

## 列表的操作-移除元素pop(重点)

```
1 pop(位置) 移除列表中指定索引处的元素，默认移除的是最后一个元素，返回的是被移除的数据
2 mylist=[1,2,3,4,5]
3 mylist.pop()
4 print(mylist)#[1, 2, 3, 4]
5
6 n=mylist.pop(2)
7 print(mylist)#[1, 2, 4]
8 print(n)#3
9
10 #案例:有列表['1','2','3','Maker','4',,'5','ceshi'],有7个元素
11 #把列表中的非数字字符串移除,并打印移除的字符串,然后计算列表的数字之和
```

## 列表的操作-移除不返回remove(重点)

```
1 remove(元素) 移除指定元素在列表中匹配到的第一个元素【从左往右】
2 mylist=[1,2,8,9,7,2,3,4,5,6,7,8,12,13,14]
3 mylist.remove(6)
4 print(mylist)#[1, 2, 8, 9, 7, 2, 3, 4, 5, 7, 8, 12, 13, 14]
5 mylist.remove(7)
6 print(mylist)#[1, 2, 8, 9, 2, 3, 4, 5, 7, 8, 12, 13, 14]
7
8 #案例:给定一个列表,首先删除以s开头的元素,删除后,修改第一个元素为"maker",并且并且把最后一个元素复制一份,放在joke的后边
9
```

## 列表的操作-清空clear

```
1 clear() 清除列表中的所有的元素
2 mylist=[1,2,3]
3 mylist.clear()
4 print(mylist)#[]
5 # mylist[0]=100报错
6
7 #案例:从终端输入一个5位数的数字,存储到列表中,如果不是数字,提示重新输入,只到输入正确为止
8 #输入正确提示输入正确并打印列表
```

## 列表的操作-获取列表信息(重点)

```
1  获取列表信息
2      1、len(列表名) 获取列表的长度
3      2、max(列表名) 获取列表中最大的值,只能都是相同类型元素的列表
4      3、min(列表名) 获取列表中最小的值,只能都是相同类型元素的列表
5      4、index(元素值) 获取指定元素值所对应的索引
6      5、count(元素值) 查找指定元素在列表中出现的次数
7  mylist=[1,2,3,4,5,8,7,8,9,10,12,13]
8  print(len(mylist))#12
9
10 print(max(mylist))#13
11
12 print(min(mylist))#1
13
14 print(mylist.index(8))#5
15
16 print(mylist.count(8))#2
17 #案例:从终端上输入N位数,组成列表,获取列表长度,最大,最小值,最大值在对应的索引,最小值出现的次数
```

## 列表的操作-排序(重点)

```
1  排序
2      1.reverse() 将列表中的元素倒序输出
3      2.sort() 排序,默认为升序排序,只能都是相同类型元素的列表
4
5  #案例:从终端输入信息,这些信息组成一个列表,然后对列表进行排序
6
7
8  #案例:将属于list1 = ["a1","a2","a3","a4","a5","a6","a7"],但不属于list2 =
    ["a1","a3","a4","a6"]的所有的元素组成一个新的列表list3。
9  list1 = ["a1","a2","a3","a4","a5","a6","a7"]
10 list2 = ["a1","a3","a4","a6"]
11 list3=[]
12 #第一种方法:
13 # for i in list1:
14 #     for j in list2:
15 #         if i==j:
16 #             break
17 #     else:
18 #         list3.append(i)
19 #
20 # print(list3)
21 #第二种方法:
22 for i in list1:
23     if i not in list2:
24         list3.append(i)
25
26 print(list3)
27
28 #案例:创建一个包含字符串和数字的列表,打印出第3到5个元素,倒数第3个元素。
29 mylist=['qianfeng',666,'aaa',999,777,888,20.22,'bbb']
```

## 随机数

```

1  import random#引入模块
2  for i in range(10):
3      print(random.randint(开始, 结束))#使用random模块中的randin函数生成随机数
4
5  随机生成数据包含开始和结束
6
7  import random
8  #生成4位随机数
9  mylist=[]
10 for i in range(4):
11     n=random.randint(0,9)
12     mylist.append(n)
13
14 print(mylist)
15
16 了解:
17 1.产生n--m范围内的一个随机数:    random.randint(n,m)
18
19 2.产生0到1之间的浮点数:    random.random()
20
21 3.产生n---m之间的浮点数:    random.uniform(1.1,5.4)
22
23 4.产生从n---m间隔为k的整数:    random.randrange(n,m,k)
24
25 5.从序列中随机选取一个元素:    random.choice([1, 2, 3, 4, 5, 6, 7, 8, 9, 0])
26
27 6.在一些特殊的情况下可能对序列进行一次打乱操作:    random.shuffle([1,3,5,6,7])
28

