
Python

串列基本 List Basic

臺北科技大學資訊工程系

內建資料型別

□ 序列資料型別 (sequence types) 共有六種

- str: 字串 (string) , 不可變 (immutable)
- bytes: 字節 (byte) , 不可變 (immutable)
- bytearray: 字節陣列 (byte array) , 可變 (mutable)
- list: 串列 , 元素有順序、可修改、可重複
- tuple: 序對 , 元素有順序、不可修改、可重複
- range: 內建函數 range() 回傳的物件 (object) , 常用於 for 迴圈 (for loop)

內建資料型別

□ Collection 資料型別

- list 串列
- tuple 序對
- set 集合: 元素無順序、可修改、無重複、無索引 indexed
- dictionary (dict) 字典: 元素無順序、可修改、無重複、有索引

序對(Tuple)

□ Tuple 的操作

- 元素有順序關係，可以有不同資料型別的元素
- 元素允許是 tuple
- 使用小括號()，索引位置可以存取元素
- 唯讀 **不可變更** 的資料結構，**不可取代** tuple 中任一元素

```
f = (2, 'a', 4, 5)
g = ()
h = (2, [3, 4], (10, 11, 12))
x = f[1]
y = f[1:3] # f[1], f[2] 也就是一到三，不包含三
z = h[2][1]
print(f)
print(g)
print(x)
print(y)
print(z)
```

| |
|----------------|
| (2, 'a', 4, 5) |
| () |
| a |
| ('a', 4) |
| 11 |

Tuple建構

❑ 造出

```
def test01():  
    thistuple = ("apple",) # 後面需加逗號,  
    print(type(thistuple))  
    thistuple = ("apple") # 不是tuple，是字串  
    print(type(thistuple))
```

test01()

```
<class 'tuple'>  
<class 'str'>
```

❑ Join Tuples (use + operator)

```
def test02():  
    tuple1 = ("a", "b", "c")  
    tuple2 = (1, 2, 3)  
    tuple3 = tuple1 + tuple2  
    print(tuple3)
```

test02()

```
('a', 'b', 'c', 1, 2, 3)
```

❑ **tuple()** 序對建構子 Constructor

```
def test03():  
    thistuple = tuple("apple", "banana", "cherry") # note the double round-brackets  
    print(thistuple)
```

test03()

```
('apple', 'banana', 'cherry')
```

Tuple建構

- Tuples 是 unchangeable，不能移除元素，能刪除整個 tuple

```
thistuple = ("apple", "banana", "cherry")  
thistuple[3] = "orange"      # This will raise an error
```

```
def test03():  
    fruit = ('apple','banana','cherry')  
    del fruit  
    print(fruit)      #沒有這個變數
```

- 判斷是否存在

```
def test02():  
    thistuple = ("apple", "banana", "cherry")  
    if "apple" in thistuple:  
        print("Yes, 'apple' is in the fruits tuple")  
  
test02()
```

Yes, 'apple' is in the fruits tuple

Tuple基本存取

□ 存取一個元素

```
def test01():  
    thistuple = ("apple", "banana", "cherry")  
    print(thistuple[1])
```

test01() banana

□ Range of indexes (-1 最後一個, -2 最後第二個)

```
def test02():  
    thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")  
    print(thistuple[2:5]) # 二到五，不包含五
```

0 1 2 3 4 5 6

test02() ('cherry', 'orange', 'kiwi')

Tuple基本方法

○ len() method

```
def test02():  
    thistuple = ("apple", "banana", "cherry")  
    print(len(thistuple))
```

3

○ count() – 回傳特定值的個數

```
def test02():  
    thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)  
    x = thistuple.count(5)  
    print(x)
```

2

○ index() – 搜尋特定值回傳(第一次出現位置)索引

```
def test02():  
    thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)  
    x = thistuple.index(8)  
    print(x)
```

