第二周平时作业

朱士杭 231300027

2024 年 3 月 6 日

1 问题一

1.1 直接赋值

将变量名指向对象，对象及其 id 不变，变的只是变量的指向

比如：x=3,y=3,y=4 一开始 x 和 y 都指向对象 3，后来 y 改变指向指向对象 4

又比如：list1=[1,2,[3,3]] list2=list1 此时 list1 和 list2 的 id 都是一样的，指向的对象是一样的，对 list1 的改变相当于在对 list2 进行改变

1.2 Copy 方法

将对象整体复制一份新的，但是对象内部的元素对象 id 不会发生改变

比如：list1=[1,2,[3,3]] list2=list1.copy() 此时 list1 和 list2 的 id 不一样，但是 for 循环输出之后会发现 list1 和 list2 里面的对象的 id 是一样的，这就意味着改变 list1 里的 1 或 2 没关系，但一旦改变 [3,3] 对象里的元素时，list2 里面的 [3,3] 也会跟着发生改变，这就牵扯到接下来需要说的有关于深拷贝 deepcopy 的问题

1.3 Deepcopy 方法

from copy import deepcopy

通过深拷贝将 list1 和 list2 里面的对象都赋予其新的 id，对其中一个的改变不会影响另一个

2 问题二

2.1 mutable 可变对象

id 是会随着操作发生改变的对象，比如说列表、字典、集合

比如：list1=[1,2,3] 其 id 为 140728001318144 进行改变操作之后 list1[1]=10 其 id 变为 1914524434752

2.2 immutable 不可变对象

id 一旦形成就是确定好的没有办法发生改变的对象

比如说整数、浮点数、元组、字符串

3 问题三

3.1 程序一

会输出：[['_','_','_'],['_','_','X'],['_','_','_']]

原因：通过列表解析，每一次创建的列表都是指向一个新的对象 ['_','_','_'] 对其中一个 ['_','_','_'] 列表的改变不会影响其他列表

3.2 程序二

会输出：[['_','_','X'],['_','_','X'],['_','_','X']]

直接通过 []*3 列表乘 3 的操作来实现二维列表的拼接，本质上是复制了 3 份指向同一个 ['_','_','_'] 对象的变量名，对其中一个的改变相当于在对另外两个列表进行相同的改变

4 问题四

4.1 Packing

打包操作，使用可迭代解包运算符在单个变量中收集多个值，个人理解就是在迭代一个对象的时候从里面将多个值囊括进一个新的对象

比如：a,*b,c=[1,2,3,4]　其中 b 就是 [2,3] 这里就是一个 packing

4.2 Unpacking

解包操作，在单个赋值语句中将可迭代的值分配给变量的元组（或列表）

比如一个基本的赋值语句就是 a,b=1,2, 将元组 (1,2) 进行解包之后分别分配给对象 (a,b) 中的元素

4.3 综合 Packing 与 Unpacking

```python
def func_tuple(*args): # packing
    print type(args)
    for i in args:
        print i

def func_dict(**dict): # packing
    print type(dict)
    print dict

if __name__ == '__main__':
    t = (1,2,3,'hello')
    d = {'a':1,'b':2,'c':3}
    func_tuple(*t) # unpacking
    func_dict(**d) # unpacking
```

图 1: Packing 和 Unpacking

4.4 *args 及其作用

*args 是一个特殊的语法，用于在函数定义中收集任意数量的位置参数。这里的 args 是参数名，可以是任何合法的变量名，通常约定使用 args，但也可以是其他名称。星号 * 表示将接收的所有位置参数打包成一个元组 tuple，这样在函数中可以遍历它来处理所有的参数

4.5 参考资料

ps. 第四问一开始不知道答案通过查找相关资料才得出结果，参考链接如下：

https://blog.csdn.net/linux4fun/article/details/16803937

https://blog.csdn.net/qq_27825451/article/details/81666601

https://zhuanlan.zhihu.com/p/639405308

5 问题五——自行创建字典

图 2: 创建简单字典

代码如下：

$dictionary = \{"num" : 1, "name" : "Edison", "age" : 19, "hobby" : "rock"\}$

print(dictionary)

dict1=eval(input(“请输入一个字典：”))

print(f“输入的字典为：$\{dict1\}''$)