1 请教一下列表与元组的区别是什么.分别在什么情况下使用

列表中的项目应该包括在方括号中，你可以添加、删除或是搜索列表中的项目。由于你可以增加或删除项目，所以列表是可变的数据类型，即这种类型是可以被改变的。

元组和列表十分类似，但是元组是不可变的.也就是说你不能修改元组。元组通过圆括号中用逗号分割的项目定义。元组通常用在使语句或用户定义的函数能够安全地采用一组值的时候，  
即被使用的元组的值不会改变。

2 字典  
 键值对的集合(map)字典是以大括号“{}”包围的数据集合。  
 与列表区别：字典是无序的，在字典中通过键来访问成员。字典是可变的，可以包含任何其他类型。

3 Python是如何进行内存管理的？  
 Python提供了对内存的垃圾收集机制，但是它将不用的内存放到内存池而不是返回给操作系统。  
Python中所有小于256个字节的对象都使用pymalloc实现的分配器，而大的对象则使用系统的 malloc。另外Python对象，如整数，浮点数和List，都有其独立的私有内存池，对象间不共享他们的内存池。也就是说如果你分配又释放了大量的整数，用于缓存这些整数的内存就不能再分配给浮点数。  
 在Python中，许多时候申请的内存都是小块的内存，这些小块内存在申请后，很快又会被释放，由于这些内存的申请并不是为了创建对象，所以并没有对象一级的内存池机制。这就意味着Python在运行期间会大量地执行malloc和free的操作，频繁地在用户态和核心态之间进行切换，这将严重影响Python的执行效率。为了加速Python的执行效率，Python引入了一个内存池机制，用于管理对小块内存的申请和释放。这也就是之前提到的Pymalloc机制。

4 如何在一个function里面设置一个全局的变量？  
 解决方法是在function的开始插入一个global声明

5 Python中pass语句的作用是什么？  
 pass语句什么也不做，一般作为占位符或者创建占位程序，pass语句不会执行任何操作，比如：

while False:  
pass  
pass通常用来创建一个最简单的类：  
class MyEmptyClass:  
pass  
pass在软件设计阶段也经常用来作为TODO，提醒实现相应的实现，比如：  
def initlog(\*args):  
pass #please implement this

6 Python里面如何拷贝一个对象？

因为在[Python](http://www.2cto.com/kf/web/Python/" \t "_blank)中,无论你把对象做为参数传递,做为函数返回值,都是引用传递的.  
 Python中的对象之间赋值时是按引用传递的，如果需要拷贝对象，需要使用标准库中的copy模块。  
1. copy.copy 浅拷贝 只拷贝父对象，不会拷贝对象的内部的子对象。  
2. copy.deepcopy 深拷贝 拷贝对象及其子对象  
一个很好的例子：  
import copy  
a = [1, 2, 3, 4, ['a', 'b']] #原始对象  
b = a #赋值，传对象的引用  
c = copy.copy(a) #对象拷贝，浅拷贝  
d = copy.deepcopy(a) #对象拷贝，深拷贝  
a.append(5) #修改对象a  
a[4].append('c') #修改对象a中的['a', 'b']数组对象

print 'a = ', a  
print 'b = ', b  
print 'c = ', c  
print 'd = ', d  
输出结果：  
a = [1, 2, 3, 4, ['a', 'b', 'c'], 5]  
b = [1, 2, 3, 4, ['a', 'b', 'c'], 5]  
c = [1, 2, 3, 4, ['a', 'b', 'c']]  
d = [1, 2, 3, 4, ['a', 'b']]