

```
1 #200923_列表与元组
2 #列表,类似于JAVA中的数组,但是它比数组更加强大,可以存放任意数据类型
3 list1=[10,20,30,40,50,60] #可以存放数字
4 # list2=['aa','bb','cc'] #也可以存放字符串
5 # list3=[10,'ABC',(10,20,30),{'A':'B'},[36,72]] #也可以存放其他类型
6 # print(list1,list2,list3)
7 #列表属于可变对象,可以添加元素,修改元素,删除元素
8 #添加元素,方式1 append
9 # list1.append(90) #在列表的末尾添加一个值
10 # print(list1)
11 #方式2 insert
12 # list1.insert(2,27.75) #在列表的指定位置添加一个值,第一个参数是下标,第二个参数是值
13 # list1.insert(0,29.975) #如果输入的的第一个参数大于列表的长度,那么就等价于append
14 # print(list1)
15 #方式3 extend 其实是列表的拼接
16 # list1.extend('abc') #实际结果是将abc拆成'a','b','c',然后放到list1的后面
17 # list1.extend([22,33])
18 # list1.extend(123) #数字不要使用extend方法,否则会报错
19 # print(list1)
20
21 #修改列表,直接输入下标就可以
22 # list1_1=[10,20,30,40,[50,60]]
23 # list1_1[-1][-1]=98 #对于子列表,使用两层下标即可
24 # print(list1_1)
25
26 #删除列表中的值
27 #方式1 pop
28 # list1.pop() #不填写参数时,默认删除列表的最后一位
29 # list1.pop(0) #也可以填写下标,删除指定位置的值
30 # print(list1)
31
32 #方式2 remove
33 # list1.remove(10) #remove根据值来删除,如果列表中没有对应的值,则报错.如果有多个值,
    则只删第一个
34 #因为remove是根据值来删除元素,每个元素都要比对一下,所以效率相对较低
35 # print(list1)
36
37 #方式3 del
38 # del list1[5]
39 # print(list1)
40
41 #列表中的切片
42 # print(list1[0:3])
43 #切片不会改变原列表
44 # print(list1)
45 #翻转列表
46 # print(list1[::-1])
47 #排序
48 # list1.sort() #升序
49 # list1.sort(reverse=True) #降序
50 # print(list1)
51
52 #元组,和列表非常相似,都可以使用下标和切片,但是元组属于不可变对象,不能进行增删改的操作
53 # tuple1=(11,22)
54 # print(tuple1[0])
```

```
55 # print(tuple1[0:2])
56 # tuple1[0]=50  #元组不能修改,这么写会报错
57 #如果元组中只有一个值,应该加上一个逗号
58 # tuple2=(80,)
59 # print(type(tuple2))  #type()返回参数的类型
60 #如果元组中有子列表,那么子列表的值是可以修改的
61 tuple3=(11,22,[33,99])
62 tuple3[-1][-1]=66
63 print(tuple3)
64 #思考题,什么是浅拷贝,深拷贝,有什么区别?
```