

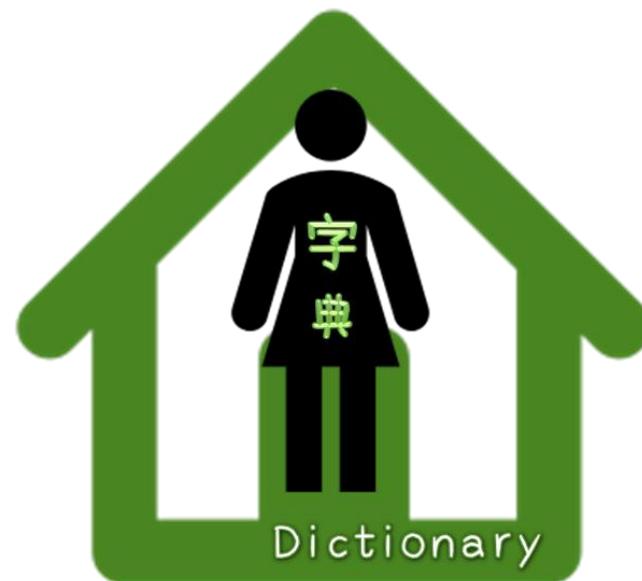
Python

集合(set)

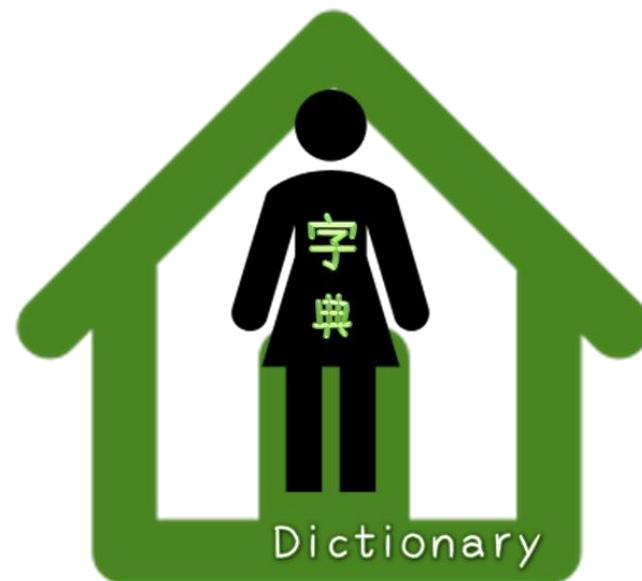
還記得這兩兄弟嗎？



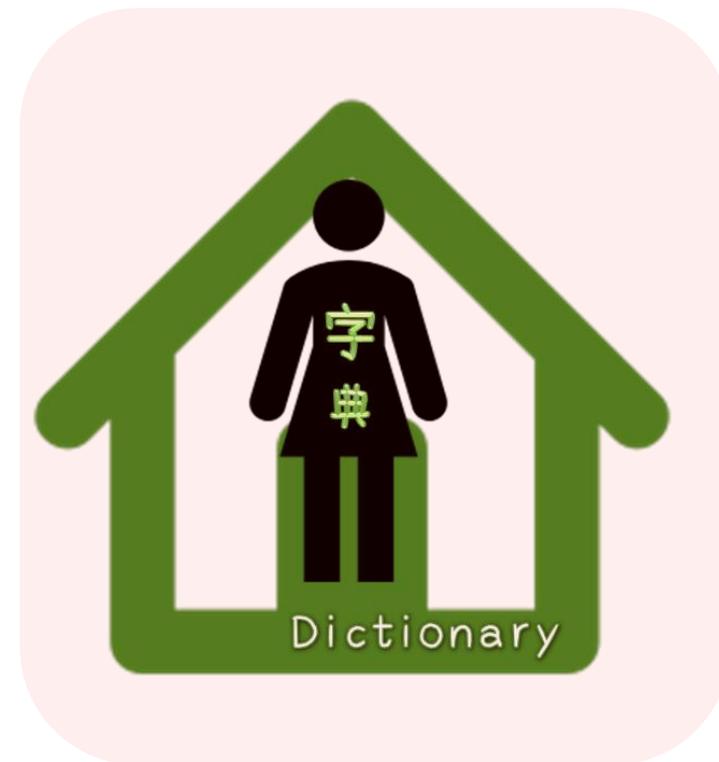
其實，他們還有親戚...



