note ： 任何一门可以访问网络的语言都可以用来进行爬虫活动，但相对而言，python的语法比较优雅简洁，且包含了很多工具库，是最方便我们进行爬虫活动的。

1 安装python的编译环境。

我选择的是anaconda（一个开包即用的的工具），下载安装程序，点击安装，一直确定下去就ok了。

下载网址：https://www.anaconda.com/download/

2 打开cmd窗口，输入 jupyter notebook 回车 ，将会打开一个网页，然后就可以进行python文件的创建，编辑，编译。

3 爬虫过程：发送网络请求-->获取响应内容-->数据清洗-->保存数据

# A最简单的爬虫程序demo , 适用于静态的,没有防爬虫策略的网站

import requests # 发送网络请求，获取响应内容

from bs4 import BeautifulSoup # 对于网站响应的html进行解析 和数据的清洗

import pandas # 数据可视化，保存数据

def simpleDemo():

url = 'https://news.qudong.com/yejie/' # 要爬取的网页的网站

# 1 获取网页内容

req = requests.get(url) # 进行网络请求，获取响应的内容

req.encoding = 'utf8' # 设置解析编码,防止中文乱码

contentAll = req.text # 解析为文本

# 2 清洗数据，处理

soup = BeautifulSoup(contentAll,'html.parser') # 创建 beautifulsoup对象

contentList = soup.select('.text h4 ') # 根据css帅选数据,得到一个嵌套着beautifulsoup对象的列表

# 为了更好的进行数据的保存与展示，将数据保存在嵌套字典的列表里

all = [] #

for item in contentList: # 循环获取文本数据

all.append({'title':item.text})

# 3 保存数据

# 使用pandas库函数，将数据导入excel

all = pandas.DataFrame(all)

all.to\_excel('./temp/content.xlsx')

# print(all)

simpleDemo()

# B添加代理

import requests

from bs4 import BeautifulSoup

import pandas

def addProxy():

# 设置代理ip，防止本地ip被封

proxyHttps = {'https': 'https://139.129.207.72:808'}

# 设置ua,伪装成游览器

userAgent = 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.102 Safari/537.36'

head = {'User-Agent': userAgent,

'Connection': 'keep-alive'}

# 1 获取网页内容

# req = requests.get(url3, headers=head, proxies = proxy) # 免费的ip代理，速度比较慢，我就不用了

url = 'https://sh.lianjia.com/ershoufang/?utm\_source=baidu&utm\_medium=pinzhuan&utm\_term=biaoti&utm\_content\

=biaotimiaoshu&utm\_campaign=sousuo&ljref=pc\_sem\_baidu\_ppzq\_x'

req = requests.get(url, headers=head)

req.encoding = 'utf8' # 设置解析编码,防止中文乱码

contentAll = req.text # 解析为文本

# 2 清洗数据，处理

soup = BeautifulSoup(contentAll,'html.parser') # 创建 beautifulsoup对象

contentList = soup.select('.info.clear .title a') # 根据css帅选数据,得到一个嵌套着beautifulsoup对象的列表

all = []

for item in contentList:

all.append({'title':item.text})

# 3 保存数据

all = pandas.DataFrame(all)

all.to\_excel('./temp/content.xlsx')

# print(all)

addProxy()

# C终结篇

# 说明：

# 1. selenium是一款用来做自动化测试的工具，可以通过加载游览器的驱动控制游览器，在这里用来控制游览器访问网页爬取数据

# 2. User Agent 是根据 操作系统，CPU，游览器类型 形成的

# 模拟游览器爬取动态网页

# !pip install selenium

from bs4 import BeautifulSoup

from selenium import webdriver

from selenium.webdriver.chrome.options import Options

import time

import random

import pandas

def final():

driverPath = 'D:/Python/chromedriver.exe'

userAgent1 = 'Mozilla/5.0(compatible;MSIE9.0;WindowsNT6.1;Trident/5.0'

# OPPO A57 Android 手机百度

userAgent2 = 'Mozilla/5.0 (Linux; Android 6.0.1; OPPO A57 Build/MMB29M; wv) AppleWebKit/537.36(KHTML, like Gecko) Version/4.0 Chrome/63.0.3239.83 Mobile Safari/537.36 T7/10.13 baiduboxapp/10.13.0.10 (Baidu; P1 6.0.1)'

chromeOptions = Options()

# 设置游览器代理

chromeOptions.add\_argument('user-agent=' + userAgent1)

chromeOptions.add\_argument('disable-infobars')

# 设置不加载图片

prefs = {"profile.managed\_default\_content\_settings.images": 2}

chromeOptions.add\_experimental\_option("prefs", prefs)

# 不显示游览器

chromeOptions.add\_argument('--headless')

chromeOptions.add\_argument('--disable-gpu')

# 设置ip代理

ipProxy = 'http://110.52.235.99:9999'

chromeOptions.add\_argument("--proxy-server=" + ipProxy)

browser = webdriver.Chrome(executable\_path = driverPath, options = chromeOptions) # 获取游览器的驱动

browser.implicitly\_wait(10) # 隐式等待10 s，即最长等待元素加载时间为10s并且一旦发现元素加载成功则执行，全局有效

url2 = 'http://139.224.115.177:7777'

url3 = 'http://httpbin.org/ip'

url4 = 'https://www.toutiao.com/'

browser.get(url2)

return

# 循环下拉加载10次

height = browser.execute\_script('return screen.height')

for \_ in range(10):

sleepTime = random.uniform(1.1,3.0) # 生成浮点型随机数

time.sleep(sleepTime) # 强制等待，防止加载过快，触发服务端防爬虫机制

js='var q=document.documentElement.scrollTop=' + str(height)

browser.execute\_script(js)

height += 700

# content = browser.find\_elements\_by\_css\_selector('.link.title')

content = browser.find\_elements\_by\_xpath("//a[@class='link title']")

for item in content:

print(content.index(item)+1,'--',item.text.replace(' ','').replace('\n',''))

browser.close() # 关闭浏览器

browser.quit() # 关闭chreomedriver进程

final()

尾声：

有两种方式可以进行爬虫活动。

其一是通过网页源码爬取。

其二是通过API爬取。

两种方法都能满足数据的获取。

此篇教程的爬虫活动都是根据网页源代码爬取的。