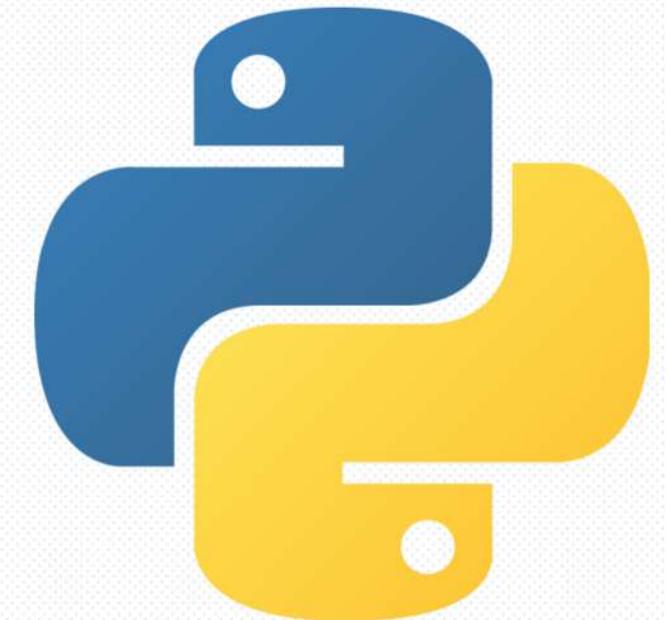


基礎程式設計 - 型態、變數與運算子

Data Type 、 Variable 、 Operator



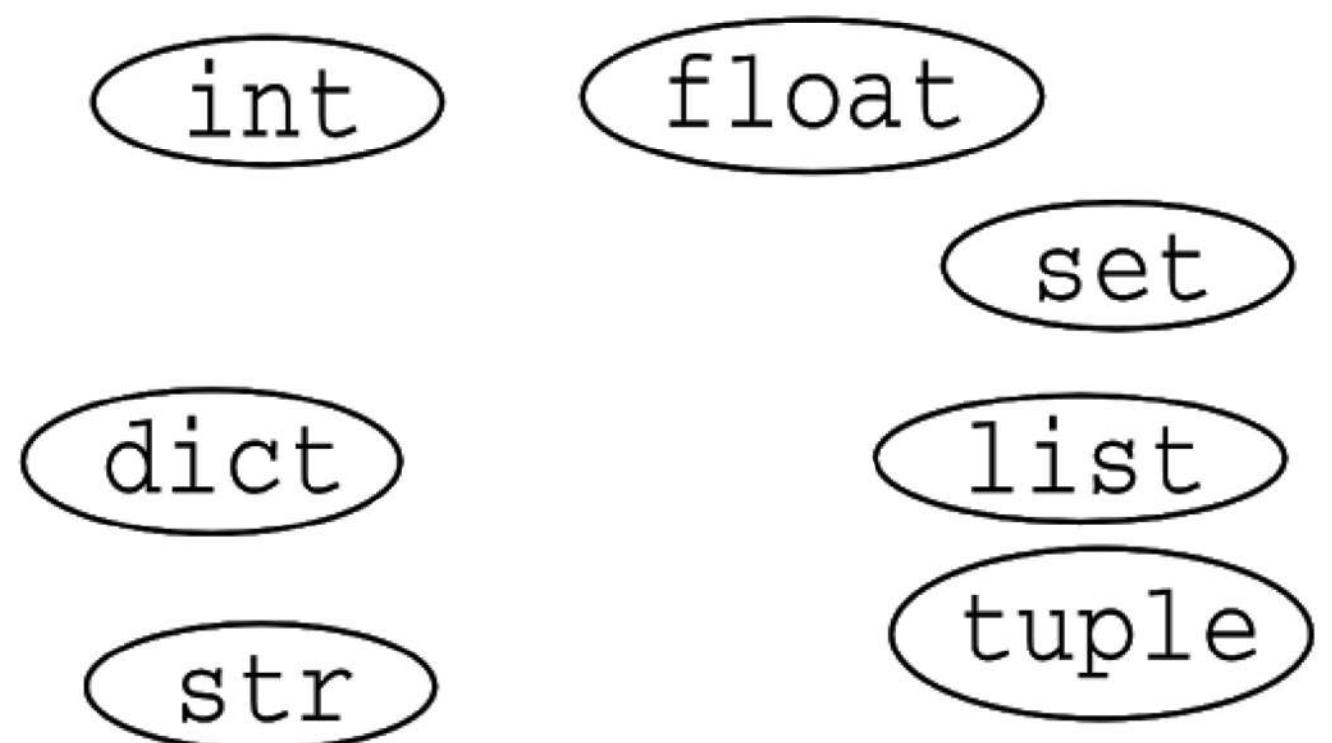
Python Fundamental



型態介紹

◆ Python 3 內建了許多資料型態，主要的資料型態如下：

- 數值型態 (numeric type) : **int**、**float**、**bool**
- 文字型態 (text sequence type) : **str**
- 容器型態 (sequence type) : **list**、**tuple**、**set**、**dict**





數值型態 - int (整數) & float (浮點數)

- ◆ int 型態表示整數 (integer)，例如 10、-100、10000。
- ◆ int 型態若加上千分位符號，例如 1,000 這樣的寫法，雖然可以執行，但是輸出結果不會是 1,000。

```
In [1]: print(1000)  
1000
```

```
In [2]: print(1000+2000)  
3000
```

```
In [3]: print(1,000)  
1 0
```

- ◆ float 型態表示浮點數 (float point number)，指的是有小數的部分，例如：3.14159、123.321。

```
In [1]: print(123.321)  
123.321
```

```
In [2]: print(123.321+123.321)  
246.642
```



數值型態 - bool (布林)

- ◆ bool 型態表示布林 (boolean) , 其內容有 True (真) 和 False (假) 兩種。
- ◆ bool() 函式會將 0 、 None 、 False 與「空的內容」轉換為 False , 其它則轉換為 True 。

```
In [1]: print(5>3)  
True
```

```
In [2]: print(5<3)  
False
```

```
In [3]: print(bool(0))  
False
```

```
In [4]: print(bool(0.0))  
False
```

```
In [5]: print(bool(0.1))  
True
```

```
In [6]: print(bool())  
False
```

```
In [7]: print(bool(None))  
False
```

```
