

**《Python数据分析与机器学习》**

**课程作业报告**

姓 名： 何欣桐

学 号： 0000000000

学 院： 商学院

专 业： 工商管理类

授课教师： 余 力

2022 年 2 月

**目录**

[1 数据准备与样例函数介绍 1](#_Toc95943154)

[1.1 crawl\_whole\_info\_table() 1](#_Toc95943155)

[2 A类题目 3](#_Toc95943156)

[2.1 题目1 3](#_Toc95943157)

[2.2 题目2 4](#_Toc95943158)

[2.3 题目3 4](#_Toc95943159)

[2.4 题目4 4](#_Toc95943160)

[2.5 题目5 4](#_Toc95943161)

[2.6 题目6 4](#_Toc95943162)

[2.7 题目7 4](#_Toc95943163)

[3 B类题目 4](#_Toc95943164)

[3.1 题目8 4](#_Toc95943165)

[3.2 题目10 4](#_Toc95943166)

[4 C类题目 5](#_Toc95943167)

[4.1 题目11 5](#_Toc95943168)

**摘要**

本项目基于《Python数据分析与机器学习》课程所学知识与网络相关教程资源，对数据集中的基金数据，进行各类数据分析，主要包括：使用requests、bs4、re、selenium爬取天天基金网上的基金总表和单个基金的相关信息（代码、名称、价格、类型等），使用numpy、pandas对得到的数据进行分析处理（筛选类型、提取需要的信息进行计算等），使用matplotlib、tkinter、wordcloud进行数据可视化（云图绘制、曲线绘制、直方图绘制等）。共爬取了x条基金，导出x条文件，x张图片。

**正文**

1. 数据准备与样例函数介绍

根据课程所给的 BigData\_course.py 样例文件，可适当修改所给样例函数，使其可以下载数据总简表和各个基金的具体数据。

* 1. crawl\_whole\_info\_table()

该函数从网上抓取基金信息总表。其步骤主要包括：设置session信息便于进入网站，随后对于访问的返回信息进行处理读取，将每一项基金存入 fund\_list 列表，创建 pandas 数据表 fund\_df，最后保存到文件 **data/wholeInfo.xlsx**中。

实际上，一个基金在网上的总表属性包含 25 项，样例函数**只取前 17 项**属性进行保存（代码、名称、英文、日期、单位净值、累积净值、日增长、近 1 周、近 1 月、近 3 月、近 6 月、近 1 年、近 2 年、近 3 年、今年来、成立来、成立时间）

此外，还需要注意的是，其中有一处遍历循环 ['hh', 'gp', 'zq', 'zs']: 地址中'ft=zq'代表爬取债券型基金，可根据情况更改为pg、gp、hh、zs、QDII、LOF（偏股型、股票型、混合型、指数型、QDII型、LOF型），所以对最后获得的总表**删除重复值**。

利用该函数保存总简表的源代码如下。

if "网上抓取基金信息总表 --------------------":

**def crawl\_whole\_info\_table():**

**if "1、网上抓取--------------------":**

fund\_list = []

try:

session = requests.session()

session.headers["Accept"] = "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8"

session.headers["Accept-Encoding"] = "gzip, deflate, br"

session.headers["Accept-Language"] = "zh-CN,zh;q=0.9"

session.headers["Connection"] = "keep-alive"

session.headers["Upgrade-Insecure-Requests"] = "1"

session.headers["User-Agent"] = \

"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36"

session.headers["Host"] = "fund.eastmoney.com"

session.headers["Referer"] = 'http://fund.eastmoney.com/data/fundranking.html'

for fund\_type in ['hh', 'gp', 'zq', 'zs']:

