



Python与金融数据挖掘(10)

文欣秀

wenxinxiu@ecust.edu.cn

如何单击按钮调用爬虫程序?



面向对象程序设计

面向对象程序设计将**数据**以及对**数据的操作**放在一起，作为一个相互依存、不可分割的整体进行处理。**对象**（包含**属性**和**方法**）是程序的**基本单元**，每个对象都可以与程序中其它对象进行交互，从而提高软件的重用性、灵活性和扩展性。

类与对象

类：建立对象的模板，它定义了事物的**属性**和事物可以执行的**行为**；利用类模板所创建的对象称为**类的实例**，类与实例之间是**抽象与具体**的关系。

同一类的不同实例之间具有如下特点：

- ◆ 相同的**操作**集合
- ◆ 相同的**属性**集合
- ◆ 不同的**对象名**

类应用示例一

```
class Animal(object):
```

```
    def __init__(self, voice='miao'):
```

```
        self.voice=voice
```

```
    def say(self):
```

```
        print(self.voice)
```

```
kitty=Animal()
```

```
kitty.say()
```

```
bob=Animal('wow')
```

```
bob.say()
```

类应用示例二

```
class animals:
```

```
    def breath(self):
```

```
        print('breathing')
```

```
class dog (animals):
```

```
    def eat(self):
```

```
        print('eating')
```

```
bob=dog()
```

```
bob. breath()
```

```
bob. eat()
```

类的三种特征

封装性：将基本类结构的细节（如实例变量）隐藏起来，
通过**方法接口**实现对实例变量的所有必要访问。

继承性：基于类的特征创建子类，子类可以继承父类的
属性和方法。

多态性：使用运算符或方法时，根据调用它们的对象类型，
执行**不同的**操作过程。

Python常用GUI库

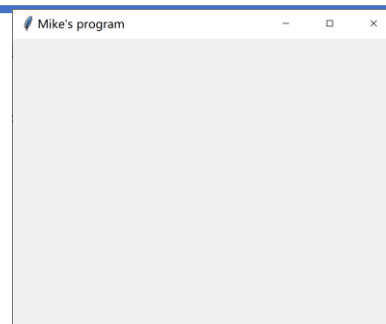
- ◆ **tkinter**
- ◆ **wxPython**
- ◆ **PyQt5**
- ◆ **PySide2**
- ◆ ...

tkinter设计步骤

- ◆ 导入tkinter模块
- ◆ 创建GUI主窗体
- ◆ 添加人机交互控件并编写相应的函数
- ◆ 在主事件循环中等待用户触发事件响应

案例分析

```
from tkinter import *  
  
root=Tk()  
  
root.title("Mike's program")  
  
root.geometry("400x300")  
  
root.mainloop()
```



GUI构件

- 按钮Button
- 单选钮Radiobutton
- 标签Label
- 多选钮Checkbutton
- 单行文本框Entry
- 菜单Menu
- 多行文本框Text
- 模式对话框...

Button构件

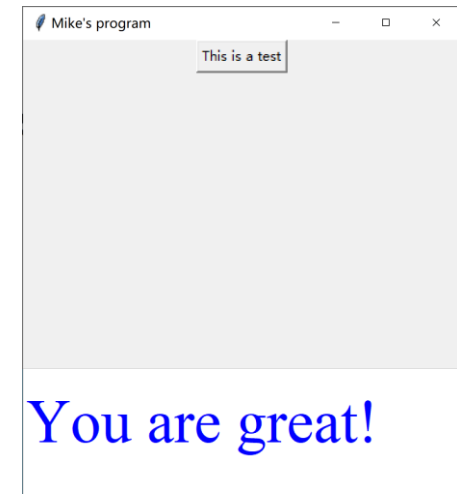
■ 类:Button

btn = Button(窗口,选项设置)

- **text:**按钮上的文本
- **height,width,fg,bg**
- **state:NORMAL/DISABLED/ACTIVE**
- **command:**点击按钮时要执行的命令

案例分析

```
from tkinter import *  
root=Tk()  
root.title("Mike's program")  
root.geometry("400x300")  
def hello():  
    print("You are great!")  
b=Button(root, text="This is a test", command=hello)  
b.pack()  
root.mainloop()
```



爬虫案例

```
# coding=utf-8
from tkinter import *
def verify():
    import cra
root=Tk()
root.title("XXX的爬虫程序")
root.geometry("300x200")
one=Button(root,text='金融信息爬虫',width=20,height=3,command=verify)
one.place(x=70,y=50)
root.mainloop()
```



课堂练习

编写程序，研发一个客户随机抽奖小程序。



随机抽一名客户模块

```
def one():  
    aList=[]  
    with open("client.csv", 'r') as handle:  
        for i in handle:  
            i=i.strip()  
            info=i.split(",")  
            aList.append(info[1])  
    time.sleep(1)  
    result=random.choice(aList)  
    showinfo("抽奖", "恭喜{}中奖.".format(result))
```

抽一名客户



抽奖



恭喜陈振宇中奖.

