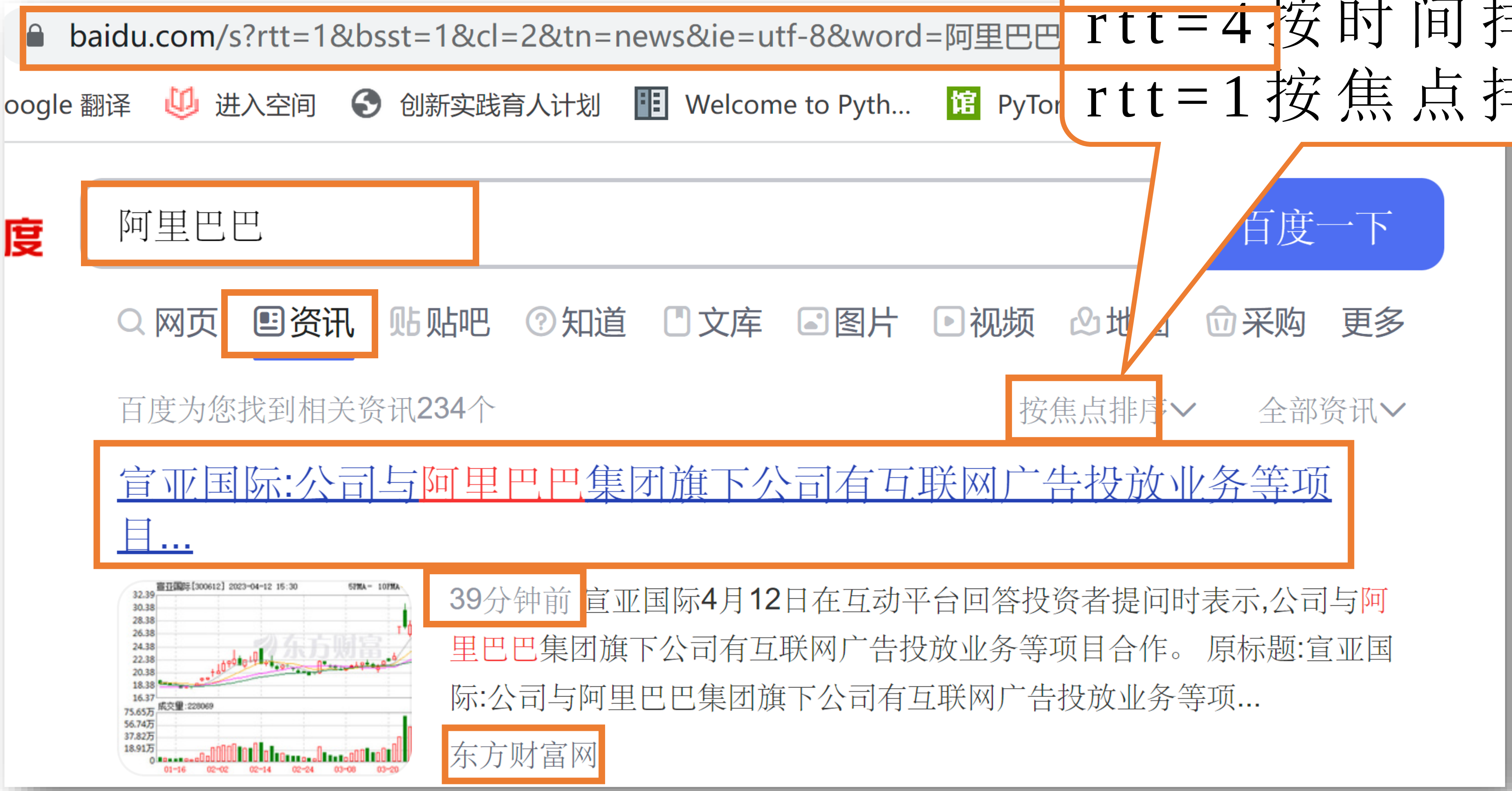
Python与金融数据挖掘(7)

文欣秀

wenxinxiu@ecust.edu.cn

案例分析



baidu.com/s?rtt=1&bsst=1&cl=2&tn=news&ie=utf-8&word=阿里巴巴

阿里巴巴

百度为您找到相关资讯234个　按焦点排序　全部资讯

宣亚国际:公司与阿里巴巴集团旗下公司有互联网广告投放业务等项目...

39分钟前 宣亚国际4月12日在互动平台回答投资者提问时表示,公司与阿里巴巴集团旗下公司有互联网广告投放业务等项目合作。 原标题:宣亚国际:公司与阿里巴巴集团旗下公司有互联网广告投放业务等项...

