

```
63     #我们创建一个字典，搭建起A、B、C、D和四个rank值的映射关系。
64     if dic[xuanze] == y['rank']:
65         #此时dic[xuanze]的内容，其实就是rank值，此时的代码含义已经和之前的版本相同了。
66         right_num += 1
67     else:
68         wrong_words.append(y)
69
70 # ----第4步-生成报告----
71 print ('现在，到了公布成绩的时刻：')
72 print ('在'+str(len(words['data']))+'个'+js_link['data'][bianhao-1][1]+'词汇当中，你认识其中'+str(len(danci))+'个，实际掌握'+str(right_num)+'个，错误'+str(len(wrong_words))+'个。')
73 #这是句蛮复杂的话，对照前面的代码和json文件你才能理解它。一个运行示例是：在50个高考词汇当中，你认识其中30个，实际掌握25个，错误5个。
74
75 save = input ('是否打印并保存你的错词集？填入Y或N： ')
76 #询问用户，是否要打印并保存错题集。
77 if save == 'Y':
78     #如果用户说是：
79     f = open('错题集.txt', 'a+')
80     #在当前目录下，创建一个错题集.txt的文档。
81     print ('你记错的单词有：')
82     f.write('你记错的单词有：\n')
83     #写入"你记错的单词有：\n"
84     m=0
85     for z in wrong_words:
86         #启动一个循环，循环的次数等于，用户的错词数：
87         m=m+1
88         print (z['content'])
89         #打印每一个错词。
90         f.write(str(m+1) +'. '+ z['content']+'\n')
91         #写入序号，写入错词。
92     print ('你不认识的单词有：')
93     f.write('你没记住的单词有：\n')
94     #写入"你没记住的单词有：\n"
95     s=0
96     for x in not_knows:
97         #启动一个循环，循环的次数等于，用户不认识的单词数。
98         print (x['content'])
99         #打印每一个不认识的单词。
100        f.write(str(s+1) +'. '+ x['content']+'\n')
101        #写入序号，写入用户不认识的词汇。
```

```

102     print ('错词和没记住的词已保存至当前文件目录下，下次见！')
103     #告诉用户，文件已经保存好。
104     #在网页版终端运行时，文件会被写在课程的服务器上，你看不到，但它的确已经存在。
105 else:
106     #如果用户不想保存：
107     print('下次见！')
108     #输出“下次见！”

```

第8关 cookies

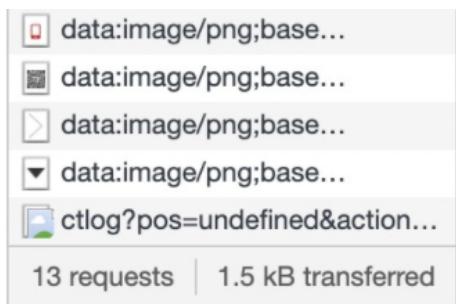
课后练习1：自制翻译器

- 题目解析：
 - 受制于篇幅，我们只介绍爬虫部分的逻辑。首先打开有道网页，同时打开检查模式，选择network。
 - 输入一个单词，然后从network里找到图中圈出的部分，点击Headers并下拉，找到From Data，其中的内容是我们需要的参数。

The screenshot shows a browser window with the Youdao Translate website open. The input field contains 'bye' and the output field shows '再见'. In the developer tools, the Network tab is selected. A specific POST request is highlighted with a red box around the 'translate_o?smartresult=dict...' parameter in the Headers section. The Headers section also shows other parameters like 'rlog.php?_npid=fanyiweb&_n...' and 'ctlog?pos=undefined&action...'. The Response section shows the JSON translation result.

