

数据结构

一、 列表

1. 列表的创建

- (1) 使用赋值运算符直接创建列表

```
Listname=[element1,element2,element3...element n]
```

其中, element 可以为不同的数据类型, 但是通常情况下为了提高程序的可读性, 我们不这样做。

- (2) 通过 list() 函数将可迭代对象转化为列表

`list(data)` 其中 data 是可以转化为列表的数据, 包括 range 对象、字符串、元组

- (3) 列表推导式

i. 生成指定范围的数值列表

```
list=[expression for var in range]
```

eg: 生成 10 个范围 10-100 的随机数列表

```
>>> import random
>>> randomnum=[random.randint(10,100) for i in range(10)]
>>> print(randomnum)
[88, 95, 100, 71, 72, 52, 79, 94, 95, 43]
```

ii. 根据列表生成指定需求的列表

```
newlist=[expression for var in list]
```

eg: 定义一个记录价格的列表, 然后生成一个全部商品打 5 折的列表

```
>>> price=[100, 90, 75, 40, 80]
>>> sale=[x*0.5 for x in price]
>>> print(sale)
[50.0, 45.0, 37.5, 20.0, 40.0]
```

iii. 从列表中选择符合条件的元素组成新的列表

```
newlist=[expression for var in list if condition]
```

eg. 商品高于 500 元时打 8 折

```
>>> price=[100, 90, 75, 40, 80, 500, 1000]
>>> sale=[x*0.8 for x in price if x>=500]
>>> print(sale)
[400.0, 800.0]
```

2. 列表的遍历

i. 直接使用 for 循环: `for item in listname:`

ii. 使用 for 循环和 enumerate() 函数: `for index, item in enumerate(listname):` 注: index 从 0 开始

3. 列表的添加和删除

- (1) 添加元素

i. 添加单个元素:

```
listname.append(obj) listname.insert(obj, index)
```

ii. 将一个列表全部元素添加到另一个列表后边

```
listname.extend(seq)
```

- (2) 删除元素

i. 根据索引删除: `del list[index]` `list.pop()` 删除最后一个元素

ii. 根据元素值删除: `list.remove(val)`

4. 列表的排序

(1). 列表对象的 sort() 方法

```
Listname.sort(key=None, reverse=False)
```

Key: 指定排序规则, 例如 key=str.lower 表示不区分大小写

Reverse: False 表示升序排列, True 表示降序排列

(2) 使用内置的 sorted() 函数

使用该函数排序后原列表中的元素顺序不变, 如法格式如下:

```
Newlist=sorted(list, key=None, reverse=False)
```

5. 二维列表的创建

(1) 直接定义二维列表

```
Listname=[[element11,element12...element 1n],[element21,...element2n]...[element  
n1,element n2,...element nn]]
```

(2) 使用 for 嵌套

```
arr=[] #创建空列表  
for i in range(4):  
    arr.append([]) #在空列表中添加一个空列表  
    for j in range(5):  
        arr[i].append(j) #为内层列表添加元素
```

(3) 列表推导式

```
>>> arr=[[j for j in range(5)] for i in range(4)]  
>>> print(arr)  
[[0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4]]
```

二、元组(tuple())

元组和列表都属于序列, 他们都是按照一定顺序存放一组数据, 类型不受限制。其实它的大部分操作与列表和字符串都是一样的, 列表和元组唯一区别就是列表支持修改操作, 而元组不支持修改, 是不可变的。具体而言, 列表和元组的区别体现在以下几方面:

1、列表属于可变序列, 它的元素可以随时修改或者删除; 元组属于不可变序列, 其中的元素不可以修改, 除非整体替换。

```
>>> coffee=(' 蓝山',' 卡布奇诺',' 曼特宁',' 摩卡',' 麝香猫',' 哥伦比亚')  
>>> coffee[4]='shexiangmao'  
Traceback (most recent call last):  
  File "<pyshell#57>", line 1, in <module>  
    coffee[4]='shexiangmao'  
TypeError: 'tuple' object does not support item assignment
```

