一,list是列表的意思。

二,底层是链表。列表是序列对象，使用一对方括号[ ]。可包含任意的Python数据信息，如字符串、数字、列表、元组等。列表的数据是可变的，我们可通过对象方法对列表中的数据进行增加、修改、删除等操作。

三,可以通过list(seq)函数把一个序列类型转换成一个列表。

列表常用方法：

1,append(y) 在列表尾部追加单个对象y。使用多个参数会引起异常。

>>> a=[1,23,45,6]

>>> a

[1, 23, 45, 6]

>>> a.append(7)

>>> a

[1, 23, 45, 6, 7]

2,count(y) 返回对象y在列表中出现的次数。

>>>b=['pussy','pussy']

>>> a.extend(b)

>>> a

[7, 6, 23, 'vagina', 'pussy', 'breset', 'lips', 'kiss', 'caress', 'touch', 'fuck', 'cum', 'pussy', 'pussy']

>>> a.count('pussy')

3

>>>

3,extend(L) 将列表L中的表项添加到列表中。返回None。

如：

>>> a=[7, 6, 23, 'vagina', 'pussy', 'breset', 'lips']

>>> b=['kiss','caress','touch','fuck','cum']

>>> a.extend(b)

>>> a

[7, 6, 23, 'vagina', 'pussy', 'breset', 'lips', 'kiss', 'caress', 'touch', 'fuck', 'cum']

>>>

4,index(y) 返回列表中匹配对象y的第一个列表项的索引。无匹配元素时产生异常。

5,insert(i,y) 在索引为i的元素前插入对象y。如list.insert(0,y)在第一项前插入对象。返回None。

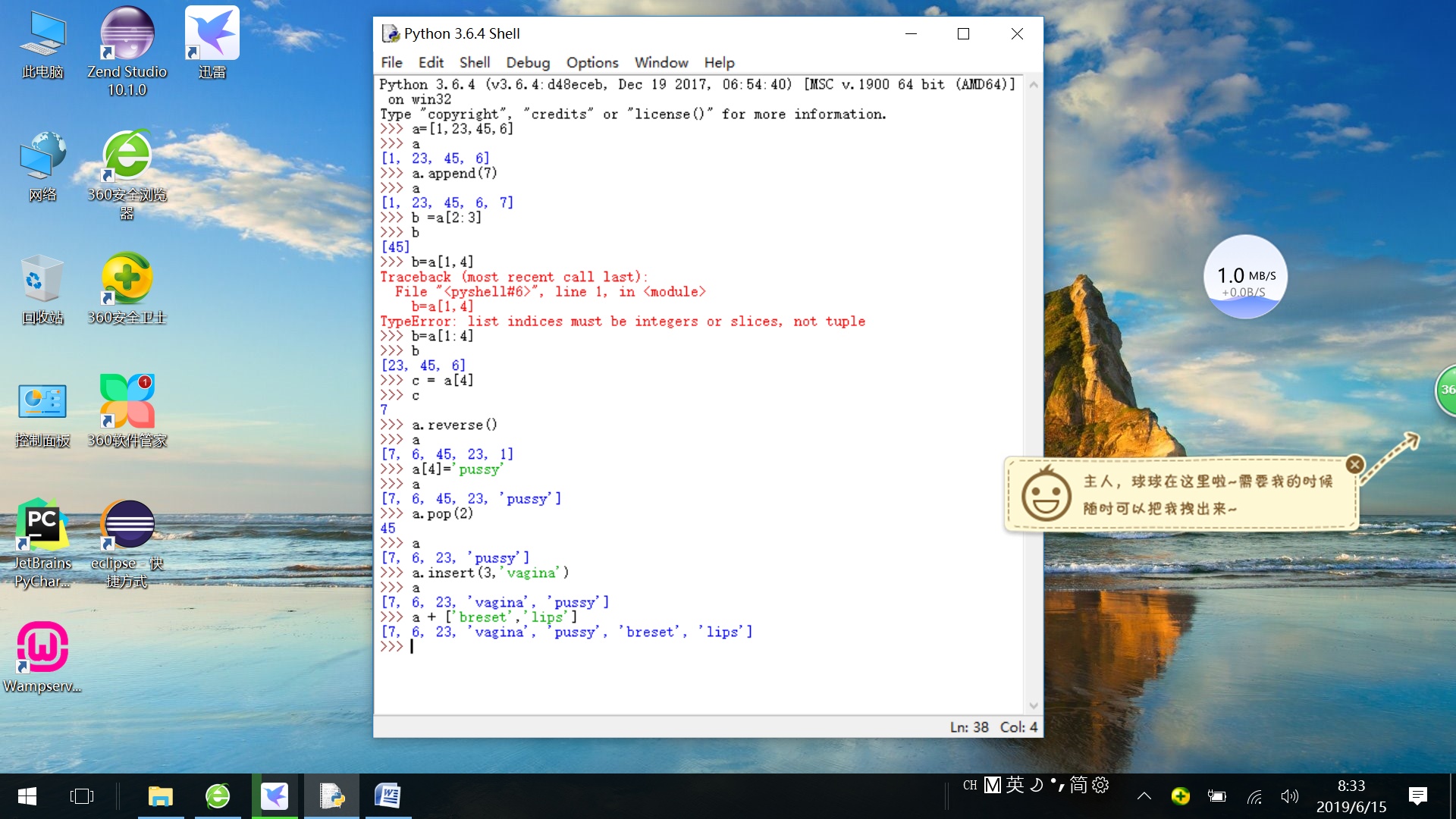
6,pop(y) 删除列表中索引为y的表项，并返回该表项的值。若未指定索引，pop返回列表最后一项。

