

Dict 字典

- Dict的概念
- Dict的使用方法
- Dict與List的差別

摘要：字典

- Who(對象)：字典Dictionary
- Why(目的緣由)：為什麼字典Dictionary資料型態
 - 賦予索引有意義的名稱，簡化多筆資料的查詢/修改/新增/刪除
- When(應用時機)/Where(應用場域)：多筆資料的建置與維護
 - 處理的多筆資料內容具備有意義的索引概念
- What(內容定義)：字典的定義結構
 - 1. 符號：{ }
 - 2. key:value配對
- How(操作機制)：1. 存取 2. 相關函式 3. 字典操作方法
 - 1. 定義存取運算：定義字典{ }，並以key來存取資料
 - 2. 相關函式：len()、dict()
 - 3. 列表運算方法：新增、刪除、修改、存取等字典操作

摘要

List資料型態可同時儲存多筆資料，再透過index取得對應的內容值。

本章節我們將介紹字典Dict的概念與常用方法，

藉由有意義的key值，在撰寫程式時更直觀的存取其對應的value。

比較Dict與List之間的差別及範例幫助熟悉Dict資料型態。

回想

在List列表中，我們介紹了小明家的五種動物：

1.狗 2.貓 3.魚 4.鳥 5.烏龜

使用List來存取五種動物，必須透過索引來對任意動物進行存取，像是`animal[0] = "dog"`，`animal[1] = "cat"` 等等，此乃因為，在列表中的順序是固定的，且因為索引是無意義的，所以必須使用其他方式才能輔助識別對應關係。

查詢動物價格

```
animal = [["dog","cat","fish","bird","turtle"],[200,200,10,100,150]]
```

```
flag = "Y"
```

```
print("an[0] : ",animal[0],"an[1] : ",animal[1])
```

```
while flag=="Y":
```

```
    animal_name = input("請輸入要查詢的動物名稱：")
```

```
    for i in range (5):
```

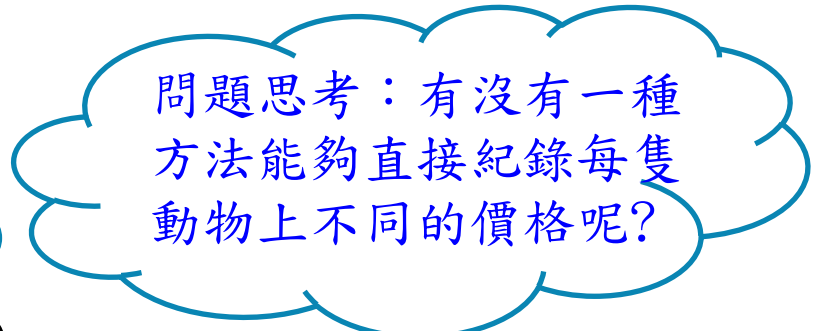
```
        name = animal[0][i]
```

```
        price = animal[1][i]
```

```
        if animal_name == name:
```

```
            print("動物 ",animal_name,"的價格是：",price)
```

```
flag = input("請問是否要繼續查詢(Y/N)：")
```



問題思考：有沒有一種方法能夠直接紀錄每隻動物上不同的價格呢？

字典

字典：dict是一種資料型態，是dictionary的縮寫。

說明：

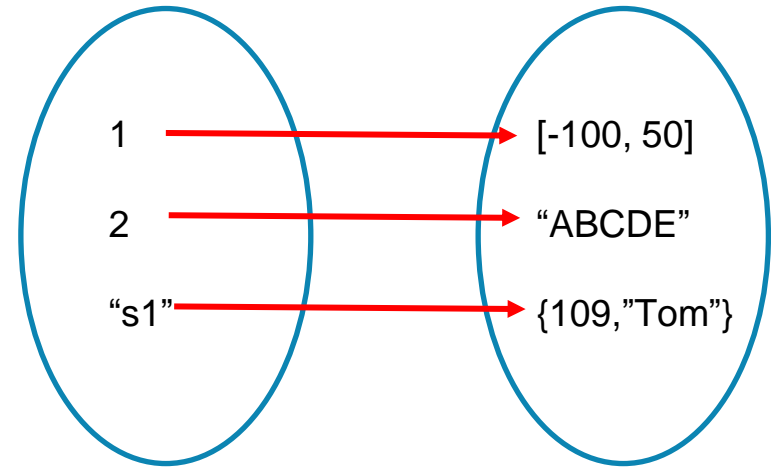
就是建立一本專屬的字典，試著想像你手上有一本字典，當你輸入英文單子的時候，可以查到它唯一的翻譯，也就是說英文單字與翻譯之間有著一對一的關係。

在字典中，輸入的單字叫做key，得到的翻譯叫做value。

➤ 我們可以說字典Dictionary就是一群key-value的集合。

Dict字典的規則

- 字典必須要用大括號{}框起來建立宣告
- 字典dict是一種資料型態，在建立時需要用一個變數去承接。
- 利用key和value的對應關係，來儲存和取得資料。
- key是唯一的，所以key不能重複。