东方财富网

rtt=4 按时间排序
rtt=1 按焦点排序

案例分析

```
import requests
headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64                HTML,
like Gecko) Chrome/69.0.3497.100 Safari/537.36'}
def baidu(company):
    url = 'http://www.baidu.com/s?tn=news&rtt=1&wd=' + company
    res=requests.get(url, headers=headers)
    res.encoding=res.apparent_encoding
    data=res.text
    print(data)

baidu("阿里巴巴")
```

打开浏览器，输入about:version

资 讯

rtt=4按时间排序
rtt=1按焦点排序

模拟浏览器的访问请求

正则表达式修饰符含义

修饰符	描述
re.I	使匹配对大小写不敏感
re.L	做本地化识别（locale-aware）匹配
re.M	多行匹配，影响 ^ 和 $
re.S	使 . 匹配包括换行在内的所有字符
re.U	根据Unicode字符集解析字符。这个标志影响 \w, \W, \b, \B.
re.X	该标志通过给予你更灵活的格式以便你将正则表达式写得更易于理解。

输出搜索到的全部链接

```
import requests
import re
import time
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'}
def baidu(company):
    url = 'http://www.baidu.com/s?tn=news&rtt=4&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)
    print(href)
    …
baidu('阿里巴巴')
```

输出搜索到的标题、日期、来源

```
    …
    p_title = '<h3 class="news-title_1YtI1 ">.*?>(.*?)</a>'
    title = re.findall(p_title, res, re.S)
    print(title)
    p_date = '<span class="c-color-gray2 c-font-normal c-gap-right-\
xsmall" .*?>(.*?)</span>'
    date = re.findall(p_date, res)
    print(date)
    p_source = '<span class="c-color-gray" .*?>(.*?)</span>'
    source = re.findall(p_source, res)
    print(source)
```

搜索结果清洗及输出

```
for i in range(len(date)):
    title[i] = title[i].strip()
    title[i] = re.sub('<.*?>', '', title[i])
    if ('小时' in date[i]) or ('分钟' in date[i]):
        date[i] = time. strftime("%Y-%m-%d")
    else:
        date[i] = date[i]
    print(str(i + 1) + '.' + title[i] + '(' + date[i] + '-' + source[i] + ')')
    print(href[i])
```

问题：如何自动生成舆情数据分析报告？

自动生成舆情数据报告

```
fobj = open('E:\\分析报告.txt', 'a', encoding='utf-8')
for i in range(len(date)):
    title[i] = title[i].strip()
    #...清洗标题与时间
    fobj. write(str(i + 1) + '.' + title[i] + '(' + date[i] + '-' + source[i] + ')' + '\n')
    fobj. write(href[i] + '\n')
fobj.close()
```

问题：如何爬取多个公司的数据？

爬取多个公司数据

```
companys = ['阿里巴巴', '万科集团', '腾讯', '京东']
for each in companys:
    baidu(each)
    print(str(each)+'成功！')
```

问题： 如何爬取多个公司的多页数据?

爬取多公司多页数据

```
# 爬取多个公司的多页, 可以给函数传入两个参数
def baidu(company, page):
    num = (page-1) * 10  # 参数规律是（页数-1）*10
    url = 'http://www.baidu.com/s?tn=news&rtt=4&wd='+company+'&pn='+str(num)
    res = requests.get(url, headers=headers).text
    print(res)
```

爬取多公司多页数据

```
companys = ['阿里巴巴', '万科集团', '百度集团', '腾讯', '京东']
for company in companys:
    for i in range(10):  # 这里一共爬取了10页
        baidu(company, i+1)  # i+1表示第几页
        print(company + '第' + str(i+1) + '页爬取成功')
        time.sleep(3)
```

问题： 如何解决爬取网页时程序崩掉?

错误处理

```
try:
    <语句块1>
except <异常类型1>:
    <语句块2>
else:
    <语句块3>
finally:
    <句块4>
```

错误处理案例

```
try:
    num1=int(input("The first number:"))
    num2=int(input("The second number:"))
    num3=num1/num2
    print(num3)
except ZeroDivisionError:
    print("除数为零错误")
```

错误处理案例

```
try:
    filename = input("input file name:")
    fobj = open(filename,"r")
    for line in fobj:
        print(line.strip())
except IOError:
    print("文件不存在")
else:
    fobj.close()
```

