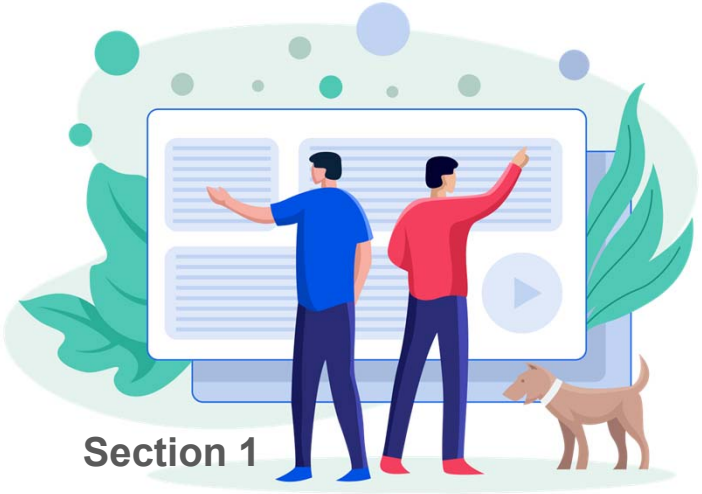
# 集合(Set)

# 學習目標

- 集合(Set) 宣告與運算
- 集合(Set) 相關的方法
- Set Comprehensions

# 集合(Set)的用法

- 介紹集合(Set) 的特性
- 如何存取集合(Set) 的內容



# 集合(Set)

- 集合沒有順序(unordered)，所以也沒有slice功能
- 項目不會重複
- 用一對大括號括起來，每個元素用逗點隔開
- 字典與集合都是使用大括號表示。若大括號內有冒號區分key與value，就代表是字典；如果是分散的資料，就代表是集合
- 或是使用內建 set() 函數建立集合
  - 必須用 set() 來建立空集合，因為 { } 代表空字典(Dict)
- 適用於成員關係的測試，儲存不重複的資料
- Set的使用時機：用於成員關係的測試，儲存不重複的資料。Ex: 用集合來處理莎士比亞的文學著作，找出哪一個字(word)被使用的最頻繁
- 集合支援算術運算，例如聯集(union)、交集(intersection)、差集(difference) 以及對稱差集(symmetric difference)

# 使用集合

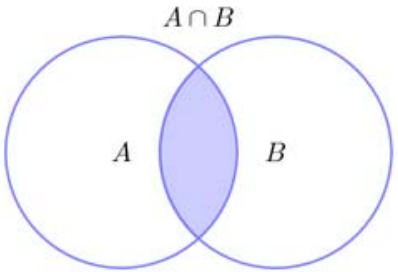
- 範例
  - basket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}
  - basket  # duplicates are removed
  - {'orange', 'banana', 'pear', 'apple'}
  - 'orange' in basket  # membership testing
  - True
  - 'grape' in basket
  - False
- 集合(Set)內不可以放串列(List)

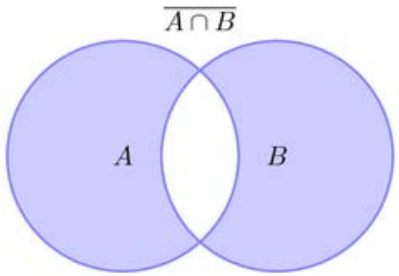
# 基本集合運算

| 運算 | 結果 | 說明 |
| --- | --- | --- |
| len({1, 2, 3}) | 3 | 求集合個數 |
| 3 in {1, 2, 3} | True | 成員運算 |
| for x in {1,5,2,3} :<br>print (x, end = ' ') | 1 2 3 5 | 迴圈運算(無順序,unordered) |
| {1,2,3} \| {2,3,4} | {1, 2, 3, 4} | 聯集(Union) |
| {1,2,3} & {2,3,4} | {2, 3} | 交集(Intersection) |
| {1,2,3} - {2,3,4} | {1} | 差集(Difference) |
| {1,2,3} ^ {2,3,4} | {1, 4} | 對稱差集(Symmetric difference) |
| {1,2} < {1,2,3,4}<br>{1,2} > {1,2,3,4} | True<br>False | 包含於(Subset)<br>包含(Superset) |

