1.python的逻辑运算符只能用 and or not ！=

2.string类内嵌了很多方法，比如：s.strip()，s.capitalize()，s.upper()等等

3.x=lis.pop()，列表的pop会返回元素

4. for idx, animal in enumerate(animals):

5. d.get('monkey', 'N/A')，如果不在，返回‘N/A’。另一种方法：‘monkey’in d

6. for I in dict： 默认返回的是key

7.字典解析：{x: x \*\* 2 for x in nums if x % 2 == 0} 同列表解析

8.python中的set和dict由于其hash特性都是无序的，而Collections则提供了有序的字典。

9.set，st={1，2}；1 in st检查是否存在，st.add(3)，st.remove(1)，set也可以用enumerate。

10.集合解析：nums = {int(sqrt(x)) for x in range(30)}

11.元组的元素不可变，与list另外一个区别是，元组可以为tuple和set的key，而list则不可以。

12.字符串前面的u代表Unicode编码，即可以出现中文，而r表示忽略转义字符。

13.sorted（ite，cmp，key，reverse）函数可以用于任何可迭代对象，它返回一个新的列表，key可以为lamba函数，返回每个元素的key，元素根据这个key是用来比较大小的。

Key=operator.itemgetter(0，1)，先由0号元素排序，相等再按1号元素排序。

cmp是比较函数，接收两个参数，注意与C++不同，return a-b，如果是负数，说明小于，为0说明相等，正数说明是大于。

14.import pprint与print的区别是，pprint输出更美观，更友好。

15.numpy的初始化数组，需要使用元组，eg：np.zeros((5,5))

16.TruncatedSVD相对于SVD不需要中心化，因此他更有利于稀疏矩阵的计算效率。

17. M = np.stack(M，axis=0)#增加维度

18.python的赋值，本质上是引用，并不会分配内存。

19. a,b=b,a元组赋值,会在右面构造一个暂时的元组变量。

20.pop（0），后面的参数是索引。remove，后面的参数是值。

21. a,b=c,d 本质上是去省略了括号的元组

22. a,\*b=[1,2,3,4] a=1,b=[2,3,4]

23. for i in iter: 中的i本质上是把可迭代对象中的每个值，依次赋值给i。并且i在之后，还可以继续被使用，这与C++的局部变量不同。

24.python中的所有变量，都是引用（本质是指针（地址）），（与 void \*指针类似，因为可以指向各种对象）。变量无类型，对象有类型。对象有cnt，表示有多少个变量引用了它，cnt为0时，则被垃圾回收。

25.==检查 值 是否相等，is检查其指针是否相等（是否引用了同一个对象）。

26. eval（str） 把str内的字符串，当成代码执行。

27.切片会创建一个新的对象，因此会浪费内存空间。

28. python中的迭代器是指，具有\_\_next\_\_()方法的的对象（可迭代对象），当\_\_next\_\_()方法调用完最后时，会产生StopIteratioin异常，for ，in ，列表推导，map等迭代工具（也叫迭代上下文），会自动地调用next，会根据异常来决定终止。

29.遍历文件时，由于文件对象就是迭代器，next总会产生下一行，最好的遍历文件的方式就是，for line in open(‘tmp.txt’)

30. 迭代协议：

可迭代对象：使用iter（可迭代对象），会调用可迭代对象.\_\_iter\_\_()方法，这个方法会返回一个“迭代器”，迭代器会有\_\_next\_\_()方法。

迭代协议：for等迭代工具，会首先调用可迭代对象的\_\_iter\_\_()方法，返回一个迭代器，接着调用迭代器的\_\_next\_\_()方法，直到StopIteration结束。这就是迭代协议。

文件既是可迭代对象，又是迭代器，即同时有\_\_next\_\_和\_\_iter\_\_方法，由于返回的迭代器是它本身，因此，只能进行一次迭代（即迭代到结尾后，空了，不能继续迭代了，除非创建一个新的迭代对象。）。而列表，只是可迭代对象，即只有\_\_iter\_\_方法，可以返回多个迭代器，进而进行多次迭代。

31. 列表推导更快，甚至比for还要快。

32. \*可迭代对象，可以用来解包，比如 def f(a,b,c) l=[1,2,3] f(\*l),通过这种方式来调用这个函数。

33.zip(\*zip(a,b)) 可以用来表示原来的a，b。zip可以看成是转置，那么zip两次，就会还原，只不过，zip(a,b)是一个可迭代对象，需要用\*进行解包。

34.对于全局变量而言，在函数中，可以像C++那样读取全局变量的值，但是，如果想要修改全局变量，需要提前在函数内部进行global声明。