错误处理案例

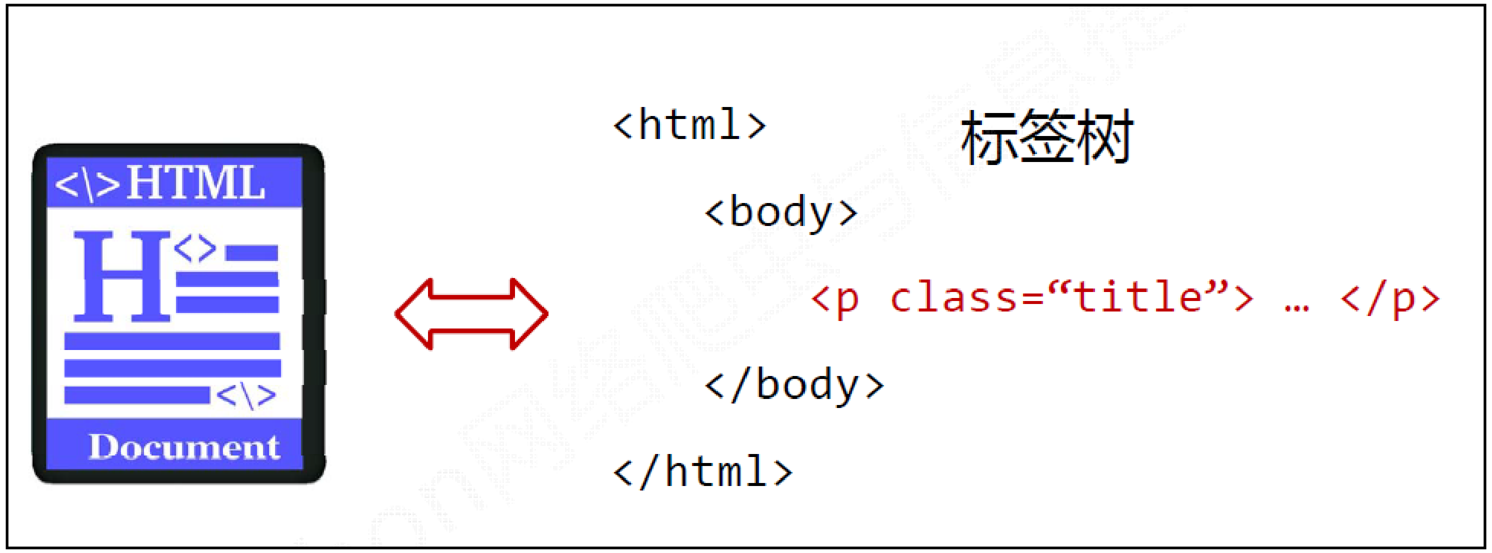
```
try:
    first=int(input("第一个数："))
    second=int(input("第二个数："))
    print(first+second)
except:
    print("未知错误")
else:
    print("成功运行")
finally:
     print("程序结束")
```

爬取多个公司多页数据(修改)

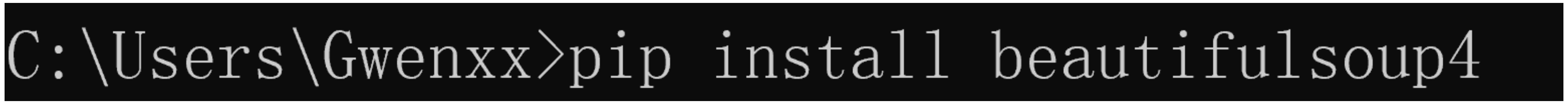
```
companys = ['阿里巴巴', '万科集团', '腾讯', '京东']
for company in companys:
    for i in range(10):
        try:
            baidu(company,i+1)
            print(company + '第' + str(i+1) + '页爬取成功')
            time. sleep(2)
        except:
            print("{}".format(company +'爬虫失败！ '))
```

