第七关 收集评论：登录凭证

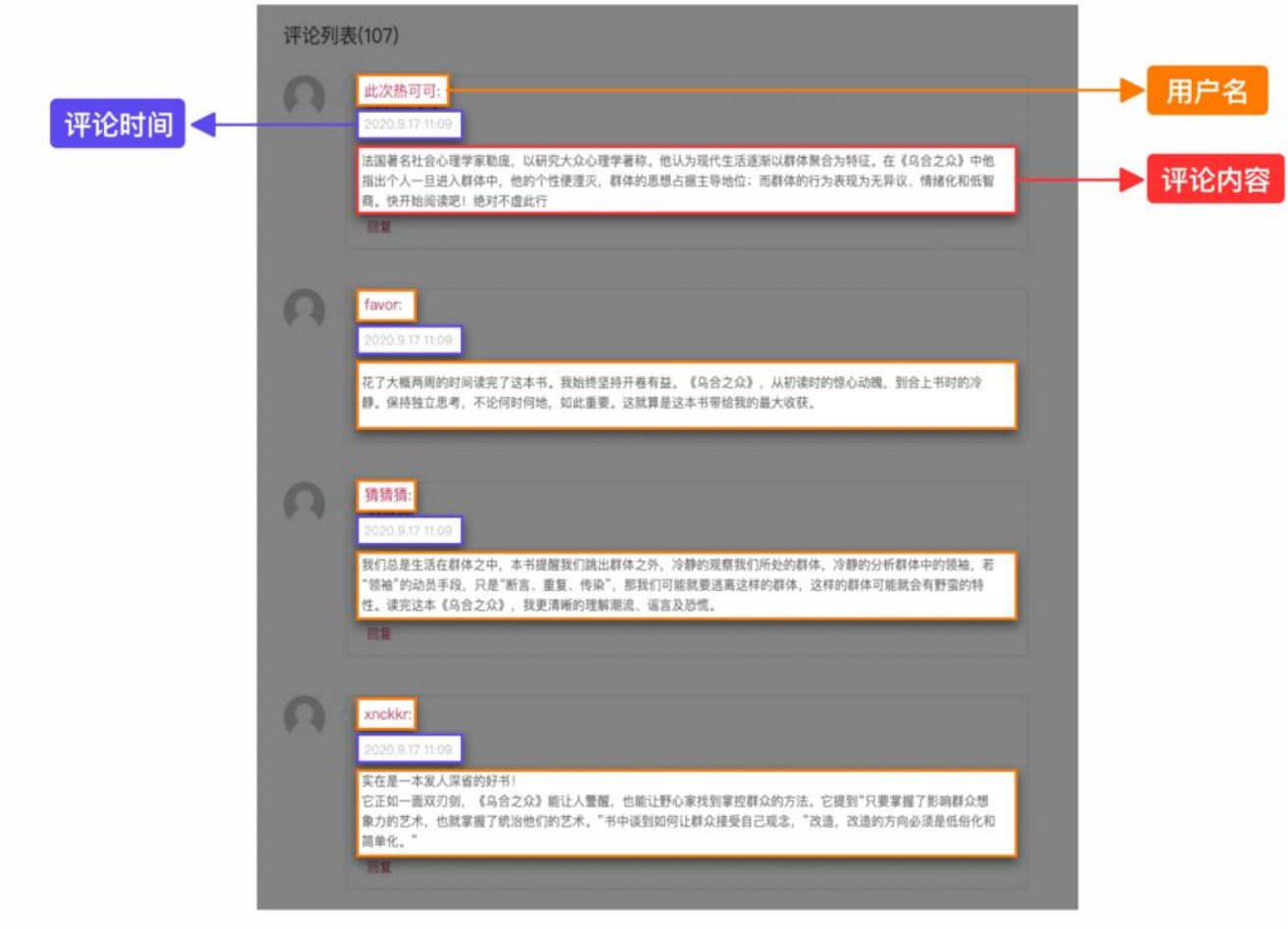
2022年12月12日 10:04

项目目标是爬取书籍《乌合之众》第一页评论页下的评论信息，网页：

https://wp.forchange.cn/psychology/11069/comment-page-1/ 。

要爬取的评论数据，被网站设置为仅登录可见，你需要在网站注册账号，注册成功并完成登录后，才能知道数据位于网页的哪些位置。

需要爬取
评论信息均包含了用户名、评论时间、评论内容。

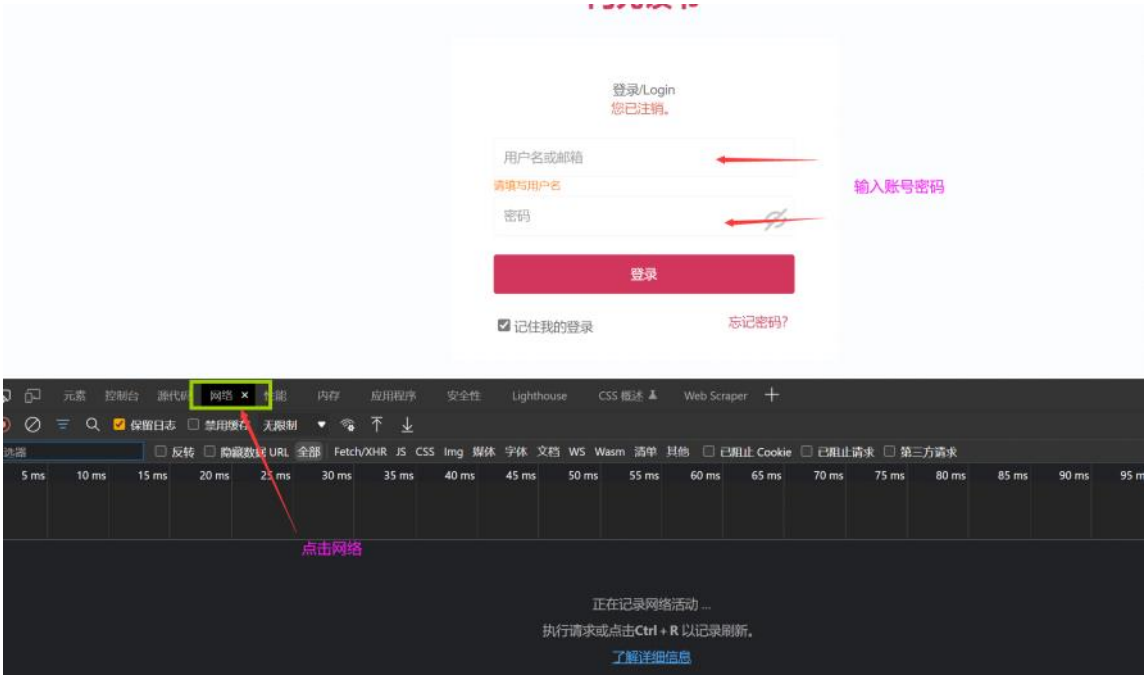


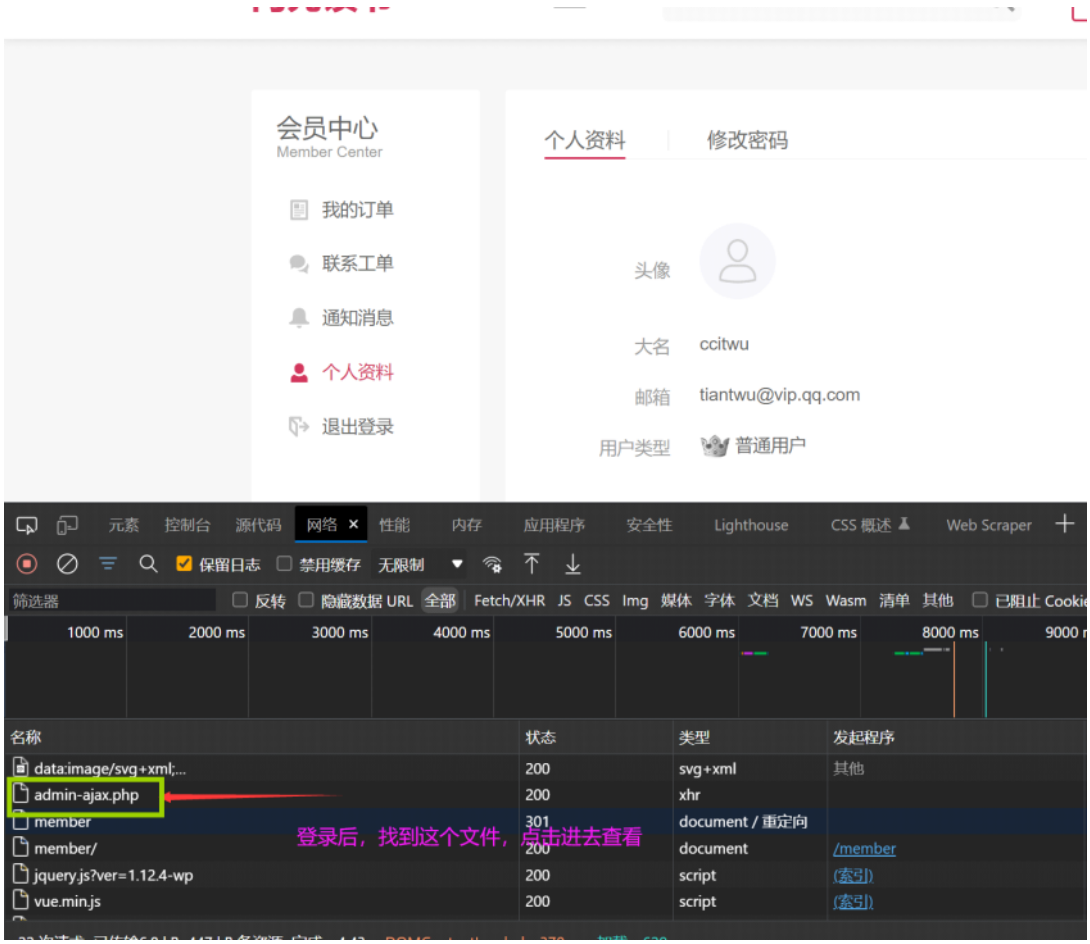
这里需要登录账号 才可以访问 评论信息

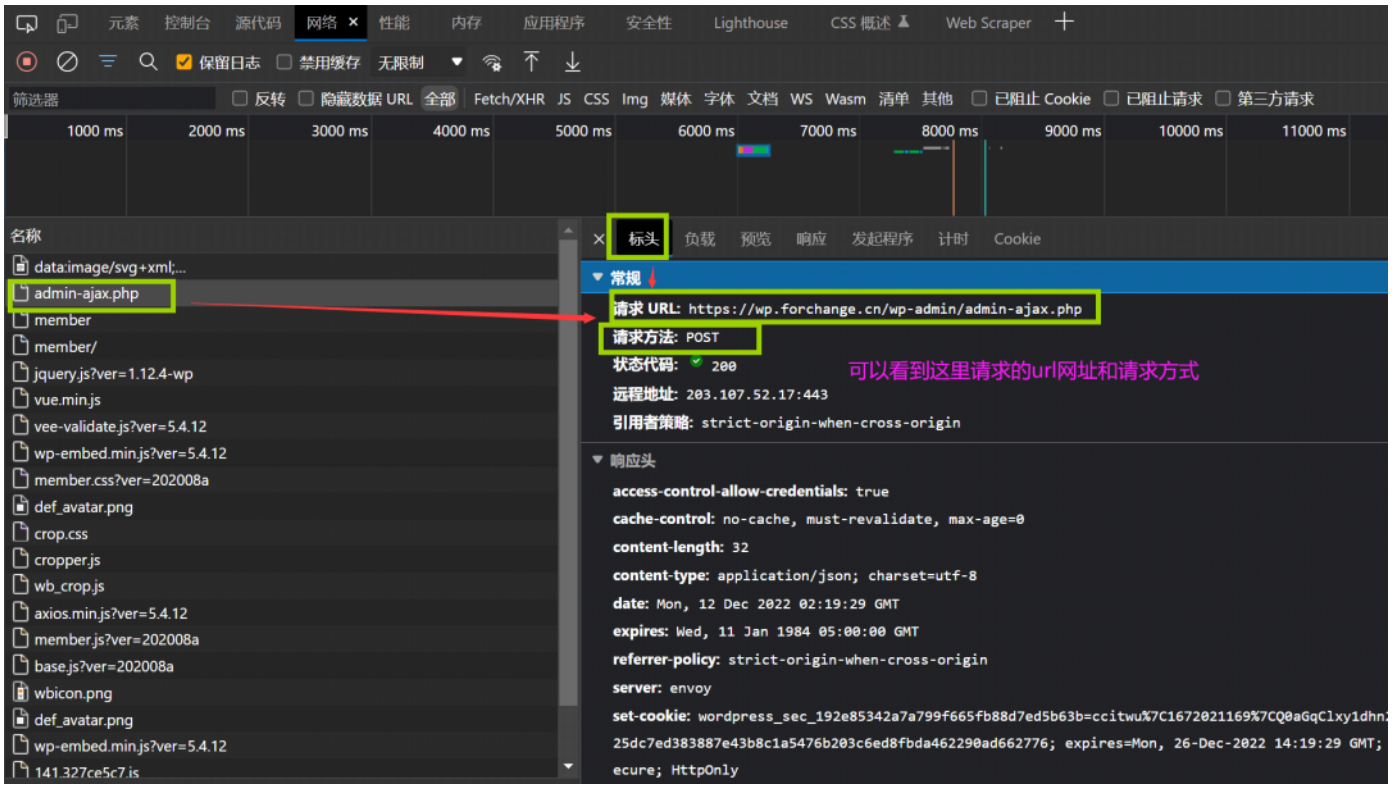
1.先登录闪光读书（forchange.cn）


闪光读书
登录/Login
您已注销。
用户名或邮箱
密码
登录
记住我的登录
忘记密码?
没有账户? 去注册

2 使用网页开发者模式


登录/Login
您已注销。
用户名或邮箱
请填写用户名
密码
登录
记住我的登录
忘记密码?
输入账号密码
元素
控制台
网络
内存
应用程序
安全性
Lighthouse
CSS 概述
Web Scraper
保留日志
禁用缓存
无限制
反转
全部
Fetch/XHR
JS
CSS
Img
媒体
字体
文档
WS
Wasm
清单
其他
已阻止 Cookie
已阻止请求
第三方请求
5 ms
10 ms
15 ms
20 ms
30 ms
35 ms
40 ms
45 ms
50 ms
55 ms
60 ms
65 ms
70 ms
75 ms
80 ms
85 ms
90 ms
点击网络
正在记录网络活动...
执行请求或点击Ctrl + R 以记录刷新。
了解详细信息