7,remove(y) 删除列表中匹配对象y的第一个元素。无匹配元素时产生异常，返回None。

8,reverse() 颠倒列表元素的顺序。 默认值为False,要想颠倒列表元素顺序，则设为True

9,sort() 对列表排序，返回none。bisect模块可用于排序列表项的添加和删除。

* 使用示例：



b=a[1:4]：这个叫做切片方法，格式—》list[开始索引:结束索引]特点包含头元素，但是不包含尾元素

切片方法的格式二，list[开始索引:结束索引:间隔] 特点包含头元素，跳过【间隔-1】个元素，但是不包含尾元素

如：

>>>a= [7, 6, 23, 'vagina', 'pussy', 'breset', 'lips']

>>> b=a[0:6:2]

>>> b

[7, 23, 'pussy'] 结果

>>>

Index方法：格式—》a.index(“需要查找的值”，开始位置，结束位置) 在list集合中查找值

a.index(6,0,5)

# Num02–>tuple的详解：

定义：

一,tuple是元组的意思.

二,元组将”多样的对象”集合到一起,不能修改,通过索引可以进行查找, 使用括号一对圆括号()。

三,因为tuple不可变，所以代码更安全。如果可能，能用tuple代替list就尽量用tuple。

一些注意事项：

1. tuple=(1,),只有1个元素的tuple定义时必须加一个逗号，来消除歧义。因为()不仅用来表示tuple数据结构，还可以用来数学中的公式。这就产生了歧义，因此，Python规定，在t(1)这种情况下，

按小括号进行计算，计算结果自然是1。

2，tuple=1，2，3，4，这也可以是一个元组，在不使用圆括号而不会导致混淆时，Python允 许不使用圆括号的元组。建议少这样写。

3，和列表一样，可对元组进行索引、分片、连接和重复。也可用len()求元组长度。

如：

>>> a=[7, 6, 23, 'vagina', 'pussy', 'breset', 'lips', 'kiss', 'caress', 'touch', 'fuck', 'cum', 'pussy', 'pussy']

>>> b=(1,2,a)

>>> b[2][1]=8

>>> b

(1, 2, [7, 8, 23, 'vagina', 'pussy', 'breset', 'lips', 'kiss', 'caress', 'touch', 'fuck', 'cum', 'pussy', 'pussy'])

>>> len(b) 求元组长度

3 （也就是将这个list看作是一个元素，但是可以用二维数组的形式来引用list中的元素）

>>>

>>> c=b[2] 索引

>>> c

[7, 8, 23, 'vagina', 'pussy', 'breset', 'lips', 'kiss', 'caress', 'touch', 'fuck', 'cum', 'pussy', 'pussy']

>>>

>>> d=b[0:2] 分片

>>> d

(1, 2)

>>>

>>> b+=(True,False) 连接

>>> b

(1, 2, [7, 8, 23, 'vagina', 'pussy', 'breset', 'lips', 'kiss', 'caress', 'touch', 'fuck', 'cum', 'pussy', 'pussy'], True, False)

>>>

>>> d\*=3 重复

>>> d

(1, 2, 1, 2, 1, 2)

>>>

4，元组的索引用tuple[y]的形式，而不是tuple(y)。切记切记。

5，和列表类似，使用tuple(arg)可把其它序列类型转换成元组。

6,最后看一个”可变”的tuple：即在包含list的tuple中，list中的元素可以改变

m = (‘aa’, ‘bb’, [‘AA’, ‘BB’])

m[2][0] = ‘XX’

m[2][1] = ‘YY’

print(m)