- 点击preview后发现需要找的翻译路径是'translateResult-0-0-tgt'，如下图。

| Name | Headers | Preview | Response | Initiator | » |
|---------------------------------|--|---------|----------|-----------|---|
| translate_o?smartresult=dict... | <pre> {translateResult: [[{tgt: "再见", src: "bye"}]] errorCode: 0 smartResult: {entries: ["", "n. 轮空 (参赛者无对"] translateResult: [[{tgt: "再见", src: "bye"}]] 0: {tgt: "再见", src: "bye"} 0: {tgt: "再见", src: "bye" src: "bye" tgt: "再见" type: "en2zh-CHS" }] } </pre> | | | | |
| rlog.php?_npid=fanyiweb&_n... | | | | | |
| ctlog?pos=undefined&action... | | | | | |
| ctlog?pos=undefined&action... | | | | | |
| data:image/png;base... | | | | | |
| data:image/png;base... | | | | | |
| data:image/png;base... | | | | | |



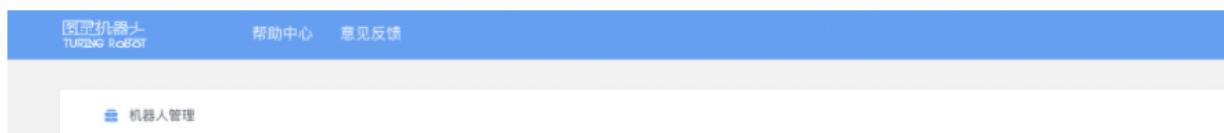
- 参考代码：

```
1 import requests, json
2 #调用了两个模块。requests负责上传和下载数据，json负责解析。
3
4
5 word = input('你想翻译什么呀？')
6 url='http://fanyi.youdao.com/translate?smartresult=dict&smartresult=rule'
7 #使用post需要一个链接。
8 data={'i': word,
9       'from': 'AUTO',
10      'to': 'AUTO',
11      'smartresult': 'dict',
12      'client': 'fanyideskweb',
13      'doctype': 'json',
14      'version': '2.1',
15      'keyfrom': 'fanyi.web',
16      'action': 'FY_BY_REALTIME',
17      'typoResult': 'false'}
18 #将需要post的内容，以字典的形式记录在data内。
19 r = requests.post(url,data)
20 #post需要输入两个参数，一个是刚才的链接，一个是data，返回的是一个Response对象。
21 answer=json.loads(r.text)
22 #你可以自己尝试print一下r.text的内容，然后再阅读下面的代码。
23 print ('翻译的结果是：'+answer['translateResult'][0][0]['tgt'])
```

课后练习2：图灵机器人

- 题目解析：

- 首先我们进入图灵机器人官网<http://www.tuling123.com/>，然后登陆注册一个账号，并创建机器人。创建好机器人之后我们会得到apikey。



The screenshot shows a list of five robots managed on the platform. Each robot has a small icon, a name, a daily chat count (all 0), an API key, and a version status (Free Edition). A red box highlights the API keys for all five robots.

| 机器人 | 今日聊天数量: 0 | apikey: [redacted] | 机器人版本: 免费版 | 设置 | 升级/续费 |
|--------|-----------|-----------------------------|------------|----|-------|
| 图灵机器人0 | 今日聊天数量: 0 | apikey: 4b3a... [redacted] | 机器人版本: 免费版 | 设置 | 升级/续费 |
| 图灵机器人2 | 今日聊天数量: 0 | apikey: 49afe... [redacted] | 机器人版本: 免费版 | 设置 | 升级/续费 |
| 图灵机器人3 | 今日聊天数量: 0 | apikey: cd90... [redacted] | 机器人版本: 免费版 | 设置 | 升级/续费 |
| 图灵机器人4 | 今日聊天数量: 0 | apikey: 2c92f... [redacted] | 机器人版本: 免费版 | 设置 | 升级/续费 |

创建机器人

- 接下来我们点开帮助中心的api文档，来查看图中的信息。

The screenshot shows the API V2.0 documentation page. The left sidebar has a navigation menu with 'API V2.0接入文档' highlighted. The right panel displays the API details:

- 接口地址:** `http://openapi.tuling123.com/openapi/api/v2`
- 请求方式:** HTTP POST
- 请求参数:** 请求参数格式为 json
- 请求示例:**

```
{
  "reqType": 0,
  "perception": {
    "inputText": {
      "text": "附近的酒店"
    },
    "inputImage": {
      "url": "imageUrl"
    },
    "selfInfo": {
      "location": {
        "city": "北京",
        "province": "北京",
        "street": "信息路"
      }
    },
    "userInfo": {
      "apiKey": "",
      "userId": ""
    }
}
```