Beautiful Soup库

Beautiful Soup： 是解析、遍历、维护"标签树"的功能库



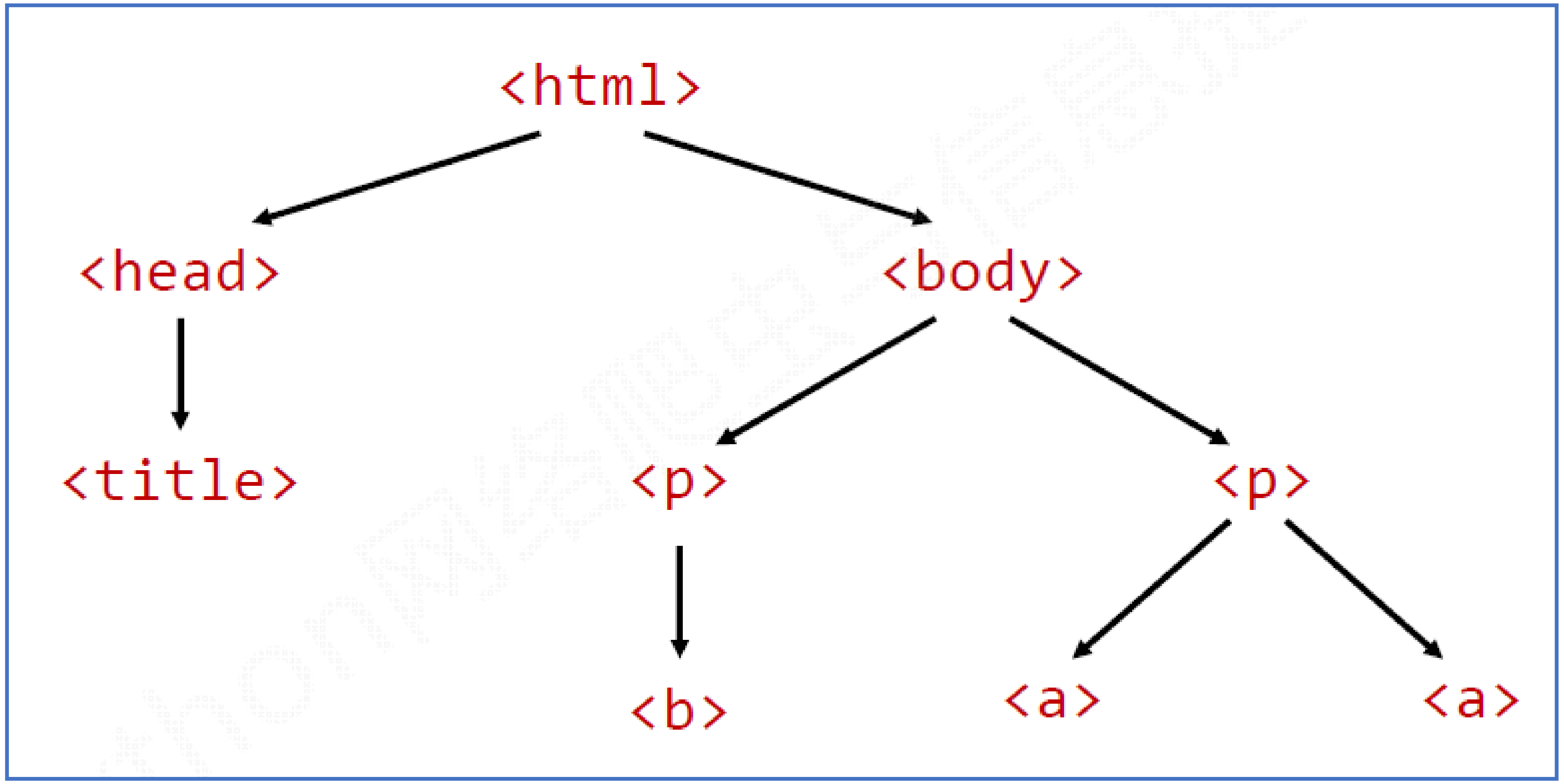
```
<html>
    <body>
        <p class="title"> … </p>
    </body>
</html>
```



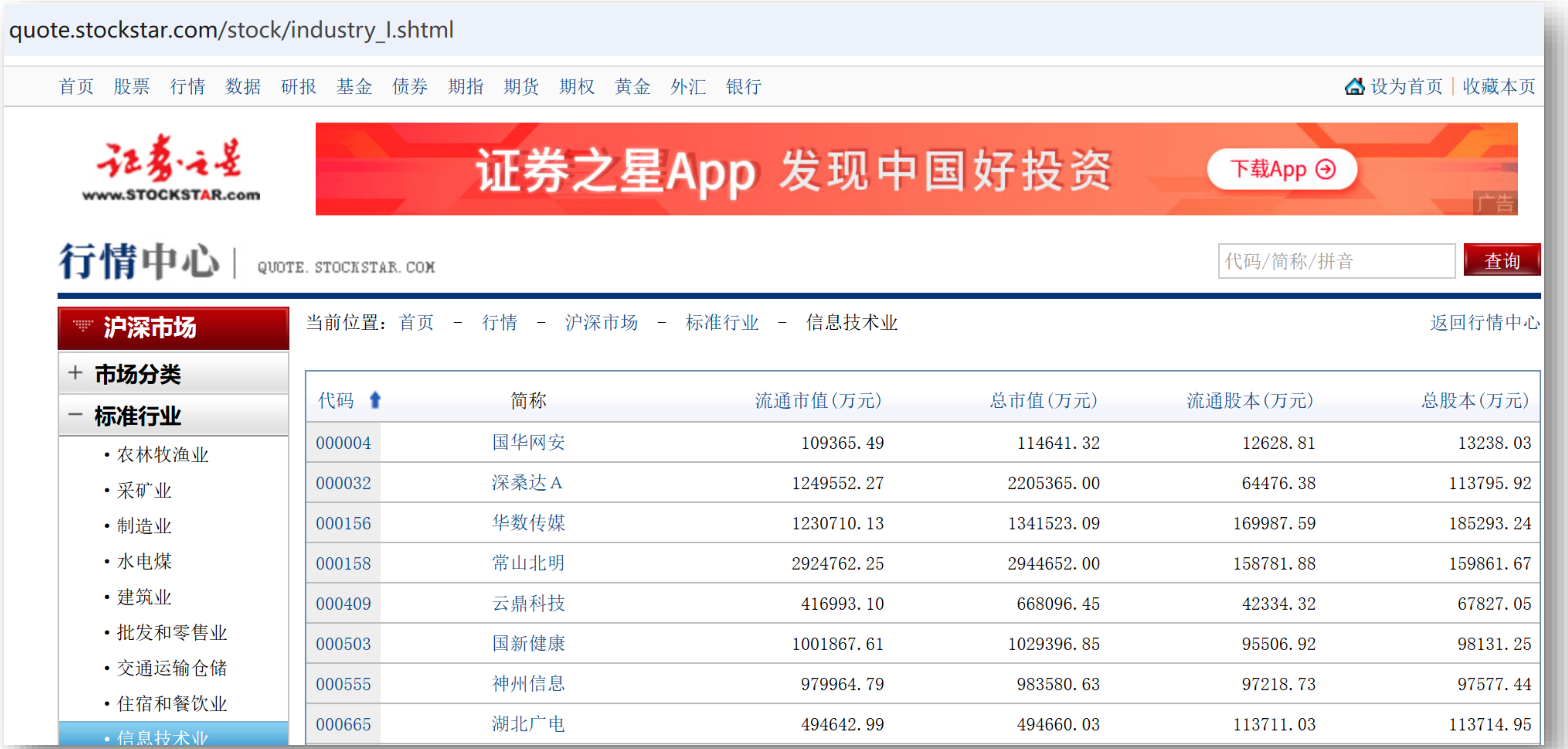
```
C:\Users\Gwenxx>pip install beautifulsoup4
```

