**测试题：**

1. **注意，这道题跟上节课的那道题有点儿不同，回答完请上机实验或参考答案。**

>>> old = [1, 2, 3, 4, 5]

>>> new = old

>>> old = [6]

>>> print(new)

如果不上机操作，你觉得会打印什么内容？

答：上机前：应该会打印[6].

上机后：它任然打印[1,2,3,4,5,6]

1. **请问如何将下边这个列表的’小甲鱼’修改为’小鱿鱼’？**

list1 = [1, [1, 2, ['小甲鱼']], 3, 5, 8, 13, 18]

要对一个列表进行顺序排序，请问使用什么方法？

答：

>>> list1 = [1, [1, 2, ['小甲鱼']], 3, 5, 8, 13, 18]

>>> del list1[1][2]

>>> list1[1].append('小鱿鱼')

先用del找到list1里面第1个元素，然后在这个列表里面再找到第2个元素‘小甲鱼’并删除，最后在list1的第1个元素中用append函数添加‘小鱿鱼’。

1. **要对一个列表进行逆序排序，请问使用什么方法？**

答：正序排序是使用sort()函数，如果要逆序排序：列表.sort(reverst=True)

1. **列表还有两个内置方法没给大家介绍，不过聪明的你应该可以自己摸索使用的门道吧：copy() 和 clear()**

答：copy()函数应该和‘列表[:]’方法差不多，都是可以直接复制一个列表。

clear函数则可以把整个列表的元素都清除。

1. **你有听说过列表推导式或列表解析吗？**

没听过？！没关系，我们现场来学习一下吧，看表达式：

>>> [ i\*i for i in range(10) ]

你觉得会打印什么内容？

>>> [i\*i for i in range(10)]

[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

居然分别打印了0到9各个数的平方，然后还放在列表里边了有木有？！

列表推导式（List comprehensions）也叫列表解析，灵感取自函数式编程语言 Haskell。Ta 是一个非常有用和灵活的工具，可以用来动态的创建列表，语法如：

[有关A的表达式 for A in B]

例如

>>> list1 = [x\*\*2 for x in range(10)]

>>> list1

[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

相当于

list1 = []

for x in range(10):

list1.append(x\*\*2)

问题：请先在 IDLE 中获得下边列表的结果，并按照上方例子把列表推导式还原出来。

>>> list1 = [(x, y) for x in range(10) for y in range(10) if x%2==0 if y%2!=0]

活学活用：请使用列表推导式补充被小甲鱼不小心涂掉的部分

在这里插入图片描述

请写下这一节课你学习到的内容：格式不限，回忆并复述是加强记忆的好方式！

总结：

比较操作符：

列表之间的比较是从第0个元素开始比较的，只要第0个比较出结果了，就不用理后边的元素了。

连接操作符：

变量x=列表1+列表2，x的结果为这两个列表元素的拼接，但是变量x会随着列表1和列表2的变化而变化，要想不变化要用列表分片的方法或者用内置函数copy()复制列表1、2的值。注意：+号不能用于列表元素的添加，列表元素添加请复习前几节课的内容。

重复操作符：

列表\*3这可以将列表内的元素复制3份，列表\*=3这样就可以把元素复制3份的同时并重新赋值。

成员关系操作符：in 和 not in

假设list1=[1,2,3,4,[5,6],7]

这样如果 1 in list1 就会返回True，如果1 not in list2 就会返回Flash。

要是要访问元素里面的元素，则要用 5 in list1[4] 才能返回True，list1后面要加索引值，要是还要访问里面的元素，则需要在list1[4]后面在接上它的索引值，如list1[4][1].

dir(list)可以查看列表的小伙伴们。

Count()函数可以查看元素重复的次数，括号里面填需要查找的元素名，如list1.count(123)。

Index()查看元素的索引值，括号内填需要查找的元素，后面可以添加需要查找的范围，如list1.index(123,2,9),则表示从列表的第2个元素开始查找到第9个元素结束，返回123所在位置的索引值。

reverse()这是让列表反转的函数，让后面的元素到最前面，然后依次翻转,这里括号内不用加参数。

Sort()是对列表进行排列，默认是从小到大排列，在括号内加入(reverse=True)就可以从大到小排序。