Python_08

結構化資料操作

串列(List)

- 串列是Python內建的資料型別，也是一種Python內建物件串列物件提供許多操作方法，讓我們來使用這個物件。
- 串列的序列結構，也稱為陣列(Array)。

串列的操作

- 串列的建立
- 將其他資料轉為串列
- 從串列取出項目
- 串列中的串列
- 更改串列項目的值
- 增加串列項目
- 串列的結合
- 取得串列項目的位移值
- 串列項目的刪除
- 串列項目的計算
- 串列的指派與複製
- 串列的排序

串列的建立

- 透過[ ] , list( ) , for

```
#使用[]
M = []
print('M:',M)

L = ['A','B','C','D']
print('L:',L)

#使用字串
W = list('ABCD')
print('W:',W)

#使用迴圈產生二維陣列
L2D = [[i,j] for i in range(0,3) for j in range(0,4)]
print('L2D:',L2D)
```

將其他資料轉為串列

```
#string to list
words = list('CAT')
print(words)

#tuple to list
a_tuple = ('a','b','c')
print(list(a_tuple))

#string to list
birthday = '1/6/1952'
blist = birthday.split('/')
print(blist)

#string to list
splitstr = 'a/b//c//d///e'
splitlist = splitstr.split('//')
print(splitlist)
```

輸出

```
['C', 'A', 'T']
['a', 'b', 'c']
['1', '6', '1952']
['a/b', 'c', 'd', '/e']
```

從串列取出項目

- 透過 [ ] 與 位移值offset 來取出項目

```
#設定值
n = [10, 11, 12, 13, 14, 15]
print(n[0])
print(n[1])
print(n[-1])

#取出到 offset = 3 停止
print(n[0:3])

#從開始以 offset = 2 為間隔
print(n[::2])

#反轉
print(n[::-1])
```

二維串列

- 串列的項目

```
n = [[10, 11], [12, 13], [14, 15], [16, 17]]

#由offset取出
print(n[0])
print(n[1])
print(n[0][1])

#從開始以2為間隔
print(n[::2])
print(n[::2][1])
print(n[::2][1][0])
```

更改串列項目的值

- 透過 [ ] 與 位移值offset 來修改項目

```
A = ['B','C','D','E']

#修改一個項目
A[0] = 'A'
A[2] = 'F'
print(A)

# 修改一段
A[1:3] = ['','']
print(A)
```

增加串列項目

- 透過 append()附加項目，insert()插入項目
- 使用 extend() 與 += 結合串列

```
L = ['A','B','C','D']

L.append('F')
print(L)

L.insert(3,'E')
print(L)

M = ['G','H']
L.extend(M)
print(L)

L += ['I','J']
print(L)

L += 'K'
print(L)
```

取得串列項目的位移值

- 使用index()取得項目的位移值

```
L = ['A','B','C','D','E']

#使用項目值取得
print('B\'s index is ',L.index('B'),sep='')

#使用迴圈逐一取得
for item in L:
    print(item,'\'s index is ',L.index(item),sep='')
```

串列項目的刪除

- remove() 、 del、 pop()

```
n = ['aa', 'bb', 'cc', 'dd', 'ee', 'ff']

# remove()
print('remove: aa')
n.remove('aa')
print(n)

#del list[index]
print('del: ff')
del n[-1]
print(n)

# pop()預設從最後一個彈出並刪除
print('pop:',n.pop())
print(n)

# pop()可指定位置
print('pop:',n.pop(2))
print(n)
```

字串與串列與整合

- Join()是字串的函式

```
Animals = ['Bat','Rat','Elephant','Cat']
seprator = ' , '

join_str = seprator.join(Animals)
print(join_str)

splitlist = join_str.split(',')
print(splitlist)
```

串列項目的計算

- In 判斷是否在串列中
- count()計算項目數量
- len()計算串列長度(總數)

```
L = ['A','A','B','C','B','A','B','A']

print('A 出現的次數:',L.count('A'))
print('L 的長度:',len(L))

if 'C' in L:
    print('C 的索引值:',L.index('C'))
```

