

PYTHON

编程基础

复制列表元素

概述

html

- Python中，通过赋值运算实际上是将两个变量指向同一个对象，而不是将一个变量的值赋给另一个变量。
- 两个变量指向同一个对象后，我们通过一个变量修改对象中元素的值，那么通过另一个变量访问对象时，访问到的对象中的元素值也是修改后的值。

概述

例：列表对象赋值示例。

```
ls1=[1,2,3] #创建列表对象并赋给变量ls1
ls2=ls1 #通过赋值运算ls2和ls1指向同一个对象
print('ls1和ls2的值分别为：',ls1,ls2)
print('ls1和ls2的内存地址分别为：',id(ls1),id(ls2))
ls1[1]=5 #将ls1中下标为1的元素值修改为5
print('ls1和ls2的值分别为：',ls1,ls2)
```

概述



ls1和ls2的值分别为：[1, 2, 3] [1, 2, 3]

ls1和ls2的内存地址分别为：1977218523720 1977218523720

ls1和ls2的值分别为：[1, 5, 3] [1, 5, 3]

提示：通过赋值运算，ls1和ls2对应了同一个列表。

通过元素截取实现列表元素复制

例：利用元素截取方法，实现修改一个对象中的元素值不会影响另一个对象。



- `ls1=[1,2,3]` #创建列表对象并赋给变量ls1
- `ls2=ls1[:]` #通过ls1[:]将ls1的所有元素截取生成新对象并
#赋给ls2
- `print('ls1和ls2的值分别为：',ls1,ls2)`
- `print('ls1和ls2的内存地址分别为：',id(ls1),id(ls2))`
- `ls1[1]=5` #将ls1中下标为1的元素值修改为5
- `print('ls1和ls2的值分别为：',ls1,ls2)`

通过元素截取实现列表元素复制

例：利用元素截取方法，实现修改一个对象中的元素值不会影响另一个对象。



ls1和ls2的值分别为：[1, 2, 3] [1, 2, 3]

ls1和ls2的内存地址分别为：2021463056968 2021463057032

ls1和ls2的值分别为：[1, 5, 3] [1, 2, 3]

通过元素截取实现列表元素复制

例：利用元素截取方法，实现列表元素复制的问题。



- `ls1=[1,[2,3]]` #创建列表对象并赋给变量ls1
- `ls2=ls1[:]` #通过`ls1[:]`将ls1的所有元素截取生成新对象并赋给ls2
- `print('ls1和ls2的值分别为：',ls1,ls2)`
- `print('ls1和ls2的内存地址分别为：',id(ls1),id(ls2))`



ls1和ls2的值分别为： `[1, [2, 3]] [1, [2, 3]]`

ls1和ls2的内存地址分别为： `1416753996424 1416754486344`

通过元素截取实现列表元素复制

例：利用元素截取方法，实现列表元素复制的问题。



- `print('ls1[1]和ls2[1]的内存地址分别为：',id(ls1[1]),id(ls2[1]))`
- `ls1[1][0]=5` #将ls1下标为1的列表元素（即ls[1]）中下标为0的元
#素值修改为5
- `print('ls1和ls2的值分别为：',ls1,ls2)`



ls1[1]和ls2[1]的内存地址分别为： 1416753996360 1416753996360
ls1和ls2的值分别为： [1, [5, 3]] [1, [5, 3]]

通过deepcopy函数实现列表元素复制

例：使用copy模块中的deepcopy函数，实现列表复制操作。



- `import copy` #导入copy模块
- `ls1=[1,[2,3]]` #创建列表对象并赋给变量ls1
- `ls2=copy.deepcopy(ls1)` #通过调用deepcopy函数复制ls1生成新对象并赋给ls2
- `print('ls1和ls2的值分别为：',ls1,ls2)`
- `print('ls1和ls2的内存地址分别为：',id(ls1),id(ls2))`



ls1和ls2的值分别为： [1, [2, 3]] [1, [2, 3]]

ls1和ls2的内存地址分别为： 1942483404104 1942484025224

通过deepcopy函数实现列表元素复制

例：使用copy模块中的deepcopy函数，实现列表复制操作。



- `print('ls1[1]和ls2[1]的内存地址分别为：',id(ls1[1]),id(ls2[1]))`
- `ls1[1][0]=5` #将ls1下标为1的列表元素（即ls[1]）中下标为0的元素值修改为5
- `print('ls1和ls2的值分别为：',ls1,ls2)`



ls1[1]和ls2[1]的内存地址分别为： 1942483405896 1942484062280
ls1和ls2的值分别为： [1, [5, 3]] [1, [2, 3]]