>>> from bs4 import BeautifulSoup

标签树的遍历



爬取表格内容


quote.stockstar.com/stock/industry_I.shtml

首页 股票 行情 数据 研报 基金 债券 期指 期货 期权 黄金 外汇 银行 设为首页 | 收藏本页

证券之星App 发现中国好投资 下载App

行情中心 | QUOTE.STOCKSTAR.COM

代码/简称/拼音 查询

沪深市场

+ 市场分类

− 标准行业

- 农林牧渔业
- 采矿业
- 制造业
- 水电煤
- 建筑业
- 批发和零售业
- 交通运输仓储
- 住宿和餐饮业
- 信息技术业

当前位置：首页 − 行情 − 沪深市场 − 标准行业 − 信息技术业 返回行情中心

代码	简称	流通市值(万元)	总市值(万元)	流通股本(万元)	总股本(万元)
000004	国华网安	109365.49	114641.32	12628.81	13238.03
000032	深桑达Ａ	1249552.27	2205365.00	64476.38	113795.92
000156	华数传媒	1230710.13	1341523.09	169987.59	185293.24
000158	常山北明	2924762.25	2944652.00	158781.88	159861.67
000409	云鼎科技	416993.10	668096.45	42334.32	67827.05
000503	国新健康	1001867.61	1029396.85	95506.92	98131.25
000555	神州信息	979964.79	983580.63	97218.73	97577.44
000665	湖北广电	494642.99	494660.03	113711.03	113714.95

爬取表格内容

每月销售数据

月份	销售额
一月	$1000
二月	$1500

```
<table width=300 border='1'>
  <caption>每月销售数据</caption>
  <thead>
    <tr>
      <th>月份</th>
      <th>销售额</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>一月</th>
      <td>$1000</td>
    </tr>
    <tr>
      <td>二月</th>
      <td>$1500</td>
    </tr>
  </tbody>
 </table>
```

爬取表格内容

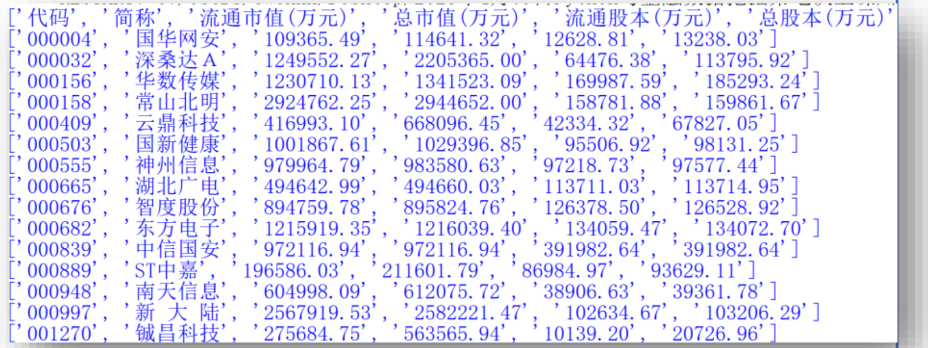
```
import requests
from bs4 import BeautifulSoup
url = "http://quote.stockstar.com/stock/industry_I.shtml"
r = requests.get(url)
r.encoding = 'gb2312'
soup = BeautifulSoup(r.text, 'html.parser')
table = soup.find('table', class_='trHover')
```

将获取到的HTML内容（r.text）通过'html.parser'解析器转换为BeautifulSoup对象

用于查找指定的HTML/XML标签

爬取表格内容

```
rows = table.find_all('tr')
for row in rows:
    cols = row.find_all(['th', 'td'])
    cols = [ele.text.strip() for ele in cols]
    if len(cols)>=6:
        print(cols)
```



