python中list和array的底层实现以及区别

（1）python中list的底层实现

list属于连续线性表，有一块连续地内存存储list中元素的地址；

list最大长度：2^(29)，即32位；或2^(60)，即64位；

list对象中包含指向list对象的指针数组和申请内存槽的个数。初始化一个空list时，会在内存中分配好内存槽空间数。一般槽的大小要大于list的大小(len)。当list为空时，分配的内存槽大小为0；当进行append操作添加一个元素时，分配的内存槽大小为4，然后直到list长度等于内存槽大小时，继续添加一个元素，此时又会重新分配一个大小为8的内存槽，依次下去（list的增长模式为：0、4、8、16、25、35、…）。当使用pop操作时，如果取出list中元素后的大小小于分配的内存槽大小的一半，那么将会缩减list的大小(缩减到当前list大小的2倍，例如：分配的内存槽大小为8，当list中剩下3个元素时，内存槽大小缩减到6)。

（2）list和array的区别

array中数据类型时一样的，而list中可以有各种类型数据；

array一般计算起来比list快，因为list涉及到类型的转换问题；

python中各种参数

定义和调用时不同类型参数的顺序：位置参数、默认参数、可变参数、关键字参数

（1）位置参数：根据函数定义的参数位置传递参数；

（2）可变参数：用一个\*表示，根据传进参数位置合并为一个元组；

（3）关键字参数：用\*\*表示，通过“键-值”形式加以指定，不需要考虑参数顺序；

（4）默认参数：函数定义时，提供了默认参数值；