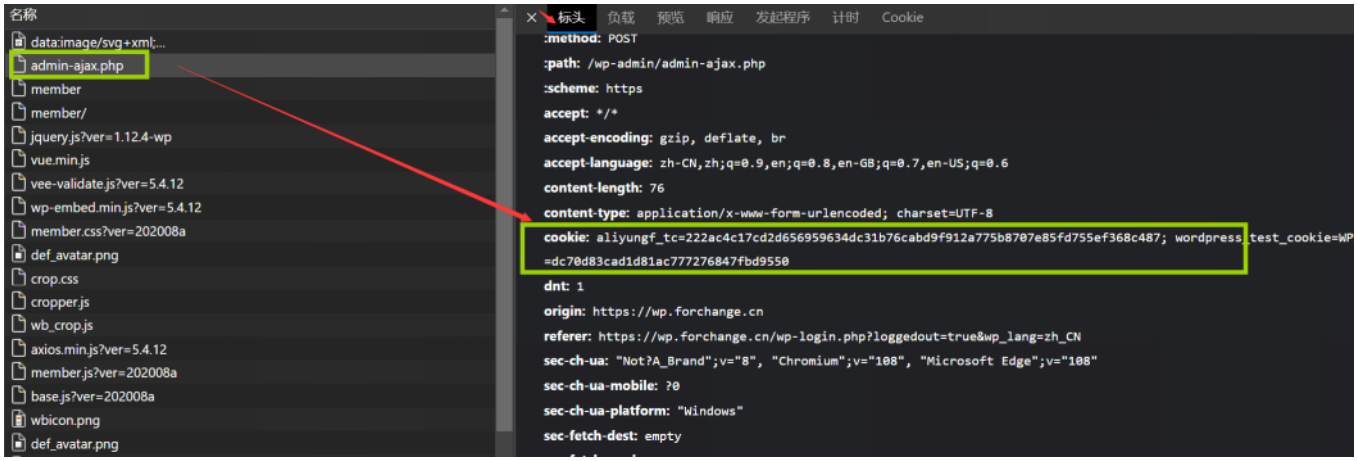




这里 请求方式注意是post 这是使用账号密码登录网站用的请求方式和get不同，在request的时候 注意使用post方法。

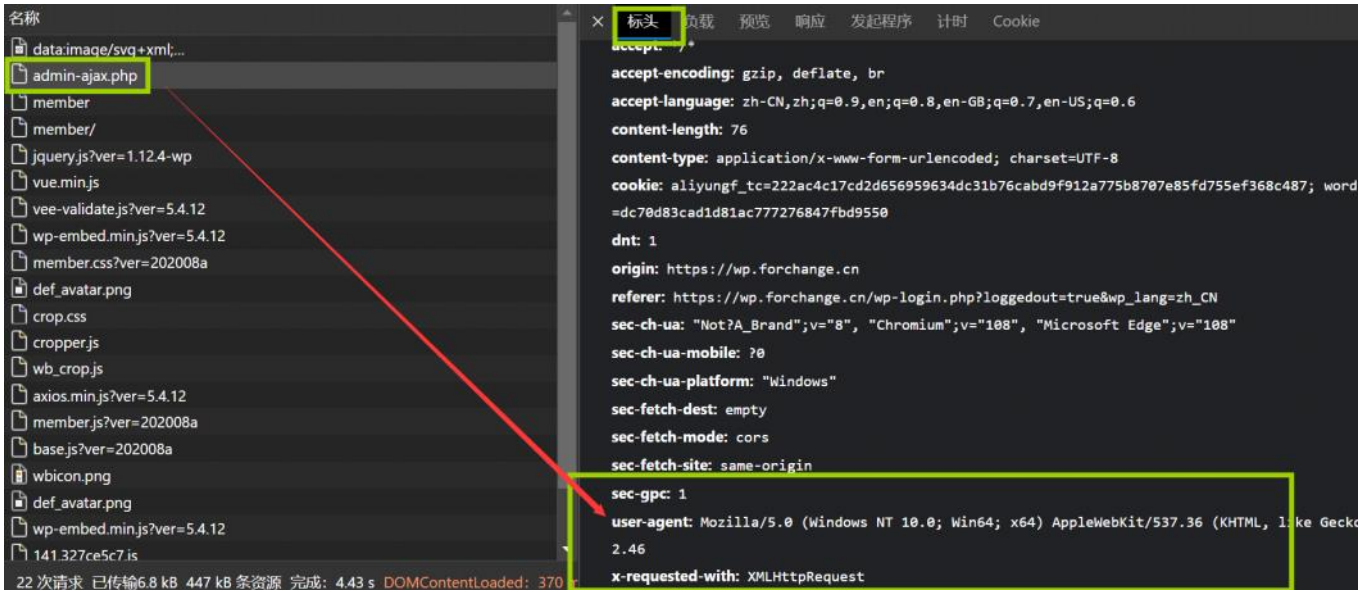
网络请求常用方法	
GET	请求从服务器获取数据
POST	请求向服务器提交数据
PUT ...	请求向服务器更新数据

一般来说，请求可以分为四部分内容：请求网址、请求方法、请求头和请求体。前三者在之前的课程也了解过它
们的作用，以及它们在请求详细信息中的位置。
而请求的最后一个部分：请求体，一般存放的是 POST 请求向服务器提交的数据。例如我们输入的账号密码，会
存储在该请求的请求体中。



这里往下拉 还可以看见cookie

最下面是 用户登录网站时候的 浏览器 操作系统版本信息

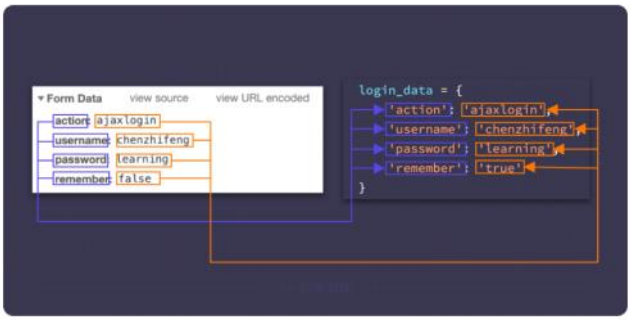


对于需要登录账号才能 访问的网站，在爬取登录的时候 需要注意以下问题

这里我们需要学习 Requests 库的另一个函数： post() 函数。来看看该函数的定义：

```
def post(url, data=None, json=None, **kwargs):
```

