



Python 3.8



# 蟒蛇

## Python 3.9

## Types



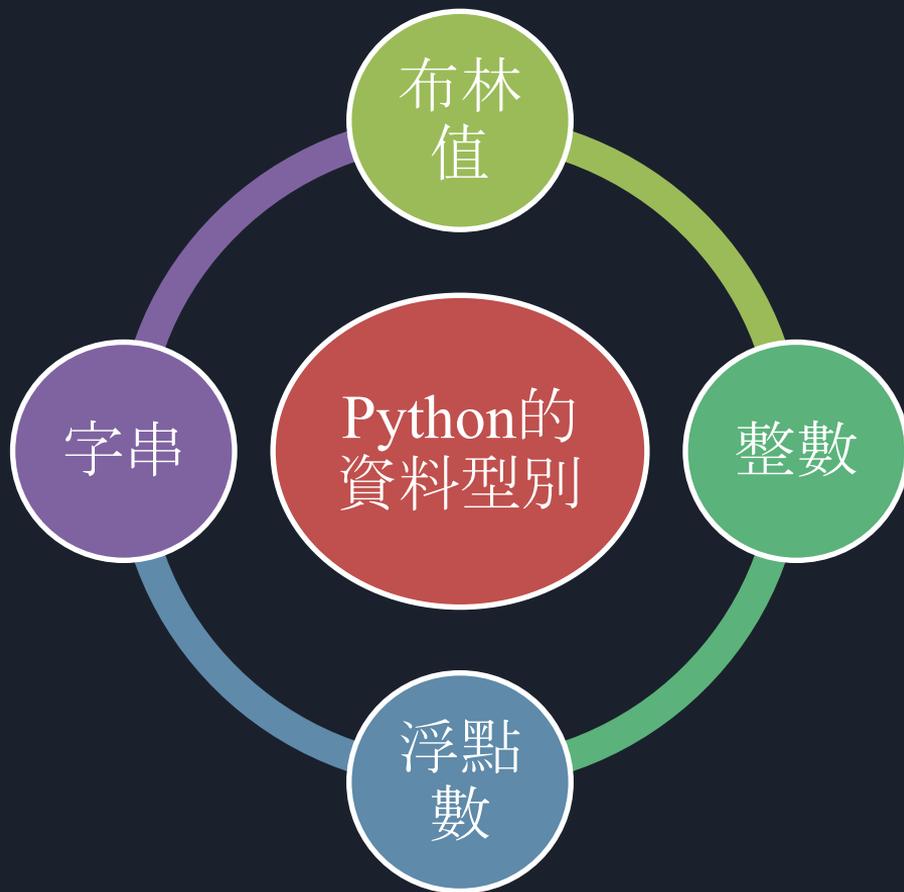
基礎程式設計

**Fundamental Programming**

-- **Python 3.8 Types**

Yung-Chen Chou Ph.D.  
iShool, Feng Chia University  
Aug. 18, 2021

# Python 資料型別





# Introduction

- Python 以 "物件" 來表示其 "識別字" 的身份、型別與值
- 什麼是物件?
  - 類別 (Class) → 製造物件的藍圖; 生物的胚胎
  - 屬性 (Attribute)
  - 方法 (Method)
  - 取用物件的屬性或方法使用的是「.」運算子



# 內建型別 (Built-in type)

- **數值型別** (Numeric types):
  - 整數 (int)、浮點數 (float)、複數 (complex)
- **序列型別** (Sequence types):
  - 字串 (str)、清單 (list)、元組 (tuple)
- **迭代型別** (Iterator type)
  - 透過容器並使用 for 迴圈做迭代操作
- **集合型別** (Set types)
  - 集合 (set)、frozenset
- **映射型別** (mapping type)
  - 字典 (dict)

# 整數 (Integer, int)

- Python 整數的數值範圍沒有限制，可以表示任意大的數字，只要能夠計算出來
  - 例如：「googol =  $10^{100}$ 」，「 $10^{100}$ 」表示10的100次方

**int()** function in Python and Python3 converts a number in given base to decimal.

## Syntax :

```
int(string, base)
```

## Parameter :

**string** : consists of 1's and 0's

**base** : (integer value) base of the number.

## Returns :

Returns an integer value, which is equivalent of binary string in the given base.