# 上面地址中'ft=zq'代表爬取债券型基金，可根据情况更改为pg、gp、hh、zs、QDII、LOF。（偏股型、股票型、混合型、指数型、QDII型、LOF型）

url = "http://fund.eastmoney.com/data/rankhandler.aspx?op=ph&ft=%s&rs=&gs=0&qdii=&tabSubtype=,,,,,&pn=10000" % (fund\_type)

response = session.get(url, verify=False, timeout=60)

response.encode = 'utf8'

response = response.text

response = response[response.find('var rankData = {datas:['): response.find(']')]

response = response.replace("var rankData = {datas:[", "")

for i in response.split('"'):

if len(i.split(',')) < 10:

continue

order = i.split(',')

for j in range(4, 16):

if "." in order[j]:

order[j] = float(order[j])

else:

order[j] = 0 # 代码位数补齐至六位

if len(order[0]) < 6:

order[0] = "0" \* (6 - len(order[0])) + order[0]

fund\_list.append(order)

except Exception as ex:

logging.exception(str(ex))

**if "2、创建pandas数据表-----------------":**

title = ["代码", "名称", "英文", "日期", "单位净值", "累积净值", "日增长", "近1周", "近1月", "近3月", "近6月", "近1年", "近2年", "近3年","今年来", "成立来", "成立时间", "未知", "成立来2", "折前手续费", "手续费", "折数", "手续费2", "折数2", "未知2"]

fund\_df = pd.DataFrame(fund\_list, columns=title)

# 只取到成立时间（含）以前的属性

fund\_df = fund\_df.iloc[:, :17]

# 按照代码升序调整

fund\_df = fund\_df.sort\_values(by='代码', axis=0, ascending=True).reset\_index(drop=True)

# 保留第一次出现的那条数据

fund\_df = fund\_df.drop\_duplicates(keep='first')

**if "3、保存到文件 ":**

file = u'data/wholeInfo.xlsx'

fund\_df.to\_excel(file, index=False, encoding='gbk')

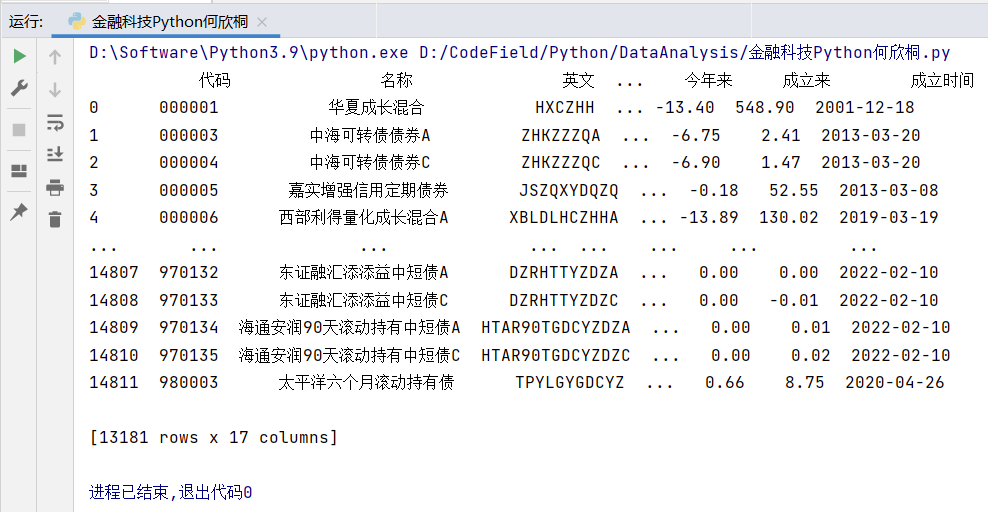
return fund\_df

if 1:

**df = crawl\_whole\_info\_table()**

print(df)

