

# 集合：去重、唯一與無序

滄海《Python 程式設計初學指引》二版·CH07

**考試範圍提醒**：老師投影片把「集合運算（交集／聯集／差集／對稱差集）、集合比較、複製、排序列舉(enumerate)、不可變集合 frozenset」都標為**（不考）**。本講義以**考試範圍**為主體（特性、宣告去重、讀取、增刪、統計），集合運算等放最後「補充（不考）」一節、僅供理解。

## § 1 名詞速查表

中文	English	一句話定義	例	易混
集合	set	無序、元素唯一、可變的容器，用 <code>{}</code> 含括	<code>{100, 200, 300}</code>	<code>{}</code> 是空字典不是空集合
元素／成員	element / member	集合裡的物件，同一集合中 <b>只能出現一次</b>	<code>'Python'</code>	重複會自動去除
無序	unordered	元素沒有排位順序， <b>不能用序號索引</b>	<code>s[0]</code> <i>x</i>	與串列不同
可雜湊	hashable	集合元素必須是 <b>不可變</b> 型別才能放入	數值、字串、元組	放串列／字典會 TypeError
去重	deduplicate	集合最常見用途：把序列的重複資料濾掉	<code>set(有重複的list)</code>	—
<code>set()</code>	—	建空集合或把可迭代物件去重轉集合	<code>set([1,1,2])</code> → <code>{1,2}</code>	空集合 <b>只能</b> 用 <code>set()</code>
<code>add()</code>	—	集合 <b>唯一</b> 的新增元素方法	<code>s.add('PHP')</code>	沒有 append/ insert
<code>discard()</code> vs <code>remove()</code>	—	刪元素；找不到時 discard 不報錯、 remove 丟 KeyError	<code>s.discard(x)</code>	差別在找不到時
<code>pop()</code>	—	<b>隨機</b> 移除並回傳一個元素 (不能指定)	<code>s.pop()</code>	與串列 <code>pop(i)</code> 不同

中文	English	一句話定義	例	易混
frozenset	—	不可變集合（ <b>老師標不考</b> ，補充）	<code>frozenset([1,2])</code>	—

## § 2 核心概念

### 核心概念

**集合 (set)** 跟數學的集合一樣，由一群元素組成，有三個關鍵特性：① **元素唯一**（同一元素只出現一次，重複會自動去除）；② **無序**（沒有排位順序，所以不能用序號 `s[0]` 取值）；③ **元素必須可雜湊 (hashable)**，也就是只能放不可變型別（數值、字串、元組），放串列或字典會 `TypeError: unhashable type`。

集合與字典同樣是**無序集合**、底層都用雜湊；差別是字典存鍵值組合、集合只存元素本身。集合最實用的兩件事：**去重**（把序列的重複資料濾掉，`set(串列)`）與集合運算（找共同／不同元素）。

