

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