最后一次爬取总简表的时间为**2022年2月16日21时**，代码运行结果如下。



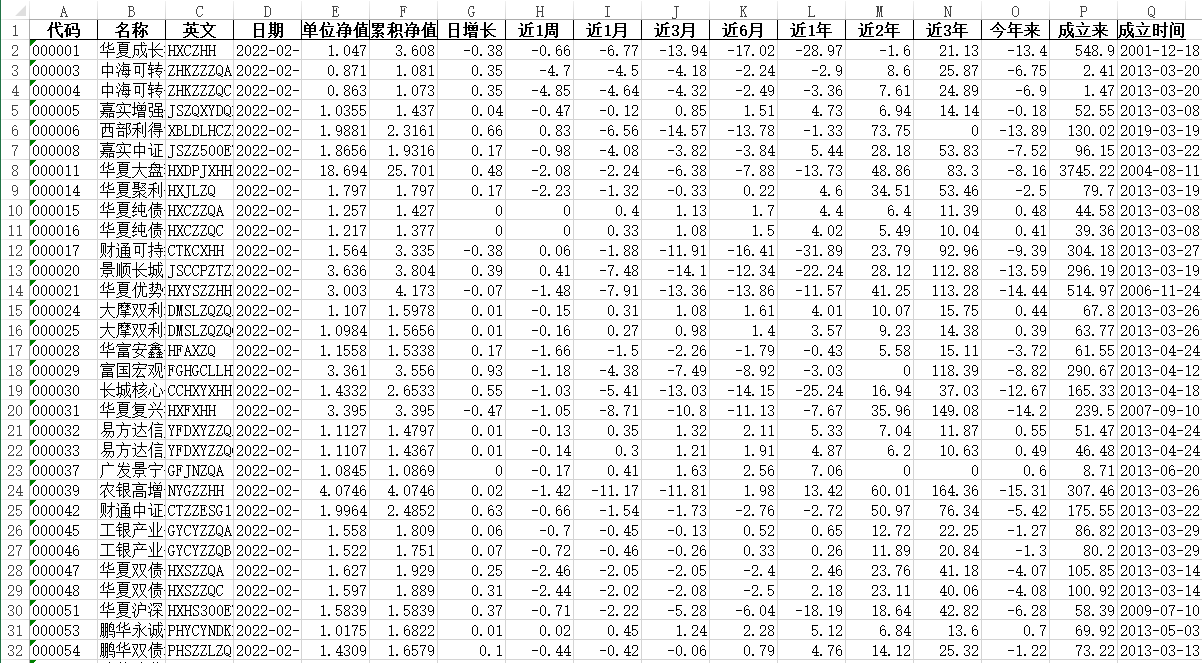


图 1 总简表的运行与导出结果

查看 excel 文件，共有 **13181** 条数据，可以对获得的数据进行验证：如果不考虑重复元素，直接导出文件，使用 excel 进行重复项删除操作，根据如下提示，可知在代码中进行的去重复操作是正确合理的。

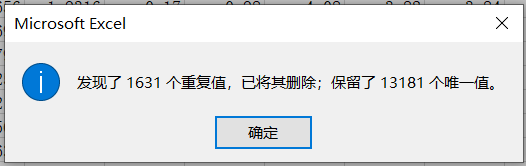


图 2 Excel 处理方法与结果验证

* 1. crawl\_one\_fund\_price(code, per, sdate, edate)

该函数根据基金代码、每页获取交易日数量、开始日期、结束日期爬取该基金设定时间内的所有交易日数据，并返回 dataframe 表格。其步骤主要包括：获取总页数和表头、抓取 records 存入 data 表格，并保存为 **data/{code}.xlsx** 表格文件。

在爬取的时候需要注意：**per 最大为 49**，也就是一页最多爬 49 个记录 如果设置为 50，只会显示 20 条，从而影响爬虫。

利用该函数保存总简表的源代码如下。

if "抓取一个基金的历史价格 --------------------------------":

def crawl\_one\_fund\_price(code, per=49, sdate='', edate=''):

**if "1、获取总页数和表头 ---------------":**

url = 'http://fund.eastmoney.com/f10/F10DataApi.aspx'

params = {'type': 'lsjz', 'code': code, 'page': 1, 'per': per, 'sdate': sdate, 'edate': edate}

rsp = requests.get(url, params)

rsp.raise\_for\_status()

html = rsp.text

# 使用正则表达式读取总页数 pages

pattern = re.compile(r'pages:(.\*),')

pages = int(re.search(pattern, html).group(1))

heads = []

soup = BeautifulSoup(html, 'html.parser')

for head in soup.findAll("th"):

heads.append(head.contents[0])

