In [1]: a = 4

In [2]: 3 < a < 6 ⟵ 是否 a 大於 3 小於 6
Out[2]: True

In [3]: 3 < a and a < 6 ⟵ 這是傳統寫法, 比較不夠直覺
Out[3]: True

In [4]: 3 < a < 6 > (a + 1) ⟵ 是否 a 大於 3 小於 6 且 6 大於 a+1
Out[4]: True

In [5]: 3 < a and a < 6 and 6 > (a+1) ⟵ 相當於傳統寫法的多個 and
Out[5]: True
```


- 複合指定算符
  - Python 提 供了一種更簡短的寫法：「a += n」。所有的算數算符 (+、-、*、/、//、%、**) 都可以這樣用，例如：



1-4 算符的優先順序

- 將常用算符的優先順序由高而低列出：

算符(由高到低)	說明
( )	小括號，若有多層括號則越內層越優先，例如 a*(b/(c+d)) 會最先算 (c+d)
**	次方
+x, -x	正、負號
*, /, //, %	乘除類的算符
+, -	加減法
<, <=, >, >=, !=, ==	關係算符
not	邏輯 not
and	邏輯 and
or	邏輯 or

```
print(3 + 2 > 5 / 2 and 7 * 8 != 6 + 7)          # 這看得懂嗎?
print((3 + 2) > (5 / 2) and (7 * 8) != (6 + 7))  # 可讀性增加了
```

1-5 Python 的內建函式

- 數值類函式

數值類函式	執行結果	功能
abs(-2.5)	2.5	取絕對值
min(1, 2)	1	取最小值, 參數可以有多個
max(1, 2, 3)	3	取最大值, 參數可以有多個
pow(2, 3)	8	2 的 3 次方
pow(2, 3, 5)	3	2 的 3 次方再除 5 取餘數
round(1.35, 1)	1.4	四捨六入到小數 1 位 (第 2 個參數表示要保留幾位小數)
round(1.35)	1	四捨六入到整數 (省略第 2 個參數時, 會進位到整數)