3

List 基本運算

- List 定義使用“中括號”，內不限定放數字或文字

- `my_list = [1, 2, 3, 4, 5, 'e', 'w', 'f']`

- List 有順序

- `print(my_list[0])` # 0為位置，第一項

- `print(my_list[-1])` # 倒數第1項

| |
|---|
| 1 |
| f |

- `len()` 計算 list 長度

- `sum()` 計算 list 中所有數值加總(但 list 中元素都須數值)

- `count` 計算 list 中某個元素出現次數

- `sort()`：將串列進行排序

- `reverse = True` 意思為反轉排列

- 不同型別排序，會出現錯誤

索引值範圍

- 回傳 2 (included) to 5 (excluded)

```
def test01():  
    thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]  
    print(thislist[2:5])
```

test01()

['cherry', 'orange', 'kiwi']

- 回傳 -4 (included) 到 -1 (excluded)

```
def test02():  
    thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]  
    print(thislist[-4:-1])
```

test02()

['orange', 'kiwi', 'melon']

-1

List 基本運算

□ List 操作

```
def testList01():  
    list1 = ['cat', 'tiger', 'dog']  
    list2 = [2, 3, 2]  
    print(list1[0])  
    print(list2[-1])  
    print(len(list1))  
    print(sum(list2))  
    print(list1.count('cat'))  
    print(list2.count(2))  
    list1[0] = 'bigCat' #改變某一個元素值  
    print(list1)
```

```
cat  
2  
3  
7  
1  
2  
  
['bigCat', 'tiger', 'dog']
```

```
list1 = [59, 33, 77, 62, 101, 243, 189, 5]  
list1.sort()  
print(list1)  
list1.sort(reverse = True)  
print(list1)
```

```
[5, 33, 59, 62, 77, 101, 189, 243]
```

```
[243, 189, 101, 77, 62, 59, 33, 5]
```

List 操作 - 新增

- ❑ + : 兩個 List 加在一起
- ❑ insert(object) : 在指定位置插入新物件元素
- ❑ append(object) : 在最後加入物件元素
- ❑ extend(object) : 逐一取出object內元素，加在最後
list
- ❑ nums = (10, 5, 7, 1, 6, 2)
 - tuple不提供排序的方法，若要排序(由小到大)該怎辦？
 - 轉成 list

```
f = (5, 7, 11, 23, 45, 2)
g = list(f)
print(g)
g.sort()
print(g)
```

```
[5, 7, 11, 23, 45, 2]
```

```
[2, 5, 7, 11, 23, 45]
```

list 操作 - 新增

□ 新增

```
def testList02():  
    list1 = ['cat']  
    list2 = [2, 3, 2]  
    list1.append('dog')  
    print(list1)  
    list1.insert(1, 'ant')  
    print(list1)  
    list1.extend('ti')  
    print(list1)  
    list3 = list1 + list2  
    list4 = [list1, list2]  
    print(list3)  
    print(list4)  
    list2.append(list1)  #list 的 list  
    print(list2)
```

['cat', 'dog']

['cat', 'ant', 'dog']

['cat', 'ant', 'dog', 't', 'i']

['cat', 'ant', 'dog', 't', 'i', 2, 3, 2]

[['cat', 'ant', 'dog', 't', 'i'], [2, 3, 2]]

[2, 3, 2, ['cat', 'ant', 'dog', 't', 'i']]

list 操作 - 刪除

- ❑ list.pop() # 刪除末尾
- ❑ list.pop(X) # 刪除指定位置的內容
- ❑ remove(item) : 刪除指定項

```
def testList():  
    list1 = ['ant', 'cat', 'dog', 'tiger', [1, 2, 'mouse']]  
    print(list1[2])  
    print(list1[4])  
    print(list1[4][2])  
    list1.remove('dog')  
    print(list1)  
    list1.pop(0) # 移除第一個  
    print(list1)  
    list1.pop(1) # 移除第二個  
    print(list1)
```

```
testList()
```

```
dog  
[1, 2, 'mouse' ]  
mouse  
  
['ant', 'cat', 'tiger', [1, 2, 'mouse']]  
  
['cat', 'tiger', [1, 2, 'mouse' ]]  
  
['cat', [1, 2, 'mouse']]
```

list 操作 – 切片

❑ 切割(slice) List

- 語法為[start : end : step]
- my_list[1:3] # print 1 ~ 2 項
- my_list[: -1] # print 0 ~ 倒數第2項
- step:(optional), 間隔取物, 如為負數則從最後元素反向取物

```
list = ['a', 'b', 'c']  
print(list[0:2])  
print(list[:2]) # 留白表示從頭到底  
print(list[: -2]) # - 代表反向 2 代表間距  
print(list[: -1])
```

```
['a', 'b']  
['a', 'c']  
['c', 'a']  
['c', 'b', 'a']
```