确定

随机抽三名客户模块

```
def three():  
    aList=[]  
    with open("client.csv", 'r') as handle:  
        for i in handle:  
            i=i.strip()  
            info=i.split(",")  
            aList.append(info[1])  
    time.sleep(1)  
    result=random.sample(aList,3)  
    for i in result:  
        showinfo("抽奖", "恭喜{}中奖.".format(result))
```



主窗体设计

```
from tkinter import *
from tkinter.messagebox import *
import time
import random
root=Tk()
btn_One=Button(root, text='抽一名客户', width=40,
                height=5, command=one)
btn_One.pack()
btn_Three=Button(root, text='抽三名客户', width=40,
                 height=5, command=three )
btn_Three.pack()
#...
root.mainloop()
```



随机抽N名客户模块

```
def call(x):  
    aList=[]  
    with open("client.csv", 'r') as handle:  
        for i in handle:  
            i=i.strip()  
            info=i.split(",")  
            aList.append(info[1])  
    time.sleep(2)  
    namelist=random.sample(aList,x)  
    for i in namelist:  
        showinfo("抽奖", "恭喜{}中奖.".format(i))
```

抽一名客户

主窗体设计 (修改)

```
from tkinter import *
from tkinter.messagebox import *
import time
import random
root=Tk()
btn_One=Button(root, text='抽一名客户', width=40,
                height=5, command=lambda: call(1))
btn_One.pack()
btn_Three=Button(root, text='抽三名客户', width=40,
                 height=5, command=lambda: call(3))
btn_Three.pack()
#...
root.mainloop()
```



案例分析

编写程序，在窗体上设计如样张所示效果。



标签构件

■ 类: Label

lb = Label(窗口,选项设置)

- **text:** 标签文本内容
- **font:** 文本字体
- **width/height:** 标签宽度、高度
- **fg/bg:** 前景色、背景色

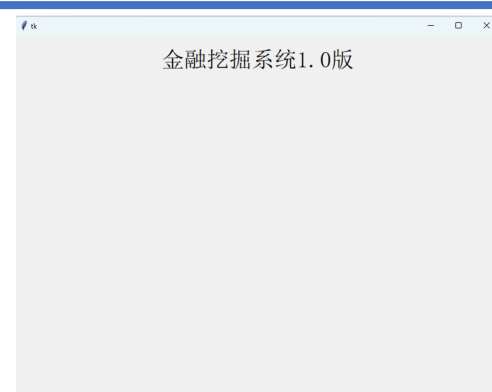
静态内容示例

编写程序，在窗体上设计标签显示系统版本信息。



静态内容示例

```
from tkinter import *  
root=Tk()  
root.geometry("800x600")  
lb=Label( root,text='金融挖掘系统1.0版',  
          font=('宋体',28),width=20,height=2)  
lb.pack()  
root.mainloop()
```



动态时钟

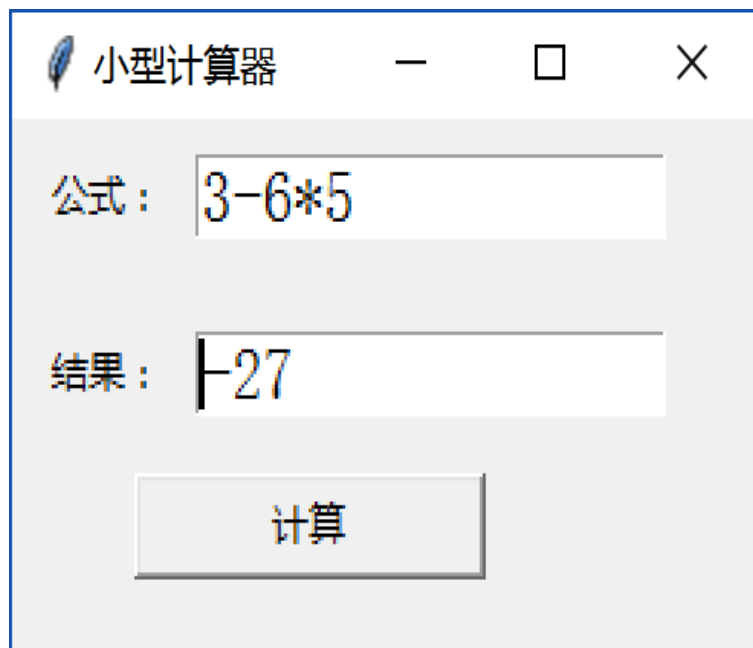
```
from tkinter import *
import time
def gettime():
    current=time.strftime("%H:%M:%S")
    lb2.configure(text=current)
    root.after(1000, gettime)

root=Tk()
root.geometry("800x600")
lb=Label(root,text='金融挖掘系统1.0版', font=('宋体',28),width=20,height=2)
lb.pack()
lb2=Label(root,text="",fg='blue',font=("Times New Roman", 22))
lb2.pack()
gettime()
root.mainloop()
```



