概述

爬虫可以简单理解为，使用程序来模拟手动获取网络信息的过程。本文主要介绍爬虫的思路，建议按照每一步的过程进行操作，加深印象。可结合当前目录下*逐行演示.ipynb* 进行阅读

第 0 步，打开 jupyter 以查看其中的爬虫程序：

在*./爬虫程序*所在文件夹下打开 cmd（命令提示符），进入已建好的 conda 环境（方式为:conda activate ***，***为环境名称），输入 jupyter notebook 打开 jupyter 编辑器（需在当前 conda 环境下已安装 jupyter，安装方式为，进入 conda 环境后，输入 pip install jupyter，等待安装完成即可）。

建议先阅读文件*逐行演示.ipynb*

```
D:\爬虫程序>conda activate pytorch_gpu

(pytorch_gpu) D:\爬虫程序>jupyter notebook
```

图 1 进入 conda 环境

以光明网为例

第 1 步，浏览网页与查看网页元素

在浏览器中打开如下网址 https://epaper.gmw.cn/gmrb/html/2021-02/02/nw.D110000gmrb_20210202_1-01.htm，并按键 F12，可显示如下画面

图 2 F12 查看网页元素

网页显示界面中的所有内容，都可以在 HTML 源码界面找到对应的内容。如何找到对应关系呢？以文字和图片为例，先在右侧 HTML 界面点击下图中所示图标。

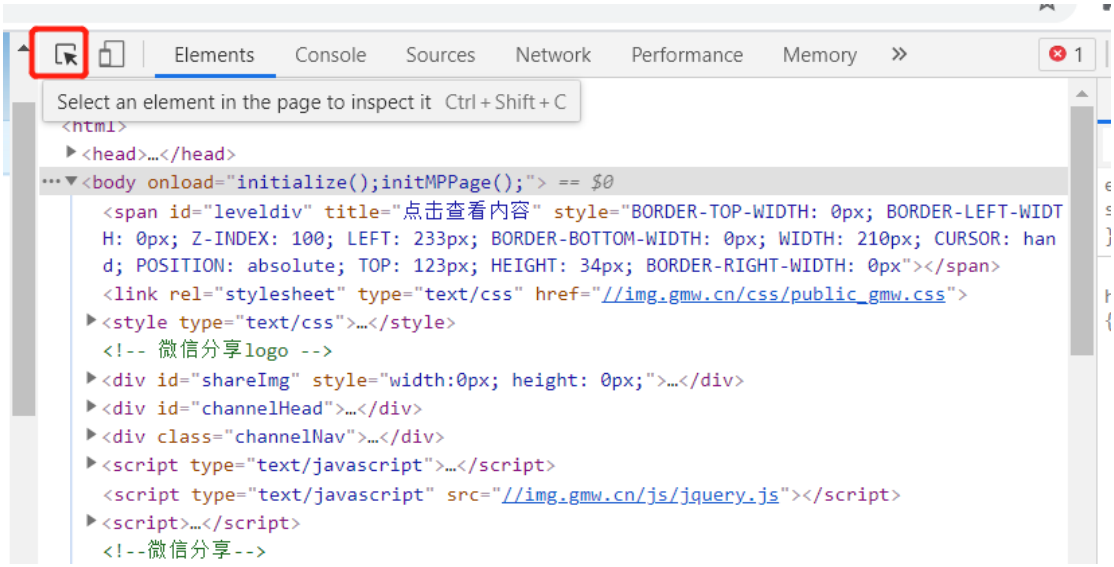


图 3 元素选取工具

然后可在网页显示界面点击任何可查看内容，即可在 HTML 源码界面显示对应的 HTML 代码部分

图 4 选取文字

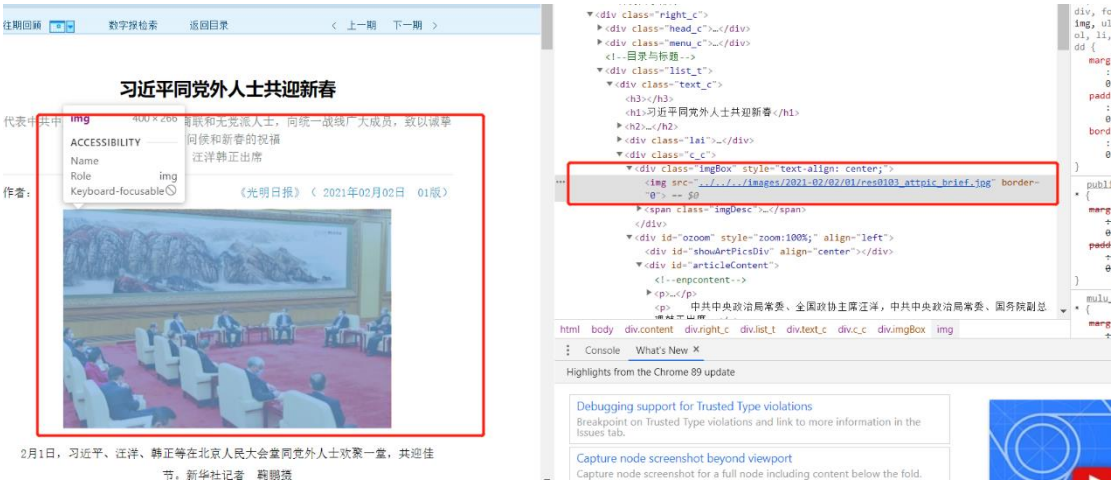


图 5 选取图片

第 2 步，通过程序访问网页，获取网页内容

通过程序访问上述提到的网址，可获取该网页的 HTML 代码。而通过该网页 HTML 代码，可获取网页中所有内容。代码如下：

分析该网页

```
# 伪装成chrome浏览器进行网页登录
headers = {'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/
# 文件保存地址
dirs_save = './爬取的文件/'

# 从指定网页url的内容读入到webdata中，以供分析使用
webdata = requests.get(url, headers=headers)
print(webdata.text)
# 通过xpath模块来分析网页内容
selector = etree.HTML(webdata.text)
```

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="content-type" content="text/html;charset=utf-8" />
<meta http-equiv="x-ua-compatible" content="ie=7" />
<meta http-equiv="content-language" content="utf-8" />
<meta name="robots" content="all" />
<meta name="author" content="" />
<meta name="copyright" content="" />
<meta name="description" content="习近平同党外人士共迎新春" />
<meta name="keywords" content="光明日报" />
<!-- TianRun tongji start-->
<meta name="webterren_speical" content="gmrbanalytics"/>
<META name="filetype" content="0">
<META name="publishedtype" content="1">
<META name="pagetype" content="1">
<META name="catalogs" content="11">
<!-- TianRun tongji end-->
<link href="../../../tplimg/mulu_4.css" type="text/css" rel="stylesheet" rev="s
<title>习近平同党外人士共迎新春-光明日报-光明网</title>
```

图 6 通过程序获得网页信息

第 3 步，通过程序爬取指定内容

获取对应的 HTML 代码后，即可通过爬虫代码爬取指定内容。如何获取呢？以选取文字为例：

图 7 新闻标题的 xpath 路径

对应程序：

获得新闻标题

```
[36]: titles = selector.xpath('//*[@class="text_c"]/h1/text()')
      #titles = selector.xpath('//*[@class="list_t"]/div/h1/text()')
      title = ''
      # 标题可能为两行
      for line in titles:
          title += line
      print(title)
```

习近平同党外人士共迎新春

图 8 通过 xpath 爬取新闻标题

以图片为例：

图 9 图片网址的 xpath 路径

对应程序：