❑ 將串列複製到一個獨立串列: copy() 函式

❑ 轉換函式: list()

```
a = [1, 2, 3, 4]  
b = a.copy()  
c = list(a)  
d = a[:]
```

```
['1', '2', '3', '4']  
['1', '2', '3', '4']  
['1', '2', '3', '4']  
['1', '2', '3', '4']
```

list 操作 – 切片

```
def myList():
```

```
    my_list = []
```

```
    my_list.append(1)
```

```
    my_list.append(2)
```

```
    my_list2 = [5.5, 22, 'Hi', 3, 99, 22, 66]
```

```
    my_list2[0] = 3.3
```

```
    print(len(my_list), sum(my_list), my_list2.count(22))
```

```
    print(my_list2[0])
```

```
    print(my_list2[2])
```

```
    print(my_list2[-1])
```

```
    print(my_list2[:-1])
```

```
    print(my_list2[1:4])
```

```
    print(my_list2[2:3])
```

```
    print(my_list2[2:])
```

```
    list2 = my_list + my_list2
```

```
    list3 = my_list * 2 # 重複二次
```

```
    print(list2, list3)
```

```
    print(list2 == list2[:])
```

```
2 3 0
```

```
3.3
```

```
Hi
```

```
66
```

```
[3.3, 22, 'Hi', 3, 99, 22]
```

```
[22, 'Hi', 3]
```

```
['Hi']
```

```
['Hi', 3, 99, 22, 66]
```

```
[1, 2, 3.3, 22, 'Hi', 3, 99, 22, 66] [1, 2, 1, 2]
```

```
True
```

```
myList()
```


list 基本操作 – 搜尋

□ 搜尋

○ 判斷值是否存在該串列

- "bb" in data
 - True
- "aa" in data
 - True

```
def listTest():  
    data = ['aa', 'bb', 'cc', 'dd']  
    t = 'bb' in data  
    print(t)  
    print('aa' in data)  
    if 'cc' in data:  
        print('cc in')  
  
listTest()
```

```
True  
True  
  
cc in
```

○ 尋找某個項目的 index

```
student = ['tom', 'job', 'jay', 'jeff']  
print(student.index('jeff'))
```