计算器案例

编写程序，设计界面并实现数学计算功能。



单行文本构件

■ 类Entry:单行文本编辑

e = Entry(窗口,选项设置)

- ❑ **bd:**边界周围的指标的大小
- ❑ **font:**字体字号
- ❑ **show:**设置显示内容是否为"*"

单行文本控件常用方法

get(): 获取文件框的值，值为字符串

insert (index, s): 向文本框中插入值，**index**为插入位置，**s**为插入值

例： `ent1.insert(0,'你好')`

delete (first, last): 删除文本框里指定位置值

例： `ent1.delete(0, END)`

计算器案例

```
#coding=utf-8
from tkinter import *
def cal():
    result=eval(E1.get())
    E2.delete(0, END)
    E2.insert(END,result)
root = Tk()
root.title("小型计算器")
root.geometry("250x150")
```

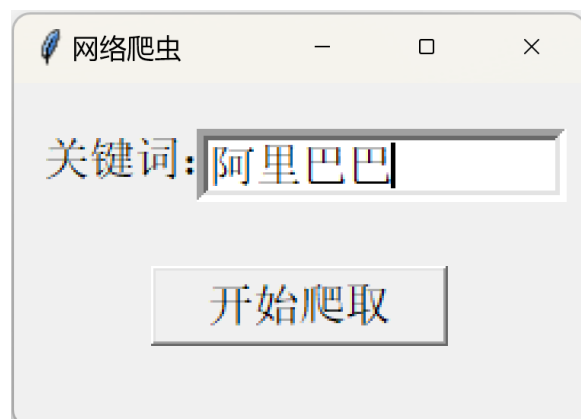
eval(): 计算字符串表达式的值并返回计算结果

计算器案例

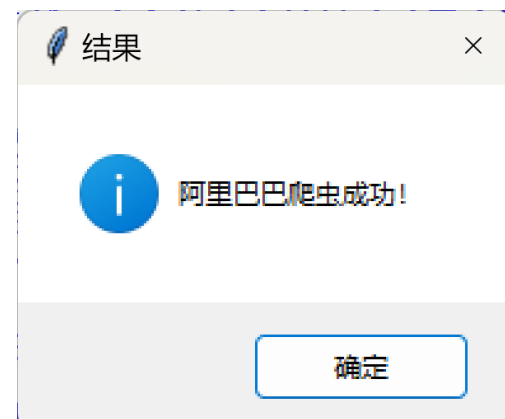
```
L1 = Label(root, text="公式: ")
L1.place(x=10,y=10)
E1 = Entry(root, bd =1,font=12,width=15)
E1.place(x=60,y=10)
L2=Label(root, text="结果: ")
L2.place(x=10,y=60)
E2 = Entry(root,bd =1,font=12,width=15)
E2.place(x=60,y=60)
B1 = Button(root, text="计算",width=15,command=cal)
B1.place(x=60,y=100)
root.mainloop()
```

案例分析

编写程序，在窗体上实现输入具体内容实现网络爬虫功能。



1. 阿里巴巴1688登顶苹果
<https://baijiahao.baidu.com>
2. 长信科技公司与阿里巴
<http://yuanchuang.10jqka.com.cn>
3. 新华都:2023年度与阿里



1. 阿里巴巴1688登顶苹果
<https://baijiahao.baidu.com/article?id=1790000000000000000>
2. 长信科技公司与阿里巴
<http://yuanchuang.10jqka.com.cn/2023/11/20/0908000000000000000.shtml>
3. 新华都：2023年度与阿里

