

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
-

Create List

```
1 squares = [1, 4, 9, 16, 25]
2 print('squares', squares)
3 empty1 = []
4 print('empty1 ', empty1)
5 empty2 = list()
6 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 index and slice Example

```
1 squares = [1, 4, 9, 16, 25]
2 print("squares      ", squares)
3 print("squares[0]   ", squares[0])    # indexing returns the first item
4 print("squares[-1]  ", squares[-1])   # indexing returns the last item
5 print("squares[1:4] ", squares[1:4])   # slicing returns list 1st to 3rd
6 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

```
1 cubes = [1, 8, 27, 65, 125] # something's wrong here
2 print('cubes', cubes)
3 cubes[3] = 64 # replace the wrong value
4 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

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

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

List operation

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

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

List add items

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

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

List delete items

```
1 | fruits = ["apple", "banana", "cherry", "pear"]
2 | print(fruits)
3 | # remove() method removes the specified item
4 | fruits.remove("banana")
5 | print(fruits)
6 | #pop() method removes the specified index
7 | fruits.pop(1)
8 | print(fruits)
9 | # clear() method delete all items
10 | fruits.clear()
11 | print(fruits)
```

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

List length

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

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

4

List count() method

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

```
1 | emptydic = {}
2 | tel = {'jack': "0933123456", 'sape': "0963123456"}
3 | fruit = dict([('apple', "蘋果"), ('orange', "橘子")])
4 | weight = dict(john=50, mary=45)
5 |
6 | print("Empty :      ", emptydic)
7 | print("telephon : ", tel)
8 | print("fruit :     ", fruit)
9 | 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

```
1 # Creating an empty Dictionary
2 Dict = {}
3 print("Dict: ", Dict)
4
5 # Adding elements one at a time
6 Dict[0] = 'Geeks'
7 Dict[2] = 'For'
8 Dict[3] = 1
9
10 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

```
1 # Creating a Dictionary
2 Dict = {0: 'Geeks', 2: 'For', 3: 1}
3 print("Dict: ", Dict)
4
5 # Updating existing Key's Value
6 Dict[2] = 'Welcome'
7 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

```
1 # Creating a Dictionary
2 Dict = {1: 'Geeks', 'name': 'For', 3: 'Geeks'}
3 print("Dict: ", Dict)
4
5 # accessing a element using key
6 print("\nAccessing a element using key: Dict['name'] = ", Dict['name'])
7
8 # accessing a element using key
9 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
```