```
img_url_part_list = selector.xpath('//*[@class="imgBox"]/img/@src')
#img_url_part_list = selector.xpath('//*[@class="c_c"]/div/img/@src')
```

图 10 通过 xpath 爬取图片网址

最后，将爬取的内容保存到本地。具体程序可见 *逐行演示.ipynb* 文件

第 4 步，自动爬取大量内容

程序运行至此，已爬取了指定网页的指定内容（新闻+图片）。但如何进行大量内容爬取呢，以前面提到的新闻网站为例，可以通过设置日期、新闻版块来爬取。比如，爬取过去一年的所有新闻与图片。这往往需要分析所爬取网页的网址特点：

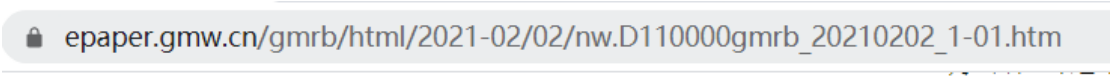


图 11　光明网 2 月 2 日新闻网址

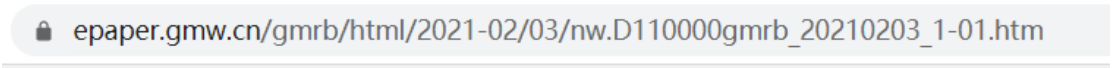


图 12　光明网 2 月 3 日新闻网址

对网址特点进行简单分析后，即可生成任意日期的新闻网址，重复 1-3 步，即可爬取任意日期的指定内容。

以起点中文网为例

对于爬取网络小说正文这一任务来说，思路是比较清楚的，即：打开网页-选择小说-选择章节-查看正文。这里的每一步都需要打开新的网页，所以，在爬取前需要分析这些元素在上一网页中的位置，以供自动爬取。

点击进入网址 https://www.qidian.com/free/all，选择免费小说以爬取全文

按 F12 进行内容选取，过程与上一例类似：

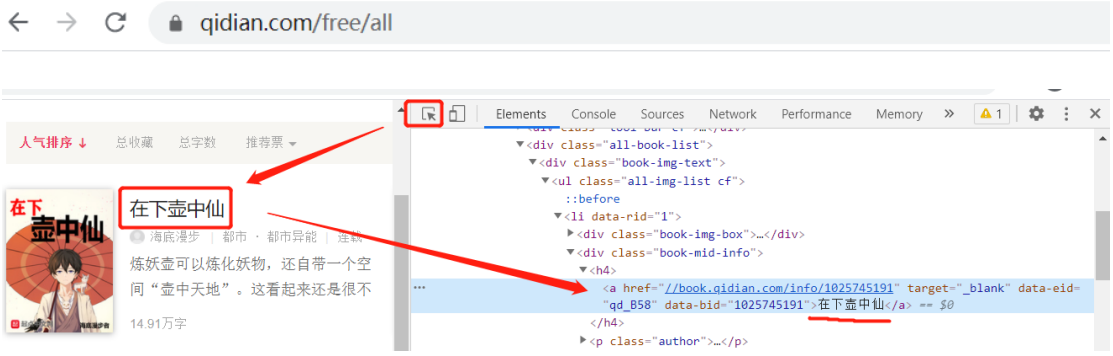


图 1　选取小说书名

图 2　小说书名的 xpath 路径

```
book_names = selector.xpath('//*[@class="book-mid-info"]/h4/a/text()')
#book_names = selector.xpath('//li/div/h4/a/text()')
book_names, len(book_names)
```

```
(['在下壶中仙',
  '我在秋斩刑场当缝尸人那些年',
```

图 3 通过 xpath 爬取书名

并爬取作者名、小说网址

```
    <a href="//book.qidian.com/info/1025745191" target="_blank" data-eid="qd_B58"
    data-bid="1025745191">在下壶中仙</a>
  </h4>
▼<p class="author">
    <img src="//qidian.gtimg.com/qd/images/ico/user.f22d3.png">
    <a class="name" href="//my.qidian.com/author/10726735" data-eid="qd_B59"
    target="_blank">海底漫步者</a>
    <em>|</em>
    <a href="//www.qidian.com/dushi" target="_blank" data-eid="qd_B60">都市</a>
```

//*[@class="author"]/a[1]/text()

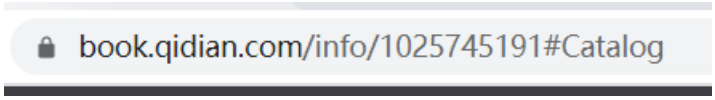
图 4 作者的 xpath 路径