宣告有個經典陷阱：`{}` 是**空字典**不是空集合，空集合只能寫 `set()`。這點期末很容易考。

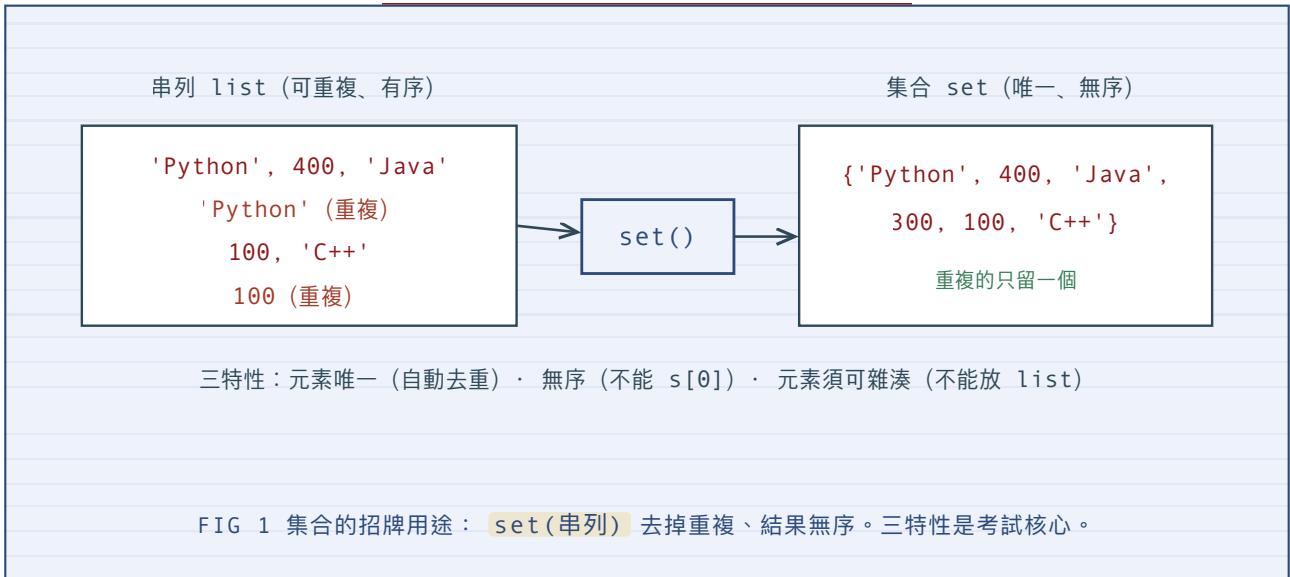
## § 3 主要內容

### 3.1 宣告集合與去重

```
MySet = {} # 注意！這是空「字典」，不是空集合
empty = set() # 空集合只能這樣建
MixSet = {'Python', 400, 'Java', 300, 'Python', 100} # 重複的 'Python' 自動去除
```

`set(可迭代物件)` 可把串列、元組、字典（取 key）、字串轉成集合，同時**去重**：

```
DupList = ['Python', 400, 'Java', 300, 'Python', 100, 'C++', 100]
set(DupList) # {'100', 'C++', 300, 400, 'Java', 'Python'} 去掉重複的 'Python'、100
set('Anaconda') # {'A', 'n', 'a', 'c', 'o', 'd'} 字串逐字元、去重
```



### 宣告兩個陷阱

- ① `{}` 是空字典，空集合要用 `set()`。② 集合元素必須可雜湊（不可變）：`{'a', [1,2]}` 會 `TypeError: unhashable type: 'list'`，因為串列可變、不可雜湊。元組可以（不可變）。

## 3.2 讀取元素：不能索引，用 for / in

集合無序，沒有序號，所以不能 `s[0]`。要走訪用 `for ... in`，要判斷存在用 `in / not in`：

```
MixSet = {'Python', 400, 'Java', 300}
for item in MixSet:
    # 可走訪（順序依雜湊結果，不保證輸入順序）
    print(item, end=' ')
'Python' in MixSet
# True 判斷是否存在
MixList = list(MixSet)
# 需要序號時，先轉成串列
MixList[-1]
# 轉成串列後才能用索引
```

## 3.3 增加與刪除元素

```
BookSet = {'Python', 'Java', 'VB'}
BookSet.add('PHP')
# add() 是唯一的新增方法（沒有 append/insert）

BookSet.discard('Java')
# 刪除；找不到「不報錯」、回傳 None
BookSet.remove('VB')
# 刪除；找不到丟 KeyError
BookSet.pop()
# 隨機移除並回傳一個元素（不能指定序號）
BookSet.clear()
# 清空成 set()
```

### discard vs remove vs pop

三者都刪元素，差別是「找不到時的反應」與「能否指定」：

- `discard(x)`：找不到 `x` **不報錯** (回 `None`)，最安全。
- `remove(x)`：找不到 `x` 丟 **`KeyError`**，會中斷程式。
- `pop()`：**隨機**刪一個並回傳，**不能**像串列那樣 `pop(序號)` (集合無序)。

## 3.4 統計資訊

集合一樣能用 `len()`、`max()`、`min()`、`sum()`：

```
PriceSet = {400, 200, 100, 300}
len(PriceSet) # 4
max(PriceSet), min(PriceSet), sum(PriceSet) # 400, 100, 1000
```

## 3.5 補充 (老師標「不考」，僅供理解)

以下老師投影片標為不考，列出讓你有概念、不必背：

- **集合運算**：交集 `a & b` / `a.intersection(b)`、聯集 `a | b` / `a.union(b)`、差集 `a - b` / `a.difference(b)`、對稱差集 `a ^ b` / `a.symmetric_difference(b)`。用范恩圖理解：交集是中間重疊、聯集是全部、差集是只屬於 `a`、對稱差集是只屬於其一。
- **集合比較**：`issubset()` (`<=`)、`issuperset()` (`>=`)、`isdisjoint()` (兩集合完全無共同元素)。
- **複製**：集合元素不可變，`.copy()` 淺複製與深複製效果相同。
- **排序與列舉**：`sorted(集合)` 回傳排序後串列；`enumerate()` 列舉編號。
- **不可變集合 `frozenset`**：`frozenset()` 建立後不能增刪改，沒有 `add()` / `remove()`。