而参数 **data** 传入的是请求体，即我们上面在请求详情页的【Form Data】中看到的所有信息。这些信息在 Python 中可以以字典的形式来存储：



在模拟网站登录的操作，我们会调用 Requests 库的 post() 函数来提交账号密码。

① post() 函数

```
1  requests.post(url, data)
```

调用 post() 函数时会给函数的参数 **url** 和参数 **data** 传值。其中参数 **url** 传入登录请求的请求网址；参数 **data** 传入登录请求的请求体，数据多以字典形式写入。（请求的请求体，可以在请求详情页的 【Form Data】 位置找到）

post() 函数执行完，会返回一个 Response 对象，通过调用 Response 对象的 cookies 属性，可以获取到服务器返回的 Cookies 信息。

完成登录操作后，会调用 Requests 库的 get() 函数来请求书籍网页。

② get() 函数

```
requests.get(url, cookies)
```

调用 get() 函数时会给函数的参数 **url** 和参数 **cookies** 传值。其中参数 **url** 传入网页请求的请求网址；参数 **cookies** 传入服务器返回的 Cookies 信息。

下面解释代码

可以看到用这个方法 可以完成登录。

登录成功后，
项目目标是爬取书籍《乌合之众》第一页评论页下的评论信息，我们先打开该网页：

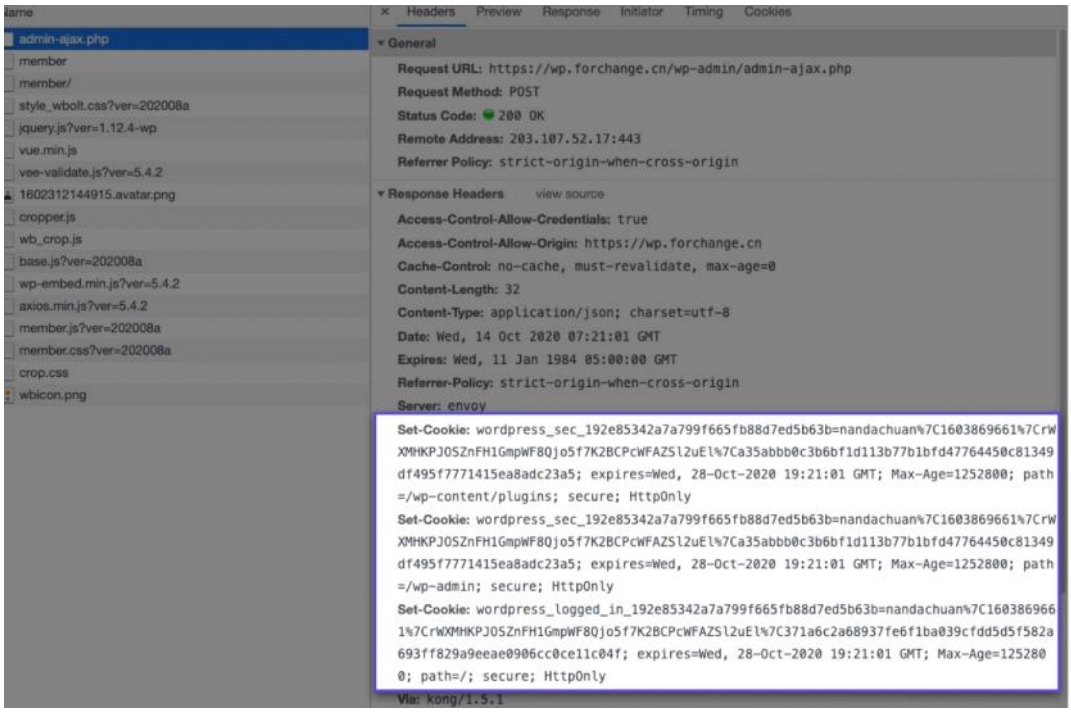
https://wp.forchange.cn/psychology/11069/comment-page-1/ 。

那我们该怎么告诉程序，请求书籍网页前，我们已经成功登录网站了呢？

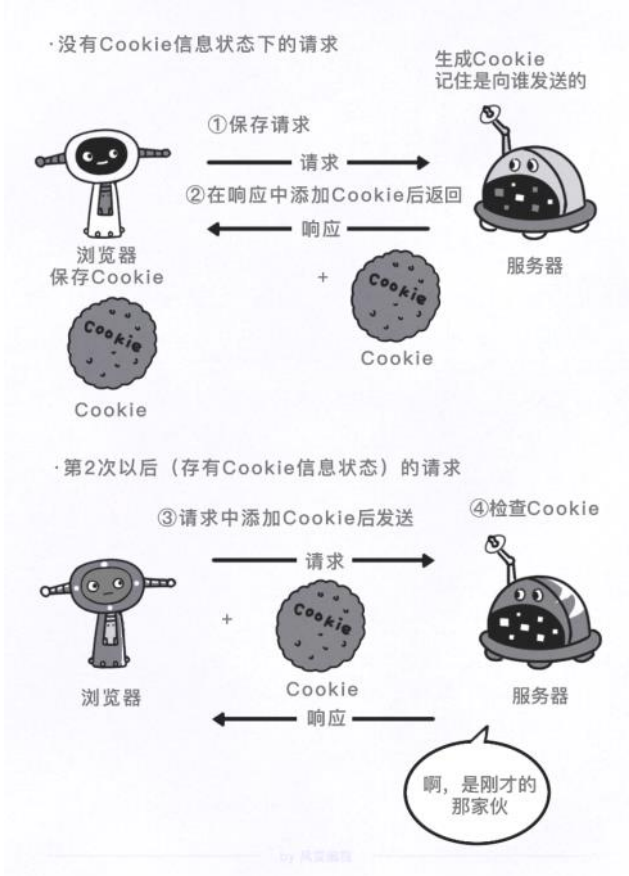
这就需要用到 get() 函数中的另一个参数 cookies，它可以设置请求中要携带的 Cookies 信息。

Cookies 是网站为了辨别用户身份，进行会话跟踪而存储在用户本地的数据，由用户客户端计算机暂时或永久保存的信息。

首次登陆网站成功后，服务器会将 Cookies 信息返回给浏览器，浏览器将 Cookies 信息保存下来。如下图，为服务器返回的 Cookies 信息。



Cookies 里含有登录相关的信息，下次浏览器请求该网站的网页时，浏览器会将 Cookies 发送给服务器，服务器通过识别 Cookies 来判断发送请求的用户是否已登录。