# 整數 (Integer)

```
>>> print( int( 110 ) )
110
>>> print( int( 3.1415926 ) )
3
>>> print( int( 4.98126 ) )
4
>>> print( int( '1101', 2 ) )
13
>>> print( int( '732', 8 ) )
474
>>> print( int( 'fe', 16 ) )
254
```

指前面資料 '1101' 是 **二進制** 資料，請 int() 函式將 資料轉換成十進制整數

指前面資料 '732' 是 **八進制** 資料，請 int() 函式將 資料轉換成十進制整數

指前面資料 'fe' 是 **十六進制** 資料，請 int() 函式將 資料轉換成十進制整數

# 整數 (Integer)

Python Main.py > ...

int 型別  
a 是個物件

```
a = 15
```

整數字

```
print(type(a))
```

使用 type() 函式查看  
a 這個物件的型別

PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE

```
(base) Yung-Chens-MacBook-Pro:1091-M1 yungchen$ /Us  
ace/1091-M1/Main.py
```

```
<class 'int'>
```

輸出

```
(base) Yung-Chens-MacBook-Pro:1091-M1 yungchen$
```

# 十進制、八進制、二進制、十六進制

```
Python Main.py > ...
int 型別 a = 255
2 print('{:08b}'.format(a, 'b'))
3 print('{:08o}'.format(a, 'o'))
4 print('{:08x}'.format(a, 'x'))
5

PROBLEMS   TERMINAL   OUTPUT   DEBUG CONSOLE

(base) Yung-Chens-MacBook-Pro:1091-M1 yungchen$ /User
ace/1091-M1/Main.py
11111111
00000377
000000ff
(base) Yung-Chens-MacBook-Pro:1091-M1 yungchen$
```

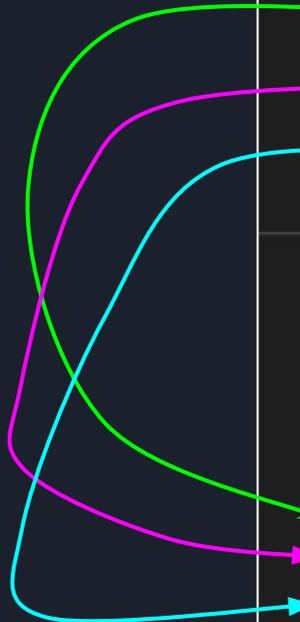
0是指有空位時補0  
8是有八位數  
b是二進制 (binary)

int 型別

b 為 二進制 (binary)

o 為 八進制 (octal)

x 為 十六進制 (Hexadecimal)





# 十進制、八進制、二進制、十六進制

十進制	二進制	八進制	十六進制
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12

# 十進制 $\Leftrightarrow$ 二進制

十進制  $\Rightarrow$  二進制

a=255

2		255	
2		127	... 1
2		63	... 1
2		31	... 1
2		15	... 1
2		7	... 1
2		3	... 1
		1	... 1

轉二進制後的結果 11111111

二進制  $\Rightarrow$  十進制

1	1	1	1	1	1	1	1
x	x	x	x	x	x	x	x
$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
128	+ 64	+ 32	+ 16	+ 8	+ 4	+ 2	+ 1 = 255

# 十進制 $\Leftrightarrow$ 八進制

十進制  $\Rightarrow$  八進制

a=255

$$\begin{array}{r|l} 8 & 255 \\ \hline 8 & 31 \quad \dots 7 \\ & 3 \quad \dots 7 \end{array}$$

轉八進制後的結果 377

八進制  $\Rightarrow$  十進制

$$\begin{array}{ccc} 3 & 7 & 7 \\ \times & \times & \times \\ 8^2 & 8^1 & 8^0 \\ \parallel & \parallel & \parallel \\ 64 \times 3 & + 8 \times 7 & + 1 \times 7 = 255 \end{array}$$

# 十進制 $\Leftrightarrow$ 16進制

十進制  $\Rightarrow$  16進制

a=255

$$\begin{array}{r|l} 16 & 255 \\ & 15 \quad \dots 15 \end{array}$$

轉 16 進制後的結果 FF

16進制  $\Rightarrow$  十進制

$$\begin{array}{r} F \\ \times \\ 16^1 \\ || \\ 16 \times 15 \end{array} \quad \begin{array}{r} F \\ \times \\ 16^0 \\ || \\ 1 \times 15 \end{array} = 255$$

Python Main.py > ...