```

## 冒泡排序(重点)

```

1  格式:
2  arr = [9, 2, 8, 6, 4]
3  for i in range(len(arr)-1):
4      for j in range(len(arr)-1-i):
5          if arr[j] > arr[j+1]:
6              arr[j], arr[j+1] = arr[j+1], arr[j]
7  print(arr)
8  记住下面四点:
9  1.外循环是长度-1
10 2.内循环是长度-1-i
11 3.是arr[j]和arr[j+1]进行对比
12 4.交互数据也是arr[j]和arr[j+1]进行交换
13 import random
14 mylist=[]

```

```

15 for i in range(10):
16     n=random.randint(1,100)
17     mylist.append((n))
18
19 print(mylist)
20
21 #冒泡排序
22 for i in range(len(mylist)-1):
23     for j in range(len(mylist)-1-i):
24         if mylist[j]>mylist[j+1]:
25             mylist[j],mylist[j+1]=mylist[j+1],mylist[j]
26
27 print(mylist)
28
29
30 案例:使用随机数给一个列表赋值,然后对列表进行冒泡排序
31

```

## 二维列表的创建

```

1 格式:
2     列表名 = [[值1,值2,...],[值1,值2,...],[值1,值2,...]...]
3  #列表里面的元素是列表
4  mylist=[
5      [1,2,3],
6      ['a','b','c']
7  ]
8
9  print(mylist)#[[1, 2, 3], ['a', 'b', 'c']]
10 print(mylist[0])#[1,2,3]
11 print(mylist[0][1])#2
12 print(mylist[1][2])#c
13 for i in mylist:
14     for j in i:
15         print(j)
16
17
18 案例:按照下方二维表创建二维列表,并用for循环输出
19 人名    年龄    年代
20 西施    16      春秋战国
21 貂蝉    18      三国
22 王昭君  20      汉
23 杨玉环  19      唐
24

```

## 元组(重点)

- 1 元组 (tuple) 与列表类似,不同之处在于元组的元素不能修改。元组写在小括号(())里,元素之间用逗号隔开
- 2 元组中的元素类型也可以不相同

```

3  1.定义元组
4  tup1=(1,2,3,'a','b')
5  tup2=1,2,3
6
7  2.只有0个或1个元素时的特殊语法
8  tup3=()
9  tup4=(1,)
10
11 3.访问元组和访问列表类似
12 print(tup1[0])
13 print(tup1[2:4])
14
15 4.元组里的元素不能修改,但元素是列表时,可以修改列表里的内容
16 tu=(1,2,3,[100,200,300])
17 tu[3][0]=1000
18 print(tu)#(1, 2, 3, [1000, 200, 300])
19 tu[3].append(2000)
20 print(tu)#(1, 2, 3, [1000, 200, 300, 2000])
21
22 # 案例: 创建一个元组A, 里面的元素是整数, 创建一个只有一个元素的元组B,
23 # 创建一个元组C, 里面的元素有其中一个是列表
24
25 # 案例: 在上个案例的基础上, 打印三个元组.然后修改C元组中的列表中的值,在打印
26 案例:现有列表:
27 name_list=["tom","kaisa","alisi",["xiaoming","songshu"]]
28 现在有个要求, 将最外层的列表转换成元组存储, 里面的小列表不变;
29 并且向小列表中添加一个元素"python"
30
31
32 5.元组的删除(删除元组,不是删除元组里的元素)
33 del 元组名
34 tup1=(1,2,3)
35 print(tup1)
36 del tup1
37 print(tup1)#报错
38
39
40 6.元组的操作
41 len,max,min,count,index也适用于元组
42 tup=(1,2,3,3,3,7,8,2)
43 print(len(tup))#8
44 print(max(tup))#8
45 print(min(tup))#1
46 print(tup.count(3))#3
47 print(tup.index(8))#6
48
49 列表和元组相互转换
50 tup=(1,2,3)
51 mylist=list(tup)
52 print(mylist)
53
54 mylist2=[3,4,5]
55 tup2=tuple(mylist2)

```

```

56 print(tup2)
57
58 案例:求元组(1,2,8,9,22,0,3)的长度,最大值,最小值,然后把元组转换为列表,打印出来,最后删除元组
59
60 7.变量带*
61 元组同时赋值给多个变量,其中一个变量带星,那么其他不带星的变量获取元组的元素,
62 其余的给这个带星的变量,这个带星的变量就变为列表
63 #注意:1,2,3,4,5这连一起是元组
64 a,*b,c=1,2,3,4,5
65 print(a)#1
66 print(b)#[2,3,4]
67 print(c)#5
68
69
70
71 元组的特点
72 1.不可改变(不支持增删改)
73 2.有序(支持下标查询)
74
75 注意
76 1、与字符串一样,元组的元素不能修改
77 2、元组也可以被索引和切片,方法一样
78 3、注意构造包含0或1个元素的元组的特殊语法规则
79 4、元组也可以使用+操作符进行拼接
80 tup=('a','b','c')
81 tup2=(1,2,3)
82 print(tup+tup2)
83
84 案例:
85 1.创建两个元组,进行连接操作。
86
87 2.创建一个列表和元组,将其连接后打印出来(用到元组转换列表),其中列表用变量代码*方式获取
88