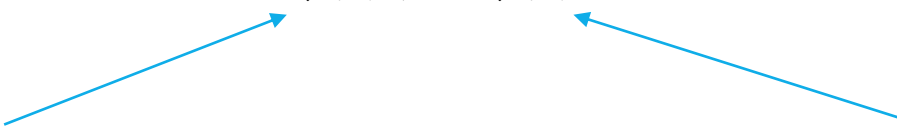
字典中可以存放的值

字典中的Key值可以是整數，浮點數、字串。

字典中的value值可以是任意的資料型態，字串、整數、列表、字典等。

舉例：

字典中的字典



```
Student = {8622501: {'name': 'Amy', 'age': 26}, 8622502: {'name': 'Bob', 'age': 33}}
```

key value key value

想想看

在python中，字典是一個非常好用的功能，
因為它可以自己設定索引值，也沒有順序的限制。

看完了字典的概念，你覺得字典可以運用在什麼地方呢？
例如：動物價格範例，動物的名稱設定為key，價格設定為value。

範例：動物價格

字典名稱

key : value

```
animal_price = {'dog' : 200, 'cat' : 200, 'fish' : 10, 'bird' : 100, 'turtle' : 150}
```

Key

Value

dog → 200

cat → 200

fish → 10

bird → 100

turtle → 150

Key與Value兩者一一對應

設定Dict空的初始值

- 語法：

字典名稱 = dict()

字典名稱 = { }

- 程式範例：

```
myDict1 = dict()
```

```
print("myDict1 :",myDict1)
```

```
myDict2 = { }
```

```
print("myDict2 :",myDict2)
```

輸出字典中的值

注意：

key必須存在於字典中

語法：

字典名稱[key名稱]

```
animal_price = {'dog': 200, 'cat': 200, 'fish': 10, 'bird': 100, 'turtle': 150}
```

```
print("animal dict初始值：\n", animal_price)
```

```
print("指定key印出：")
```

```
print("dog：", animal_price['dog'])
```

```
print("cat：", animal_price['cat'])
```

animal dict初始值：

```
{'dog': 200, 'cat': 200, 'fish': 10, 'bird': 100, 'turtle': 150}
```

指定key印出：

```
dog：200
```

```
cat：200
```

新增/修改字典中的值

- 新增：key 目前不存在字典中
- 修改：key 目前存在於字典中
- 語法：

字典名稱[key 名稱] = value

新增/修改字典中的值

```
animal_price = {'dog': 200, 'cat': 200, 'fish': 10, 'bird': 100, 'turtle': 150}
```

```
print("animal dict初始值：\n", animal_price)
```

```
animal_price["monkey"] = 1000
```

```
print("animal dict新增monkey後：\n", animal_price)
```

針對字典animal_price新增：
key為monkey；value為1000

```
animal_price["cat"] = 600
```

```
print("animal dict修改cat後：\n", animal_price)
```

針對字典animal_price修改：
原本key為cat，且對應的value為200
修改後key為cat，且對應的value為600

animal dict初始值：

```
{'dog': 200, 'cat': 200, 'fish': 10, 'bird': 100, 'turtle': 150}
```

animal dict新增monkey後：

```
{'dog': 200, 'cat': 200, 'fish': 10, 'bird': 100, 'turtle': 150, 'monkey': 1000}
```

animal dict修改cat後：

```
{'dog': 200, 'cat': 600, 'fish': 10, 'bird': 100, 'turtle': 150, 'monkey': 1000}
```

新增/修改字典中的值

```
animal_price = {'dog': 200, 'cat': 200, 'fish': 10, 'bird': 100, 'turtle': 150}
```

```
print("animal dict初始值：\n", animal_price)
```

```
key, value = input().split()
```

讀取使用者的輸入做為新增的內容值

```
animal_price[key] = int(value)
```

```
print("animal dict 新增後：", animal_price)
```

animal dict初始值：

```
{'dog': 200, 'cat': 200, 'fish': 10, 'bird': 100, 'turtle': 150}
```

MONKEY 300

使用者的輸入

animal dict 新增後： {'dog': 200, 'cat': 200, 'fish': 10, 'bird': 100, 'turtle': 150, 'MONKEY': 300}

C7_P1：自我介紹

創建一個空字典

輸入自我介紹內容

name、high_school、interests

將自我介紹儲存在字典中

C7_P2：速食店的外送聯絡人

創建外送員字典 `delivery_contacts`

輸入外送員名稱 `delivery_person`

輸出外送員電話

刪除字典中的值

- 注意:

key必須存在字典中

- 語法:

del 字典名稱[key名稱]

```
animal_price = {'dog': 200, 'cat': 200, 'fish': 10, 'bird': 100, 'turtle': 150}
```

```
print("animal dict初始值 : \n", animal_price)
```

```
del animal_price["fish"]
```

```
print("animal dict刪除fish後 : \n", animal_price)
```

del animal_price['fish']