1 a = 255

2 b = bin(a)

3 c = oct(a)

4 d = hex(a)

5 print(b)

6 print(c)

7 print(d)

8 print('{:08b}'.format(a, 'b'))

9 print('{:08o}'.format(a, 'o'))

10 print('{:08x}'.format(a, 'x'))

11

將 a 轉換成二進制, 輸出資料以 0b 開頭且 b 的型態為 str

將 a 轉換成八進制, 輸出資料以 0o 開頭且 c 的型態為 str

將 a 轉換成十六進制, 輸出資料以 0x 開頭且 d 的型態為 str

PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE

(base) Yung-Chens-MacBook-Pro:1091-M1 yungchen\$  
ace/1091-M1/Main.py

0b11111111

0o377

0xff

11111111

00000377

000000ff

(base) Yung-Chens-MacBook-Pro:1091-M1 yungchen\$

Python Main.py > ...

```
1 b = '0b11111110'
```

用字串(str)表示 二進制 值為 0b11111110, 放在變數 b

```
2 c = '0o373'
```

用字串(str)表示 八進制 值為 0o373, 放在變數 c

```
3 d = '0xef'
```

用字串(str)表示 十六進制 值為 0xef, 放在變數 d

```
4 print('{}'.format(int(b, 2)))
```

```
5 print('{}'.format(int(c, 8)))
```

```
6 print('{}'.format(int(d, 16)))
```

int(str, base) 函式是用來將字串依, base 的數位進制進行轉換。這個範例是將 b 這個二進制值 0b11111110 以 2 進制進行轉換成 int 就變成 254

PROBLEMS

TERMINAL

OUTPUT

DEBUG CONSOLE

```
(base) Yung-Chens-MacBook-Pro:1091-M1 yungchen$  
ace/1091-M1/Main.py  
254  
251  
239
```

# 另一個整數 Boolean (布林)

- boolean 布林變數只有兩種值, True 與 False
- 布林除了值為 True 或 False 之外也可以是數字。當數值為 0 時是 False, 其他的值都是屬於 True

將 a 的值轉換成 布林值, 所以 b 的型態是 boolean, 其所帶的值是 True, 當把 b 拿來計算時因為是 True 所以數字的值是 1

```
Python Main.py > ...
1 a = 3
2 b = bool(a)
3 c = b + 3
4 print(bool(a))
5 print(type(a))
6 print(type(b))
7 print(b)
8 print(c)
```

PROBLEMS    TERMINAL    OUTPUT

```
(base) Yung-Chens-MacBook-Pro:
ace/1091-M1/Main.py
True
<class 'int'>
<class 'bool'>
True
4
```

# Boolean (布林)

## Falsy Values

### Sequences and Collections:

- Empty lists `[]`
- Empty tuples `()`
- Empty dictionaries `{}`
- Empty sets `set()`
- Empty strings `""`
- Empty ranges `range(0)`