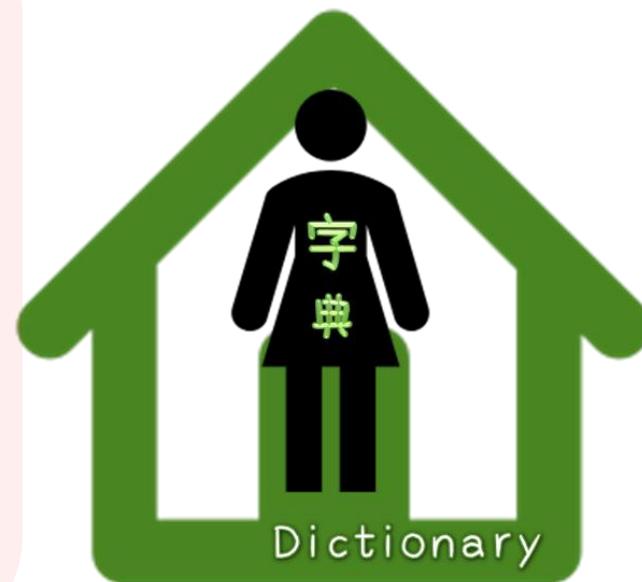
串列跟元組都可以存很多資料
串列可更動內容，元組不行



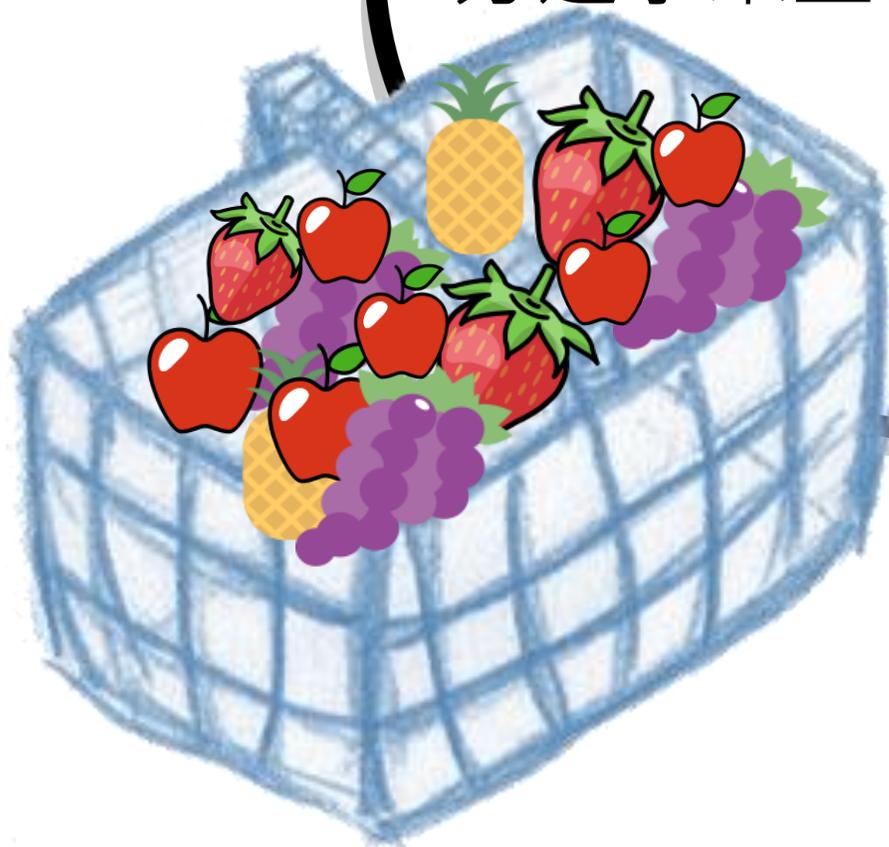
字典在下一章節介紹



先來看看集合(SET)



