1.变量赋值

a = 3

a 是变量名， 3 是一个对象，在内存中的一块地址

a = 3的过程就是把3的首地址传递给了a ----- 引用

b = a 把a的首地址传递给了b，a和b共用了一个首地址

我们可以称呼 b 拷贝了a的首地址

```
a = [1,2]
b = a
a[0] = 3  # [3,2]  a的首地址没变
print(b)  # [3,2]
```

可变对象和不可变对象的区别

```
1. 可变对象的首地址不变的情况下，对象的内容可以发生改变
2. 不可变对象内容发生改变，整体的首地址发生改变
```

不可变对象没有深浅拷贝一说，或者说不可变对象所有的拷贝都是深拷贝。

浅拷贝

对对象拷贝了一份，组成一个新的对象，新的对象的里元素依然引用了原对象元素的首地址。

```
# l1 = [1,2,3,4]
# l2 = l1[:]
# l1[0] = 5
# print(l1)
# print(l2)

l1 = [[1,2],[3,4]]
l2 = l1[:]
l1[0][0] = 5
print(l1)
print(l2)
```

浅拷贝的实现方式

1. 使用copy模块的copy方法　（通用方法）
2. 使用对象本身的copy方法
3. 使用切片方式　list1 = list2[:]
4. 使用复制语句

```
#
# s = "a = [1,2]
# # b = a
# # a[0] = 3  # [3,2]
# # print(b)abcde"
# s1 = s.upper()
```

```
# print(s1)

# a = (1,2)
# b = a
# a1 = list(a)
# a1[1] = 3
# a = tuple(a1)
# print(b)

# l1 = [1,2,3,4]
# l2 = l1[:]
# l1[0] = 5
# print(l1)
# print(l2)
# a = [1,2]
# b = a
# a[0] = 3  # [3,2]
# print(b)
#
# l1 = [[1,2],[3,4]]
# l2 = l1[:]
# l1[0][0] = 5
# print(l1)
# print(l2)


#  浅拷贝的实现方式
# 1.copy方法
```

```
# import copy
# l1 = [[1,2],[3,4]]
# l2 = copy.copy(l1)
# l1[0][0] = "hello"
# print(l1)
# print(l2)

# 2.copy方法
# l1 = [[1,2],[3,4]]
# l2 = l1.copy()
# l1[1][1] = "world"
# print(l1)
# print(l2)

#  3. 切片

# 复制方法
# l1 = [1,2,3,[4,5]]
# l2 = l1 * 1
# l1[-1][0] = "6"
# print(l1)
# print(l2)
```

深拷贝

对对象拷贝了一份，组成一个新的对象，新的对象的里元素也拷贝了原对象的元素，组成一个新的元素。

深拷贝的实现方式

使用copy模块的deepcopy方法

```
# 深拷贝
import copy
l1 = [1,2,3,[4,5,[6,7]]]
l2 = copy.deepcopy(l1)    # 深拷贝
l1[-1][-1][0] = "列表"
print(l1)
print(l2)
```

总结

1. 深浅拷贝都是对原对象的拷贝，与原对象的首地址不同
2. 浅拷贝拷贝的是原对象，对里面的小对象是进行了引用
3. 深拷贝拷贝的是原对象及里面小对象。
4. 浅拷贝： 节约资源和内存空间
5. 深拷贝： 数据更安全（数据容灾特性）
   电磁波，辐射，电子对抗，地震，火山喷发，战争,服务器宕机...