- 从参数说明中可以看到，只有参数 perception 和 userinfo 才是必须的。
- 继续看下面的输出示例，就是我们请求之后的输出形式，text数据就是我们需要的数据。

输出参数

输出示例:

```
{
  "intent": [
    {
      "code": 10005,
      "intentName": "",
      "actionName": "",
      "parameters": {
        "nearby_place": "酒店"
      }
    }
  ],
  "results": [
    {
      "groupType": 1,
      "resultType": "url",
      "values": [
        {
          "url": "http://m.elong.com/hotel/0101/nlist/#inDate=2016-12-10&outDate=2016-12-11&keyw"
        }
      ]
    }
  ]
}
```

```
        "groupType": 1,
        "resultType": "text",
        "values": [
            {
                "text": "亲，已帮你找到相关酒店信息"
            }
        ]
    ]
```

- 参考代码：

```
1 import requests
2 import json
3
4 userid = str(1)
5 # 1 可以替换成任何长度小于32的字符串哦
6 apikey = str('A')
7 # 这里的A, 记得替换成你自己的apikey哦~
8
9 # 创建post函数
10 def robot(content):
11     # 图灵api
12     api = r'http://openapi.tuling123.com/openapi/api/v2'
13     # 创建post提交的数据
14     data = {
15         "perception": {
16             "inputText": {
17                 "text": content
18             }
19         },
20         "userInfo": {
21             "apiKey": apikey,
22             "userId": userid,
23         }
24     }
25     # 转化为json格式
26     jsondata = json.dumps(data)
27     # 发起post请求
28     response = requests.post(api, data = jsondata)
29     # 将返回的json数据解码
30     robot_res = json.loads(response.content)
31     # 提取对话数据
32     print(robot_res["results"][0]['values']['text'])
33
34 for x in range(10):
35     content = input("talk:")
36     # 输入对话内容
```

```

37     robot(content)
38     if x == 10:
39         break
40     # 十次之后就结束对话，数字可以改哦，你想几次就几次
41
42 #当然咯，你也可以加一些stopwords，只要说了这些词就可以终止聊天
43 while True:
44     content = input("talk:")
45     # 输入对话内容
46     robot(content)
47     if content == 'bye':
48         # 设置stopwords
49         break
50
51 #但是，我觉得吧，喜欢和聊天机器人玩的都是话痨，所以，可以最后加个死循环，如下：
52
53 # 创建对话死循环
54 while True:
55     # 输入对话内容
56     content = input("talk:")
57     robot(content)

```

课后练习3：nlpirc人工智能

- 题目解析：

- 第0步，打开网页与检查，输入关键字‘音乐剧’，点击网页中的“Word2vec”按钮（联想词汇）。
- 第1步，观察XHR中新增加的请求“getWord2Vec.do”，点击Headers，发现关键字‘音乐剧’是请求参数data里面参数值。
- 第2步，点击Perview，观察网页的返回数据，发现是json数据，需提取数据在请求中的‘v2wlist’列表中。
- 总体思路：使用post请求获取json数据（带data参数）-将json数据转换为字典-使用for循环遍历‘v2wlist’列表-分别提取近义词和相关度

- 参考代码：

```

1 import requests, json
2
3 # 请求并数据（带data参数）
4 url = 'http://ictclas.nlpir.org/nlpir/index6/getWord2Vec.do'
5 headers = {'user-agent':'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36'}
6

```

```

7 words = input('请输入你想查询的词汇：')
8 data = {'content':words}
9 res = requests.post(url,data=data,headers=headers)
10 data=res.text
11
12 # 把json数据转换为字典
13 data1=json.loads(data)
14
15 用for循环遍历'w2vlist'列表
16 f=0 # 变量f的作用是统计共有多少个近义词
17 for i in data1['w2vlist']: # i提取出来是字符串，如i='近义词.相关度'
18     f=f+1 # 近义词数量加1
19     word = i.split(',') # 以','号切割字符串，并转为列表，如word=['近义词','相关度']
20     print ('('+str(f)+')+word[0]+', 其相关度为'+word[1]) # 如word[0]是近义词，word[1]是相关
度