```

## 集合(重点)

```

1 集合:
2     1.集合 ( set ) 是一个无序不重复元素的序列(意思是有重复的会自动删除,每次打印元素的位置不一定)
3     2.基本功能是进行成员关系测试和删除重复元素
4     3.可以使用大括号 { } 或者 set() 函数创建集合,注意:创建一个空集合必须用 set() 而不是 { },
   因为 { } 是用来创建一个空字典
5     4.因为set存储的是无序集合,所以我们没法通过索引来访问。访问 set中的某个元素实际上就是判断一个
   元素是否在set中。
6
7 1.创建集合
8 set1={'a','1',1,'c',66,'d'}
9 print(set1)
10 set2=set()#定义空集合
11
12 2.删除重复的元素
13 set1={'a','1',1,'c',66,'d',1,66}
14 print(set1)#{'d', 1, 66, 'a', 'c', '1'}

```

```

15
16 3.成员关系测试(检测某个成员是否在集合中)
17 print(88 in set1)
18
19 作用:
20 mylist=[1,2,2,1,3,4,2,1,2,3,3,44,4]
21 set2=set(mylist)#先把有重复的列表转换为集合
22 mylist=list(set2)#再把集合转换为列表
23 print(mylist)#[1, 2, 3, 4, 44]
24
25
26 案例:从终端输入5位数,以空格隔开,把这5位数存入列表中,剔除重复的数据,打印出来
27 #提示,列表转集合,集合转列表
28
29 4.集合常用操作
30     1.添加元素
31     2.删除元素
32     3.删除整个集合
33     4.获取集合的长度
34     5.不支持改元素
35
36

```

## 集合操作-添加元素(重点)

```

1 格式:
2 集合.add(元素)
3 set1=set()#空集合
4 set1.add(100)
5 print(set1)#{100}
6 set1.add((1,2,3))
7 print(set1)#{(1,2,3)}
8 # set1.add([10,20])#报错
9 set1.add("hello")
10 print(set1)#{'hello', 100, (1, 2, 3)}
11
12 格式:
13 集合.update(元素)
14 set1=set()
15 set1.update("hello")
16 print(set1)#{'l', 'e', 'o', 'h'}
17 set1.update([10,20,30])
18 print(set1)#{'e', 'o', 10, 'h', 20, 30, 'l'}
19 set1.update((100,200,300))
20 print(set1)#{'l', 100, 200, 'o', 10, 300, 'e', 20, 'h', 30}
21
22 集合可以存储集合
23 set1={10,20,30}
24 set2={1,2,3}
25 # set1.add(set2)#报错
26 set1.update(set2)
27 print(set1)#{1, 2, 3, 20, 10, 30}

```

```
28
29 区别:
30 add不能添加列表,update可以
31 update是将字符串中的拆分成字符进行追加
32 add, 是当做整体追加在集合中
33 #案例:从终端输入一个数字就加入集合中,然后打印,又输入一个数字,然后打印,直到输入-1,退出输入
34 #然后打印最终的集合
35 #案例:从终端输入一个字符串就加入集合中,然后打印,又输入一个字符串,然后打印,直到输入q,退出输入
36 #然后打印最终的集合,字符串要分开存储,如输入hello,存储的列表是{'h','e','l','l','o'}
37
38
```

## 集合操作-删除元素及删除整个集合(重点)

```
1 格式:
2 集合.remove(元素) # 元素如果不存在会引发KeyError
3 集合.discard(元素) # 元素如果不存在不发生任何事
4 集合.pop() # 集合元素是字符串类型时删除随机元素。 集合元素是其他数据类型时,删除左边第一个元素
5 del 集合名 #删除整个集合
6 set1={12,1,3,4}
7 set1.remove(12)
8 print(set1){1, 3, 4}
9 set1.discard(3)
10 print(set1){1, 4}
11 set1.pop()
12 print(set1){4}
13
14 del set1
15 print(set1)#报错
16
17
18 #案例:有集合{1,2,3,'hello','world'},删除全部字符串,然后最左边第一个元素
19
20
```

## 集合操作-获取集合的长度(重点)

```
1 格式:
2 len(集合名)
3 set1={1,2,3,4}
4 print(len(set1))#4
5
6
7 #案例:有集合{1,2,3,4,'hello','a'},保留1个元素,其他的删除,假如元素个数不确定
8 import random
9 set1=set()
10 for i in range(20):
```



```
11     n=random.randint(1,100)
12     set1.add(n)
13
14     print(set1)
15
16     for i in range(len(set1)-1):
17         set1.pop()
18
19     print(set1)
```