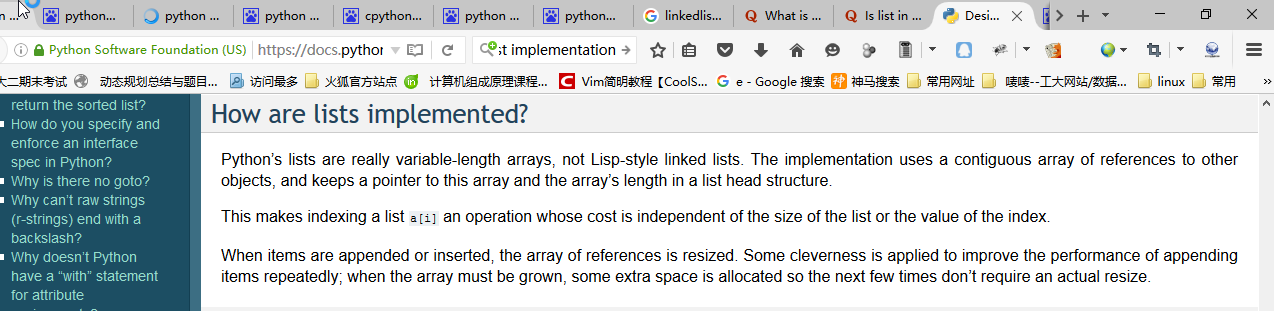
# 问题0：python list的内部实现是什么样的？

是数组而不是链表。即arraylist并非linkedlist

list只能叫做表，表有多种实现方式。

**python list is an array of C pointers**



# 问题1 ：python有没有链表(linkedList)？

For some needs, a [deque](https://docs.python.org/library/collections.html#collections.deque) may also be useful. You can add and remove items on both ends of a deque at O(1) cost.

from collections import deque

d = deque([1,2,3,4])

print d

for x in d:

print x

print d.pop(), d

因此，可以使用deque来实现链表

或者，使用传统的方式，定义Node 类，然后操作链表。因为数组才是涉及批量内存管理的数据结构，必须通过原生语言来实现。

class LinkedListNode():

val=None

nextNode=None

def \_\_init\_\_(self,val,next):

self.val=val

self.nextNode=next

def insert(self,insertion):

self.nextNode,insertion.nextNode=insertion,self.nextNode

…

或者使用lambda语法

cons = lambda el, lst: (el, lst)

mklist = lambda \*args: reduce(lambda lst, el: cons(el, lst), reversed(args), None)

car = lambda lst: lst[0] if lst else lst

cdr = lambda lst: lst[1] if lst else lst

nth = lambda n, lst: nth(n-1, cdr(lst)) if n > 0 else car(lst)

length = lambda lst, count=0: length(cdr(lst), count+1) if lst else count

begin = lambda \*args: args[-1]

display = lambda lst: begin(w("%s " % car(lst)), display(cdr(lst))) if lst else w("nil\n")

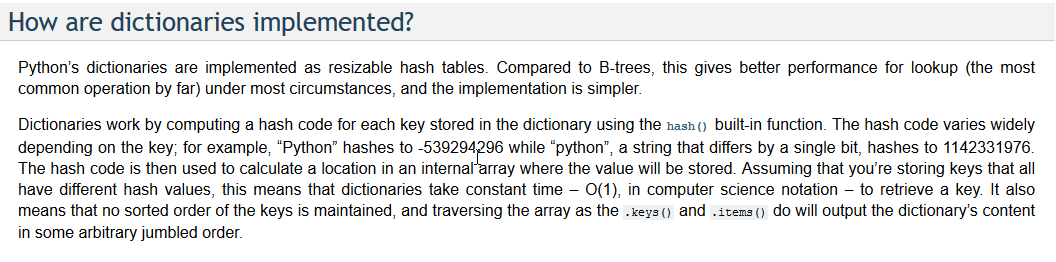
where w = sys.stdout.writ

**Linked lists have no practical value in Python. I've never used a linked list in Python for any problem except educational.**

**在python中，除了教育意义，我没有使用过链表**

# **问题：dict如何实现**

使用的是hash表



因此，dict的key必须是immutable的

Hash lists by their address (object ID). This doesn’t work because if you construct a new list with the same value it won’t be found; e.g.:

mydict = {[1, 2]: '12'}

print mydict[[1, 2]]

比较key的时候使用的是==而不是is

is 当且仅当 id1==id2 及地址相等

# del语法 和 .\_\_del()\_\_

del 只是删除一个元素的引用计数，但是.\_\_del\_\_()却是在真正地销毁一个元素.