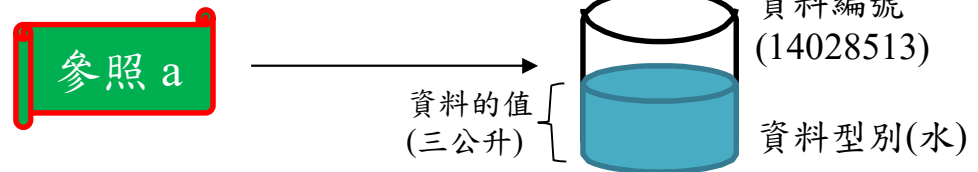
3

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/指向一個資料物件
- 資料物件

- 在電腦記憶體 (RAM) 中某個位址，存在一個值，
- 具有資料型別與編號，例如容器有水三公升



❑ Python 中所有資料 (data) 都是物件。

❑ 物件有：

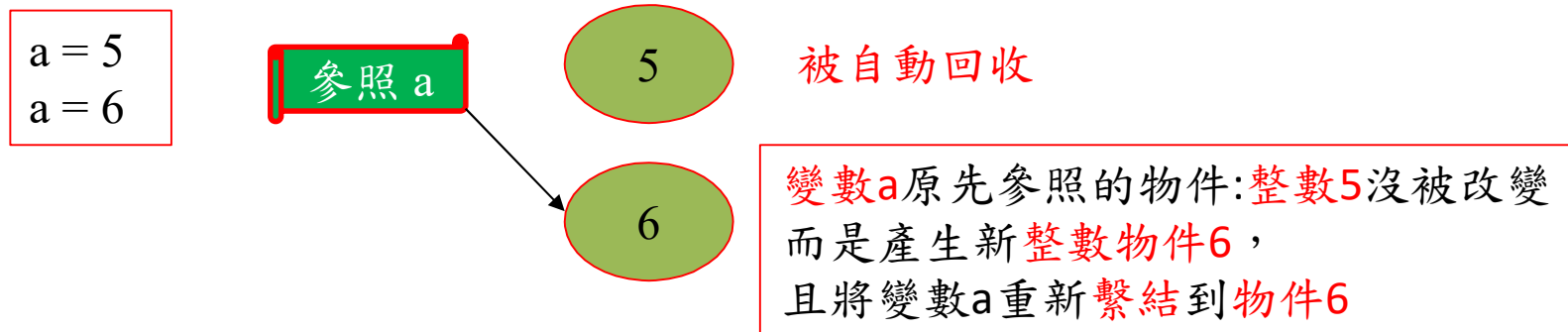
- id() 編號
- type() 資料型別
- value 值

```
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。



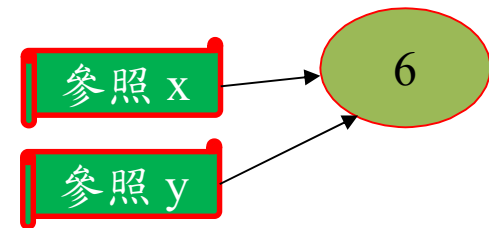
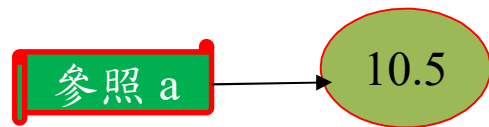
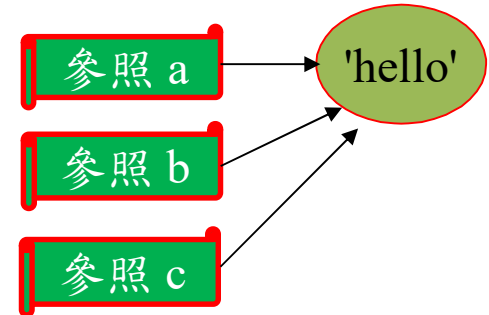
不可變物件

□ 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
```



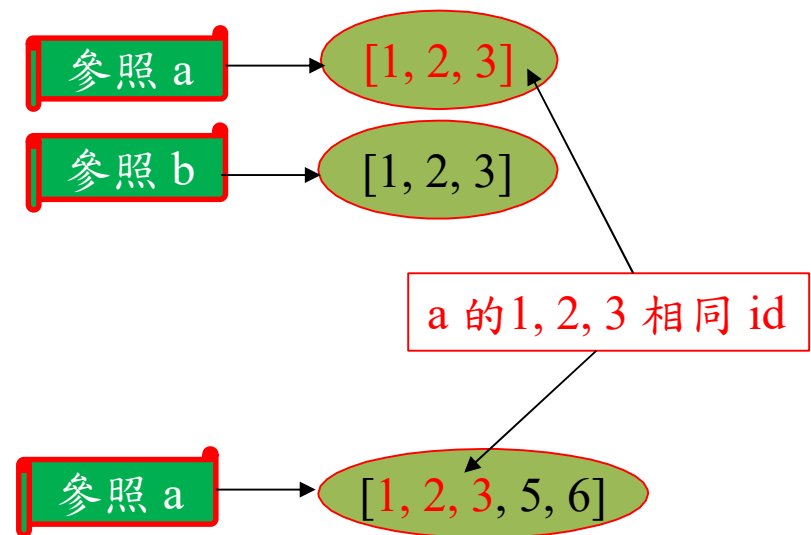
可變物件

□ 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
```



END