串列的指派與複製

- = 用於指派
- copy()、list()、[:] 用於複製

```
a = [1, 2, 3, 4, 5]
b = a
print('a id:', id(a))
print('b id:', id(b))

c = a.copy()
d = list(a)
e = a[:]

print('c id:', id(c))
print('d id:', id(d))
print('e id:', id(e))

b[2] = 10
print('a',a)
print('b',b)
print('c',c)
print('d',d)
print('e',e)
```

串列項目的排序

- Sort()

```
a = [6, 2, 5, 4, 3, 1]

a.sort()
print(a)

a.sort(reverse=True)
print(a)

a = sorted(a)
print(a)

b = ['b', 'd', 'e', 'a', 'c', 'f']

b.sort()
print(b)

b.sort(reverse=True)
print(b)

b = sorted(b)
print(b)
```

值組(Tuple)

- 值組是Python內建的資料型別，也是一種Python內建物件，物件提供許多操作方法。
- 值組常用於函式的不限數量參數宣告。
- 與串列不同，值組是不可修改的，宣告之後的數值無法修改。

值組的操作

- 值組的建立
- 值組的項目
  - count()
  - index()
  - 替變數賦值
  - 不定數量參數

值組的建立

- 用()來建立值組(tuple)

```
t = ()
print(type(t),t)

t = ('aa')  #不是值組
print(type(t),t)

t = 'aa',
print(type(t),t)

t = 'aa', 'bb', 'cc'
print(type(t),t)

t = ('aa', 'bb', 'cc')
print(type(t),t)

t = 'aa, bb, cc' #不是值組
print(type(t),t)
```

值組項目

- count()、index()、替變數賦值、不限數目參數

```
t = 10, 30, 20, 40, 50, 10, 20, 30

print('值組中20的數量：',t.count(20))
print('值組中50的索引值', t.index(50))

v = 1, 2, 3
a, b, c = v
print('a+b+c =',a+b+c)

def tuple_test(*p):
    for item in p:
        print(item)

tuple_test(11, 22, 33)
tuple_test(44, 55)
```

字典(Dictionay)

- 字典是Python內建的資料型別，也是一種Python內建物件，物件提供許多操作方法。
- 與串列不同，字典不是序列的，每個內部項目，由鍵(Key)與值(Vale)組成。
- 鍵(Key)必需是唯一，由鍵(Key)找到配對的值(Vale)。

字典的操作

- 字典的建立
- 將資料轉換成字典
- 取得字典裡的資料
- 使用鍵值變更字典項目
- 合併兩份字典資料
- 刪除字典資料
- 判斷是否為字典鍵值
- 字典的指派與複製

字典的建立

- 使用{ Key:Vaule}建立字典、使用迴圈生成字典

```
#空字典
d = {}
print(type(d))

#字典
d = {'a': 5,
     'b': 4,
     'c': 5}

#透過迴圈產生字典
k = ['a','b','c']
v = [10,20,30]

d = {k[i]:v[i] for i in range(len(k))}
print(d)
```

將資料轉換成字典

- 透過dict()把成對的資料轉成字典

```
L = [[0,0],[1,1],[2,2]]
a = dict(L)
print(a)

L = [('a1','b1'),('a2','b2'),('a3','b3')]
b = dict(L)
print(b)

L = [[0,[1,3]],
     [1,[20,4]],
     [2,5]]

c = dict(L)
print(c)
```

將資料轉換成字典

- 透過 zip()將兩個串列轉為字典

```
f = ['Apple','Orange','Banana']
f_zh = ['蘋果', '橘子','香蕉']

print(list(zip(f,f_zh)))

d = dict(zip(f,f_zh))
print(d)
```

取得字典裡的資料

- 透過 key、get()、keys、values()、items() 取得字典的資料

```
d = { 'Jim' : '135', 'Amy': '150',
      'Tom': '150', 'Bob': '142' }

#使用 key 取出一筆資料
print(d['Bob'])

#使用 get()
print(d.get('Jim'))

#取出所有的鍵
print(d.keys())

#取得所有的值
print(d.values())

#取得所有鍵值配對
print(d.items())
```

使用鍵值變更字典項目

- 透鍵值可以修改或新增字典項目，字典的每個鍵值都要獨一無二。

```
d = { 'Jim' : '135', 'Amy': '150',
      'Tom': '150', 'Bob': '142' }

#更新字典
d['Amy'] = 140

#增加新的項目
d['Ann'] = 151

print(d)
```

