07 元組(數組) 、集合、字典

1. 元組 (tuple)

Python 的元組 (tuple) 和串列很相似，但有幾點不同：

* 1. 元組 的元素值不可以改變。
  2. 在元組 中無法刪除個別元素和取代元組 中的資料，但可以刪除或覆寫整個元組 的所有元素。
  3. 沒有提供類似串列加入的方法如 append 與 insert，但是可以利用 + 把元素加入元組 或是利用 \* 複製元素。

元組 是以小括號建立，元素間用逗號隔開。例：

tuple1 = (2,4,1,3,9,5)

print (tuple1)

(2, 4, 1, 3, 9, 5)

若小括號內沒有元素，則表示為空元組 。例：

tuple2 = ()

print(tuple2)

()

從串列中建立元組 。例：

tuple3 = tuple([i for i in range(1,6)])

print(tuple3)

(1, 2, 3, 4, 5)

這表示建立一組元組 tuple3，元素計有(1,2,3,4,5)。注意，要加上**tuple**。也可以從字串建立元組 ，其元組 是這字串中字元所組成的。例：

沒有加上 tuple 之前

tuple4 = ('python')

print(tuple4)

python

加上tuple之後

tuple4 = tuple('python')

print(tuple4)

('p', 'y', 't', 'h', 'o', 'n')

元組 可以用 len、max 、min、sum等串列使用的函式，以及in、not in、\*、+ 運算子，這些功能和串列相似。例：

a = len (tuple1)

print(a)

6

a = max (tuple1)

print(a)

9

a = min (tuple1)

print(a)

1

a = sum (tuple1)

print(a)

24

a = 8 in tuple1

print(a)

False

a = 9 in tuple1

print(a)

True

a = 8 not in tuple1

print(a)

True

a = 9 not in tuple1

print(a)

False

元組 內的元素不能被刪除，但可以增加，增加方式使用 += 運算符號，要加入的元素放在小括號之中。例：

tuple1 += (6,)  #增加一個元素時後面加逗號

print(tuple1)

如內容所述，只增加一個元素時，元素內容後面加上逗號

(2, 4, 1, 3, 9, 5, 6)

tuple1 += (7,8)

print(tuple1)

兩個以上元素，最後一個元素後面逗號可以省略

(2, 4, 1, 3, 9, 5, 6, 7, 8)

與串列相同，可以使用索引值來取得所在位置的元素內容，注意使用中括號。例：

a = tuple1[2]

print(a)

1

tuple2 = tuple1[3:6]

print(tuple2)

(3, 9, 5)

tuple2 = 2\* tuple1

print(tuple2)

(2, 4, 1, 3, 9, 5, 2, 4, 1, 3, 9, 5)

使用迴圈印出元組 中的所有資料：

for i in tuple1:

    print(i, end = ' ')

2 4 1 3 9 5

一開始就提到元組 沒有刪除個別元素的方式，但可以刪除整組元組 ，刪除元組 是透過del 指令執行。

del tuple1

print(tuple1)

NameError: name 'tuple1' is not defined

1. 集合

若是注意元組 的特點是無法刪除個別元素的話，集合的特點是集合內容當中，不會出現重複的元素。

* 1. 集合的建立方式

集合以**大括號**建立，元素間以逗號區隔。例：

set1 = {1,3,5}

print(set1)

{1, 3, 5}

建立空集合不能直接使用空的大括號，否則會被當成字典(後面地3段說明)，而是使用**小括號**，注意小括號前面加上set否則會被當成tuple，這是較為特別的地方。

set2 = set()  #注意用到"set"否則會被當成字典

集合可以透過串列或是元組 建立資料。

從串列建立集合資料方式：

set3 = set([i for i in range(1,6)])

print(set3)

{1, 2, 3, 4, 5}

list2 = [1,2,3,4,5]

set3 = set(list2)

print(set3)

{1, 2, 3, 4, 5}

從元組 建立集合資料方式：

set4 = set((1,2,3))

print(set4)

{1, 2, 3}

tuple1 = (2,4,1,3,9,5)

set4 = set(tuple1)

print(set4)

輸出結果與原元組 排序並不相同

{1, 2, 3, 4, 5, 9}

開始說過集合不會出現重複資料，所以當產生重複資料時：

set5 = set((1,1,2,2,3)) #以元組 值置入集合

print(set5)

{1, 2, 3}

若將存入集合資料再放回元組

tuple2 = (1,1,2,2,3,3,4,5)

tuple2 = tuple(set5)  #將集合放入元組

print(tuple2)

(1, 2, 3)

tuple4 = tuple('python')

print(tuple4)

set6 = set(tuple4)

print(set6)

('p', 'y', 't', 'h', 'o', 'n')

{'h', 't', 'o', 'y', 'p', 'n'}

* 1. 集合加入及刪除

使用add(x)將x加入集合中，使用remove(x)將x從集合中刪除。

set10 = {1,3,6}

set10.add(30)

print(set10)