數數看
旁邊水果籃裡有幾種水果呢？



問題



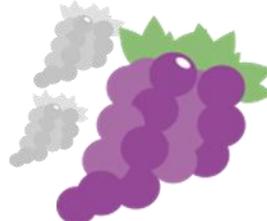
用串列數數看？

`fruits_list = [🍍, 🍏, 🍏, 🍓, 🍍, 🍓, 🍇, 🍏, 🍇, 🍏, 🍓, 🍏, 🍇, 🍏]`

`len(fruits_list) = 14`

總共有 14 顆(串)水果，但到底是有幾種呢？

同一種水果只能算一次

fruits_set = {  ,  ,  ,  }

len(fruits_set) = 4

總共有 4 種水果

同一種只會算一次的串列就稱為 **集合(set)**

集合數數

```
fruit_list = ['鳳梨', '蘋果', '蘋果', '草莓', '鳳梨', '草莓', '葡萄', '蘋果', '葡萄', '蘋果', '草莓', '葡萄', '葡萄', '蘋果']  
fruit_set = {'鳳梨', '蘋果', '蘋果', '草莓', '鳳梨', '草莓', '葡萄', '蘋果', '葡萄', '蘋果', '草莓', '葡萄', '葡萄', '蘋果'}  
print('串列個數: {fruit_list}, 集合個數:  
{fruit_set}'.format(fruit_list=len(fruit_list),  
fruit_set=len(fruit_set)))  
print('集合內容: {fruit_set}'.format(fruit_set=fruit_set))
```



執行結果

```
串列個數: 14, 集合個數: 4  
集合內容: {'蘋果', '草莓', '葡萄', '鳳梨'}
```



集合(set)

用大括號{}框起來一
堆資料稱為**集合(set)**

- 集合跟串列一樣都可以放很多資料
- 但是！集合沒有索引值！
- 所以，集合不能用索引(index)與切片(slicing)方式取得內含的資料

串列: []

元組: ()

集合: {}

別搞混了喔！



集合操作

```
thisset = {'蘋果', '草莓', '葡萄', '鳳梨'}  
print('1. 有{cnt}種水果'.format(cnt=len(thisset)))  
print('2. 芒果是否在集合中? {check}'.format(check=('芒果' in thisset)))  
# 新增一個  
thisset.add('柳丁')  
# 新增多個  
thisset.update(['西瓜', '芒果', '櫻桃'])  
print('3. 有{cnt}種水果'.format(cnt=len(thisset)))  
print('4. 芒果是否在集合中? {check}'.format(check=('芒果' in thisset)))  
# 移除芒果  
thisset.remove('芒果')  
print('5. 芒果是否在集合中? {check}'.format(check=('芒果' in thisset)))  
# 清空集合  
thisset.clear()  
print('6. 有{cnt}種水果'.format(cnt=len(thisset)))
```



集合最主要的功能是在判斷物品是否存在，所以新增/刪除/檢查是否存在等功能仍跟串列的操作相似喔！

執行結果

1. 有4種水果
2. 芒果是否在集合中? False
3. 有8種水果
4. 芒果是否在集合中? True
5. 芒果是否在集合中? False
6. 有0種水果



PYTHON

延伸的概念

概念 1: 小心刪除總沒錯

`thisset.remove(item)`

- 刪除集合內的某項項目
- 但項目一定要存在，不存在時會出現錯誤訊息

```
thisset = {'蘋果', '草莓', '葡萄'}  
thisset.remove('芒果')  
print(thisset)
```