```
<div class="book-img-text">
▼<ul class="all-img-list cf">
    ::before
  ▼<li data-rid="1">
    ▶<div class="book-img-box">…</div>
    ▼<div class="book-mid-info"> == $0
      ▼<h4>
        <a href="//book.qidian.com/info/1025745191" target="_blank" data-eid="qd_B58"
        data-bid="1025745191">在下壶中仙</a>
      </h4>
```

//*[@class="book-mid-info"]/h4/a/@data-bid

图 5 小说网址 xpath 路径

进入小说所在网址



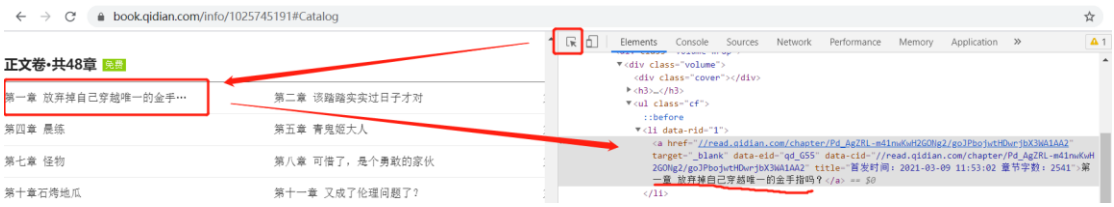


图 6 选取章节

重复之前的内容选取操作