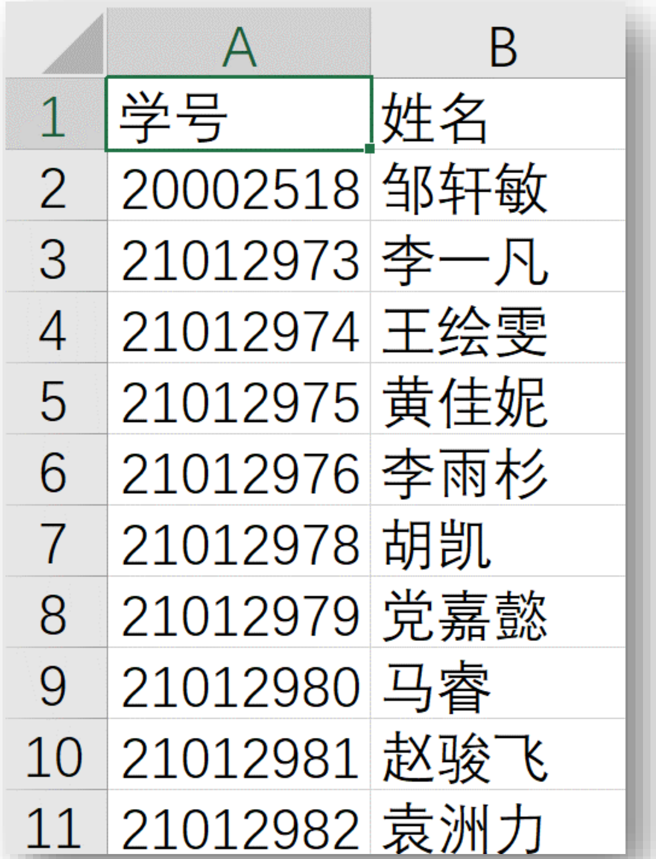
```
['代码', '简称', '流通市值(万元)', '总市值(万元)', '流通股本(万元)', '总股本(万元)'
['000004', '国华网安', '109365.49', '114641.32', '12628.81', '13238.03']
['000032', '深桑达A', '1249552.27', '2205365.00', '64476.38', '113795.92']
['000156', '华数传媒', '1230710.13', '1341523.09', '169987.59', '185293.24']
['000158', '常山北明', '2924762.25', '2944652.00', '158781.88', '159861.67']
['000409', '云鼎科技', '416993.10', '668096.45', '42334.32', '67827.05']
['000503', '国新健康', '1001867.61', '1029396.85', '95506.92', '98131.25']
['000555', '神州信息', '979964.79', '983580.63', '97218.73', '97577.44']
['000665', '湖北广电', '494642.99', '494660.03', '113711.03', '113714.95']
['000676', '智度股份', '894759.78', '895824.76', '126378.50', '126528.92']
['000682', '东方电子', '1215919.35', '1216039.40', '134059.47', '134072.70']
['000839', '中信国安', '972116.94', '972116.94', '391982.64', '391982.64']
['000889', 'ST中嘉', '196586.03', '211601.79', '86984.97', '93629.11']
['000948', '南天信息', '604998.09', '612075.72', '38906.63', '39361.78']
['000997', '新 大 陆', '2567919.53', '2582221.47', '102634.67', '103206.29']
['001270', '铖昌科技', '275684.75', '563565.94', '10139.20', '20726.96']
```

问题： 如何将爬取的数据存入数据库?

大数据定义

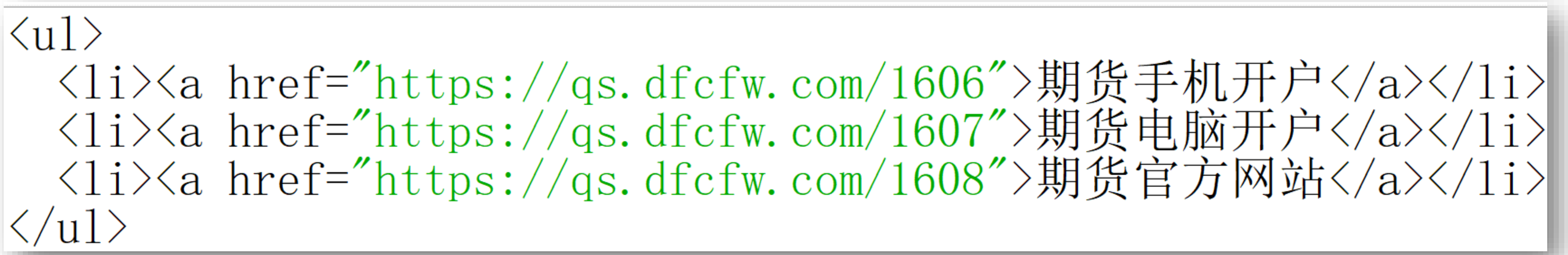
大数据（big data）：一种数据规模大到在获取、存储、管理、分析方面大大超出了传统数据库软件工具能力范围的数据集合。它包括结构化、半结构化和非结构化数据，非结构化数据越来越成为数据的主要部分。