```

第9关 selenium

课后练习1：博客达人

- 题目解析：
 - 使用selenium实例化一个浏览器对象，再用其get()方法获取登录页。
 - 接下来我们定位到登录使用的用户名输入框和密码输入框输入账号信息。本习题我们可以不用注册账号，直接使用课程中提供的账号密码进行操作。
 - 点击《未来已来（三）——同九义何汝秀》文章标题，进入文章页面。注意从博客首页进入文章页面时，需要用到 find_element_by_partial_link_text 通过链接的部分文本获取超链接。
 - 定位到评论区的标签，模拟输入评论，点击发表评论。
- 参考代码：

```

1 import time
2 from selenium import webdriver
3
4 # 获取用户输入的评论内容
5 while True:
6     comment_content = input('请输入你想要的评论的内容，按回车提交：')
7     if comment_content == '':
8         print('&' * 5, '评论内容不允许为空', '&' * 5)
9     else:

```

```
10     break
11
12 driver = webdriver.Chrome() # 实例化浏览器对象
13 driver.get('https://wordpress-edu-3autumn.localprod.oc.forchange.cn/wp-login.php') # 访问页面
14 time.sleep(2)
15
16 # 定位到用户名输入框，输入用户名
17 login_name = driver.find_element_by_id('user_login')
18 login_name.send_keys('forchangeman')
19 time.sleep(1)
20 # 定位到密码输入框，输入密码
21 password = driver.find_element_by_id('user_pass')
22 password.send_keys('forchange123')
23 # 定位到登录按钮，并点击按钮
24 submit_btn = driver.find_element_by_id('wp-submit')
25 submit_btn.click()
26 time.sleep(2)
27
28 # 通过链接的部分文本定位到‘《未来已来（三）——同九义何汝秀》’这篇文章
29 # 获取到该文章对应的a标签（超链接），并点击链接进入文章详情页
30 article_link = driver.find_element_by_partial_link_text('同九义何汝秀')
31 article_link.click()
32
33 # 进入文章详情页，定位到该页面下编写评论的文本框，输入内容
34 comment_area = driver.find_element_by_id('comment')
35 comment_area.send_keys(comment_content)
36 time.sleep(2)
37 # 定位到提交按钮，点击该按钮提交评论
38 comment_submit = driver.find_element_by_id('submit')
39 comment_submit.click()
40
41 # 评论成功10秒后关闭浏览器
42 time.sleep(10)
43 driver.close()
44 print('#' * 6, '评论成功，浏览器已关闭', '#' * 6)
```

课后练习2：Python之禅

- 题目解析：
 - 使用selenium实例化一个浏览器对象，再用get()方法获取“你好，蜘蛛侠”的网页。

- 找到输入框的标签，输入你喜欢的老师和助教，点击提交。
- 网页跳转到Python之禅后，提取Elements的源代码里Python之禅文本的标签。
 - 方法一：使用selenium库自身提供的标签定位方法定位标签。
 - 方法二：使用selenium库中浏览器对象的page_source属性获取经过渲染的网页源代码，然后使用BeautifulSoup库的标签定位方法去定位。

- 参考代码：

```

1 # 方法一: selenium
2 from selenium import webdriver # 从selenium库中调用webdriver模块
3 import time
4
5 driver = webdriver.Chrome() # 声明浏览器对象
6 driver.get('https://localprod.pandateacher.com/python-manuscript/hello-spiderman/') #
访问页面
7 time.sleep(2) # 暂停两秒，等待浏览器缓冲
8
9 teacher = driver.find_element_by_id('teacher') # 找到【请输入你喜欢的老师】下面的输入框位置
10 teacher.send_keys('必须是吴枫呀') # 输入文字
11 assistant = driver.find_element_by_name('assistant') # 找到【请输入你喜欢的助教】下面的输入框
位置
12 assistant.send_keys('都喜欢') # 输入文字
13 button = driver.find_element_by_class_name('sub') # 找到【提交】按钮
14 button.click() # 点击【提交】按钮
15 time.sleep(1)
16
17 contents = driver.find_elements_by_class_name('content') # 定位到Python之禅所在的标签
18 for content in contents:
19     title = content.find_element_by_tag_name('h1').text # 提取标题
20     chan = content.find_element_by_tag_name('p').text # 提取正文
21     print(title + '\n' + chan + '\n') # 打印标题与正文
22 driver.close()