{1, 3, 6, 30}

set10.remove(3)

print(set10)

{1, 6, 30}

計算長度的len()、計算總和的sum()、以及求出最大和最小值的max()與min()

set20 = {1,3,6,8,10}

a = len(set20)

print(a)

5

a = sum(set20)

print(a)

28

a = max(set20)

print(a)

10

a = min(set20)

print(a)

1

使用 in 與 not in 檢視元素是否在集合中

set20 = {1,3,6,8,10}

a = 4 in set20

print(a)

False

a = 8 in set20

print(a)

True

a = 4 not in set20

print(a)

True

a = 8 not in set20

print(a)

False

使用for迴圈印出集合中所有元素

for i in set20:

    print(i, end = ' ')

1 3 6 8 10

* 1. 集合聯集、交集、差集、對稱差

集合在運用上，常見使用聯集(union)、交集(intersection)、差集(difference)、對稱差集(symmetric difference)四種。

以下以A集合set20為主，進行A、B(set25)兩個集合運算

聯集：將A集合與B集合內容集合起來，扣除掉重複項目

set20 = {1,6,8,10,20}

set25 = {1,3,8,10}

set20 = set20|set25

# set20 = set20.union(set25)

print(set20)

{1, 3, 6, 8, 10, 20}

交集：保留A、B兩集和共有內容

set20 = set20 & set25

# set20 = set20.intersection(set25)

print(set20)

{8, 1, 10}

差集：將A、B集合共有有部分去除，剩餘A的元素

set20 = set20 - set25

# set20 = set20.difference(set25)

print(set20)

{20, 6}

對稱差集：將A、B集合共有有部分去除，剩餘A與B的元素

set20 = set20 ^ set25

# set20 = set20.symmetric\_difference(set25)

print(set20)

{3, 6, 20}

* 1. 子集合、超集合、 集合內容比較

兩個集合A與B若A集合內容是B集合的一部分，則A是B的部分集合，B是A的超集合。例：

set15 = {1,3,8,10}

set20 = {1,3,8}

a = set20.issubset(set15)

b = set15.issuperset(set20)

print(a,b)

True True

集合可以利用 == 、 != 進行兩者間內容是否相等。例：

set30 = {1,8,3}

a = set20 == set30

print(a)

a = set20 != set30

print(a)

a = set15 == set30

print(a)

True

False

False

1. 字典

字典(dictionary)類似集合使用大括號，與串列、元組 、集合不同的是，每個元素內容都是由一個鍵值(key)與數值(value)組成的數對。

字典的特點在於，串列資料是透過索引碼如： lst2 [0]的方式去取得內容，而字典是透過自行設定易辨識的鍵值取得資料。

* 1. 建立字典

透過大括號建立字典，若大括號內容是空的，表示建立一個空的字典。例：

dic10 = {}      #建立一組空字典

print (dic10)

{}

指定內容，假設建立一個城市地標的資料組，以城市當作鍵值，地標為內容數值，建立方式如下：

dic10 = {'Taipei':'101','Paris':'Tour Eiffel','London':'Big Ben'}

print (dic10)

{'Taipei': '101', 'Paris': 'Tour Eiffel', 'London': 'Big Ben'}

初始資料建立後，若要新增一筆資料，則是透過鍵值與數值的對應新增，例如新增一筆城市為柏林，地標為圍牆的資料，語法如下：

dic10['Berlin'] = 'Wall'

print (dic10)

{'Taipei': '101', 'Paris': 'Tour Eiffel', 'London': 'Big Ben', 'Berlin': 'Wall'}

前面說過利用鍵值取得資料，比透過索引碼容易識別，方式如下：

a = dic10['Taipei']

print(a)

101

與前面幾項功能相同，也可以透過for迴圈輸出數對資料：

for i in dic10:

    print('%8s : %s' % (i , dic10[i]))

Taipei : 101

Paris : Tour Eiffel

London : Big Ben

Berlin : Wall

lst10 = sorted(dic10)  #將"鍵值"排序後存到串列

print('lst10=',lst10)

for i in lst10:       #使用串列內容當key調閱出字典內容值

    print('%8s : %s' % (i , dic10[i]))

lst10= ['Berlin', 'London', 'Paris', 'Taipei']

Berlin : Berlin Wall

London : Big Ben

Paris : Tour Eiffel

Taipei : 101

字典可以使用 len 來計算字典內有多少項目，利用in 和 not in 判斷某一鍵值是否存在於字典中，利用 == 、 != 條件判斷檢視兩個字典內容項目是否相等，(內容相同，不論順序是否一樣)。

計算字典內有多少項目

a = len(dic10)

print(a)

4

確認指定項目是否在字典內，注意檢查的值是**鍵值**

a = 'Taipei' in dic10

print(a)

True

a = 'Tainan' in dic10

print(a)

False

a = 'Taipei' not in dic10

print(a)