结构化数据



	A	B
1	学号	姓名
2	20002518	邹轩敏
3	21012973	李一凡
4	21012974	王绘雯
5	21012975	黄佳妮
6	21012976	李雨杉
7	21012978	胡凯
8	21012979	党嘉懿
9	21012980	马睿
10	21012981	赵骏飞
11	21012982	袁洲力

半结构化数据



```
<ul>
  <li><a href="https://qs.dfcfw.com/1606">期货手机开户</a></li>
  <li><a href="https://qs.dfcfw.com/1607">期货电脑开户</a></li>
  <li><a href="https://qs.dfcfw.com/1608">期货官方网站</a></li>
</ul>
```

非结构化数据

大数据的特点

Volume(大量)： 存储单位至TB、PB、EB级别

Velocity(高速)： 处理速度**快**、时效性要求**高**

Variety(多样)： **结构化**、**半结构化**及**非结构化**

Value(价值)： 数据价值密度**低**、需要**算法**挖掘

Python支持的数据库

- SQLite
- MySQL
- MongoDB
- Redis
- Microsoft SQL Server 2000
- ….

常用数据库

SQLite： 是一个开源的关系型数据库，具有零配置、自我包含、便于传输等优点。它将整个数据库的表、索引、数据都存储在一个**单一的.db文件**中，不需要网络配置和管理，没有帐户和密码，数据库访问依赖于文件所在的操作系统。

SQLite数据库连接

- 和数据库建立连接
- 执行sql语句，接收返回值
- 关闭数据库连接

常用SQL语句

- 创建一个新的数据表

```
import sqlite3
conn=sqlite3.connect("trade.db")
SQL='''create table stock (code char(8) not null,
          name char(10),price float, primary key("code"))'''
conn.execute(SQL)
conn.commit()
conn.close()
```

常用SQL语句

◆ 往一个表中插入数据

```
import sqlite3
conn=sqlite3.connect("trade.db")
SQL='''insert into stock (code, name, price)
                            values('2349', '精华制药' , 10.49)'''
conn.execute(SQL)
conn.commit()
conn.close()
```

常用SQL语句

◆ 更新数据表中的数据

```
import sqlite3
conn=sqlite3.connect("trade.db")
SQL='''update stock set price=11.5
                                        where code='2349' '''
conn.execute(SQL)
conn.commit()
conn.close()
```

常用SQL语句

◆ 从一个表中删除数据

```
import sqlite3
conn=sqlite3.connect("trade.db")
SQL='''delete from stock where code='2349' '''
conn.execute(SQL)
conn.commit()
conn.close()
```

常用SQL语句

◆ 删除表

```
import sqlite3
conn=sqlite3.connect("trade.db")
SQL='''drop table if exists stock'''
conn.execute(SQL)
conn.commit()
conn.close()
```

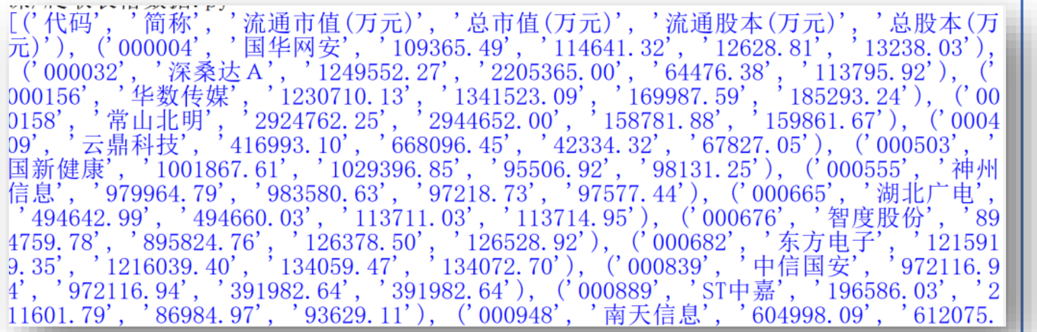
爬取表格数据存入数据库中

爬取表格数据并存入表中(1)

```
import requests
from bs4 import BeautifulSoup
import sqlite3
url = "http://quote.stockstar.com/stock/industry_I.shtml"
r = requests.get(url)
r.encoding = 'gb2312'
soup = BeautifulSoup(r.text, 'html.parser')
table = soup.find('table', class_='trHover')
rows = table.find_all('tr')
```

爬取表格数据并存入表中(2)

```
result=[]
for row in rows:
    cols = row.find_all(['th', 'td'])
    cols = [ele.text.strip() for ele in cols]
    if len(cols)>=6:
        info=tuple(cols)
        result. append(info)
del result[0]
```