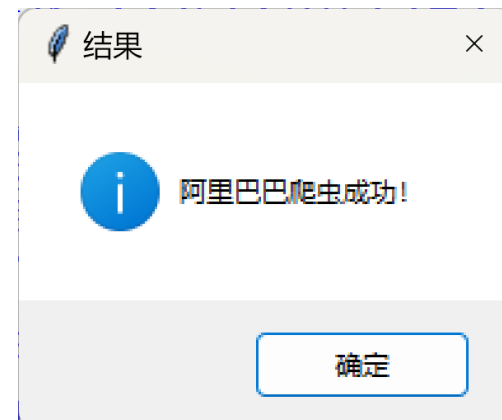
```
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36'}
```

```
url = 'http://www.baidu.com/s?tn=news&rtt=1&wd=' + company
```

```
res = requests.get(url, headers=headers).text
```

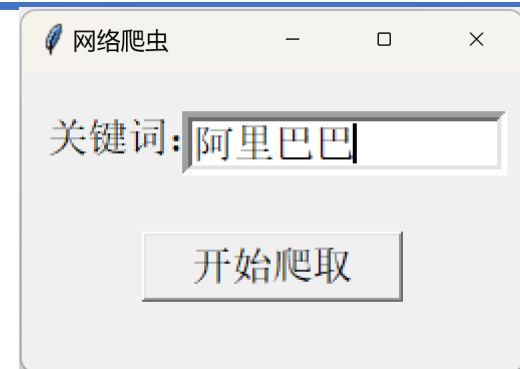

案例分析

```
p_href = '<h3 class="news-title_1YtI1 "><a href="(.*?)"'
href = re.findall(p_href, res, re.S)
p_title = '<h3 class="news-title_1YtI1 ">.*?>(.*?)</a>'
title = re.findall(p_title, res, re.S)
for i in range(len(title)):
    title[i] = title[i].strip()
    title[i] = re.sub('<.*?>', '', title[i])
    print(str(i + 1) + '.' + title[i])
    print(href[i])
showinfo("结果", "{ }".format(company+'爬虫成功! '))
except:
showinfo("结果", { } .format(company+'爬虫失败! '))
```



案例分析

```
root = Tk()
root.title("网络爬虫")
root.geometry("250x150")
L1 = Label(root, text="关键词: ", font=20)
L1.place(x=10, y=20)
E1 = Entry(root, bd=5, font=20, width=15)
E1.place(x=80, y=20)
B1 = Button(root, text="开始爬取", font=20, width=12, command=crawler)
B1.place(x=60, y=80)
root.mainloop()
```



扩展练习

编写程序，单击按钮实现网络爬虫和存入数据库功能。

网络爬虫

关键词:

开始爬取

存入数据库

结果

京东爬虫成功!

确定

存储

存入数据库成功

确定

1. 抖音京东加码小时达
<https://baijiahao.ba>
2. 京东31亿元北京拿地
<https://baijiahao.ba>
3. 京东股权曝光刘强东
<https://baijiahao.ba>
4. fabrique京东官方旗

company	title	href
京东	抖音京东加码小时达大厂打响即时	https://baijiaha
京东	京东31亿元北京拿地规划图曝光员	https://baijiaha
京东	京东股权曝光刘强东是最大股东拥	https://baijiaha
京东	fabrique京东官方旗舰店盛大开业	http://www.dzw

扩展练习答案

```
from tkinter import *  
from tkinter.messagebox import *  
import requests  
import re  
title=[]  
href=[]  
company=""
```

扩展练习答案

```
def crawler():
```

```
    try:
```

```
        headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100
Safari/537.36'}
```

```
        global company,title,href
```

```
        company=E1.get()
```

```
        url = 'http://www.baidu.com/s?tn=news&rtt=1&wd=' + company
```

```
        res = requests.get(url, headers=headers).text
```

```
        p_href = '<h3 class="news-title_1YtI1 "><a href="(.*?)"'
```

```
        href = re.findall(p_href, res, re.S)
```

```
        #...
```

扩展练习答案

```
#...
p_title = '<h3 class="news-title_1YtI1 ">.*?>(.*?)</a>'
title = re.findall(p_title, res, re.S)
for i in range(len(title)):
    title[i] = title[i].strip()
    title[i] = re.sub('<.*?>', "", title[i])
    print(str(i + 1) + '.' + title[i])
    print(href[i])
showinfo("结果", "{ }".format(company+'爬虫成功! '))
except:
    showinfo("结果", "{ }".format(company+'爬虫失败! '))
```