不过 Cookies 是有时效性的，可能一段时间后就会失效。例如在很多网站登录页面，会有类似“记住我的登录”的
选项。

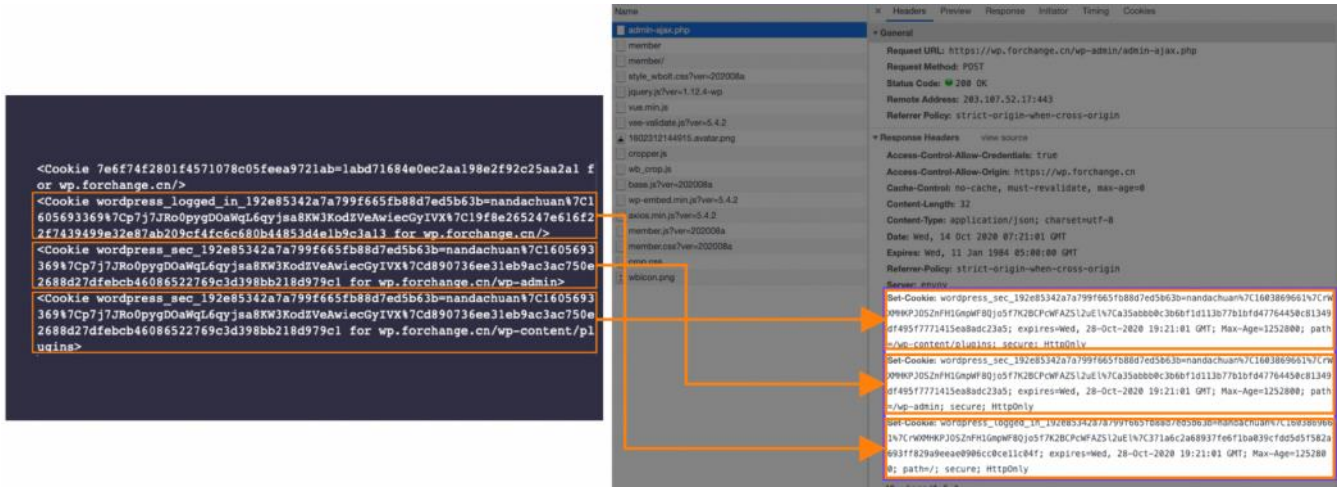
勾选了该选项可以让 Cookies 信息的有效期更长。但哪怕是勾选了该选项，一段时间过去后，该 Cookies 可能还
是因失效而无法使用，需要重新登录生成新的 Cookies。

怎么才能拿到服务器返回的 Cookies 信息呢?
我们只需要在“登录网站”的代码末尾，使用 post() 函数返回的 Response 对象，去调用它的 cookies 属性即
可。（调用 cookies 属性时，注意开头的 c 为小写）

```python
import requests

# 设置登录请求的请求网址
login_url = 'https://wp.forchange.cn/wp-admin/admin-ajax.php'
# 输入用户的账号密码
username = input('请输入用户名: ')
password = input('请输入密码: ')

# 设置登录请求的请求体数据
login_data = {
    'action': 'ajaxlogin',
    'username': username,
    'password': password,
    'remember': 'true'
}

# 请求登录网站
login_res = requests.post(login_url, data=login_data)

# 循环遍历获取到的 Cookies 信息
for cookie in login_res.cookies:  调用cookies
    # 打印 Cookies 信息
    print(cookie)
```

```
bash:root$ python /home/python-class/root/main56.py
请输入用户名：tiantwu@vip.qq.com
请输入密码：123456
<Cookie aliyungf_tc=41107c50925a627c05d54650aba88c069fa8fd71e216a77141eeb42c63382d02 for wp.forchange.cn/>
<Cookie wordpress_logged_in_192e85342a7a799f665fb88d7ed5b63b=ccitwu%7C1672034003%7COWz8fczab6T85toggGXm9pmqtZoBX7kqG4KDhKd9mBy%7C6162d86e7c7321fdecd29c648a183a99fc5b49c8f639a5642b73496ee9bb4100 for wp.forchange.cn/>
<Cookie wordpress_sec_192e85342a7a799f665fb88d7ed5b63b=ccitwu%7C1672034003%7COWz8fczab6T85toggGXm9pmqtZoBX7kqG4KDhKd9mBy%7Cb62ab3f60a3dda5e4fecff26de5a71a404ad13daab2c5df5ffd0b932bb001c01 for wp.forchange.cn/wp-admin>
<Cookie wordpress_sec_192e85342a7a799f665fb88d7ed5b63b=ccitwu%7C1672034003%7COWz8fczab6T85toggGXm9pmqtZoBX7kqG4KDhKd9mBy%7Cb62ab3f60a3dda5e4fecff26de5a71a404ad13daab2c5df5ffd0b932bb001c01 for wp.forchange.cn/wp-content/plugins>
```

将它的部分数据与请求详情页的信息做对比



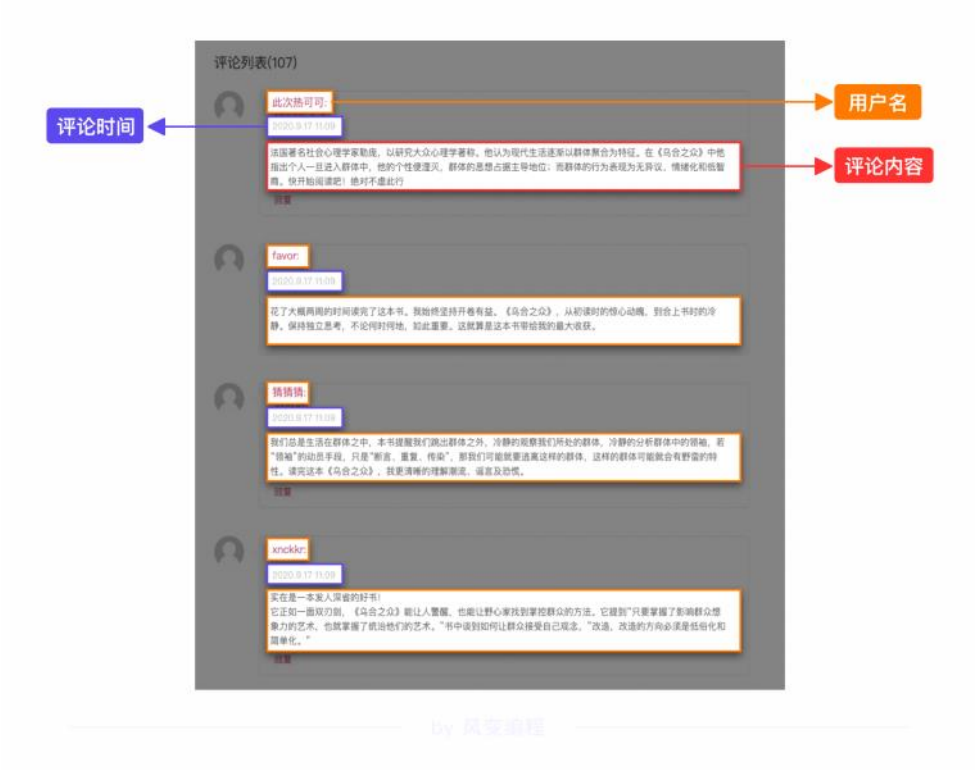
```python
import requests

# 设置登录请求的请求网址
login_url = 'https://wp.forchange.cn/wp-admin/admin-ajax.php'
# 输入用户的账号密码
username = input('请输入用户名: ')
password = input('请输入密码: ')

# 设置登录请求的请求体数据
login_data = {
    'action': 'ajaxlogin',
    'username': username,
    'password': password,
    'remember': 'true'
}

# 请求登录网站
login_res = requests.post(login_url, data=login_data)

# 设置要请求的书籍评论页链接
comment_url = 'https://wp.forchange.cn/psychology/11059/comment-page-1/'
# 携带获取到的 Cookies 信息请求书籍评论页
comment_res = requests.get(comment_url, cookies=login_res.cookies)
# 打印获取到的网页内容
print(comment_res.text)
```