```

```

1 # 方法二: selenium+BeautifulSoup
2 from selenium import webdriver # 从selenium库总调用webdriver模块
3 import time
4 from bs4 import BeautifulSoup
5
6 driver = webdriver.Chrome() # 声明浏览器对象
7 driver.get('https://localprod.pandateacher.com/python-manuscript/hello-spiderman/') #
访问页面
8 time.sleep(2) # 暂停两秒，等待浏览器缓冲

```

```
9
10 teacher = driver.find_element_by_id('teacher') # 定位到【请输入你喜欢的老师】下面的输入框位置
11 teacher.send_keys('必须是吴枫呀') # 输入文字
12 assistant = driver.find_element_by_name('assistant') # 定位到【请输入你喜欢的助教】下面的输入
    框位置
13 assistant.send_keys('都喜欢') # 输入文字
14 button = driver.find_element_by_class_name('sub') # 定位到【提交】按钮
15 button.click() # 点击【提交】按钮
16 time.sleep(1) # 等待一秒
17
18 pageSource = driver.page_source # 获取页面信息
19 soup = BeautifulSoup(pageSource, 'html.parser') # 使用bs解析网页
20 contents = soup.find_all(class_="content") # 找到源代码Python之禅中文版和英文版所在的元素
21 for content in contents: # 遍历列表
22     title = content.find('h1').text # 提取标题
23     chan = content.find('p').text.replace(' ', '') # 提取Python之禅的正文，并且去掉文字前面
        的所有空格
24     print(title + chan + '\n') # 打印Python之禅的标题与正文
25 driver.close()
```

第10关 定时与邮件

课后练习：周末吃什么

- 题目解析：
 - 爬虫：按照爬虫第3关爬取下厨房的方法进行序号、菜名、链接和原料的爬取。
 - 发送邮件：使用email和smtplib两个库按照基础语法第17关的方法进行邮件发送。
 - 设置定时任务：使用schedule库设置定时任务，注意：可以不把发送邮件的时间设置为固定时间。
- 参考代码：

```
1 import requests
2 import smtplib
3 import schedule
4 import time
5 from bs4 import BeautifulSoup
6 from email.mime.text import MIMEText
7 from email.header import Header
8
```

```
9 account = input('请输入你的邮箱: ')
10 password = input('请输入你的密码: ')
11 receiver = input('请输入收件人的邮箱: ')
12
13 def recipe_spider():
14     headers={'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_5)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/85.0.4183.121 Safari/537.36'}
15     res_foods = requests.get('http://www.xiachufang.com/explore/',headers=headers)
16     bs_foods = BeautifulSoup(res_foods.text,'html.parser')
17     list_foods = bs_foods.find_all('div',class_='info pure-u')
18     list_all = ''
19     num=0
20     for food in list_foods:
21         num=num+1
22         tag_a = food.find('a')
23         name = tag_a.text.strip()
24         url = 'http://www.xiachufang.com'+tag_a['href']
25         tag_p = food.find('p',class_='ing ellipsis')
26         ingredients = tag_p.text.strip()
27         food_info = '''
28             序号: %s
29             菜名: %s
30             链接: %s
31             原料: %s
32             '''%(num,name,url,ingredients)
33         list_all=list_all+food_info
34     return(list_all)
35
36 def send_email(list_all):
37     global account,password,receiver
38     mailhost='smtp.qq.com'
39     qqmail = smtplib.SMTP()
40     qqmail.connect(mailhost,25)
41     qqmail.login(account,password)
42     content= '亲爱的，本周的热门菜谱如下'+list_all
43     message = MIMEText(content, 'plain', 'utf-8')
44     subject = '周末吃个啥'
45     message['Subject'] = Header(subject, 'utf-8')
46     try:
47         qqmail.sendmail(account, receiver, message.as_string())
48         print ('邮件发送成功')
```

```

49     except:
50         print ('邮件发送失败')
51     qqmail.quit()
52
53 def job():
54     print('开始一次任务')
55     list_all = recipe_spider()
56     send_email(list_all)
57     print('任务完成')
58
59 schedule.every().tuesday.at("11:27").do(job) #部署每周三的13: 15执行函数的任务
60 while True:
61     schedule.run_pending()
62     time.sleep(1)