不过，并非每个类都定义了\_\_del\_\_()函数，如list就没有定义

# **其他实现细节**

**CPython实现细节：** 在CPython中，id(x) 返回x 的内存地址。

**在a = 1; b = 1之**后，a 和b可能是或者可能不是引用同一个值为1的对象，这依赖于实现，但c = []; d = []之后，c 和d可以保证是引用两个不同的、唯一的、新创建的空列表。 （注意c = d = []是把相同的对象赋给c 和d。）

google了下，搞清楚了中文术语。数组就是arraylist，链表是linkedlist。python list的确是arraylist。所以access by offset效率高。   
: 【 在 wincss 的大作中提到: 】   
: : 没人会用链表实现存储很多元素的数据结构的，O(n)时间访问完全不可容忍   
: : 除非确定这个链表一定不太大（比如链表经常被用来解决Hash表冲突）   
: : 数组的插入是慢点，不过现在内存操作很快的，无所谓了。

# 一些FAQ

## Why are there separate tuple and list data types?

Lists and tuples, while similar in many respects, are generally used in fundamentally different ways. Tuples can be thought of as being similar to Pascal records or C structs; they’re small collections of related data which may be of different types which are operated on as a group. For example, a Cartesian coordinate is appropriately represented as a tuple of two or three numbers.

Lists, on the other hand, are more like arrays in other languages. They tend to hold a varying number of objects all of which have the same type and which are operated on one-by-one. For example, os.listdir('.') returns a list of strings representing the files in the current directory. Functions which operate on this output would generally not break if you added another file or two to the directory.

Tuples are immutable, meaning that once a tuple has been created, you can’t replace any of its elements with a new value. Lists are mutable, meaning that you can always change a list’s elements. Only immutable elements can be used as dictionary keys, and hence only tuples and not lists can be used as keys.

元组实际上是不同类型数据的集合，但是它们经常被组合在一起。举个例子，人身体指标，由身高，体重，三围，体质等组成，属于不同的数据类型，但是应当集合在一起。

list列表是数组，同类元素的集合

## heapq的实现

如果你使用内置类型，heapq默认使用元素的’ < ‘来排序元素。

如果你想要使用外部类型，请定义\_\_lt\_\_函数

## lambda表达式

lambda表达式只能有一个表达式语句，但其实这个表达式语句还可以是一个函数调用

## functools库

1. 在list的sort方法中，由于python3已经停用了cmp关键字参数，所以必须使用key来自定义排序。现在有下面几个需求：

1**.简单一维数组**把[1,3,5,7,9] 倒序排序 实现方法有 key=lambda x:-x , reverse=True key=functools.cmp\_to\_key(lambda x,y:y-x)

2.**多维数组** 如果是多维数组，要确定以哪个键为标准排序，并且确定该键是否逆序。

最通用的方式是是cmp函数，使用functools.cmp\_to\_key函数，将cmp与减法对应起来。

如果多维数组的元素的元素(如[[3,5]])支持负号操作，可以使用key=lambda x:x[0],-x[1] 表示按照第一维升序，第二维降序的方式排序。

而通用的cmp会如下：functools.cmp\_to\_key(f):

def f(x,y):

if x[0]==y[0]:

return y[0]-x[0]

else:

return x[0]-y[0]

1. **引入operator**还可以引入operator库中的itemgetter函数生成key，itemgetter(I,j,k…)表示依次获取每个元素。唯一不足的是，它只支持默认的升序排序。

**因此我们赞成使用cmp函数**

## 使用dict，multidict完成指定范围的查找

**一键多值**的情况：

key:{value-set}

这种dict，逐个加入元素，当加入失败的时候，把这个件并入key的set中，当然，此时的set不能再有相同元素了

dict对应的操作：

x in dict 键x是否存在

d [ x ] = y  若键 x 存在，则修改 x 对应的值为 y， 若键 x 不存在，则在字典 d 中增加键值对 x : y

set对应的操作：

add

remove

[not] in

issubset

union,intersection,difference,symmetric\_difference