

2.数据结构

列表

- 定义列表

- 列表的增

- 列表的删除

- 列表修改

- 查找元素

- 其他操作

元组

- 可变元组

字典

- 增加键值

- 删除/修改键值

- 查找键值

- 其他操作

集合

- 定义，增加，删除，查找元素

- 关系

- 不可变集合

循环

- 终止语句

- 随机数

列表

是一个有序的可变集合，可以存储不同类型的元素。

定义列表

```
1 fruits = ["apple", "banana", "cherry"]
2 print(fruits)
3 print(fruits[1])#输出banana
```

列表的增

1.索引位置增加list.insert()

2.末尾增加list.append()

3.迭代增加list.extend()

```
1 fruits = ["apple", "banana", "cherry"]
2 fruit.insert(1,"kiwi")#索引位置增加
3 fruit.append("orange")#末尾添加
4 fruit.extend(['a','b','c'])#迭代添加
5
```

列表的删除

1.删除具体值list.remove()

2.删除索引位置list.pop()

3.删除切片范围del list[x:y]

4.清空列表list.clear

5.删除列表del list

▼ 删除元素

Python |

```
1 fruits = ["apple", "banana", "cherry"]
2 fruits.remove("cherry")
3 fruits.pop(1)#删除索引位置
4 del fruits[0:2]#删除切片范围
5 fruits.clear()#清空列表
6 del fruits#删除列表
7 print (fruits)
```

列表修改

1.按索引修改元素

2.按切片范围修改

▼ 修改元素

Python |

```
1 fruits = ["apple","banana","cherry"]
2 fruits[1]="orange"
3 print(fruits)
4 fruits[0:2]=["kiwi","orange"]
5 print(fruits)
```

查找元素

1.找到元素的索引fruits.index("y"),只能找到第一个y的位置，其他的找不到

2.统计y出现的次数fruits.count("y")

▼ 查找和统计

Python |

```
1 fruits = ["apple", "banana", "cherry","cherry"]
2 print(fruits.index("banana"))
3 print(fruits.count("cherry"))
```

其他操作

1.列表切片

2.排序列表list.sort()

3.列表中的元素反过来list.reverse()

```
1  #1.列表切片
2  list=[1,2,6,9,8,4,5]
3  print(list[0:5:2])#步长2
4  #2.排序列表
5  list=[1,2,6,9,8,4,5]
6  list.sort()
7  print(list)
8  #3.列表中的元素反过来
9  list=[1,2,6,9,8,4,5]
10 list.reverse()
11 print(list)
```

元组

有序的不可变集合，只能读的列表。

可变元组

tuple 其实不可变的是地址空间，列表可变

```
1  tuple = (1,2,3,'a','b','c')
2  tuple1 = (1,2,3,['a','b','c'])
3  print(tuple1[3])
4  tuple1 [3][0]='h'#把tuple1中['a','b','c']中的第0个元素改成'h'
5  print(tuple1)
```

字典

是映射类型，键值对（key-value）的形式存储数据。

key具有唯一性，不可重复，值无所谓

可哈希：数据类型不可变，如数字，字符串，元组

不可哈希：可变数据类型

增加键值

1.直接通过键值对来增加

2.指定的键已经存在则不做操作，原字典中不存在指定的键值对，则会添加。

▼ 定义和增加

Python |

```
1  #定义字典
2  dic = {"name":"nls", 'age':18, 'job':'teacher'}
3  print(dic["name"])
4
5  #增加键值
6  dic = {"name":"nls", 'age':18, 'job':'teacher'}
7  #1. 直接通过键值对来增加
8  dic['sex']="男"
9  print(dic)
10 #2. 不存在指定的键值对，则会添加
11 dic.setdefault("name", "zls")
12 dic.setdefault("hobby", "玩")
13 print(dic)
```

删除/修改键值

1.删除指定键，返回其值dic.pop('键名')

2.删除指定键值对del dic['key']

3.删除最后插入的键值对dic.popitem

4.清空字典dic.clear

修改键值dic['key']=value

```
1 dic = {'name':'nls','age':18,'job':'teacher'}
2 #1. 删除指定的键,返回其值
3 name = dic.pop('job')
4 print(name)
5 hobby = dic.pop('hobby','查无此项')
6 print(hobby)
7 #2. 删除指定键值对
8 dic = {'name':'nls','age':18,'job':'teacher'}
9 del dic['name']
10 print(dic)
11 #3. 删除最后插入的键值对
12 dic = {'name':'nls','age':18,'job':'teacher'}
13 a=dic.popitem()
14 print(a)
15 #4. 清空字典
16 dic = {'name':'nls','age':18,'job':'teacher'}
17 dic.clear()
18 print(dic)
19 #修改键值
20 dic = {'name':'nls','age':18,'job':'teacher'}
21 dic['age'] = 25
22 print(dic)
```

查找键值

1. 直接通过键名获取
2. 使用get方法获取键值,若不存在则返回 None, 可以自定义异常返回值
3. IN关键字, 存在返回True, 反之False

```
1 dic = {'name':'nls','age':18,'job':'teacher'}
2 print(dic['age'])#直接键名获取
3 print(dic.get('job','查无此项'))#get方法获取
4 exists="name" in dic
5 print(exists)
```

其他操作

可以通过keys()和values()方法，获取键和值

Python

```
1 dic = {'name': 'nls', 'age': 18, "phone": ['18888888888', '0511-10101010']}
2 # 1. 对键和值进行迭代操作
3 for i in dic.keys():
4     print(i)
5 # 只迭代键
6 for i in dic.values():
7     print(i)
```

集合

无序的，不重复，确定性的数据集合，元素可哈希。

定义，增加，删除，查找元素

定义：set1 = {1,2,3,4,5,'a','b'}

增加：1.set1.add('x')

2.set1.update() #update接收的参数应该是可迭代的数据类型

删除：1.set1.remove('x')

2.set1.pop() #随机删除某个元素

查找：exists = "a" in set1 #判断a是否在集合里

关系

交集：print(set1 & set2) 反交集：print(set1 ^ set2)

并集：print(set1 | set2) 差集：print(set1 - set2)

子集：print(set1 < set2) 超集：print(set2 > set1)

不可变集合

set2=frozenset(set1)

循环

终止语句

break: 用于完全结束一个循环, 跳出循环体执行循环后面的语句

continue: 只是终止本次循环, 接着还执行后面的循环, break 则完全终止循环

while...else: 当 while 循环执行完, 中间没被 break 中止的话, 就会执行 else 后面的语句

随机数

导入模块import random

num=random.randint(1,100)#给定int型的数