In [8]: print(bool(True))  
True
```

```
In [9]: print(bool(False))  
False
```

```
In [10]: print(bool(()))  
False
```

```
In [11]: print(bool([]))  
False
```

```
In [12]: print(bool({}))  
False
```



文字型態 - str (字串)

◆ Python 使用字串(str) 處理文字、數字、符號等文字資料，以下列三種語法表示字串：

- 單引號 (')：例如 'Python程式設計'。(多數 Python 開發者的習慣)
- 雙引號 (")：例如 "Python程式設計"。
- 三個單引號 ('''')、三個雙引號 ("""")：這種語法允許多行字，中間的空白亦包含在內。

```
In [1]: print('it is a good day')
it is a good day
```

```
In [2]: print("it is a good day")
it is a good day
```

```
In [3]: print(''' today is Wednesday
...: it is a good day''')
today is Wednesday
it is a good day
```



跳脫字元(Escape character)

- ◆ 跳脫字元是指該字元具有特殊含義，必須特別對待。通常以反斜線「\」加上某一固定字元表示。
- ◆ Python 在遇到某些特定的字元組合時，會把它辨認成其他的意思(轉義)。常見的跳脫字元如下。

符號	說明
\\	反斜線。
\'	單引號，當你使用 '' 來表示字串，又要表示單引號時使用，例如 'Justin\'s Website'。
\"	雙引號，當你使用 "" 來表示字串，又要表示單引號時使用，例如 "\text\" is a string"。
\n	換行。
\t	Tab。

```
In [1]: print('C:\todo')  
C:      todo
```

```
In [2]: print('C:\\\\todo')  
C:\\todo
```

```
In [3]: print('it\' is')  
it' is
```

```
In [4]: print('\\\"it\\" is')  
"it" is
```

```
In [5]: print('is\\tit?')  
is      tit?
```

```
In [6]: print('is it ok?\nYes!')  
is it ok?  
Yes!
```