```
  ▶<h3>…</h3>
  ▼<ul class="cf">
      ::before
    ▼<li data-rid="1">
      <a href="//read.qidian.com/chapter/Pd_AgZRL-m41nwKwH2GONg2/goJPbojwtHDwrjbX3WA1AA2"
      target="_blank" data-eid="qd_G55" data-cid="//read.qidian.com/chapter/Pd_AgZRL-m41nwKwH
      2GONg2/goJPbojwtHDwrjbX3WA1AA2" title="首发时间：2021-03-09 11:53:02 章节字数：2541">第
      一章 放弃掉自己穿越唯一的金手指吗？</a> == $0
```

//*[@class="cf"]/li/a/text()

图 7 章节名字的 xpath 路径

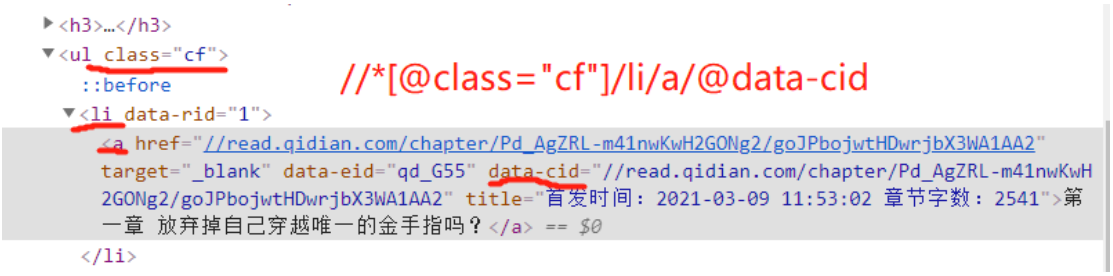


图 9 章节网址的 xpath 路径

进入章节所在网址，爬取正文

图 10 选取正文

图 11 正文的 xpath 路径

以今日头条为例

今日头条网站的内容显示与报刊新闻网站的内容显示有一定差异，报刊新闻网站在打开指定页面后，可以在 HTML 页面中爬取所有内容。而头条等网站则不是这样的，头条网站会在网址不变的情况下，随着用户下拉页面，源源不断地加载新闻。

下面一起来分析一下

第 1 步，浏览网页与查看网页元素

打开今日头条网址：https://www.toutiao.com/，选择不同的新闻版块并查看网址变化

图 1 游戏版块网址

图 2 新闻版块网址

会发现，在网页持续加载新闻时，该页网址不会发生变化。所以在爬取该版块内容时，需要试图模拟翻阅网页的加载过程。

那么是如何加载的呢？仍是按 F12 检查网页元素，选择 Network—XHR，在滑动加载新闻的同时，观察 Name 处的信息变化。可以发现，在加载新闻的同时，有新的网页元素加载进来。

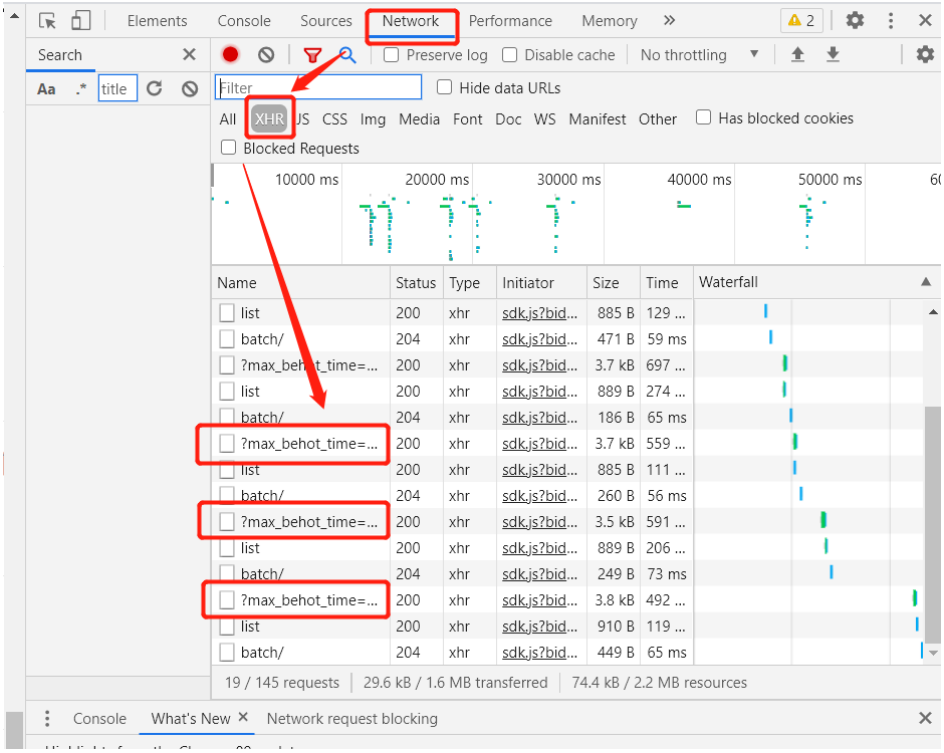