**if "2、开始抓取 -----------------":**

records = []

page = 1

while page <= pages:

params = {'type': 'lsjz', 'code': code, 'page': page, 'per': per, 'sdate': sdate, 'edate': edate}

rsp = requests.get(url, params=params)

rsp.raise\_for\_status()

html = rsp.text

soup = BeautifulSoup(html, 'html.parser')

for row in soup.findAll("tbody")[0].findAll("tr"):

row\_records = []

for record in row.findAll('td'):

val = record.contents

# 处理空值

if not val:

row\_records.append(np.nan)

else:

row\_records.append(val[0])

records.append(row\_records)

page = page + 1

if len(records) == 0:

records = [[np.nan, np.nan, np.nan, np.nan, np.nan, "---", "---", np.nan]]

**if "3、得到数据表 ---------------":**

np\_records = np.array(records)

data = pd.DataFrame()

for col, col\_name in enumerate(heads):

data[col\_name] = np\_records[:, col]

data['单位净值'] = data['单位净值'].astype(float)

data['累计净值'] = data['累计净值'].astype(float)

data['日增长率'] = data['日增长率'].str.strip('%').astype(float)

data = data.sort\_values(by='净值日期', axis=0, ascending=True).reset\_index(drop=True)

**if "4、保存到文件 ":**

file = u'data/{}.xlsx'.format(code)

data.to\_excel(file, index=False, encoding='gbk')

if 1:

file = u"data/wholeInfo.xlsx"

df = pd.read\_excel(file, usecols=['代码'], dtype={'代码': str})

total = len(df)

count = 0

start = datetime.datetime.now()

wrong\_list = []

# 读取各个基金信息，同时对每条信息计算所消耗的时间，进行格式化输出

for i in range(total):

count += 1

if not os.path.exists("data/" + df.iloc[i, 0] + '.xlsx'):

this\_start = datetime.datetime.now()

try:

**crawl\_one\_fund\_price(df.iloc[i, 0])**

this\_end = datetime.datetime.now()

print("Done [{}]. ({:5} / {:5}) items finished. ({} / {}) items used time.".

format(df.iloc[i, 0], count, total, this\_end - this\_start, this\_end - start))

# 防止因为出错而中断了程序

except requests.exceptions.Timeout or requests.exceptions.ConnectionError:

print("wrong [{}]. test again later.".format(df.iloc[i, 0]))

wrong\_list.append(df.iloc[i, 0])

sleep(10)

continue

else:

print("exists [{}]. ({:5} / {:5}) items finished.".format(df.iloc[i - 1, 0], count, total))

print(wrong\_list, "should crawl again.")

由于只使用一个程序爬虫的速度过慢，考虑使用 Pycharm 按照表格从前往后爬取，使用百度飞桨 BML CodeLab （一种全功能 AI 开发平台、机器学习集成开发环境，可运行 .ipynb 文件，excel文件可运行完后下载）从后往前爬取。