變數介紹與命名規則

- ◆ 變數是我們在程式中所使用一個有意義的名稱(name)，電腦會配置記憶體位置跟空間給這個名稱，然後我們可以來存放數值、字串、布林，稱為變數的值(value)，而且值可以重新設定或經由運算更改。
- ◆ 第一個字元可以是英文字母、中文或底線(_)，第二個字元之後可以是英文字母、數字、中文或底線。
- ◆ 英文字母有大小寫之分。
- ◆ 不能使用關鍵字，內建常數、內建函式、內建類別等的名稱。
- ◆ 在多數 Python 整合開發環境(IDE)中，都會以特殊顏色顯示關鍵字。

合法的變數名稱

```
studentID  
_studentID  
studentName  
student_name  
myCar1
```

不合法的變數名稱

```
class          # 不能使用關鍵字  
7eleven       # 不能以數字開頭  
my      Name   # 不能包含空白  
customer@ID    # 不能使用特殊符號@  
!myName        # 不能使用特殊符號！
```



變數指派方式「=」 - 賦值運算

- ◆ 「=」是賦值運算，代表將右邊運算的結果(右值)賦予(存到)左邊(左值)，跟數學上「相等」的意義有所不同。

left value = right value



- ◆ Python 屬於動態型態程式語言，變數在使用之前無須宣告型態。
- ◆ 我們使用「=」(指定運算子)來指定變數的值，而且是以最近一次指定的資料為準。輸入變數名稱可看到變數的值。

In [1]: myName='王大明' In [6]: x=y=z=123

In [2]: myName
Out[2]: '王大明'

In [7]: x
Out[7]: 123

In [3]: x=3

In [8]: y
Out[8]: 123

In [4]: x=x+1

In [9]: z
Out[9]: 123

In [5]: x
Out[5]: 4



變數刪除

- ◆ 刪除變數：`del` 變數

```
In [1]: A=1
```

```
In [2]: B=2
```

```
In [3]: C=3
```

```
In [4]: print(A)  
1
```

```
In [5]: print(B)  
2
```

```
In [6]: print(C)  
3
```

```
In [7]: del A
```

```
In [8]: del C
```

```
In [9]: print(A)  
Traceback (most recent call last):
```

```
File "<ipython-input-9-939330b76721>",  
      print(A)
```

```
NameError: name 'A' is not defined
```

```
In [10]: print(B)  
2
```

```
In [11]: print(C)  
Traceback (most recent call last):
```

```
File "<ipython-input-11-085b32debabd>",  
      print(C)
```

```
NameError: name 'C' is not defined
```



變數型態

◆ Python 的變數可以即時更換型態：

- C=5 → 從 5 推論型態是整數 int
- C='ABC' → 從 'ABC' 推論型態是字串 str
- C=1.234 → 從 1.234 推論型態是浮點數 float

◆ 我們可以使用 **type(變數名稱)** 函式知道變數的資料型態。

In [1]: X=3

In [2]: print(type(X))
<class 'int'>

In [3]: X='ABC'

In [4]: print(type(X))
<class 'str'>

In [5]: X=True

In [6]: print(type(X))
<class 'bool'>

In [7]: X=789.987

In [8]: print(type(X))
<class 'float'>



型態轉換

- ◆ 字串轉整數

```
In [1]: int('10')
Out[1]: 10
```

- ◆ 浮點數轉整數

```
In [2]: int(3.14)
Out[2]: 3
```

- ◆ 布林轉整數

```
In [3]: int(True)
Out[3]: 1
```

```
In [4]: int(False)
Out[4]: 0
```

- ◆ 將數值轉換為字串

```
In [1]: str(1.23)
Out[1]: '1.23'
```



運算子

- ◆ 「運算式」是由「運算元」和「運算子」組成。

運算式 = 運算元 + 運算子

Expression = Operands + Operators

- ◆ 運算子 (operator) 是用來進行運算的符號。
- ◆ 運算元 (operand) 是單元、元素，也就是運算子進行運算的對象。
- ◆ 運算式 (expression) 則是運算元與運算子組成的敘述。
- ◆ 右側的例子中， $3 * 6$ 是運算式，其中「*」為乘法運算子，「3」和「6」則為運算元。