调用登录网页的cookies信息

```
="nofollow"><svg class="wb-icon wbsico-weibo"><use xlink:href="
#wbsico-weibo"></use></svg></a><a class="share-logo icon-qzone"
 data-cmd="qzone" title="分享到QQ空间" rel="nofollow"><svg clas
s="wb-icon wbsico-qzone"><use xlink:href="#wbsico-qzone"></use>
</svg></a><a class="share-logo icon-qq" data-cmd="qq" title="分
享到QQ" rel="nofollow"><svg class="wb-icon wbsico-qq"><use xlin
k:href="#wbsico-qq"></use></svg></a>';
</script>
<script type='text/javascript' src='https://wp.forchange.cn/wp-
content/themes/elib-pro-patched20210727/js/base.js?ver=202008a'
 async></script>
<script type='text/javascript' src='https://wp.forchange.cn/wp-
includes/js/wp-embed.min.js?ver=5.4.12'></script>
<script type='text/javascript' src='https://wp.forchange.cn/wp-
content/themes/elib-pro-patched20210727/js/qrious/qrious.min.js
?ver=5.4.12'></script>
<script type='text/javascript' src='https://wp.forchange.cn/wp-
content/themes/elib-pro-patched20210727/js/single.js?ver=5.4.12
'></script>

<div class="tool-bar" id="J_toolBar">

    <a class="tb-item item-bt" id="J_backTop" href="javascript:
;" rel="nofollow" title="返回页顶">
            <svg class="wb-icon wbsico-backtop"><use xlink:href
="#wbsico-backtop"></use></svg>    </a>
</div>

</body>
</html>
```

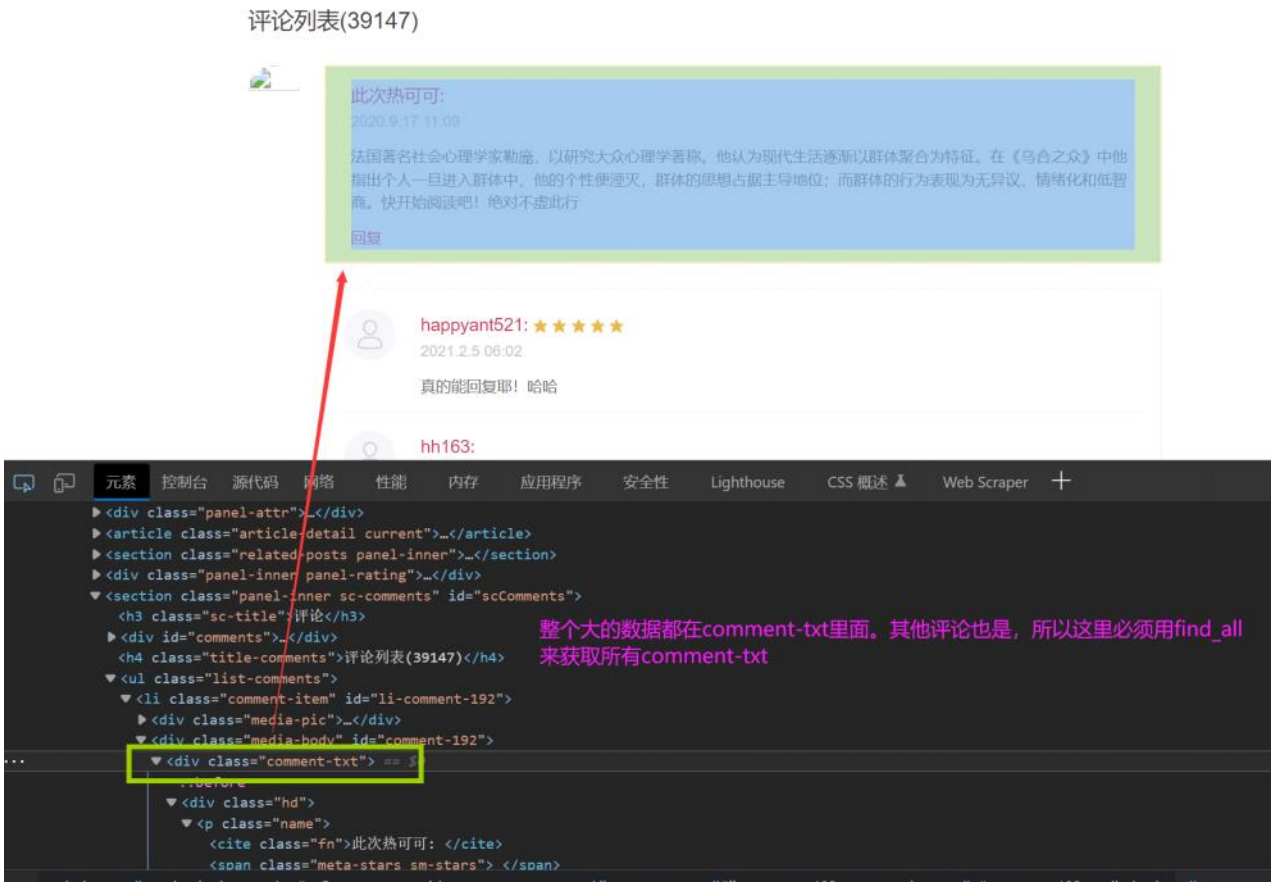
到下方评论区

一页评论页下包含 10 条评论信息，每一条评论信息均包含了用户名、评论时间、评论内容。



我们需要爬取的是 这一页 评论信息均包含了用户名、评论时间、评论内容

下面用开发工具，分析爬取页的信息



3.在解析爬取评论页时，如果解析网页的html代码出不来，则需要添加cookies信息

```python
# 登录后需要爬取评论信息的网址
comment_url = 'https://wp.forchange.cn/psychology/11069/comment-page-1/'
comment_res = requests.get(comment_url,headers = hearer,cookies = post_res.cookies)

if comment_res.status_code == 200:
    print("请求爬取网址成功")

# 解析网页
comment_soup = BeautifulSoup(comment_res.text,'html.parser')
```

