python中兩種基礎資料結構：list、字典

0_2_Data Structures

List 串列

- a list of comma-separated values (items) between square brackets [xxx, ooo, ***, ...]
- contain items of **different** types
- mutable type **– changeable**
  - it is possible to change their content
- allow **duplicate** values
- ordered

程式碼中靈活的清單：用方括號包起來
裡面可以放數字、文字
可變的：可以隨時增加、刪除、修改裡面的東西
裡面的項目有固定順序

Create List

```
squares = [1, 4, 9, 16, 25]
print('squres', squares)
empty1 = []
print('empty1 ', empty1)
empty2 = list()
print('empty2', empty2)
```

```
squares [1, 4, 9, 16, 25]
empty1 []
empty2 []
```

List index and slice

> list[n] : index
> returns the nth item n : 0 - len(list)-1
> list[n:m] : slice
> returns a new list from n to m-1 elements

List idex and slice Example

```
squares = [1, 4, 9, 16, 25]
print("squares      ", squares)
print("squares[0]   ", squares[0])    # indexing returns the first item
print("squares[-1]  ", squares[-1])   # indexing returns the last item
print("squares[1:4] ", squares[1:4])  # slicing returns list 1st to 3rd
print("squares[-3:] ", squares[-3:])  # slicing returns list -3 to last
```

```
squares      [1, 4, 9, 16, 25]
squares[0]   1
squares[-1]  25
squares[1:4] [4, 9, 16]
squares[-3:] [9, 16, 25]
```

List change content

```
cubes = [1, 8, 27, 65, 125]  # something's wrong here
print('cubes', cubes)
cubes[3] = 64  # replace the wrong value
print('after cubes[3] = 64\ncubes', cubes)
```

```
cubes [1, 8, 27, 65, 125]
after cubes[3] = 64
cubes [1, 8, 27, 64, 125]
```

List change slices

```
letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
print(letters)
letters[2:5] = ['1', '2', '3'] # replace some values
print(letters)
letters[2:5] = [] # now remove them
print(letters)
# clear the list by replacing all the elements with an empty list
letters[:] = []
print(letters)
```

```
['a', 'b', 'c', 'd', 'e', 'f', 'g']
['a', 'b', '1', '2', '3', 'f', 'g']
['a', 'b', 'f', 'g']
[]
```

List operation

```
print(['Yes! '] + [1, 2, 3]) # operation +
print(['Yes! '] * 3) # operation *
```

```
['Yes! ', 1, 2, 3]
['Yes! ', 'Yes! ', 'Yes! ']
```

List add items

```
squares2 = [1, 2, 4, 8, 16]
print(squares2)
squares2.append(64)    # add item at the end of the list
print(squares2)
squares2.insert(5, 32) # add itenm at the index of the list
print(squares2)
```

```
[1, 2, 4, 8, 16]
[1, 2, 4, 8, 16, 64]
[1, 2, 4, 8, 16, 32, 64]
```

List delete items

```
fruits = ["apple", "banana", "cherry", "pear"]
print(fruits)
# remove() method removes the specified item
fruits.remove("banana")
print(fruits)
#pop() method removes the specified index
fruits.pop(1)
print(fruits)
# clear() method delete all items
fruits.clear()
print(fruits)
```

```
['apple', 'banana', 'cherry', 'pear']
['apple', 'cherry', 'pear']
['apple', 'pear']
[]
```

List length

len() Return the length (the number of items) of an object

```
letters = ['a', 'b', 'c', 'd']
len(letters)
```

```
4
```

List count() method

```
points = [1, 4, 2, 9, 7, 8, 9, 3, 1]
num_9 = points.count(9)
print(num_9)
```

```
2
```

Dictionary(字典)

> {key1:value1, key2:value2, ...}
>
> Dictionary is a comma-separated list of key:value pairs within the braces.

Creating a Dictionary

```
emptydic = {}
tel = {'jack': "0933123456", 'sape': "0963123456"}
fruit = dict([('apple', "蘋果"), ('orange', "橘子")])
weight = dict(john=50, mary=45)

print("Empty :     ", emptydic)
print("telephon : ", tel)
print("fruit :     ", fruit)
print("weight :    ", weight)
```

```
Empty :      {}
telephon :  {'jack': '0933123456', 'sape': '0963123456'}
fruit :      {'apple': '蘋果', 'orange': '橘子'}
weight :     {'john': 50, 'mary': 45}
```

Adding elements to a Dictionary

```python
# Creating an empty Dictionary
Dict = {}
print("Dict: ", Dict)

# Adding elements one at a time
Dict[0] = 'Geeks'
Dict[2] = 'For'
Dict[3] = 1

print("\nDictionary after adding 3 elements:\nDict: ", Dict)
```

```
Dict:  {}

Dictionary after adding 3 elements:
Dict: {0: 'Geeks', 2: 'For', 3: 1}
```

Updating Dictionary Key's Value

```python
# Creating a Dictionary
Dict = {0: 'Geeks', 2: 'For', 3: 1}
print("Dict: ", Dict)

# Updating existing Key's Value
Dict[2] = 'Welcome'
print("\nUpdated key value: Dict[2] = 'Welcome' \nDict ", Dict)
```

```
Dict:  {0: 'Geeks', 2: 'For', 3: 1}

Updated key value: Dict[2] = 'Welcome'
Dict {0: 'Geeks', 2: 'Welcome', 3: 1}
```

Accessing elements of a Dictionary

```python
# Creating a Dictionary
Dict = {1: 'Geeks', 'name': 'For', 3: 'Geeks'}
print("Dict: ", Dict)

# accessing a element using key
print("\nAccessing a element using key: Dict['name'] = ", Dict['name'])

# accessing a element using key
print("\nAccessing a element using key: Dict[1] = ", Dict[1])
```

```
Dict:  {1: 'Geeks', 'name': 'For', 3: 'Geeks'}

Accessing a element using key: Dict['name'] =  For
```

```
Accessing a element using key: Dict[1] =  Geeks
```