列表的主要属性：

1. 任意对象的有序集合。
2. 支持随机访问（下标方式）。
3. 可变对象（支持原处修改），支持索引、切片以及合并。
4. 支持任意嵌套，长度可变。
5. 列表包含0个或多个其他对象的引用。（类似C语言中的指针数组）

Python中的列表和C++中的vector类似，都是动态数组。最初会给列表分配空间，[1, 2, 3]这样一个列表申请的空间可能有3个，也可能有8个，如果一直进行append，空间会填满，这时系统就会给列表重新分配一个两倍的空间，将原来空间复制到新的空间，

例：程序list\_test/list\_test4.py

l1 = [1, 2, 3]  
print(l1 + ['a', 'b']) # [1, 2, 3, 'a', 'b']  
print(l1) # [1, 2, 3]  
l1 += ['a', 'b'] # l1 = l1 + ['a', 'b']  
print(l1)  
print()  
  
# 列表中可以包含不同类型的元素  
# 但部分列表方法就无法使用了  
# print(l1.sort()) # Error  
  
# 列表嵌套  
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]  
print(matrix[0]) # [1, 2, 3]  
  
# 列表解析  
print([value for value in matrix])  
print()  
# print(value) # Error，value未定义  
col2 = [row[1] for row in matrix]  
print(col2) # [2, 5, 8]  
print([row[1] for row in matrix if row[1] % 2 == 0]) # [2, 8]  
print([matrix[i][i] for i in [0, 1, 2]]) # [1, 5, 9]

list中的append和+

a = [1, 2, 3]  
b = [4, 5, 6]  
a.append(b)  
print(a) # [1, 2, 3, [4, 5, 6]]  
  
a1 = [1, 2, 3]  
b1 = [4, 5, 6]  
a1 = a1 + b1  
print(a1) # [1, 2, 3, 4, 5, 6]

列表解析：生成二维列表

例：udacity\_test/udacity\_test9.py

marks = [[0 for j in range(columns)] for i in range(columns)]  
print(marks)

输出为：

[[0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0]]

双重for循环解析：

d1 = {'1': 0, '2': 1, '3': 2, '4': 3, '5': 4}

l1 = [1, 2]

#找出与l1中与字典相等value对应的key

s = [key for key, value in d1.items()for i in l1 if i == value]

print(s)

输出为：

['2', '3']