此外，在 BML CodeLab 上运行时，需要另行下载 BeautifulSoup4 库，添加一个 code 块，输入如下代码，进行项目所需库的持久化安装。

# 如果需要进行持久化安装，需要使用持久化路径，示例如下（加!读取为命令行语句）

!mkdir /home/aistudio/external-libraries

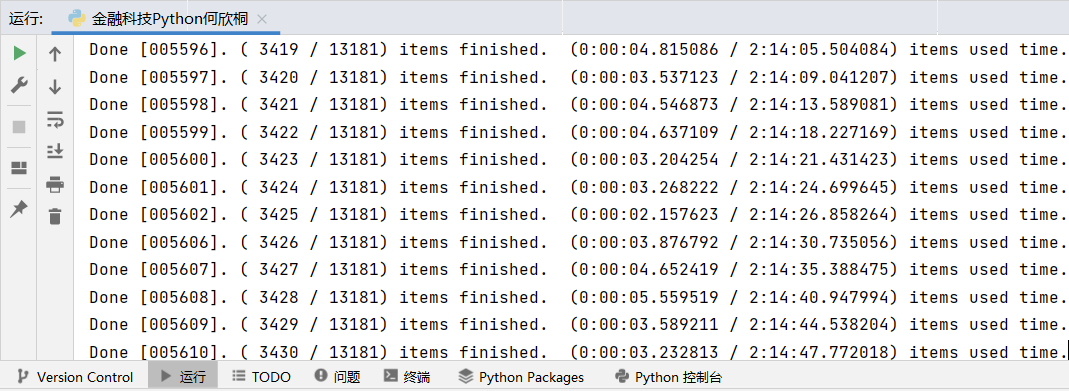
!pip install beautifulsoup4 -t /home/aistudio/external-libraries

# 然后在每次环境启动的时候，运行如下的代码即可

import sys

sys.path.append('/home/aistudio/external-libraries')

在检查到数据会合时，停止爬取，爬虫总计用时大约9小时，合并两部分的数据后，基金价格表格数量为**13181**个，与总简表项目数量一致，最后一次爬取信息的时间段为2022年2月16日22时——2022年2月1日15时，确保了绝大多数基金爬取到的最后结算日期是**2022年2月16日**，其运行截图和 data 文件夹保存情况如下。



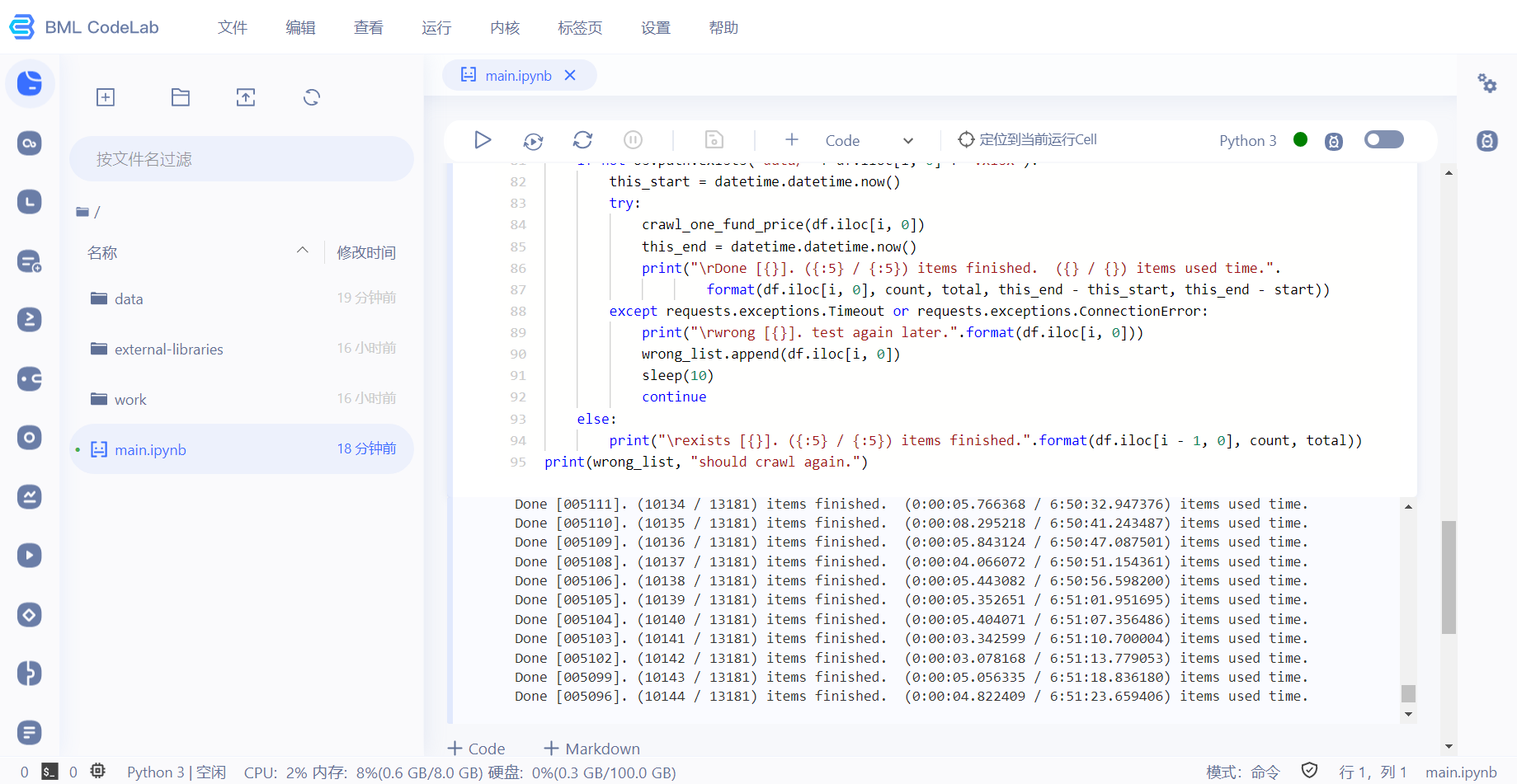
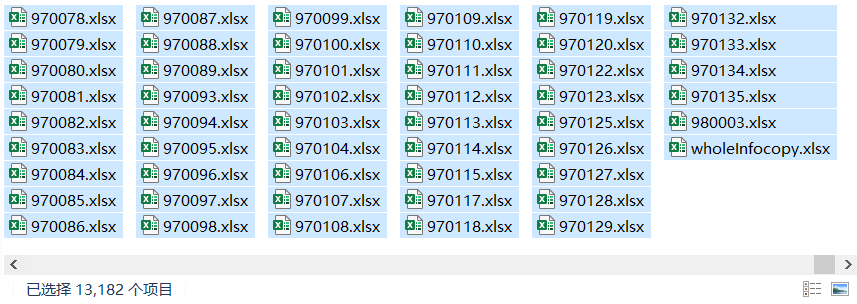