扩展练习答案

def save():

```
import pymysql
```

```
global company,title,href
```

```
try:
```

```
    conn = pymysql.connect(host="localhost", user="root",  
        password="123456", database="test")
```

```
    cur = conn.cursor()
```

```
    sql = """CREATE TABLE result (company CHAR(20),  
        title CHAR(100), href CHAR(100))"""
```

```
    cur.execute(sql)
```

```
    conn.commit()
```

```
    conn.close()
```

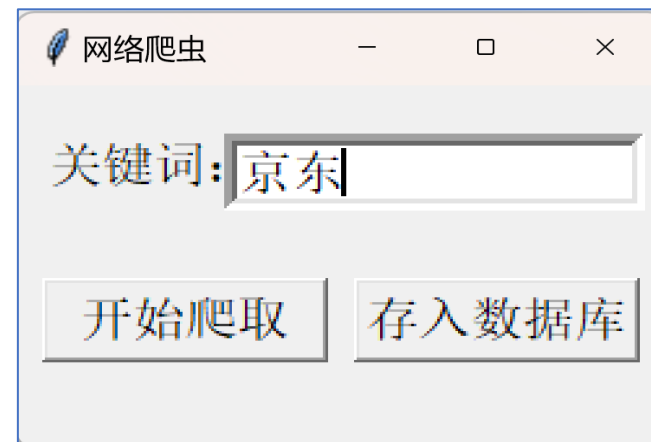
```
#...
```

扩展练习答案

```
# 将数据存入数据库
conn = pymysql.connect(host='localhost', port=3306, user='root',
password='123456', database='test')
cur = conn.cursor()
for i in range(len(title)):
    sql = "INSERT INTO result(company,title,href)
          VALUES (%s,%s,%s) "
    cur.execute(sql, (company, title[i], href[i]))
    conn.commit()
cur.close()
conn.close()
showinfo("存储","存入数据库成功")
except:
    showinfo("存储","存入数据库失败")
```


扩展练习答案

```
root = Tk()
root.title("网络爬虫")
root.geometry("250x150")
L1 = Label(root, text="关键词: ", font=20)
L1.place(x=10, y=20)
E1 = Entry(root, bd=5, font=20, width=15)
E1.place(x=80, y=20)
B1 = Button(root, text="开始爬取",
            font=20, width=10, command=crawler)
B1.place(x=10, y=80)
B2 = Button(root, text="存入数据库",
            font=20, width=10, command=save)
B2.place(x=130, y=80)
root.mainloop()
```



案例分析

胖猫（本名：刘杰，2003年-2024年4月23日），男，湖南省郴州市临武县人，因为游戏天赋过人，是一名游戏代练，喜欢玩王者荣耀的角色“梦奇”，于是给自己的游戏id名字起名“胖猫”。

2024年3月，胖猫疑似因和女友谭竹在感情上出现摩擦造成双方分手；4月11日，胖猫跳江；于4月23日离世。5月2日，胖猫姐姐晒出胖猫和女友的聊天记录，随后引发热议。5月3日，“胖猫”的姐姐发布了弟弟的火化证照，证实已被火化。

舆情数据按标题评分

```
score = []
title=["XX饼干成分不合格", "XX研发新产品","XX有偷税漏税行为"]
keywords = ['违约','不合格','偷税']
for i in range(len(title)):
    num = 10
    for k in keywords:
        if k in title[i]:
            num-=10
    score.append(num)
for i in range(len(title)):
    print("{} 评分为{}分".format(title[i],score[i]))
```

舆情数据按标题和内容评分

```
score = []
keywords = ['违约','不合格','偷税']
for i in range(len(title)):
    num = 10
    try:
        article=requests.get(href[i],headers=headers,timeout=10).text
    except:
        article="单个新闻爬取失败"
    for k in keywords:
        if (k in article) or (k in title[i]):
            num-=10
    score.append(num)
```

本段代码不能独立运行，需要标题和网页数据

舆情数据评分系统搭建

- ◆ 创建窗体和控件，用于输入新闻主题
- ◆ 编写爬虫模块，用于数据采集和清洗
- ◆ 编写舆情分析模块，用于数据的评分
- ◆ 编写数据库模块，用于存储统计数据
- ◆ 编写绘图模块，用于展示及相关性分析
- ◆ 编写机器学习算法模块，用于结果预测

