

# 列表

liuneng=80

xiaoshenyang=90

....+10分会很麻烦，占内存不好管理

用 [] 包起来的是列表

列表：可以存放任何类型的数据（最常用的数据类型之一）

eg：存整型，数字，字符串

关键字：list（ ）

- 列表的创建

三种列表：

1. 普通列表：l=[1,2,3]
2. 混合列表：l=[1, 'hehe', [2, 'hehe']]
3. 空列表：l=[]  
len(l)

- 向列表添加元素

列表中的每一个数据---元素

1.append（object）：

object：任何对象（万物皆对象）

向列表中添加一个元素，并将该元素添加到末尾

l.append(元素)

只能传递一个参数

2.extend(iterable):

iterable:可迭代对象

将可迭代对象中的多个元素逐个添加到列表中

区别：

1.append([1,2,3])    l=[1,2,[1,2,3]]

2.extend([1,2,3])    l=[1,2,3,1,2,3]

3. insert（索引,数据）

索引：index----下标

- 从列表中获取元素

### 1. 获取一个值

先获取下标，如果超出的列表的下标范围，那么会报错  
根据下标获取想要的值  
不能越界  
取多值

### 2. 遍历

```
for
    列表的遍历

互换
    找个中间人
    l[0],l[1]=l[1],l[0]
```

## • 从列表中修改元素

```
l[]=新值
l=[1,2,3]
for i in l:
    i+=1
    print(i)
print(l)
如果对列表做运算：
    for i in range(len(l)):
        l[i]+=1
    print(l)
```

## • 从列表中删除

1.remove(元素)如果列表中没有这个元素会报错  
l.remove(5)

2.del 是一个删除语句，不是一个方法  
del l.[0] del l 删除整个列表 删除的是变量：l2=l1

3.pop() 栈！先进后出，也就是后进先出  
画个井 s|\_|  
l.pop()不加参数时时从后面删除一个  
l.pop(下标)删除下标所对应的元素  
返回值：s=l.pop()  
再删除元素时会返回所删除的元素  
下标越界或者是空列表时报错

life is short,you need python  
意译：生命诚可贵，python价更高

以上是列表的基本使用方法

## 列表的高级特性列表的分片b

### 切片

l[起始下标：结束下标] （从起始下标到终止下标的所有元素，但是终止下标取不到）  
不会出现下标越界错误  
分片实质上是对原有列表中的元素进行的拷贝，原列表并没有发生改变

简化：

```
l[:5]    起始值是0时可以不写
l[4:]    从起始下标开始取到最后，那么最后可以省略不写
l[:]     输出整个列表元素拷贝了一份
```

区别：

```
l1=[1,2,3]
l2=l1
l3=l1[:]
l1[0]=520
print(l1,l2,l3)
```

跳过！分片进阶：

```
l[起始:结束:步长]
l=[1,2,3,4]
print(l[:2])
l[::-1]倒序输出
```

- 列表中的运算符

- 1.比较运算符（bool运算符）

先比较第一个，如果第一个元素相等，进而比较第二个，以此类推

```
l=['abc','acd']
l2=['abc','bcd']
```

- 2.逻辑运算符

```
and or not
l>l1 and l2<l3
```

- 3.列表的拼接

列表的加法

```
l=[1,2,3]+l1=[4,5,6]
extend([4,5,6])
```

- 4.重复运算符

```
l=[1,2,3]
l*3
```

- 5.成员关系运算符

in在---中      not in

- 列表的函数

dir（列表）

- 1.count(元素)：查看元素出现的次数

```
l.count(3)
```

- 2.index（元素）

返回元素在列表中的位置（默认第一次出现的位置）

index(元素，起始下标，结束下标)

- 3.reverse()

将整个列表进行原地反转

```
l=[1,2,3,4,5]
l.reverse()
print(l)
```

- 4.sort()

快速排序，默认是从小到大

考虑：如何到大排序

```
1.reverse
2.sort([key],reverse)
    func : 排序算法
    key : 关键字
    reverse : 默认是False , 给True
1.sort ( reverse=True )
5.clear()
    清空列表中所有元素
1.clear()
```

- 二维列表

列表中的元素也是个列表，每个元素有两个下标

```
l=[[1,2,3],[4,5,6],[7,8,9]]
```

```
l[][]
```

\* 遍历二维列表