運算子

◆ Python的運算子主要分為下列幾種類型：

- 算術運算子 (arithmetic operator) : + 、 - 、 * 、 / 、 // 、 % 等
- 比較運算子 (comparison operator) : > 、 < 、 >= 、 <= 、 != 等
- 邏輯運算子 (logical operator) : and 、 or 、 not 等
- 指派運算子 (assignment operator) : += 、 -= 、 *= 、 /= 、 >>= 、 %= 等



算術運算子

◆ + (加法) : $a + b$

```
In [1]: print(1+2)  
3
```

```
In [2]: print(1.5+2.3)  
3.8
```

◆ - (減法) : $a - b$

```
In [1]: print(15-3)  
12
```

```
In [2]: print(20-True)  
19
```

◆ * (乘法) : $a * b$

```
In [1]: print(12*21)  
252
```

```
In [2]: print('ABC'*2)  
ABCABC
```

◆ / (浮點數除法) : a / b

```
In [1]: print(6/3)  
2.0
```

```
In [2]: print(12/True)  
12.0
```

◆ // (整數除法) : $a // b$, 取商數

```
In [1]: print(8//3)  
2
```

```
In [2]: print(108//11)  
9
```

◆ % (餘數) : $a \% b$, 取餘數

```
In [1]: print(28%10)  
8
```

```
In [2]: print(12.5%10)  
2.5
```

◆ ** (指數) : $a ** b$, a 的 b 次方

```
In [1]: print(5**3)  
125
```

```
In [2]: print(3**5)  
243
```



比較(關係)運算子

- ◆ 比較(關係)運算子是用來比較兩個運算元的大小或相等與否，若是真的回傳 True，否則就回傳 False。

運算子	語法	說明	範例
>	$a > b$	若 a 大於 b ，就傳回 True，否則傳回 False	$18 + 3 > 18$ 會得到 True
<	$a < b$	若 a 小於 b ，就傳回 True，否則傳回 False	$18 + 3 < 18$ 會得到 False
\geq	$a \geq b$	若 a 大於等於 b ，就傳回 True，否則傳回 False	$18 + 3 \geq 21$ 會得到 True
\leq	$a \leq b$	若 a 小於等於 b ，就傳回 True，否則傳回 False	$18 + 3 \leq 21$ 會得到 True
\equiv	$a \equiv b$	若 a 等於 b ，就傳回 True，否則傳回 False	$21 + 5 \equiv 18 + 8$ 會得到 True
\neq	$a \neq b$	若 a 不等於 b ，就傳回 True，否則傳回 False	$21 + 5 \neq 18 + 8$ 會得到 False

```
In [1]: print(123=='123')
False
```

```
In [2]: print('ABC'=='abc')
False
```

```
In [3]: print(True==1)
True
```

```
In [4]: print(False==0)
True
```

```
In [5]: print((3>2)==True)
True
```

```
In [6]: print(3>2==True)
False
```



邏輯運算子

- ◆ 邏輯運算子可以用來進行邏輯運算，Python 提供如下的邏輯運算子。

- and 語法為 `a and b`，若兩者均為True就傳回True，否則傳回False。

```
In [1]: print((1>2) and (3>2))  
False
```

a	b	a and b
True	True	True
True	False	False
False	True	False
False	False	False

- or語法為 `a or b`，若兩者均為False就傳回False，否則傳回True。

```
In [2]: print((1>2) or (3>2))  
True
```

a	b	a or b
True	True	True
True	False	True
False	True	True
False	False	False

- not語法為`not a`，若a的值為True，就傳回False，否則傳回True。

```
In [3]: print(not(1>2))  
True
```

a	not a
True	False
False	True



指派運算子

- ◆ 指派運算子用來進行指派運算，以下為 Python 所提供的指派運算子。

運算子	範例	說明
=	a = b	將 b 指派給 a，也就是將 a 的值設定為 b 的值
+=	a += b	相當於 $a = a + b$ ，+ 為加法運算子， 也就是將 a 的值設定為 a 原來的值加 b 的值
-=	a -= b	相當於 $a = a - b$ ，- 為減法運算子
*=	a *= b	相當於 $a = a * b$ ，* 為乘法運算子
/=	a /= b	相當於 $a = a / b$ ，/ 為浮點數除法運算子
//=	a //= b	相當於 $a = a // b$ ，// 為整數除法運算子
%=	a %= b	相當於 $a = a \% b$ ，% 為餘數運算子
=	a **= b	相當於 $a = a ** b$ ， 為指數運算子