图 3 查看网页加载的元素

尝试在新的页面打开该元素

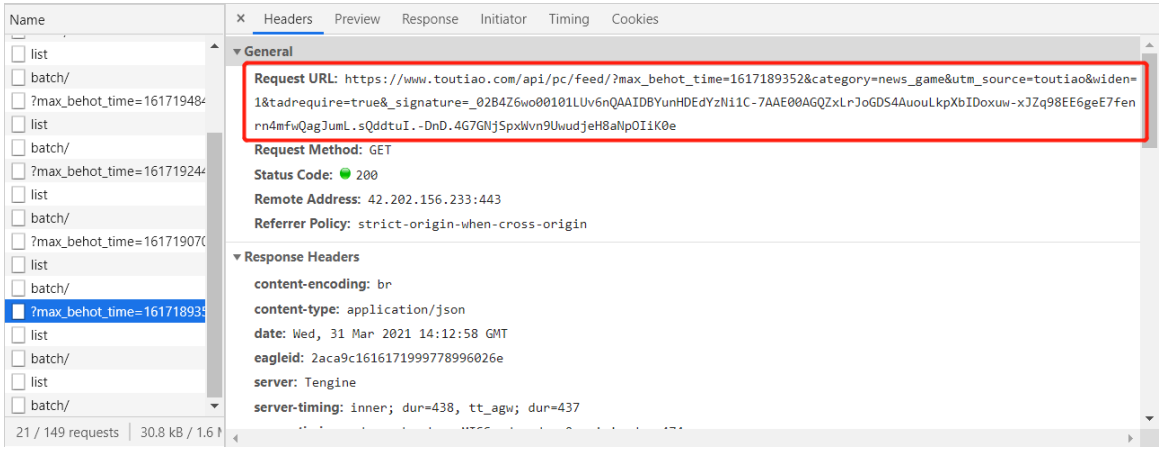


图 4 选中弯个腰加载的元素并查看信息

这其实是 json 格式的数据，在这些数据中就包含了新加载新闻的所有信息。尝试用程序来获取此处加载的信息，即可进一步爬取新闻的相关信息。

图 5 打开所加载元素的网页

在浏览器上查看这些信息不够直观，下图是读取这些信息后在 jupyter 上的显示。可以看出有新闻标题、新闻来源等信息。

图 6 通过程序查看加载的内容，可获取新闻标题等信息

第 2 步，通过程序访问网页，获取网页内容

分析 json 网址组成:

https://www.toutiao.com/api/pc/feed/?max_behot_time=1617189352&category=news_game&utm_source=toutiao&widen=1&tadrequire=true&_signature=_02B4Z6wo00101LUv6nQA

AIDBYunHDEdYzNi1C-7AAE00AGQZxLrJoGDS4AuouLkpXbIDoxuw-xJZq98EE6geE7fenrn4mfwQagJumL.sQddtuI.-DnD.4G7GNjSpxWvn9UwudjeH8aNpOIiK0e

https://www.toutiao.com/api/pc/feed/?max_behot_time=1617190042&category=__all__&utm_source=toutiao&widen=1&tadrequire=true&_signature=_02B4Z6wo00901k1oy8QAAIDDmq7mvpVRs1pNTM9AAPNNGESBlp4h8JenueQPSm4B.9f9HCskJQPe-BO2d4EYPk0sf1KfhnoaC8ddbIRgRUv7cXhu-0AKQGMYbsdFpnIFvihgpNwlKRbPt6ff92

根据系统当前时间和新闻版块名称，合成待访问的网址（网址中_signature 部分可删除）：