添加cookies信息

最后详细代码：

```python
import requests
from bs4 import BeautifulSoup
import csv

# 登录网站的请求网址
post_url = 'https://wp.forchange.cn/wp-admin/admin-ajax.php'

hearer = {'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.42'}



username = input('输入用户名：')
password = input('输入密码：')

post_data ={
            'action': 'ajaxlogin',
            'username': username,
            'password': password,
            'remember':'true'
            }

comment_list = []
```

```python
# 登录网站
post_res = requests.post(post_url, headers=hearer,data=post_data)

if  post_res.status_code == 200:

    print("登录成功")
```

```python

# 登录后需要爬取评论信息的网址
comment_url = 'https://wp.forchange.cn/psychology/11069/comment-page-1/'
comment_res = requests.get(comment_url,headers = hearer,cookies = post_res.cookies)

if comment_res.status_code == 200:
    print("请求爬取网址成功")

# 解析网页
comment_soup = BeautifulSoup(comment_res.text,'html.parser')
```

```python
comment_data = comment_soup.find_all('div',class_='comment-txt')

for comment in comment_data:
    name = comment.find('cite',class_='fn').text
    date = comment.find('p',class_='date').text
    content = comment.find('div',class_='bd').find('p').text

    content_dict = {'评论人': name, '评论时间': date, '评论内容': content}
    comment_list.append(content_dict)
    print(comment_list)
```

```
# 写入csv
with open('D:\PythonTest\风变python学习资料\Python爬虫\评论信息.csv','w',newline = '',
encoding='utf-8-sig') as f:

    content_csv = csv.DictWriter(f,fieldnames=['评论人','评论时间','评论内容'])
    content_csv.writeheader()
    content_csv.writerows(comment_list)

print("评论信息写入csv成功")
```

```
import requests
from bs4 import BeautifulSoup
import csv
# 登录网站的请求网址
post_url = 'https://wp.forchange.cn/wp-admin/admin-ajax.php'
hearer = {'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
Edg/108.0.1462.42'}


username = input('输入用户名: ')
password = input('输入密码: ')
post_data ={
            'action': 'ajaxlogin',
            'username': username,
            'password': password,
            'remember':'true'
            }

comment_list = []
# 登录网站
post_res = requests.post(post_url, headers=hearer,data=post_data)
if  post_res.status_code == 200:
    print("登录成功")
# 登录后需要爬取评论信息的网址
comment_url = 'https://wp.forchange.cn/psychology/11069/comment-page-1/'
comment_res = requests.get(comment_url,headers = hearer,cookies =
post_res.cookies)
if comment_res.status_code == 200:
    print("请求爬取网址成功")
# 解析网页
comment_soup = BeautifulSoup(comment_res.text,'html.parser')
comment_data = comment_soup.find_all('div',class_='comment-txt')
for comment in comment_data:
    name = comment.find('cite',class_='fn').text
    date = comment.find('p',class_='date').text
    content = comment.find('div',class_='bd').find('p').text

    content_dict = {'评论人': name, '评论时间': date, '评论内容': content}
    comment_list.append(content_dict)
    print(comment_list)
# 写入csv
with open('D:\PythonTest\风变python学习资料\Python爬虫\评论信
息.csv','w',newline = '',encoding='utf-8-sig') as f:

    content_csv = csv.DictWriter(f,fieldnames=['评论人','评论时间','评论内
容'])
    content_csv.writeheader()
    content_csv.writerows(comment_list)
print("评论信息写入csv成功")
```

6.2 知识归纳与总结

本节课主要以案例练习为主，新增的知识点不多，只有以下几个：

POST 请求

1）POST 请求一般用来向服务器提交数据；
2）POST 请求与 GET 请求的区别在于，POST 请求更多用于提交私密信息到服务器，而 GET 请求一般用于获取数据。

请求的组成部分

1）请求一般有四个组成部分：请求网址、请求方法、请求头、请求体；
2）请求体的查看步骤
  a.打开开发者工具（快捷键 + 菜单选择）；
  b.点击 Network 选项卡；
  c.在网页上输入数据后，点击相应按钮，再在 Network 选项卡寻找目标请求链接；
  d.查看请求中的【Form data】部分。

Requests 库

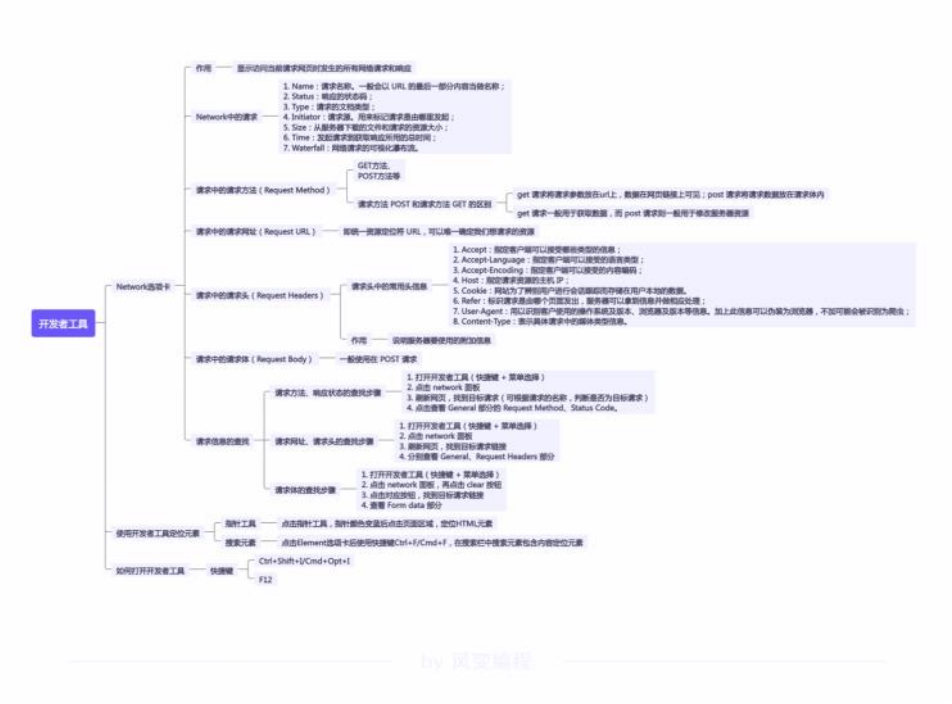
1）requests.post(url, data)：使用 POST 请求向服务器提交数据；
2）requests.get(url, cookies)：向服务器发起携带 Cookies 信息的 GET 请求；
3）Response.cookies：获取 Cookies 信息。

7.1 爬虫相关知识了解

首先，了解爬虫的第一步，什么是爬虫？
爬虫指的是能够自动化访问网站的程序。

网站上的信息一般是 HTML 代码，它是一种用来告知浏览器如何组织界面的标记语言，能够控制整个网页界面上的显示出来的内容。
我们大部分爬虫访问网站的目的也就是为了获取 HTML 中的信息。