指派運算子

```
In [1]: x,y,z=10,20,30 # 將變數 x, y, z 設定為 10, 20, 30
```

```
In [2]: x*=y # 相當於 x=x*y
```

```
In [3]: print(x) # 顯示變數 x 的值  
200
```

```
In [4]: z%=5 # 相當於 z=z%5
```

```
In [5]: print(z)  
0
```



Python 程式碼註解

- ◆ Python 提供下列兩種註解符號：

- **#**：標示單行註解

```
# 我的第一個Python程式
print("Hello, World!")
```

- **''' ...xoooo..... '''**：標示多行註解

```
''' 我的第一個Python程式
    它將會印出Hello, World!
'''
```



輸入

- ◆ 我們可以使用 `input()` 函式取得使用者輸入的資料，語法如下：

`input('提示句')`

- ◆ 提示句是選擇性參數，用來設定提示文字。若省略不寫，表示沒有提示文字。
- ◆ 輸入下面第一行敘述，然後按「Enter」鍵，此時會出現提示文字「請輸入姓名：」。
- ◆ 輸入「Peter」，然後按「Enter」鍵，所輸入的姓名就會被指派給變數`name`。

```
In [1]: name=input('請輸入姓名：')  
[請輸入姓名 : Peter]
```

```
In [2]: print(name)  
Peter
```



輸入

- ◆ `input()`函式所輸入的值**均為字串**，若要將輸入的值進行運算時，可以使用`eval()`函式來處理。
- ◆ `eval()`函式：將 `input()`函式取得的字串轉換成數值型態(整數、浮點數或 `bool`)或是變數(需先設定)。
- ◆ 輸入範例：

```
PI = 3.14159
radius = eval(input("請輸入圓半徑："))
print("半徑為", radius, "的圓面積為", PI * radius * radius)
```

請輸入圓半徑：5

半徑為 5 的圓面積為 78.53975



輸出

- ◆ 大部份程式在執行完畢後會將結果輸出到螢幕。
- ◆ 使用內建的 `print()` 函式在螢幕上印出指定的字串，語法如下：

`print(value, value,....., sep = '空一格', end = '\n', file= sys.stdout)`

- `value`：這個參數用來設定要印出的值。若有多個值，中間以逗號隔開。
- `sep`：選擇性參數，用來隔開兩個值的字串，可省略不寫，表示採取預設空格。
- `end`：選擇性參數，用來印出最後一個值之後所要加上的字串，可省略不寫，表示採取預設值 '\n' (換行)。
- `file`：選擇性參數，用來輸出的裝置，可以省略不寫，表示採取預設值 `sys.stdout` (標準輸出，即螢幕)。



輸出

print(value, values ,..., sep = '空一格', end = '\n', file= sys.stdout)

- ◆ value(s)：必要參數，用來設定要印出的資料。若有多個資料，中間以逗號隔開，資料型態可以不同。

```
In [1]: print(1)  
1
```

```
In [2]: print(1,2.5,True,'xyz')  
1 2.5 True xyz
```

- ◆ sep：選擇性參數，用來隔開兩個值的字串，可省略不寫，表示採取預設空格。

```
In [1]: name='Bob'
```

```
In [2]: print('Hello',name,sep='***')  
Hello***Bob
```



輸出

print(value, values ,..., sep = '空一格', end = '\n', file= sys.stdout)

