PYTHON
编程基础

可变类型和不可变类型

可变类型和不可变类型

可变类型

即可以对该类型对象中保存的元素值做修改，如列表、字典都是可变类型。

不可变类型

即该类型对象所保存的元素值不允许修改，只能通过给对象整体赋值来修改对象所保存的数据。但此时实际上就是创建了一个新的不可变类型的对象、而不是修改原对象的值，如数字、字符串、元组都是不可变类型。

可变类型和不可变类型

例：可变类型对象和不可变类型对象示例

```
1    n1,n2=1,1 #定义两个整型变量n1和n2，都赋值为1
2    print('第 2行n1和n2的内存地址分别为：', id(n1), id(n2))
3    n2=3 #将n2重新赋值为3
4    print('第 4行n1和n2的内存地址分别为：', id(n1), id(n2))
5    n1=3 #将n1重新赋值为3
6    print('第 6行n1和n2的内存地址分别为：', id(n1), id(n2))
```

可变类型和不可变类型

例：可变类型对象和不可变类型对象示例

第2行n1和n2的内存地址分别为： 140724681233440 140724681233440

第4行n1和n2的内存地址分别为： 140724681233440 140724681233504

第6行n1和n2的内存地址分别为： 140724681233504 140724681233504

可变类型和不可变类型


```
7    s1,s2='Python','Python' #定义两个字符串变量s1和s2，都赋值为
                             #'Python'
8    print('第 8行s1和s2的内存地址分别为：', id(s1), id(s2))
9    s2='C++' #将s2重新赋值为'C++'
10   print('第10行s1和s2的内存地址分别为：', id(s1), id(s2))
11   s1='C++' #将s1重新赋值为'C++'
12   print('第12行s1和s2的内存地址分别为：', id(s1), id(s2))
```

可变类型和不可变类型

第8行s1和s2的内存地址分别为： 2484401162312 2484401162312
第10行s1和s2的内存地址分别为： 2484401162312 2484402647480
第12行s1和s2的内存地址分别为： 2484402647480 2484402647480

可变类型和不可变类型


```
13      t1,t2=(1,2,3),(1,2,3) #定义两个元组变量t1和t2，都赋值为(1,2,3)
14      print('第14行t1和t2的内存地址分别为：', id(t1), id(t2))
15      t2=(1,2,3) #t2被重新赋值为(1,2,3)
16      print('第16行t1和t2的内存地址分别为：', id(t1), id(t2))
```

```
第14行t1和t2的内存地址分别为： 2484401955176 2484401955392
第16行t1和t2的内存地址分别为： 2484401955176 2484401955464
```

可变类型和不可变类型


```
17    ls1,ls2=[1,2,3],[1,2,3] #定义两个列表变量ls1和ls2，都赋值为
                              #[1,2,3]
18    print('第18行ls1和ls2的内存地址分别为：', id(ls1), id(ls2))
19    ls2[1]=5 #将列表ls2中下标为1的元素值重新赋值为5
20    print('第20行ls1和ls2的内存地址分别为：', id(ls1), id(ls2))
21    ls2=[1,2,3] #ls2被重新赋值为[1,2,3]
22    print('第22行ls1和ls2的内存地址分别为：', id(ls1), id(ls2))
```

可变类型和不可变类型

第18行ls1和ls2的内存地址分别为： 2484400710216 2484400710280

第20行ls1和ls2的内存地址分别为： 2484400710216 2484400710280

第22行ls1和ls2的内存地址分别为： 2484400710216 2484401200264

可变类型和不可变类型

提示：可变类型的对象和不可变类型的对象的区别在于是否可修改对象中的元素值。对于可变类型的对象，如果对可变类型对象中的元素做修改，则不会创建新对象；而如果直接对其赋值，则也会创建一个新的对象。