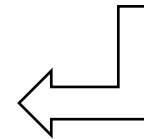
代表刪除 key為fish的key-value

animal dict初始值 :

```
{'dog': 200, 'cat': 200, 'fish': 10, 'bird': 100, 'turtle': 150}
```

animal dict刪除fish後 :

```
{'dog': 200, 'cat': 200, 'bird': 100, 'turtle': 150}
```



刪除字典中的值：

```
animal_price = {'dog': 200, 'cat': 200, 'fish': 10, 'bird': 100, 'turtle': 150}
```

```
print("animal dict初始值：\n", animal_price)
```

```
del animal_price["monkey"]
```

```
print("animal dict刪除monkey後：\n", animal_price)
```

當key不存在字典中，
會出現error message

```
animal dict初始值：
```

```
{'dog': 200, 'cat': 200, 'fish': 10, 'bird': 100, 'turtle': 150}
```

```
-----  
KeyError
```

```
Traceback (most recent call last)
```

```
<ipython-input-2-2bdf5f6460e7> in <module>
```

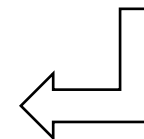
```
2 animal_price = {'dog': 200, 'cat': 200, 'fish': 10, 'bird': 100, 'turtle': 150}
```

```
3 print("animal dict初始值：\n", animal_price)
```

```
----> 4 del animal_price["monkey"]
```

```
5 print("animal dict刪除monkey後：\n", animal_price)
```

```
KeyError: 'monkey'
```



字典的複製

```
student1 = dict()
student1 = {108001:{"name":"s1","Test1":26,"Mid":60},108002:{"name":"s2","Test1":36,"Mid":50}}
student2 = {108001:{"name":"s1","Test1":26,"Mid":60},108002:{"name":"s2","Test1":36,"Mid":50}}
student3 = student1
student4 = student2.copy()
print("student1 : ",student1,"\n\n", "student2 : ",student2,"\n\n")
print("student3 : ",student3, "\n\n", "student4 : ",student4,"\n\n\n")
student3[108003]={"name":"s3","Test1":45,"Mid":70}
student4[108004]={"name":"s4","Test1":55,"Mid":70}
print("student1 : ",student1,"\n")
print("student2 : ",student2,"\n")
print("student3 : ",student3,"\n")
print("student4 : ",student3,"\n")
```

兩種複製字典的方法，
有甚麼差異？



List vs Dict

- Dict跟List一樣，不限定只能放同一種型態的資料。
- List使用中括號， Dict中所有的key和value都要放在大括號{}裡，並用冒號：來指定一組key和value。
- Dict由key及value配對組合而成，順序不重要，但key有分大小寫。
- Dict以Key值當索引來存取資料，List以index當索引來存取資料。

Dict常用方法

method	說明
len()	計算Dict的個數(key-value的組數)
in / not in	檢查Key是否存在該dictionary裡，回傳Boolean值
keys()	傳回Dict的key部份，資料型態為<class 'dict_keys'>
values()	傳回Dict的value部份，資料型態為<class 'dict_values'>
get()	取得指定key的值，第二個參數則是指定當找不到該key的時候所回傳的預設值

Dict常用方法

```
fruit = {'apple':20, 'orange':10, 'grape':80}
print("fruit key-value的組數:", len(fruit))
print("fruit中有lemon:", "lemon" in fruit)
print("fruit中沒有lemon:", "lemon" not in fruit)
print("fruit所有的key:", fruit.keys())
print("key轉成list:", list(fruit.keys()))
print("fruit所有的values:", fruit.values())
print("value轉成list:", list(fruit.values()))
print("fruit中apple的值:", fruit.get("apple"))
print("fruit中lemon的值:", fruit.get("lemon"))
print("fruit中lemon的值:", fruit.get("lemon",
"Lemon not found!!!"))
```

```
fruit key-value的組數: 3
fruit中有lemon: False
fruit中沒有lemon: True
fruit所有的key: dict_keys(['apple',
'orange', 'grape'])
key轉成list: ['apple', 'orange', 'grape']
fruit所有的values: dict_values([20, 10,
80])
value轉成list: [20, 10, 80]
fruit中apple的值: 20
fruit中lemon的值: None
fruit中lemon的值: Lemon not found!!!
```

C7_P3：查詢字典中的最大值

創建空字典 data_dict

讀取三組資料並加入字典

回傳字典中的最大值

OJ 習題

Problem

- C7_P1：自我介紹內容儲存
- C7_P2：速食店的外送聯絡人
- C7_P3：查詢字典中的最大值

Eercise

- C7_E1：計算字典中值的平均值
- C7_E2：修改及計算學生成績
- C7_E3：檢查字典中是否包含某個key
- C7_E4：字典的刪除
- C7_E5：計算單字出現次數

總結



本章關鍵字
本章重點概念簡述



請將你的章節總結寫在Memo上，
Memo的title請設定為**總結**