# 集合運算

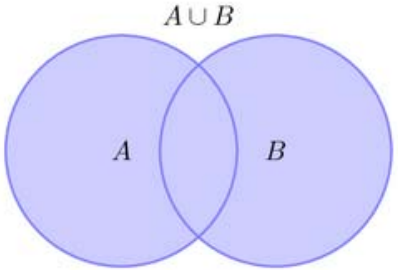
A & B

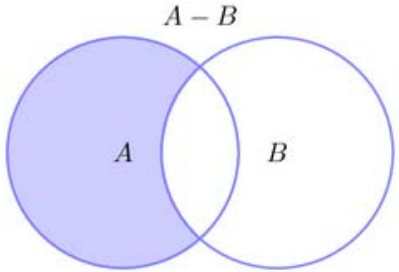



A ^ B

A | B




A - B

# 集合(Set)的用法(Demo)

- 如何使用集合(Set)
- 集合內，可放數字、文字。但不可以放置List。
- 但通常會放置相同資料型態的資料，以方便後續的資料處理

# 集合(Set)相關函數與方法

- 介紹集合(Set) 的函數與方法

Section 2

# 內建集合相關函數 -1

- len(set)
  - 取得 set 項目個數
- max(set)
  - 傳回 set 內的最大值
  - 所有項目必須是相同的資料型態
- min(set)
  - 傳回 set 內的最小值
  - 所有項目必須是相同的資料型態

# 內建集合相關函數 -2

- sum(set)
  - 傳回 set 所有元素的總和
  - 所有項目必須是數值資料型態
- sorted(set)
  - 預設傳回升冪排序的串列(List)，可用 reverse=True 引數做降冪排序
  - 所有項目必須是相同的資料型態
- set(seq)
  - 將資料轉成 set

# 集合相關方法 -1

- s.isdisjoint(other)
  - 如果兩集合沒有交集則傳回 True，否則傳回 False
- s.issubset(other)
  - 檢查 s 集合是否包含於 other 集合，相當於 s <= other
- s.issuperset(other)
  - 檢查 s 集合是否包含 other 集合，相當於 s >= other
- s.union(*others)
  - 傳回 s 與其他集合的聯集，相當於 s | other | ...

# 集合相關方法 -2

- s.intersection(*others)
  - 傳回 s 與其他集合的交集，相當於 s & other & ...
- s.difference(*others)
  - 傳回 s 與其他集合的差集，相當於 s - other - ...
- s.symmetric_difference(other)
  - 傳回 s 與其他集合的對稱差集，相當於 s ^ other
- s.copy()
  - 複製 s 集合

# 集合相關方法 -3

- s.update(*others)
  - 將 s 與其他集合的聯集存入 s，相當於 s |= other | ...
- s.intersection_update(*others)
  - 將 s 與其他集合的交集存入 s，相當於 set &= other & ...
- s.difference_update (*others)
  - 將 s 與其他集合的差集存入 s，相當於 s -= other | ...
- s.symmetric_difference_update (other)
  - 將 s 與其他集合的對稱差集存入 s，相當於 s ^= other

# 集合相關方法 -4

- s.add(elem)
  - 將 elem 加入 s
- s.remove(elem)
  - 將 elem 從 s 移除。如果 elem 不存在 s 內，會觸發 KeyError 的例外
- s.discard(elem)
  - 如果 elem 存在於 s ，則將其移除。此方法不會觸發 KeyError 的例外
- s.pop()
  - 從 s 內隨意取出並移除某一元素。如果 s 是空集合，會觸發 KeyError 的例外
- s.clear()
  - 移除 s 所有元素

# 集合(Set)相關函數與方法(Demo)

- 如何使用集合(Set) 相關函數與方法

# Set Comprehensions

- 介紹 Set comprehensions 的語法
- 如何使用 Set comprehensions

Section 3

# 使用Set Comprehensions

- Python 的集合一樣也有 Comprehensions
- 語法
  - { expression for expression in iterable }
- 範例
  - s = {num for num in range(1,6) if num % 3 == 1}
  - s
    - {1, 4}

# 本章重點精華回顧

- 集合(Set) 的特性與用途
- Set comprehensions 的用法

# Lab: 集合

- Lab01: 使用集合(Set)
- Lab02: 使用 Set comprehensions

# Lab01: 使用集合(Set)

- 啟動Python互動式執行環境，做以下練習

- >>> s1 = set()
- >>> s1
- >>> type(s1)
- >>> s2 = {1,2,3,3,5,6,5,1,7}
- >>> s2
- >>> type(s2)
- >>> len(s2)
- >>> 2 in s2
- >>> 5 not in s2

- >>> s1 = {1,2,3,4,5}
- >>> s2 = {4,5,6,7,8}
- >>> s1
- >>> s2
- >>> s1 | s2
- >>> s1.union(s2)
- >>> s1 & s2
- >>> s1.intersection(s2)
- >>> s1 - s2
- >>> s1.difference(s2)

- >>> s1 ^ s2
- >>> s1.symmetric_difference(s2)
- >>> {2,3} < s1
- >>> s2 > {6,7}
- >>> s1. intersection_update(s2)
- >>> s1
- >>> s2.pop()
- >>> s2
- >>> s2.clear()
- >>> s2

# Lab02: 使用Set comprehensions

- 啟動Python互動式執行環境，做以下練習

- >>> s = {num for num in range(1,6) if num % 3 == 1}
- >>> s