輿情数据评分系统 (1)

```
from tkinter import *  
from tkinter.messagebox import *  
import requests  
import re  
#定义三个全局变量在函数之间共享数据  
title=[]  
href=[]  
company=""
```

舆情数据评分系统 (2)

```
def crawler():
    try:
        headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100 Safari/537.36'}
        global company,title,href
        company=E1.get()
        url = 'http://www.baidu.com/s?tn=news&rtt=1&wd=' + company
        res = requests.get(url, headers=headers).text
        p_href = '<h3 class="news-title_1 YtI1 "><a href="(.*?)"'
        href = re.findall(p_href, res, re.S)
        p_title = '<h3 class="news-title_1 YtI1 ">.*?>(.*?)</a>'
        title = re.findall(p_title, res, re.S)
        for i in range(len(title)):
            title[i] = title[i].strip()
            title[i] = re.sub('<.*?>', '', title[i])
            print(str(i + 1) + '.' + title[i])
            print(href[i])
        showinfo("结果",{ }.format(company+'爬虫成功! '))
    except:
        showinfo("结果",{ }.format(company+'爬虫失败! '))
```

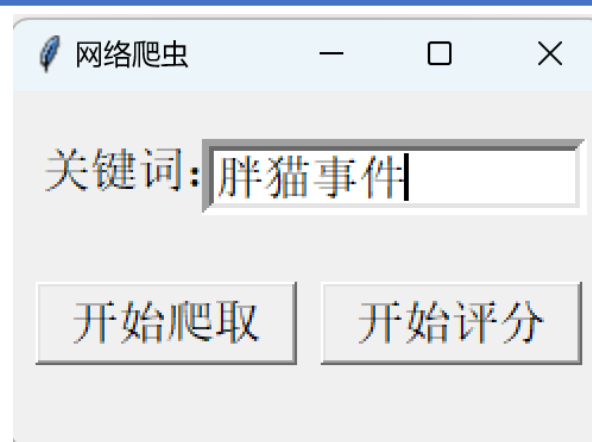
舆情数据评分系统 (3)

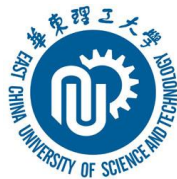
```
def grade():  
    global company,title,href  
    score = []  
    keywords = ['空包','白水'] # 这个关键词列表可以自己定义，这里只是为了演示  
    for i in range(len(title)):  
        num = 0  
        # 获取新闻正文  
        try:  
            article = requests.get(href[i], headers=headers, timeout=10).text  
        except:  
            article = '爬取失败'  
        # 只筛选真正的正文内容，旁边的滚动新闻之类的内容忽略  
        p_article = '<p.*?>(.*?)</p>' # 有的时候p标签里还有class等无关内容  
        article_main = re.findall(p_article, article) # 获取<p>标签里的正文信息  
        article = ''.join(article_main) # 将列表转换成为字符串  
        for k in keywords:  
            if (k in article) or (k in title[i]):  
                num -= 5  
        score.append(num)  
    for i in range(len(title)):  
        print(title[i],score[i])
```

胖猫事件引爆!空杯、空包、白水,多个知名品牌回应 -10
多家品牌就胖猫事件道歉 涉事门店关闭,订单10倍退款 0
话题:胖猫事件引爆!空杯、空包、白水,多个知名品牌回应 -10
21岁重庆胖猫跳江事件反转,真相却只有一个 0
“胖猫”事件引爆网络!空杯、空包、白水!茶百道、蜜雪冰城...
-10
21岁“胖猫”事件持续引发关注,律师建议提起刑事立案申请 0
胖猫谭竹情感PUA事件全过程,包括后续退款 0
... 赔偿金另外算;华为内部权力重新分配?知情人士回应;胖猫事件
闹... 0
网友祭奠“胖猫”点餐空包事件,朱小小螺蛳粉向公众致歉 -5
【茶百道、华莱士就「空包白水」事件回应】 -10

舆情数据评分系统 (4)

```
root = Tk()
root.title("网络爬虫")
root.geometry("250x150")
L1 = Label(root, text="关键词: ", font=20)
L1.place(x=10, y=20)
E1 = Entry(root, bd=5, font=20, width=15)
E1.place(x=80, y=20)
B1 = Button(root, text="开始爬取", font=20, width=10, command=crawler)
B1.place(x=10, y=80)
B2 = Button(root, text="开始评分", font=20, width=10, command=grade)
B2.place(x=130, y=80)
root.mainloop()
```





谢 谢