



06：數組、字典、集合

2018.1.30

主題

- 數組(Tuple)
- 字典(Dict)
- 進階字典操作
- 集合(Set)

數組(Tuple)

- 數組的結構與串列完全相同，不同處在於數組的元素個數與元素值皆不能改變，所以稱數組是「不能修改的串列」。
- 數組的使用方式是將元素置於小括號中 (串列是中括號)，元素之間以逗號分隔，語法為：
 - 數組名稱 = (元素1, 元素2, ...)
 - `tup = (“魯夫”, “索隆”, “烏索普”)`
 - `tup[1] = “香吉士” # error`

3

數組的優點

執行速度較串列快

數組的資料較安全

4

• 串列 & 數組

- 串列與數組可以互相轉換

```
3 tup = (1, 2, 3, 4, 5, 6)
4
5 lst = list(tup)
6 print(lst.pop(3))
7
8 tup = tuple(lst)
9 print(tup)
```

5

• 主題

- 數組(Tuple)
- 字典(Dict)
- 進階字典操作
- 集合(Set)

6

● 怎麼做？

- 想要在程式裡使用下列資料，該怎麼做？

- 魯夫 → ATK = 90
- 娜美 → ATK = 50
- 喬巴 → ATK = 70

```
atk = [90, 50, 70]
```

```
name = ["魯夫", "娜美", "喬巴"]
```

7

● 還有一招

```
atk = [ ["魯夫", 90],  
        ["娜美", 50],  
        ["喬巴", 70]]
```

```
print(atk[2][1]) # 顯示甚麼？
```

8

字典(Dict)

- 字典的結構也與串列類似，其元素是以「**鍵 - 值**」對方式儲存，這樣就可使用「**鍵**」來取得「**值**」。
- 字典是將元素置於一對大括號「**{}**」中，其語法為：
 - 字典名稱 = { 鍵1:值1, 鍵2:值2, ... }

9

範例



atk = { “魯夫”:90, “娜美”:50, “喬巴”:70 }

The diagram shows a blue rectangular box containing the dictionary definition. Two green speech bubbles point to parts of the definition: one points to the keys (魯夫, 娜美, 喬巴) and is labeled '鍵(key)', and the other points to the values (90, 50, 70) and is labeled '值(value)'.



```
print(atk[“娜美”])
```

The diagram shows a red rectangular box containing the code snippet for accessing a value from the dictionary.

10

字典操作1

- 元素在字典中的排列順序是隨機的，與設定順序不一定相同
 - `atk = {"魯夫":90, "娜美":50, "喬巴":70}`
 - `print(atk)`

11

字典操作2

- 新增元素的方法是設定新「鍵」及新「值」，例如：
 - `atk = {"魯夫":90, "娜美":50, "喬巴":70}`
 - `atk["羅賓"] = 65`
- 修改元素值的方法是對「鍵」設定新「值」，新元素值會取代舊元素值，例如：
 - `atk["喬巴"] = 80`

12

字典操作3

- 刪除字典則有三種情況
 - 第一種是刪除字典中特定元素，語法為：
 - `atk = { “魯夫”:90, “娜美”:50, “喬巴”:70 }`
 - `del atk[“魯夫”]`
 - 第二種是刪除字典中所有元素，語法為：
 - `atk.clear()`
 - 第三種是刪除字典，字典刪除後該字典就不存在，語法為：
 - `del atk`

13

主題

- 數組(Tuple)
- 字典(Dict)
- 進階字典操作
- 集合(Set)

14

進階字典操作1

- 許多進階方法可操作字典，下表為常用方法：
 - `dict1={"joe":5, "mary":8}`，`n` 為整數，`b` 為布林變數

方法	意義	範例	範例結果
<code>len(dict1)</code>	取得字典元素個數	<code>n=len(dict1)</code>	<code>n=2</code>
<code>dict1.clear()</code>	移除所有字典元素	<code>dict2=dict1.clear()</code>	<code>dict2</code> 為空字典
<code>dict1.copy()</code>	複製字典	<code>dict2=dict1.copy()</code>	<code>dict2={"joe":5, "mary":8}</code>
<code>dict1.get(鍵, 值)</code>	取得「鍵」對應的「值」，若「鍵」不存在就傳回參數中的「值」	<code>n=dict1.get("joe")</code>	<code>n=5</code>
★ <code>鍵 in dict1</code>	檢查「鍵」是否存在	<code>b="joe" in dict1</code>	<code>b=True</code>

15

進階字典操作2

- 許多進階方法可操作字典，下表為常用方法：
 - 表中 `dict1={"joe":5, "mary":8}`，`n` 為整數，`b` 為布林變數

方法	意義	範例	範例結果
★ <code>dict1.items()</code>	取得以「鍵 - 值」組為元素的組合	<code>dict2=dict1.items()</code>	<code>dict2=[("joe":5), ("mary":8)]</code>
<code>dict1.keys()</code>	取得以「鍵」為元素的組合	<code>dict2=dict1.keys()</code>	<code>dict2=["joe", "mary"]</code>
<code>dict1.setdefault(鍵, 值)</code>	與 <code>get()</code> 類似，若「鍵」不存在就以參數的「鍵 - 值」建立新元素	<code>n=dict1.setdefault("joe")</code>	<code>n=5</code>
<code>dict1.values()</code>	取得以「值」為元素的組合	<code>dict2=dict1.values()</code>	<code>dict2=[5,8]</code>

16

• keys(), values(), items()

- 字典的 **keys()** 方法可取得所有「**鍵**」組成的組合，資料型態為 `dict_keys`；**values()** 方法可取得所有「**值**」組成的組合，資料型態為 `dict_values`。
- 可將 **keys()**, **values()** 方法取得的資料以 **list()** 轉換為串列，轉成串列才能取得元素值，將兩者組合就可列印字典全部內容。
- e.g., `atk = { "魯夫":90, "娜美":50, "喬巴":70 }`

17

• 主題

- 數組(Tuple)
- 字典(Dict)
- 進階字典操作
- **集合(Set)**

18

集合(Set)

- 集合是無序、元素不重複的群集，語法如下：
 - 集合名稱 = {"元素1", "元素2", ...}

```
3 one_piece = {"魯夫", "索隆", "娜美"}
4 user = {"娜美", "喬巴"}
5
6 print(one_piece)
7 print(one_piece & user)      # 交集
8 print(one_piece | user)     # 聯集
9 print(one_piece - user)     # 差集
10 print(user - one_piece)    # 差集
11 print(user ^ one_piece)    # 對稱差集
12
13 s = set("Hello")
14 print(s)
```



```
{'索隆', '魯夫', '娜美'}
{'娜美'}
{'魯夫', '喬巴', '娜美', '索隆'}
{'索隆', '魯夫'}
{'喬巴'}
{'喬巴', '索隆', '魯夫'}
{'l', 'H', 'e', 'o'}
```

19

集合的操作

- 若一個集合名稱為 test_set，則利用
 - test_set.add(e) 對集合進行加入元素 e
 - test_set.remove(e) 對集合進行移除元素 e
- 也可在集合中使用 len(), min(), max(), sum() 或 for 迴圈拜訪集合內的所有元素

20

● PY3-0002

七、集合條件判斷

2. 設計說明：

- (1) 請撰寫一程式，輸入數個整數並儲存至集合，以輸入-9999 為結束點（集合中不包含-9999），最後顯示該集合的長度（Length）、最大值（Max）、最小值（Min）、總和（Sum）。

TO BE CONTINUED...