合併兩份字典資料

- 使用update()合併字典資料

```
d = {'Jim':135,'Amy': 150,'Tom': 150, 'Bob': 142}

e = {'Joe':138,'Tim': 152,'Bob': 145}

d.update(e)

print(d)
```

刪除字典資料

```
d = {'Jim':135,'Amy': 150,'Tom': 150, 'Bob': 142,
     'Joe':138,'Tim': 152}

#使用 del 刪除
del d['Jim']
print('del Jim :',d)


#使用 pop()
print('pop Amy:',d.pop('Amy'))
print(d)

#使用 popitem()
print('popitem:',d.popitem())
print(d)

#clear()
d.clear()
print('clear:',d)
```

判斷是否為字典鍵值

- 使用 in判斷資料是否為字典鍵值

```
d = {'Jim':135,'Amy': 150,'Tom': 150, 'Bob': 142}

if 'Jim' in d:
    print(d['Jim'])
```

字典的指派與複製

- 用 = 來指派, 用copy()來複製

```
d = {'Jim':135,'Amy': 150,'Tom': 150, 'Bob': 142}

#Assign
e = d

#建立複製
f = d.copy()

d['Amy'] = 155

print('d=',d)
print('e=',e)
print('f=',f)
```

集合的操作

- 集合的建立
- 資料轉換成集合
- 集合元素的新增與刪除
- 集合的計算
- 集合與運算子

集合的建立

- 使用 set( ) 來建立或 { 鍵值,...}
- 物件值可修改，指派 = 與複製 copy() 類似字典
- 用 in來測試鍵值是否在集合中

```
#空集合 要用set(), 不是{}
e = set()
print(type(e))

#集合的每一個鍵都是獨一無二的
even = {0, 2, 4, 6, 8, 10, 0, 2}
print(even)

# print(even_set)
odd ={1, 3, 5, 7, 9}
print(odd)
```

資料轉換成集合

- 透過set()轉換成集合刪除重複的值

```
#字串轉set
set_A = set('letters')
print(set_A )

#串列轉set
set_B = set([10,15,20,20,30,40])
print(set_B)

#值組轉set
set_C = set((10,15,20,20,30,40))
print(set_C)

#字典轉set
set_D = set({'a':10,'b':20,'c':30})
print(set_D)
```

集合元素的新增與刪除

- 集合新增使用add() 、刪除可用remove()、discard()、clear()、pop()

```
set_A = {2, 8, 10, 22, 11, 12}

set_A.add(25)
print(set_A)

set_A.remove(2)
print(set_A)

set_A.discard(2) #OK,不存在可以 discard
print(set_A)

set_A.discard(8)
print(set_A)

set_A.pop()
print(set_A)

set_A.remove(8)  #Error,不存在不能 remove
print(set_A)
```

集合的計算

- 交集、聯集、差集、對稱差集

```
set_A = {2, 8, 10, 22, 11, 12}
set_B = { 6, 7, 10, 11, 15 }
set_C = {8,12}

print('set_A 交集 set_B',set_A.intersection(set_B))
print('set_A 聯集 set_B',set_A.union(set_B))
print('set_A 差集 set_B',set_A.difference(set_B))
print('set_A 與 set_B 對稱差 ',set_A.symmetric_difference(set_B))
print('set_C 是 set_A 的子集嗎?',set_C.issubset(set_A) )
```

集合與運算子

- &(交集) , | (聯集) , - (差集) , ^ (對稱差), <= (子集), >= (父集)

```
set_A = {2, 8, 10, 22, 11, 12}
set_B = {6, 7, 10, 11, 15 }
set_C = {8, 12, 22, 24}


if set_C & set_A:
    print('set_C & set_A is', set_C & set_A)
else:
    print('set_C & set_A is empty set')

if set_C & set_B:
    print(print('set_C & set_B is', set_C & set_B))
else:
    print('set_C & set_B is empty set')
```

單元作業

- 請將下列字典型態資料，按照鍵值的英文字母順序排列。
  {'Jim':135,'Amy': 150,'Tom': 150, 'Bob': 142}