# 第三章 列表

## 1.list 的增删改

**增**：list.append() 在末尾增加元素

list.insert(0,x) 在位置0添加元素x，添加完成后所有元素向后移位。

**删**：list.pop() 弹出末尾元素

List.pop(x) 弹出位置x的元素

del list[0] 删除位置0的元素

**何时用pop何时用del**

如果你要从列表中删除一个元素， 且不再以任何方式使用它， 就使用del 语句； 如果你要在删除元素后还能继续使用它， 就使用方法pop() 。

**根据值来删除元素：**

List.remove(x) 删除值为x的元素

## 2.组织list

### 1.永久性的排序：list.sort（）

### 2. 使用函数sorted() 对列表进行临时排序

要保留列表元素原来的排列顺序， 同时以特定的顺序呈现它们， 可使用函数sorted() 。 函数sorted() 让你能够按特定顺序显示列表元素， 同时不影响它们在列表中的原始排

列顺序

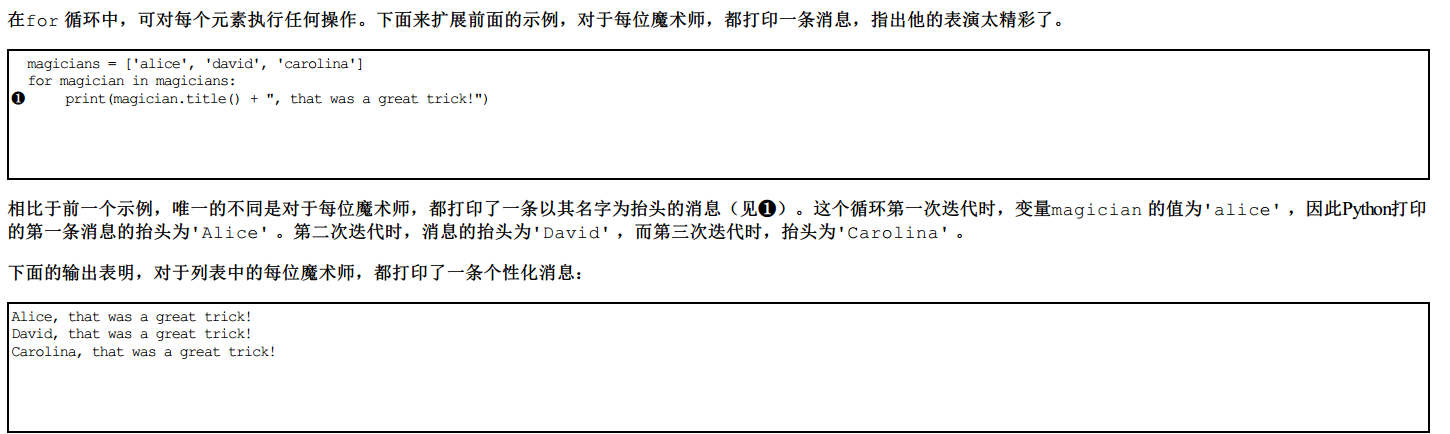
### 3.列表翻转：list.reverse()

### 4.列表长度：len(list)

# 第四章：操作列表

## 遍历：

### 在for 循环中执行更多的操作：list.title() 表示当前元素



## 数值列表：

### Range：

Range(a,b) 从a开始一直到b前一个数结束

Range(a,b,c) 步长为c

### squares = [value\*\*2 for value in range(1,11)]

[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

## List的复制与关联

my\_foods = ['pizza', 'falafel', 'carrot cake']

复制：

friend\_foods = my\_foods[:]

此时两个list是不相关的list，改变my\_foods不改变friend\_foods

关联：

friend\_foods = my\_foods

这种语法实际上是让Python将新变量friend\_foods 关联到包含在my\_foods 中的列表， 因此这两个变量都指向同一个列表。