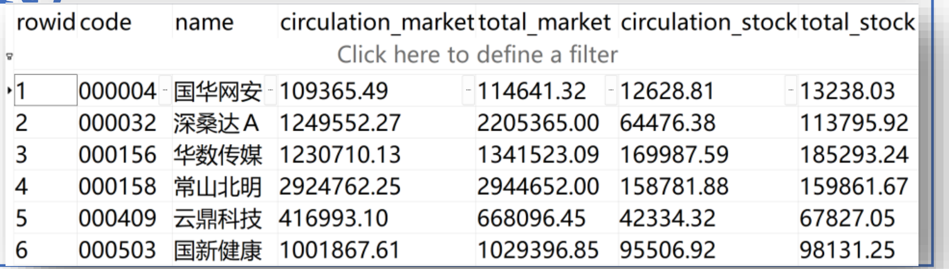
```
[('代码', '简称', '流通市值(万元)', '总市值(万元)', '流通股本(万元)', '总股本(万
元)'), ('000004', '国华网安', '109365.49', '114641.32', '12628.81', '13238.03'),
 ('000032', '深桑达Ａ', '1249552.27', '2205365.00', '64476.38', '113795.92'), ('
000156', '华数传媒', '1230710.13', '1341523.09', '169987.59', '185293.24'), ('00
0158', '常山北明', '2924762.25', '2944652.00', '158781.88', '159861.67'), ('0004
09', '云鼎科技', '416993.10', '668096.45', '42334.32', '67827.05'), ('000503', '
国新健康', '1001867.61', '1029396.85', '95506.92', '98131.25'), ('000555', '神州
信息', '979964.79', '983580.63', '97218.73', '97577.44'), ('000665', '湖北广电',
 '494642.99', '494660.03', '113711.03', '113714.95'), ('000676', '智度股份', '89
4759.78', '895824.76', '126378.50', '126528.92'), ('000682', '东方电子', '121591
9.35', '1216039.40', '134059.47', '134072.70'), ('000839', '中信国安', '972116.9
4', '972116.94', '391982.64', '391982.64'), ('000889', 'ST中嘉', '196586.03', '2
11601.79', '86984.97', '93629.11'), ('000948', '南天信息', '604998.09', '612075.
```

爬取表格数据并存入表中(3)

```
conn=sqlite3.connect("information.db")
SQL= ''' drop table if exists stock'''
conn.execute(SQL)
conn.commit()
SQL='''create table stock (code char(10) not null,
        name char(20),circulation_market char(20),
        total_market char(20), circulation_stock char(20),
        total_stock char(20), primary key("code"))'''
conn.execute(SQL)
conn.commit()
```

爬取表格数据并存入表中(4)

```
SQL='''insert into stock (code, name, circulation_market,
total_market, circulation_stock,total_stock)
values(?,?,?,?,?,?)'''
conn. executemany(SQL, result)
conn.commit()
conn.close()
```



rowid	code	name	circulation_market	total_market	circulation_stock	total_stock
			Click here to define a filter			
1	000004	国华网安	109365.49	114641.32	12628.81	13238.03
2	000032	深桑达A	1249552.27	2205365.00	64476.38	113795.92
3	000156	华数传媒	1230710.13	1341523.09	169987.59	185293.24
4	000158	常山北明	2924762.25	2944652.00	158781.88	159861.67
5	000409	云鼎科技	416993.10	668096.45	42334.32	67827.05
6	000503	国新健康	1001867.61	1029396.85	95506.92	98131.25

交易表数据存入数据库中

链接数据库并创建表

```
import sqlite3
conn=sqlite3.connect("trade.db")
SQL= ''' drop table if exists stock'''
conn.execute(SQL)
conn.commit()
SQL='''create table stock (code char(8) not null,
          name char(10),price float, primary key("code"))'''
conn.execute(SQL)
conn.commit()
```

从文件读取数据并存入数据库中

```
with open("C:\\trade.csv","r") as fobj:
    for i in fobj:
        if i[:4]=="code":
            continue
        i=i.strip();  info=i.split(",")
        SQL='''insert into stock (code, name, price)
           values('%s','%s',%f)''' %(info[0],info[1],float(info[2]))
        conn.execute(SQL)
        conn.commit()
conn.close()
```

东方财富网爬虫存入数据库中

爬取东方财富网链接和标题

```
import requests
import re
url="https://www.eastmoney.com/"
html=requests. get(url)
html. encoding=html. apparent_encoding
data=html. text
reg=r'<a href="(https://.*?)".*?>(.*?)</a>'
urls=re.findall(reg, data)
```

爬虫结果存入数据库

```
import sqlite3
conn=sqlite3.connect("web.db")
SQL='''drop table if exists information'''
conn.execute(SQL)
conn.commit()
SQL='''create table information(code integer not null,name char(30),
        link char(20), primary key("code"))'''
conn.execute(SQL)
conn.commit()
```

爬虫结果存入数据库

```
count=1
for item in urls:
    SQL='''insert into information(code,name,link)
                values(%d,'%s', '%s')''' %(count,item[1],item[0])
    conn.execute(SQL)
    conn.commit()
    count+=1
conn.close()
```

从数据库中查询部分记录

```
import sqlite3
conn=sqlite3.connect("web.db")
SQL='''select name,link from information where name like "东方%" '''
aList=list(conn.execute(SQL))
conn.commit()
for line in aList:
    print(line)
conn.close()
```

谢　谢