### Numbers

- Zero of any numeric type.
- Integer: `0`
- Float: `0.0`
- Complex: `0j`

### Constants

- `None`
- `False`

## Truthy Values

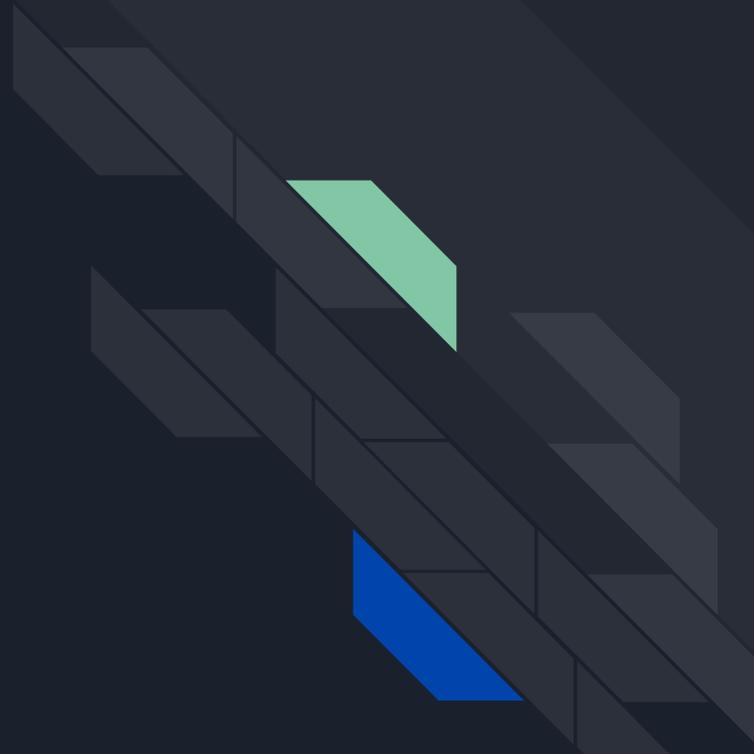
According to the [Python Documentation](#):

By default, an object is considered true.

### Truthy values include:

- Non-empty sequences or collections (lists, tuples, strings, dictionaries, sets).
- Numeric values that are not zero.
- `True`

# 浮點數 (Float)





# 浮點數 (Float)

- 含有小數位數的數值，也稱為「實數」
- 三種資料型別
  - **float**: 儲存 倍精準 浮點數，**精確程度**則依執行環境的**作業系統而定**
  - **complex**: 處理 複數數值資料之用，由**實數**與**虛數**組成
  - **decimal**: 非內建，數值要有精確小數位數，需自標準函式庫 decimal.Decimal 類別支援

```
Python Main.py > ...
1 total = 2.0 - 1.4
2 a = '0xff.ff'
3 b = 15.1
4 c = b.hex()
5 print(total)
6 print(type(total))
7 print(float.fromhex(a))
8 print(c)
9 print(float.fromhex(c))
```

b 是浮點數字, 透過 b.hex() 可以得到 b 的值轉換成 16 進制後的結果放到 c

將 16 進制 0xff.ff 放到變數 a, 透過 float.fromhex(a) 類別函數就可以將 16 進制字串轉回 十進制的值了

印出 c 的值 16 進制)

c 是 16 進制的值, 透過 float.fromhex(c) 將 c 的值轉換回 float 數字

PROBLEMS TERMINAL OUTPUT DEBUG CONS

```
(base) Yung-Chens-MacBook-Pro:1091-M1 yungche
ace/1091-M1/Main.py
0.6000000000000001
<class 'float'>
255.99609375
0x1.e3333333333333p+3
15.1
```



# Decimal

- 更精確的含小數位數的數值資料
- 10/3 用浮點數呈現的小數點位數較少，decimal 可以讓計算結果的小數位數更精準

```
Python Main.py > ...
1 import decimal
2 v1 = 10/3
3 v2 = decimal.Decimal(10/3)
4 print(v1)
5 print(v2)
```

PROBLEMS    TERMINAL    OUTPUT    DEBUG CONSOLE

```
(base) Yung-Chens-MacBook-Pro:1091-M1 yungchen$ /Users/yungchen/1091-M1/Main.py
3.3333333333333335
3.333333333333333481363069950020872056484222412109375
```

Python Main.py > ...

```
1 import decimal
2 import math
3 v1 = 10/3
4 v2 = decimal.Decimal(10/3)
5 print(v1)
6 print(v2)
7 print(round(v2, 3))
8 print(math.floor(v2))
9 print(math.ceil(v2))
```

小數點後第3位四捨五入

無條件捨去法取整數

無條件進位法取整數

PROBLEMS TERMINAL OUTPUT DEBUG CONSOLE

```
(base) Yung-Chens-MacBook-Pro:1091-M1 yungchen$ /Users/
ace/1091-M1/Main.py
3.3333333333333335
3.333333333333333481363069950020872056484222412109375
3.333
3
4
```