`thislist.discard(item)`

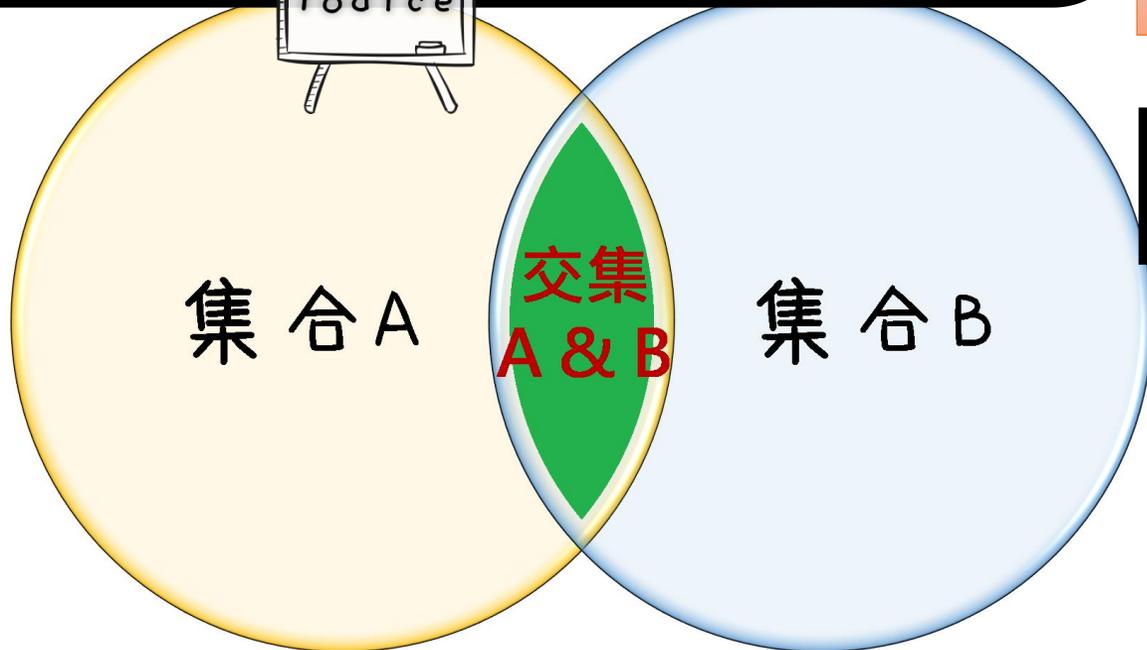
- 刪除集合內的某項項目
- 項目不存在也沒關係，只是做白工而以！

```
thisset = {'蘋果', '草莓', '葡萄'}  
thisset.discard('芒果')  
print(thisset)
```



概念 2: 交集 – 同時存在於兩個集合

```
thisset = {'蘋果', '葡萄', '鳳梨'}  
thatset = {'草莓', '鳳梨'}  
print(thisset & thatset)
```



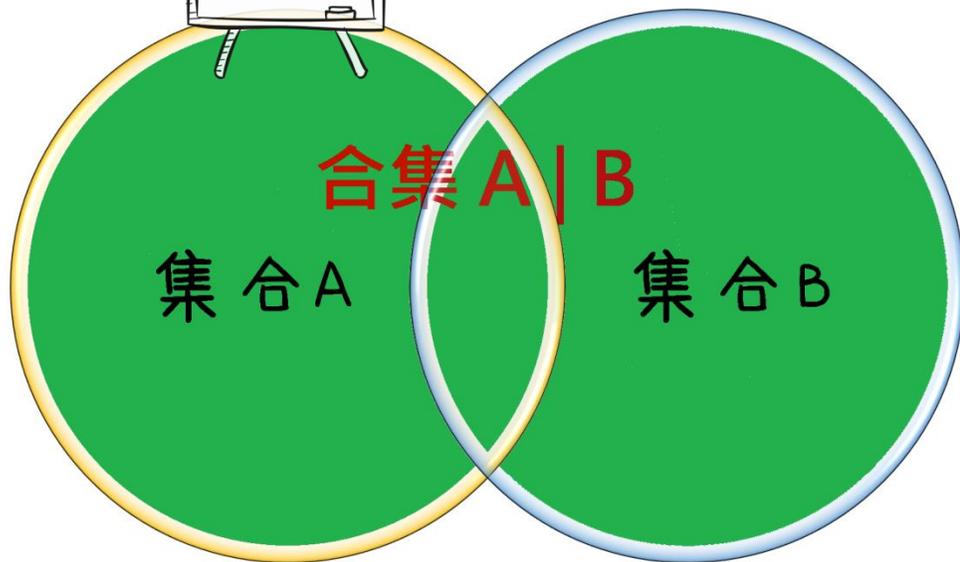
執行結果

{'鳳梨'}



概念 3: 合集 – 兩個集合當一個看

```
thisset = {'蘋果', '葡萄', '鳳梨'}  
thatset = {'葡萄', '鳳梨'}  
print(thisset | thatset)
```



執行結果

```
{'蘋果', '葡萄', '鳳梨'}
```



概念 4: 差集 – 一個集合有 一個集合沒有

```
thisset = {'蘋果', '葡萄', '鳳梨'}  
thatset = {'鳳梨', '芒果'}  
print(thisset - thatset)  
print(thatset - thisset)
```



執行結果