当然，直接从一片密密麻麻的 HTML 代码中提取出我们想要信息是一件需要勇气的事。
所以，大部分网络浏览器里都包含着一套强大的开发者工具，这个工具可以帮助我们轻松定位到当前网站上加载的网页元素。

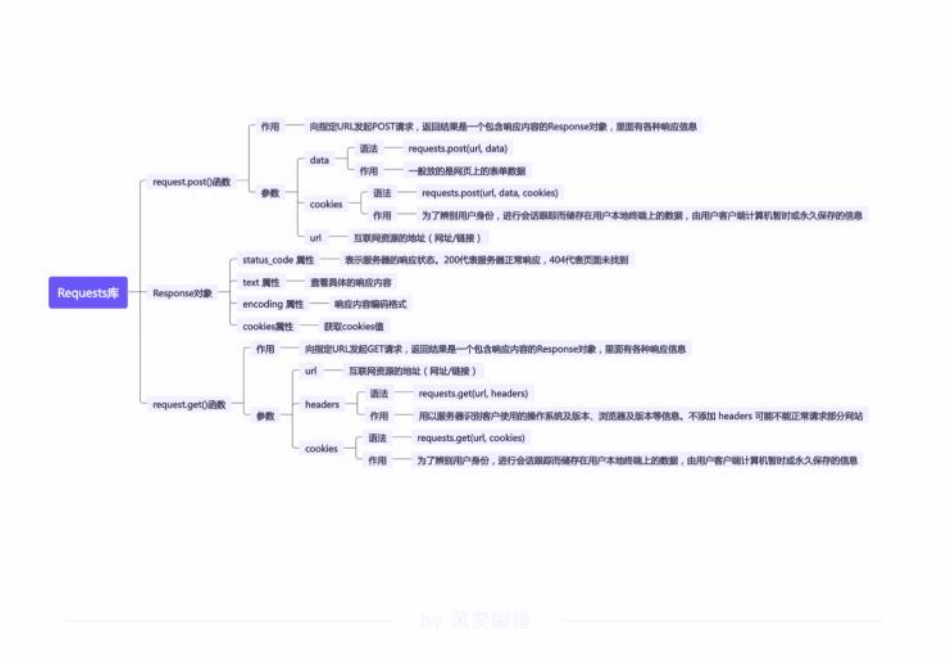


从网页元素中找到对应的请求网址后就正式进入我们的爬虫流程。

获取网页

有了请求网址后，我们先用 `requests` 库来对指定 `URL` 发起请求。

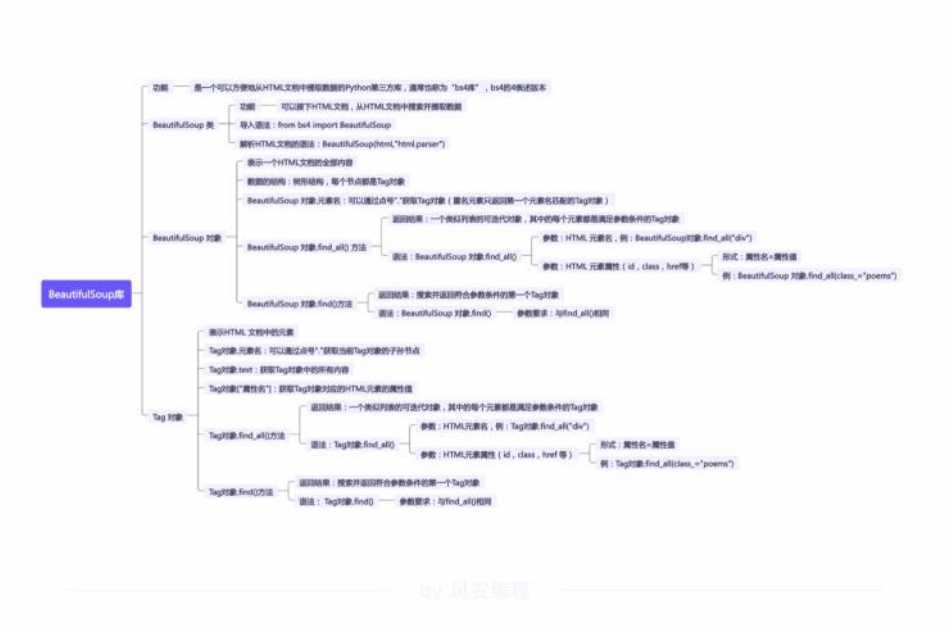
`requests` 发起请求后的返回结果是一个 `Response` 对象，这个对象中包含我们从网页中获取的信息，我们可以通过调用这个对象里的属性取出相应的信息（响应状态、具体的响应内容、Cookies值等）。



解析网页

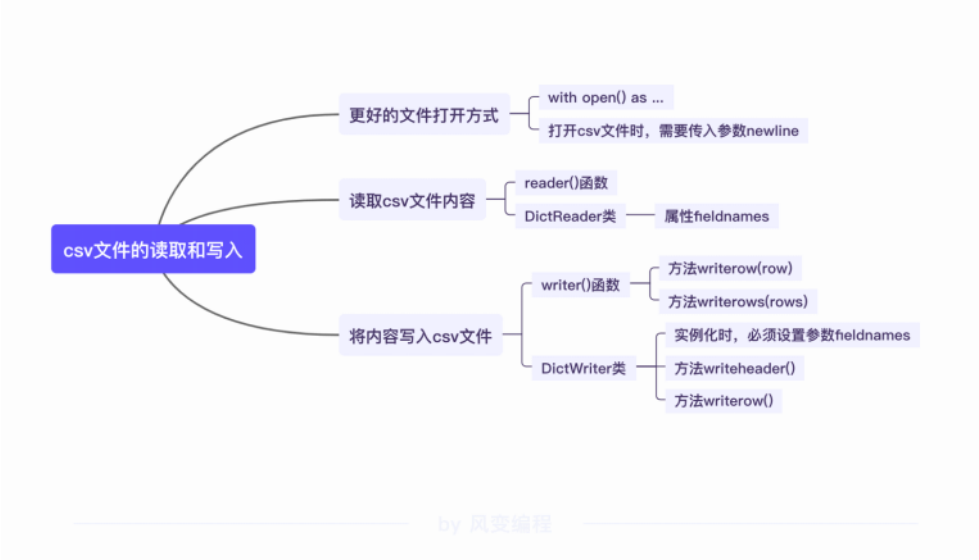
得到的响应内容如果是 HTML 格式的文档，我们可以用 BeautifulSoup 来对 HTML 文档里我们需要的数据进一步解析、提取。

BeautifulSoup库可以快速、方便、简单地提取网页中指定的内容。



存储数据

需要的数据都提取出来了，我们就可以用 csv 库将这些数据都存储起来。



总结完这个阶段的爬虫知识，也意味着你掌握了网络爬虫的基本思路和方法。接下来，你或许还会遇到新的爬虫难关，但请放心，我会继续陪着你学习下去。加油吧！