

4.1.2 列表

- 列表 (list)

数据类型可存储由多个值组成的序列。在列表中，值可以是任何数据类型，称为元素 (element) 或项 (item)。

- Python的列表是有序的
- Python的列表是动态的





4.1.2 列表

- 创建列表

将逗号分隔的不同的数据项使用方括号 “[]”括起来即可创建列表。

```
>>>list1 = ['physics', 'chemistry', 1997, 2000]
```

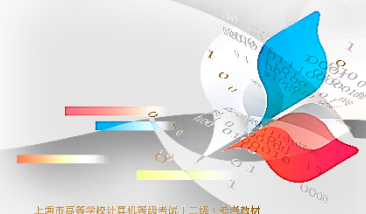
```
>>>list2 = [1, 2, 3, 4, 5 ]
```

```
>>>list3 = ["a", "b", "c", "d"]
```

列表允许嵌套，也就是说列表中的成员同样可以是列表。

```
>>>olist=[1,'str',['name','goofy'],...]
```

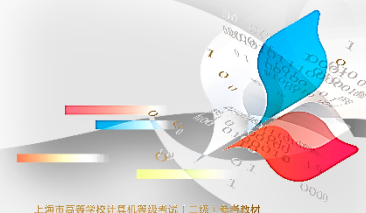
```
>>>nlist=[] #定义一个空的list:
```



4.1.2 列表

• 列表的基本操作

操 作	含 义
<code><seq>[i]</code>	索引 (求<seq>中位置索引为 i 的元素)
<code><seq>[i:j:k]</code>	切片 (求<seq>的位置索引为 i~j-1 的子列表)
<code><seq1>+<seq2></code>	将<seq1>和<seq2>连接
<code><seq>*<int-expr>或<int-expr>*<seq></code>	将<seq>复制<int-expr>次
<code>len(<seq>)</code>	求<seq>长度
<code>for<var>in<seq>:</code>	对<seq>中元素循环
<code><expr>in<seq></code>	查找<seq>中是否存在<expr>, 返回值为布尔类型
<code>del <seq></code>	删除列表
<code>del <seq>[i]</code>	删除列表中位置索引为 i 的元素
<code>max(<seq>)</code>	返回列表中最大值
<code>min(<seq>)</code>	返回列表中最小值



4.1.2 列表

• 【例4-3】 列表切片举例。

```
>>>list2 = [1, 2, 3, 4, 5, 6, 7 ]
```

```
>>>print ("list2[1:5]: ", list2[1:5])
```

```
list2[1:5]: [2, 3, 4, 5]
```

```
>>>l1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

```
>>>l1[0:2]
```

```
[1, 2]
```

```
>>>l1[:2]
```

```
[1, 2]
```

```
>>>l1[2:]
```

```
[3, 4, 5, 6, 7, 8, 9, 10, 11]
```

```
>>>l1[2:-1]
```

```
[3, 4, 5, 6, 7, 8, 9, 10]
```

```
>>>l1[:]
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
```

```
>>>l1[::2]
```

```
[1, 3, 5, 7, 9, 11]
```

```
>>>l1[0:7:2]
```

```
[1, 3, 5, 7]
```

```
>>>l1[7:0:-2]
```

```
[8, 6, 4, 2]
```

4.1.2 列表

- 【例4-4】 列表元素的更改举例。

注意：列表与字符串的重要区别是，列表中的元素可被更改，因此可以使用赋值语句改变列表中任意元素的值。

```
>>>lst = [1, 2, 3, 4]
```

```
>>>lst [3]
```

```
4
```

```
>>>lst [3] = "Hello"
```

```
>>>lst
```

```
[1, 2, 3, 'Hello']
```

```
>>>lst [2] = 7
```

```
>>>lst
```

```
[1, 2, 7, 'Hello']
```

```
>>>lst [1:3] = ["Slice", "Assignment"]
```

```
>>>lst
```

```
[1, 'Slice', 'Assignment', 'Hello']
```





4.1.2 列表

- 【例4-4】 列表元素的更改举例(续)

```
>>> mlist = [['ColA', 'ColB', 'ColC'], [1, 2, 3], [4, 5, 6], [7, 8, 9]]
>>> mlist[0][1] = 'Col_D'
>>> mlist[2][1] = 50
>>> mlist
[['ColA', 'Col_D', 'ColC'], [1, 2, 3], [4, 50, 6], [7, 8, 9]]
```



4.1.2 列表

- 运算符 '+' (拼接) 和 '*' (复制)

