第2章 列表和元组

1.序列

列表和元组的区别: 列表可以修改,元组不可以修改

(列表几乎都可代替元组,除了字典里面的键必须是元组)

数据结构容器: 容器是包含其他对象的任意对象 两类主要容器: 序列 映射

序列包括列表和元组 映射包括字典 集合既不是序列也不是映射

2.通用序列操作

索引

序列中的所以元素都是有编号的,都是从0开始递增 使用负数表示从右边,就是最后一个元素开始计数。最后一个元素的编号是-1 可以对序列直接索引。如: a=raw_input("year:")[3]

分片

分片通过相隔的两个索引来实现。第一个索引包括在内,第二个不包括在内。

部分索引: number=[1,2,3,4,5] number[-3:] number[:3] number[:] [3,4,5] [1,2,3] 整个序列

步长: number[0:4:2] [1,3] 步长为 2, 将跳过某些元素

number[5:1:-1] [5,4,3] 步长为负数,从右往左取数

加

只有类型相同的序列才可以进去连接。 如: [1,2,5]+ [3,4] [1,2,5,3,4]

乘

返回一个新序列,原先序列被重复 x 次

None,空列表和初始化

None: 内建值 表示这里什么都没有,初始化一个长度为 10 的列表

seq=[None]*10

空列表: [] 里面什么东西也没有

成员资格

使用in运算符来检查一个值是否在序列中。

如: p='hello' 'h' in p user=['hello','world'] 'hello' in user

长度,最小值和最大值

len min max

3.列表

1.list 函数

字符串和列表互转

a=list("hello") b=''.join(a)

2.基本的列表操作

赋值: 不能给一个不存在的元素赋值

删除: del 语句

分片赋值: name=list("perl") name[1:]=list("ython")

3.列表方法

append: 直接修改原来的列表,并不是返回一个新列表

count: 统计出现的次数

extend: 在末尾一次性追加另一个列表的多个值

extend: 扩展了原来的列表 连接:返回一个新的列表

index: 找出某个值第一个匹配项的索引位置

insert: 将对象插入到列表中 num=[1,2,3,4,5] num.insert(2,6)

pop: 移除列表中的一个元素(默认为最后一个),并返回该值

?怎么用 append/insert/pop 来实现队列和栈

remove: 移除列表中某个值的第一个匹配项

reverse: 将列表元素反向存放

sort: 用于在原位置对列表进行排序(改变了列表) 不返回值

需要原列表和排序后的列表:

x=[2,1,3] y=x[:] y.sort() 或者使用 sorted 函数

高级排序: num=[2,1,3] num.sort(cmp)

4.元组

元组不能修改。创建一个值的元组方法:(3,)

1.tuple 函数(是一种类型)

将一个序列作为参数转换为一个元组

2.基本元组操作

主要是创建元组和访问元组

3.元组的意义

元组可以在映射(和集合的成员)中国当做键使用,而列表不行 元组作为很多内建函数和方法的返回值存在