False

a = 'Tainan' not in dic10

print(a)

True

dic12 = {10:'John',30:'Peter',20:'Mary'}    #內容相同，不限順序

dic22 = {10:'John',20:'Mary',30:'Peter'}

a = dic12 == dic22

print('兩個字典相等：',a)

a = dic12 != dic22

print('兩個字典不等：',a)

兩個字典相等： True

兩個字典不等： False

字典可以透過del 刪除特定內容。例：

del dic10['Taipei']

print(dic10)

{'Paris': 'Tour Eiffel', 'London': 'Big Ben', 'Berlin': 'Wall'}

直接刪除整個字典：

del dic10

幾個專門檢查與調用字典內容的功能如下：

鍵值 keys()

print(dic10.keys())

dict\_keys(['Taipei', 'Paris', 'London', 'Berlin'])

數值 values()

print(dic10.values())

dict\_values(['101', 'Tour Eiffel', 'Big Ben', 'Wall'])

字典項目 items()

print(dic10.items())

dict\_items([('Taipei', '101'), ('Paris', 'Tour Eiffel'), ('London', 'Big Ben'), ('Berlin', 'Wall')])

將結果輸入到tuple

print(tuple(dic10.keys()))

('Taipei', 'Paris', 'London', 'Berlin')

print(tuple(dic10.values()))

('101', 'Tour Eiffel', 'Big Ben', 'Wall')

print(tuple(dic10.items()))

(('Taipei', '101'), ('Paris', 'Tour Eiffel'), ('London', 'Big Ben'), ('Berlin', 'Wall'))

字典也可以透過pop()，刪除某一鍵值的項目，popitem()表示刪除字典中最後一個項目，clear()表示清除字典中所有項目。

dic10.pop('Paris')

print(dic10)

{'London': 'Big Ben', 'Berlin': 'Berlin Wall'}

dic10['Taipei'] = '101'

print(dic10)

dic10.popitem()

print(dic10)

{'London': 'Big Ben', 'Berlin': 'Berlin Wall', 'Taipei': '101'}

{'London': 'Big Ben', 'Berlin': 'Berlin Wall'}

dic10.clear()

print(dic10)

{}

字典複製 copy() 與合併更新 update()

copy()是將一個字典複製到另一個字典，要注意的是，若原字典有資料，”會被完全覆蓋掉”

將dic1 複製到 dic3：

dic1 = {1:'rad',2:'Yellow',3:'Green'}

print(dic1)

dic2 = {4:'black',1:'Blue'}

print(dic2)

dic3 = dic1.copy()

print(dic3)

{1: 'rad', 2: 'Yellow', 3: 'Green'}

{4: 'black', 1: 'Blue'}

{1: 'rad', 2: 'Yellow', 3: 'Green'}

update()是將兩個字典合併，若有兩個相同鍵值，後面資料會覆蓋原資料。

將dic2，合併到dic3

dic3.update(dic2)

print(dic3)

{1: 'Blue', 2: 'Yellow', 3: 'Green', 4: 'black'}

練習題

1. 輸入數個整數並儲存至**串列**中，直到輸入-9999結束，再將此串列轉換成**元組** ，最後顯示該元組 以及長度、最大值、最小值、總和。
2. 輸入數個整數並建立兩個**元組** ，輸入-9999結束，將此兩元組 合併並從小到大排序，顯示排序前的元組 和排序後的串列。
3. 輸入至少五個字串至**元組** ，直到輸入’end’結束，接著輸出這個元組 ，再分別顯示元組 的第一個元素到第三個元素，和倒數第三個元素。
4. 輸入數個整數儲存到**集合**，直到-9999結束，顯示這個集合的長度、最大值、最小值、總和。
5. 依序輸入五個、三個、九個整數，儲存到set1、set2、set3當中，查詢set2是否為set1的子集合?set3是否為set1的超集合?
6. 課程分組，分為X與Y兩組，輸入X和Y兩組需要學習的科目到集合中，以字串”end”作為結束，請依序分行顯示
7. X組和Y組所有的科目
8. X組和 Y組共同科目
9. Y組有但是X組沒有的科目
10. X有Y沒有以及Y有X沒有的科目
11. X組: 國文、英文、數學乙、地理、歷史。 Y組：國文、數學甲、英文、物理、化學
12. 自行輸入兩個字典，以輸入值end作為結束，將這兩個字典合併，根據key值字母，由小到大排序輸出，如有重複key值，後輸入的key值覆蓋前面的key值。
13. 輸入顏色字典color\_dict 內容，直到鍵值輸入end為止。再依據鍵值字母由小到大排序輸出
14. 輸入資料到字典中，直到輸入鍵值為end結束，再輸入一個鍵值，進行搜尋這個鍵值是否存在字典中。
15. 輸入五筆資料放在 tuple10 的元組 ，之後印出元組 每一個元素，以及找出此元組 最大值、最小值、總和。