```
news_type = 'news_tech'
max_behot_time = str(round(time.time()))   # 链接参数
url_to_get = \
f'https://www.toutiao.com/api/pc/feed/?max_behot_time={max_behot_time}&category={news_type}&utm_source=toutiao&widen=1&tadrequire=true'
print(url_to_get)
```

```
https://www.toutiao.com/api/pc/feed/?max_behot_time=1617200106&category=news_tech&utm_source=toutiao&widen=1&tadrequire=true
```

图 7 json 网址组成

访问网址，获得 json 数据

```
webdata = requests.get(url_to_get, headers=random.choice(headers), cookies=cookies)
demo = json.loads(webdata.text)
```

```
demo
```

```
    'has_video': True,
    'chinese_tag': '视频',
    'source': '芝士来喇',
    'group_source': 2,
    'has_gallery': False,
```

图 8 通过程序获得 json 数据

第 3 步，通过程序爬取指定内容

已获得 josn 数据，可进一步获取新闻标题：

```
print(demo['data'][0]['title'])
```

```
中芯国际：2020年归母净利润43.32亿元，同比增长141.5%
```

图 9 爬取新闻标题

获取新闻的网址：

```
print(demo['data'][0]['source_url'])
```

```
/group/6945779554036384263/
```

```
print('https://www.toutiao.com'+demo['data'][0]['source_url'])
```

```
https://www.toutiao.com/group/6945779554036384263/
```

图 10 爬取新闻网址

打开 https://www.toutiao.com/group/6945779554036384263/：

图 11 进入程序爬取的网址，和从浏览器点击进入的网址一致

此时便可按照爬取光明网文字与图片的方法来爬取该页面的内容。

第 4 步，自动爬取大量内容

如何不断获取有效的 json 网址以进一步爬取呢，关键在于 max_behot_time 的选择。而新的用于加载的 max_behot_time 其实已在上一次 json 数据中给出了：

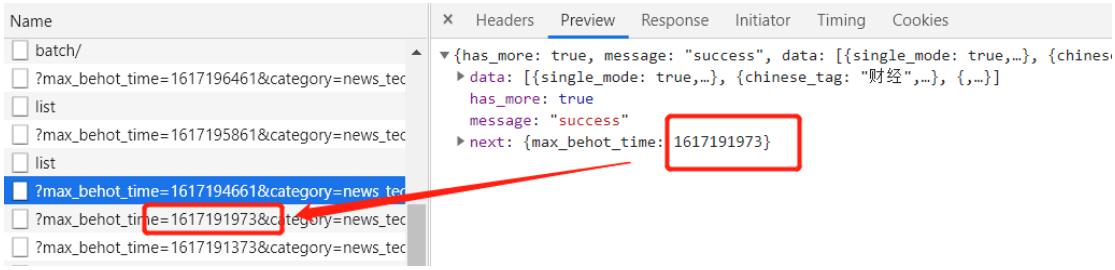


图 12 相邻两个加载元素间的联系

获取该数据，用于组合新的 json 数据请求即可：

```
print(demo['next'],'\n',demo['next']['max_behot_time'])
```

```
{'max_behot_time': 1617198261}
 1617198261
```

图 13 在 json 数据中获取 max_behot_time

根据以上分析可知，只要爬取到一个 josn，就可以从中提取 max_behot_time 来获取下一个 josn，即模拟在浏览器中不断滚动页面的操作。因此，只要将爬取程序放入一个循环中，即可不断爬取。可设置一个爬取数量作为爬取结束条件。