- ◆ end : 選擇性參數，用來輸出最後一個value之後所要加入的設定。若省略不寫，會採取預設值 '\n' (換行)。

```
1 print(123,23.56,True, "abc", sep=":::")
2 print("testa")
3
4 print(123,23.56,True, "abc", sep=":::", end="---")
5 print("testb")
6 print("testc")
```

```
In [1]: runfile('C:/Users/ASUS/.spyder-py3/temp.py',
               wdir='C:/Users/ASUS/.spyder-py3')
123:::23.56:::True:::abc
testa
123:::23.56:::True:::abc---testb
testc
```



格式化字串輸出(一)

- ◆ 和其它語言類似，Python 可使用 `string % data` 或 `string % (data1, data2, ...)` 的字串格式化方式。

In [1]: '哈囉！%s' % 'Python'

Out[1]: '哈囉！Python'

In [2]: '你目前的存款只剩%d元' % 666

Out[2]: '你目前的存款只剩666元'

In [3]: '%d除%d是%f' % (20,7,20/7)

Out[3]: '20除7是2.857143'

In [4]: '%5d除%5d是%.3f' % (20,7,20/7)

Out[4]: ' 20除 7是2.857'

In [5]: '%-5d除%-5d是%.3f' % (20,7,20/7)

Out[5]: '20 除7 是2.857'

In [6]: '%-8d除%-8d是%10.3f' % (20,7,20/7)

Out[6]: '20 除7 是 2.857'

符號	說明
%d	10 進位整數。
%f	10 進位浮點數。
%s	字串格式符號。



格式化字符串输出(二)

- ◆ Python 有自己的字符串格式化方式，语法如下：

'{格式化方式}'.format(数字/字符串)

```
In [1]: print('{:.2f}'.format(3.1415926))  
3.14
```

```
In [2]: print('{:10s}'.format('ABCDE'))  
ABCDE
```



格式化字符串輸出(二)

數字	格式	輸出	說明
3.1415926	{:.2f}	3.14	保留小數點後兩位
3.1415926	{:+.2f}	+3.14	帶正負符號保留小數點後兩位
-3		-3.00	
6.71828	{:.0f}	7	不帶小數
6	{:0>3d}	006	數字補零(填充左邊, 寬度為3)
10	{:x<5d}	10xxx	數字補x (填充右邊, 寬度為5)
1000000	{:,}	1,000,000	以逗號分隔的數字格式
18	{:10d}	18	右對齊(默認預設值, 寬度為10)
18	{:<10d}	18	左對齊(寬度為10)
18	{:^10d}	18	中間對齊(寬度為10)
0.35	{:.2%}	35.00%	百分比格式
2000000000	{:.2e}	2.00e+09	科學指數表示法



格式化字符串輸出(二)

- ◆ 將數值轉換為不同的進制。
- ◆ 可以使用如下字母來將數字轉換成字母代表的進制：

- **d**ecimal：十進位
- **b**inary：二進位
- **o**ctal：八進位
- **he****x**：十六進位

```
In [1]: print('{0:d}-{2:b}-{1:o}-{0:x}'.format(10,20,30))
10-11110-24-a
```



格式化字符串输出(二)

```
In [1]: print('{0:d} 除以 {1:d}是{2:f}'.format(10, 3, 10/3))  
10 除以 3是3.333333
```

```
In [2]: print('{0:5d} 除以 {1:5d}是{2:10.2f}'.format(10, 3, 10/3))  
10 除以 3是 3.33
```

```
In [3]: print('{n1:5d} 除以 {n2:5d}是{n3:.2f}'.format(n1=10, n2=3, n3=10/3))  
10 除以 3是3.33
```

```
In [4]: print('{n1:<5d} 除以 {n2:<5d}是{n3:.2f}'.format(n1=10, n2=3, n3=10/3))  
10    除以 3    是3.33
```

```
In [5]: print('{n1:>5d} 除以 {n2:>5d}是{n3:.2f}'.format(n1=10, n2=3, n3=10/3))  
10 除以 3是3.33
```

```
In [6]: print('{n1:!^5d} 除以 {n2:?^5d}是{n3:.2f}'.format(n1=10, n2=3, n3=10/3))  
!10!! 除以 ??3??是3.33
```

Q & A