(‘aa’, ‘bb’, [‘XX’, ‘YY’])

表面上看，tuple的元素确实变了，但其实变的不是tuple的元素，而是list的元素。tuple一开始指向的list并没有改成别的list，所以，tuple所谓的“不变”是说，tuple的每个元素，指向永远不变。

Num03–>dict的详解：

定义：

1,dict是字典的意思。

2,字典是一组键(key)和值(value)的组合,通过键(key)进行查找,没有顺序, 使用大括号”{}”。

3,Dictionary存储了从一种对象（key）到另一种对象（value）的映射关系，key必须是不可变的对象（字符串、数字或者tuple），而value对象则是可变的任意数据类型。

4,dict中的key的顺序不是固定的，即没有确定的元素顺序，编程的时候也不应该依赖于dict中的元素顺序。

一些注意事项：

1. 向字典中追加元素

**增加Python字典元素：两种方法**

1. **>>>** dict1['a']=1 #第一种
2. **>>>** dict1
3. {'a': 1}
4. #第二种：setdefault方法
5. **>>>** dict1.setdefault('b',2)
6. 2
7. **>>>** dict1
8. {'a': 1, 'b': 2}

1，dict.keys(),dict.values(),dict.items()

如：

>>> dic={'name':'kenny','age':18,'height':168}

>>> dic

{'name': 'kenny', 'age': 18, 'height': 168}

>>> dic.keys()

dict\_keys(['name', 'age', 'height'])

>>> dic.values()

dict\_values(['kenny', 18, 168])

>>> dic.items()

dict\_items([('name', 'kenny'), ('age', 18), ('height', 168)])

>>>

2，hash(obj)返回obj的哈希值，如果返回表示可以作为dict的key

>>> hash('name')

6998080228242846472 //正确，因为str类可以作为字典的key

>>> hash([1,2,3,4,5])//出错因为list类型不能作为字典的key

Traceback (most recent call last):

File "<pyshell#19>", line 1, in <module>

hash([1,2,3,4,5])

TypeError: unhashable type: 'list'

3，del 或 dict.pop可以删除一个item,clear清除所有的内容

4，sorted(dict)可以吧dict排序

5，dict.get()可以查找没存在的key。也可以使用in判断，key是否存在，但不建议。

6，dict.setdefault() 检查字典中是否含有某键。 如果字典中这个键存在，你可以取到它的值。 如果所找的键在字典中不存在，你可以给这个键赋默认值并返回此值。

7，{}.fromkeys()创建一个dict，例如: {}.fromkeys((‘love’, ‘honor’), True) =>{‘love’: True, ‘honor’: True}

8，不允许一个键对应多个值

9，键值必须是哈希的，用hash()测试

10，一个对象，如果实现hash()方法可以作为键值使用

Num04–>set的详解：

定义：

1，set是集合的意思。

2，无序,元素只出现一次, 自动去重,使用”set([ ])”。在创建set的时候，需要提供一个list作为输入。

3,因此set中的元素和dict中的key一样，只能是string，number和tuple等不可变的元素，而不能是list。

一些注意事项：

1，set和list可以自由转换，在删除list中多个/海量重复元素时，可以先转换成set，然后再转回list并排序(set没有排序)。此种方法不仅方便且效率较高.

2，set.add(),set.update.set.remove，添加更新删除，-= 可以做set减法

3,set.discard 和 set.remove不同在于如果删除的元素不在集合内，discard不报错，remove 报错

< <= 表示 子集，> >=表示超集

| 表示联合, & 表示交集 ,- 表示差集, ^ 差分集

list和dict的比较如下：

dict有以下几个特点：

查找和插入的速度极快，不会随着key的增加而增加；

需要占用大量的内存，内存浪费多。

而list相反：

查找和插入的时间随着元素的增加而增加；

占用空间小，浪费内存很少。

所以，dict是用空间来换取时间的一种方法。

dict可以用在需要高速查找的很多地方，在Python代码中几乎无处不在，正确使用dict非常重要，需要牢记的第一条就是dict的key必须是不可变对象。这是因为dict根据key来计算value的存储位置，如果每次计算相同的key得出的结果不同，那dict内部就完全混乱了。这个通过key计算位置的算法称为哈希算法（Hash）。要保证hash的正确性，作为key的对象就不能变。在Python中，字符串、整数等都是不可变的，因此，可以放心地作为key。而list是可变的，就不能作为key。

可以使用dir(dict)和help(dict.xxx)来显示更详细的帮助信息。