

本章总结

基本数据类型

| 可变数据类型 | 不可变数据类型 |
|--------|-----------------------------|
| list | 数字类(bool,int,float,complex) |
| dict | str |
| set | tuple |
| | frozenset |

扩展数据类型collectins

- 1.namedtuple(): 生成可以使用名字来访问元素内容的tuple子类
- 2.deque: 双端队列，可以快速的从另外一侧追加和推出对象
- 3.Counter: 计数器，主要用来计数
- 4.OrderedDict: 有序字典
- 5.defaultdict: 带有默认值的字典

练习

- 1、请用代码实现：利用下划线将列表的每一个元素拼接成字符串，li=['alex','eric','rain']
- 2、查找列表中元素，移除每个元素的空格，并查找以a或A开头并且以c结尾的所有元素。

```
li = ["alec", " aric", "Alex", "Tony", "rain"]  
  
tu = ("alec", " aric", "Alex", "Tony", "rain")  
  
dic = {'k1': "alex", 'k2': ' aric', 'k3': "Alex", 'k4': "Tony"}
```

- 3、写代码，有如下列表，按照要求实现每一个功能

```
li=['alex','eric','rain']
```

- 计算列表长度并输出
- 列表中追加元素“seven”，并输出添加后的列表
- 请在列表的第1个位置插入元素“Tony”，并输出添加后的列表
- 请修改列表第2个位置的元素为“Kelly”，并输出修改后的列表
- 请删除列表中的元素“eric”，并输出修改后的列表
- 请删除列表中的第2个元素，并输出删除的元素的值和删除元素后的列表
- 请删除列表中的第3个元素，并输出删除元素后的列表
- 请删除列表中的第2至4个元素，并输出删除元素后的列表
- 请将列表所有的元素反转，并输出反转后的列表
- 请使用for、len、range输出列表的索引
- 请使用enumerate输出列表元素和序号（序号从100开始）
- 请使用for循环输出列表的所有元素

4、写代码，有如下列表，请按照功能要求实现每一个功能

```
li = ["hello", 'seven', ["mon", ["h", "kelly"], 'all'], 123, 446]
```

- 请根据索引输出“Kelly”
- 请使用索引找到'all'元素并将其修改为“ALL”，如：li[0][1][9]...
- 写代码，有如下元组，按照要求实现每一个功能

5、写代码，有如下元组，请按照功能要求实现每一个功能

```
tu=('alex','eric','rain')
```

- 计算元组长度并输出
- 获取元组的第2个元素，并输出
- 获取元组的第1-2个元素，并输出
- 请使用for输出元组的元素
- 请使用for、len、range输出元组的索引
- 请使用enumerate输出元组元素和序号（序号从10开始）

6、有如下变量，请实现要求的功能

```
tu = ("alex", [11, 22, {"k1": 'v1', "k2": ["age", "name"], "k3": (11,22,33)}, 44])
```

- 讲述元组的特性
- 请问tu变量中的第一个元素“alex”是否可被修改？
- 请问tu变量中的“k2”对应的值是什么类型？是否可以被修改？如果可以，请在其中添加一个元素“Seven”

- 请问tu变量中的"k3"对应的值是什么类型？是否可以被修改？如果可以，请在其中添加一个元素“Seven”

7、字典

```
dic = {'k1': "v1", "k2": "v2", "k3": [11,22,33]}
```

- 请循环输出所有的key
- 请循环输出所有的value
- 请循环输出所有的key和value
- 请在字典中添加一个键值对, "k4": "v4", 输出添加后的字典
- 请在修改字典中“k1”对应的值为“alex”，输出修改后的字典
- 请在k3对应的值中追加一个元素44，输出修改后的字典
- 请在k3对应的值的第1个位置插入个元素18，输出修改后的字典

8、转换

- 将字符串s = "alex"转换成列表
- 将字符串s = "alex"转换成元祖
- 将列表li = ["alex", "seven"]转换成元组
- 将元祖tu = ('Alex', "seven")转换成列表
- 将列表li = ["alex", "seven"]转换成字典且字典的key按照10开始向后递增

9、元素分类

有如下值集合[11,22,33,44,55,66,77,88,99,90]，将所有大于66的值保存至字典的第一个key中，将小于66的值保存至第二个key的值中。

即：{'k1':大于66的所有值, 'k2':小于66的所有值}

10、输出商品列表，用户输入序号，显示用户选中的商品

```
商品li = ["手机", "电脑", '鼠标垫', '游艇']
```

- 允许用户添加商品
- 用户输入序号显示内容

11、用户交互显示类似省市县N级联动的选择

- 允许用户增加内容
- 允许用户选择查看某一个级别内容

12、列举布尔值是False的所有值

13、有两个列表

l1 = [11,22,33]

l2 = [22,33,44]

- 获取内容相同的元素列表
- 获取l1中有，l2中没有的元素列表
- 获取l2中有，l3中没有的元素列表
- 获取l1和l2中内容都不同的元素

14、利用For循环和range输出

- For循环从大到小输出1 - 100
- For循环从小到大输出100 - 1
- While循环从大到小输出1 - 100
- While循环从小到大输出100 - 1

15、利用for循环和range输出9 * 9乘法表

作业

一、三级菜单：

数据结构：

```
menu = {  
    '北京': {  
        '海淀': {  
            '五道口': {  
                'soho': {},  
                '网易': {},  
                'google': {}  
            },  
            '中关村': {  
                '爱奇艺': {},  
                '汽车之家': {},  
                'youku': {},  
            },  
            '上地': {  
                '百度': {},  
            },  
        },  
        '昌平': {
```

```

        '沙河':{
            '老男孩':{},
            '北航':{},
        },
        '天通苑':{},
        '回龙观':{},
    },
    '朝阳':{},
    '东城':{},
},
'上海':{
    '闵行':{
        "人民广场":{
            '炸鸡店':{}
        }
    },
    '闸北':{
        '火车战':{
            '携程':{}
        }
    },
},
    '浦东':{},
},
    '山东':{},
}

```

需求：

- 可依次选择进入各子菜单
- 可从任意一层往回退到上一层
- 可从任意一层退出程序

所需新知识点：列表、字典

二、购物车程序：

数据结构：

```

goods = [
    {"name": "电脑", "price": 1999},
    {"name": "鼠标", "price": 10},
    {"name": "游艇", "price": 20},
    {"name": "美女", "price": 998},
    .....
]

```

功能要求：

- 1、启动程序后，输入用户名密码后，让用户输入工资，然后打印商品列表
- 2、允许用户根据商品编号购买商品
- 3、用户选择商品后，检测余额是否够，够就直接扣款，不够就提醒
- 4、可随时退出，退出时，打印已购买商品和余额
- 5、在用户使用过程中， 关键输出，如余额，商品已加入购物车等消息，需高亮显示

扩展需求：

- 1、用户下一次登录后，输入用户名密码，直接回到上次状态，即上次消费的余额什么的还是那些，再次登录可继续购买
- 2、允许查询之前的消费记录