元组是不可变序列, 所以我们不能对其单个元素进行修改, 我们只能通过重新赋值的方式来修改。

```
>>> coffee=(' 蓝山',' 卡布奇诺',' 曼特宁',' 摩卡',' 麝香猫',' 哥伦比亚')  
>>> coffee=(' 蓝山',' 卡布奇诺',' 曼特宁',' 摩卡',' shexiang',' 哥伦比亚')  
>>> print(coffee)  
( ' 蓝山', ' 卡布奇诺', ' 曼特宁', ' 摩卡', ' shexiang', ' 哥伦比亚')
```

2、列表可以使用 append、extend、insert、remove、pop 等方法修改元素, 而元组没有这些方法。

```
>>> coffee=coffee+(' shex',)  
>>> print(coffee)  
( ' 蓝山', ' 卡布奇诺', ' 曼特宁', ' 摩卡', ' 麝香猫', ' 哥伦比亚', ' shex')
```

元组支持多个元组相加的操作, 可以向元组后边加入新元组

3、元组比列表的访问和处理速度更快, 所以当只是需要对其中的元素进行访问, 而不进行任何修改时, 建议使用元组。

4、列表不能作为字典的键, 而元组可以。

三、字典

1、字典的创建

(1) 直接创建字典: `dictionary={'key1'='value1', 'key2'='value2' ...}`

(2) 通过映射函数创建字典: `dictionary=dict(zip(list1,list2))`

(3)通过给定关键字参数创建字典: `dictionary=dict(key1=value1, key2=value2...)`

(4)通过元组和列表创建字典:

```
>>> coffee=('蓝山','卡布奇诺','曼特宁','摩卡','麝香猫','哥伦比亚')
>>> list=[1,2,3,4,5,6]
>>> dic={coffee:list}
>>> print(dic)
{('蓝山', '卡布奇诺', '曼特宁', '摩卡', '麝香猫', '哥伦比亚'): [1, 2, 3, 4, 5, 6]}
```

2、字典的访问

(1)`dictionary['key']`

(2)`dictionary.get(key)`

3、字典的遍历

使用字典的 `item()` 方法可以获取字典的键值对, 语法格式为: `dictionary.item()`

```
>>> dic={'qq':'4006751066','明日科技':'0431-84978983','无语':'0213-23421243'}
>>> for item in dic.items():
...     print(item)
...
('qq', '4006751066')
('明日科技', '0431-84978983')
('无语', '0213-23421243')

>>> dic={'qq':'4006751066','明日科技':'0431-84978983','无语':'0213-23421243'}
>>> for key,value in dic.items():
...     print(key,value)
...
qq 4006751066
明日科技 0431-84978983
无语 0213-23421243
```

注: `dictionary` 提供了 `values()` 和 `keys()` 方法, 用于返回字典的值和键的列表

4、字典的添加、修改、删除

(1)添加和修改: `dictionary[key]=value` 当 `key` 不存在时即为添加, 存在时为修改

(2)删除: `del dictionary[key]`

5、字典推导式

Eg: `ran={i:random.randint(10,100) for i in range(5)}`

```
Name=['香凝','黛蓝','伊一']
Sign=['水瓶','射手','双鱼']
Dic={i:j+'座' for i,j in zip(name,sign)}
```

四、集合

1、集合的创建

(1)直接创建: `set={element1,element2,element3...}`

(2)`set` 函数创建: `setname=set(iteration)` 其中 `iteration` 可以是列表、元组、`range` 对象等

2、集合的添加和删除

(1)添加: `setname.add(element)`

(2)删除: `setname.remove(element)`//删除指定元素 `setname.pop()`//删除最后元素

`setname.clear()`//清空集合所有元素

3、集合的交并和差集运算

交集运算: `&` 并集运算: `|` 差集运算: `-` 对称差运算: `^`