```
>>> zeroes = [0]*6
```

```
>>> print (zeroes)
```

```
[0,0,0,0,0,0]
```

```
>>> len(zeroes)
```

```
6
```

```
>>> print (2*['a']+['c']*3)
```

```
['a', 'a', 'c', 'c', 'c']
```

- 列表的成员检查(in):

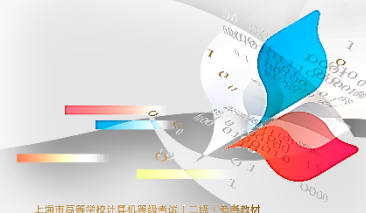
```
>>> lst = [1,2,3,4]
```

```
>>> 3 in lst
```

```
True
```

```
>>> 5 in lst
```

```
False
```



4.1.2 列表

如果将一个序列变量赋值给另外一个变量，则这2个变量表达了同一个序列。

```
a = [2, 3, 5, 7, 11, 13]
```

```
b = a
```

```
b[0] = 1
```

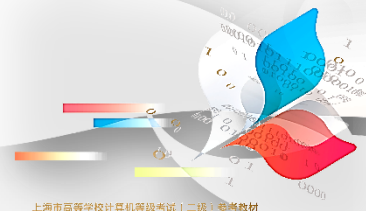
```
print(a)
```

```
[1, 3, 5, 7, 11, 13]
```

如果希望2个变量各自拥有独立的序列，可使用切片。

```
a = [2, 3, 5, 7, 11, 13]
```

```
b = a[:]
```

4.1.2 列表

- **【例4-5】 列表元素求和举例。**

用对列表中元素循环操作可实现列表元素求和。

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

```
sum=0
```

```
for i in s:
```

```
    sum=sum+i
```

```
print ("sum is",sum)
```

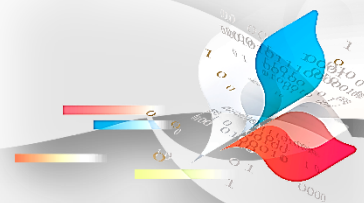
sum is 45

4.1.2 列表

- 【例4-6】我国居民身份证号码由17位数字和1位校验码组成。其中，前6位为所在地编号，第7~14位为出生年月日，第15~17位为登记流水号，其中第17位偶数为女性，奇数为男性。校验码的生成规则为：将前面的身份证号码17位数分别乘以系数：7, 9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10, 5, 8, 4, 2，然后将这17个乘积相加，结果与11求模，余数只可能是0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10这11种数字，它们分别对应的最后一位身份证的号码为1, 0, X, 9, 8, 7, 6, 5, 4, 3, 2。例如，余数是2，最后一位数字就是罗马数字X，如果余数是10，则身份证的最后一位就是2。

现设计程序实现输入18位身份证号，辨别其真伪。若为真，则进一步判断性别；若不是18位或身份证号非法，则提示重新输入。





上海市高等学校计算机等级考试(二级)参考教材

Python程序设计基础

(第2版)

主 编 李东方 支晓勇

4.1.2 列表

```
factor=[7,9, 10, 5, 8, 4, 2, 1, 6, 3, 7, 9, 10, 5, 8, 4, 2]
```

```
last=['1', '0', 'X', '9', '8', '7', '6', '5', '4', '3', '2']
```

```
while True:
```

```
    id=input('请输入身份证号, 0则退出')
```

```
    if id=='0':
```

```
        break
```

```
    if len(id)!=18:
```

```
        print('输入位数不对, 请重新输入')
```

```
        continue
```

```
    else:
```

```
        sum=0
```

```
        for i in range(17):
```

```
            sum+=int(id[i])*factor[i]
```

```
        m=sum%11
```

```
        lastchar=id[-1]
```

```
        lastchar=lastchar.upper()
```

```
        if lastchar!=last[m]:
```

```
            print(id,'为合法身份证号码, ',end='')
```

```
            if int(id[-2])%2==0:
```

```
                print('为女性')
```

```
            else:
```

```
                print('为男性')
```

```
        else:
```

```
            print(id,'为非法号码')
```

请输入身份证号, 0则退出310110200007058616

310110200007058616 为非法号码

请输入身份证号, 0则退出310110200007058617

310110200007058617 为合法身份证号码, 为男性

请输入身份证号, 0则退出0