图 3 爬取各个基金交易日数据的程序截图（Pycharm 与 BML CodeLab）



121311

1. A类题目
   1. 题目1

编写函数，输出基金名字中包含有“混合”两字的基金名称及其代码（从总简表中）。

* 1. 题目2

编写函数，输出所有基金名称的云图（从总简表中）。

* 1. 题目3

编写函数，输出跌天数比率最大的 10 个基金的代码（从价格文件中）。

* 1. 题目4

编写函数，可以画出多个基金的价格变化曲线（从价格文件中）。

* 1. 题目5

编写函数，输出近 1 个月来跌幅最大的 10 个基金（从总简表中）。

* 1. 题目6

编写函数，可实现对于指定的一些基金 codes，输出近一个月来涨的天数比率最大的 10 个基金的代码。

* 1. 题目7

编写函数，输出近 1 周和近 1 个月跌幅都排名在前 20 的基金（从总简表中）。

1. B类题目

主要展示你作了哪些数据分析，比如下面的，下面只是提示

* 1. 题目8

编写函数 rising\_days\_distribution(code)，可以统计某一个基金 code 的连续涨跌天数，并以直方图的形式画出其连续涨跌天数分布直方图（从价格文件中）。

* 1. 题目10

编写函数 gain\_loss\_max(codes, money, from\_date, end\_date), 计算如果分别都投入资金数额为 money 的资金给代码为 codes 的多个基金（codes 为基金代码序列），投资时间范围是从 from\_date 到 end\_date，计算投资哪支基金的收益将会最大，输出该基金代码、名称，及相应的收益额（从价格文件中）.

1. C类题目
   1. 题目11

抓取主题基金：编写程序抓取各个主题或行业的基金代码。

vd

import urllib.request

import urllib.parse

url = 'https://www.baidu.com/s?wd='

headers = {

'User‐Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36'

}

url = url + urllib.parse.quote('小野')

request = urllib.request.Request(url=url,headers=headers)

response = urllib.request.urlopen(request)

print(response.read().decode('utf‐8'))