## § 4 語法與方法速查

```

# 建立
{元素, ...}          set(可迭代物件)      set()      # 空集合只能用 set(), {} 是空字典

# 讀取 (不能索引)
for x in s:          x in s      x not in s      list(s)      # 要序號先轉串列

# 增刪
s.add(x)             # 唯一的新增方法
s.discard(x)         # 找不到不報錯 (回 None)
s.remove(x)          # 找不到 KeyError
s.pop()              # 隨機刪一個並回傳 (不能指定)
s.clear()            # 清空成 set()

# 統計
len(s)      max(s)      min(s)      sum(s)

```

操作	找不到元素	能否指定
<code>s.discard(x)</code>	不報錯、回 <code>None</code>	指定值
<code>s.remove(x)</code>	<b>KeyError</b>	指定值
<code>s.pop()</code>	(集合空才 <code>KeyError</code> )	<b>隨機</b> 、不能指定

## § 5 常見錯誤

### 常見錯誤

- **{}** 當空集合：`{}` 是空字典，空集合要用 `set()`。
- **集合元素放可變型別**：`{[1,2]}` 或 `{ {...}:... }` 會 `TypeError: unhashable type`；元素只能放數值／字串／元組等不可變型別。
- **想用序號取集合元素**：集合無序，`s[0]` 會 `TypeError`；要走訪用 `for`、要序號先 `list(s)`。
- **`remove()` 刪不存在的元素**：丟 `KeyError`；不確定在不在用 `discard()`。
- **以為 `pop()` 能指定**：集合的 `pop()` 隨機刪，不能像串列 `pop(序號)`。
- **集合只有 `add()`**：沒有 `append()`／`insert()`（那是串列的）。
- **依賴集合的順序**：印出順序依雜湊、不保證等於輸入順序，不要假設。

## § 6 練習題

### 練習 1 (一般題)：去重

#### 引導步驟

1. 建一個有重複元素的串列。
2. 用 `set()` 轉成集合、再用 `len()` 算去重後剩幾個。
3. 預測：原串列順序會保留嗎？


#### 解答

```
dup = [1, 2, 2, 3, 3, 3]
s = set(dup)
print(s, len(s))      # 例如 {1, 2, 3} 3; 順序不保證
```

## 練習 2（重要題）：空集合與可雜湊陷阱

### 引導步驟

1. `type({})` 是什麼？空集合該怎麼建？
2. 試 `{1, 2, [3, 4]}` 會發生什麼？為什麼？


### 解答

```
print(type({}))      # <class 'dict'> {} 是空字典
empty = set()        # 空集合
# {1, 2, [3,4]}      # TypeError: unhashable type: 'list' (串列可變、不可雜湊)
```

### 練習 3（一般題）：走訪與判斷存在

#### 引導步驟

1. 集合能不能用 `s[0]`？要走訪用什麼？
2. 用 `in` 判斷某元素是否存在。
3. 需要序號時怎麼辦？


#### 解答

```
s = {'Python', 'Java', 'VB'}  
for x in s: print(x, end=' ') # 走訪（順序不定）  
print('Java' in s)          # True  
lst = list(s); print(lst[-1]) # 要序號先轉串列
```



### 練習 5（一般題）：統計

#### 引導步驟

1. 給一個數值集合，用 `len/max/min/sum` 算數量、最大、最小、總和。


#### 解答

```
prices = {400, 200, 100, 300}
print(len(prices), max(prices), min(prices), sum(prices)) # 4 400 100 1000
```

## § 7 自我檢核

- 能說出集合三特性：元素唯一、無序、元素須可雜湊（不可變）。
- 知道 `{}` 是空字典，空集合要用 `set()`。
- 能用 `set(串列)` 去重，並知道結果無序。
- 知道集合不能用序號索引，走訪用 `for`、判斷用 `in`、要序號先 `list()`。
- 記得集合唯一的新增方法是 `add()`（沒有 `append/insert`）。
- 能分辨 `discard()`（找不到不報錯）、`remove()`（找不到 `KeyError`）、`pop()`（隨機刪）。
- 能用 `len/max/min/sum` 取集合統計資訊。
- 知道集合運算、比較、複製、`frozenset`、`enumerate` 老師標**不考**，有概念即可。