```

第11关 协程

课后练习：煲剧达人

- 题目解析：
 - 该练习和第3关课后练习豆瓣爬虫比较相似，面对多个网址，分析思路也是一样，翻页查看网址规律，该练习第一页的网址不同于其他9页，需要单独写，其他九页的网址，只有一个数字不同，可以通过for循环来控制。
 - 找标签先使用小箭头定位，然后测试爬取。测试成功之后，再加上gevent库相关代码，进行多爬虫爬取数据。
 - 不同的是，该网址中的电影，缺失内容（导演、主演、推荐语）的数量比较多，所以在取的时候，直接把所有都取到即可。找到标签<div class_="mov_con">下的所有<p>标签，使用加号 + 进行拼接。
 - 注意：存储时可以使用utf-8-sig编码防止csv在Excel中打开出现乱码，uft-8-sig带有签名（Bom），Bom一般出现在文本文件头部，Unicode编码标准中用于标识文件是采用哪种格式的编码。Excel 在读取 csv 文件的时候是通过读取文件头上的 BOM 来识别编码的，如果文件头无 BOM 信息，则默认按照 Unicode 编码读取，如果使用 utf-8 编码生成 csv 文件（没有生成BOM信息），Excel 就会自动按照 Unicode 编码读取，就会出现乱码问题了。
- 参考代码：

```

1 from gevent import monkey
2 monkey.patch_all()
3 import gevent, requests, bs4, csv

```

```

4  from gevent.queue import Queue
5
6  work = Queue()
7  url_1 = 'http://www.mtime.com/top/tv/top100/'
8  work.put_nowait(url_1)
9
10 url_2 = 'http://www.mtime.com/top/tv/top100/index-{page}.html'
11 for x in range(2,11):
12     real_url = url_2.format(page=x)
13     work.put_nowait(real_url)
14
15 def crawler():
16     headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36'}
17     while not work.empty():
18         url = work.get_nowait()
19         res = requests.get(url,headers=headers)
20         bs_res = bs4.BeautifulSoup(res.text,'html.parser')
21         datas = bs_res.find_all('div',class_="mov_con")
22         for data in datas:
23             TV_title = data.find('a').text
24             data = data.find_all('p')
25             TV_data =''
26             for i in data:
27                 TV_data =TV_data + i.text + ' '
28             writer.writerow([TV_title,TV_data])
29             print([TV_title,TV_data])
30
31 csv_file = open('timetop.csv','w',newline='',encoding='utf-8-sig')
32 writer = csv.writer(csv_file)
33
34 task_list = []
35 for x in range(3):
36     task = gevent.spawn(crawler)
37     task_list.append(task)
38 gevent.joinall(task_list)
39 csv_file.close()

```

第13关 Scrapy框架

课后练习：当当图书馆爬虫

- 题目解析：

- 分析网址：翻页查看网址规律，该网站的网址规律比较统一，只有一个数字不同，可以使用for循环控制变量来给网址赋值该数字；
- 定位标签：借助小箭头进行定位，查找标签所有数据都在标签`<ul class="bang_list clearfix bang_list_mode">`下，详细数据标签如下：
 - 书名在`<div class="name">`
 - 作者在`<div class="publisher_info">`
 - 价格在``
- 文件编写：
 - 在`items.py`文件里定义item；
 - 在`bestsellers.py`文件里编写spider；
 - 在`settings.py`文件里修改设置；
 - 在`main.py`文件里写入运行scrapy的代码，并点击运行。
- 注意：课堂系统里并没有显示Scrapy的项目里的所有文件，只显示了需要修改或编辑的文件。

- 参考代码：

```
1 # items.py文件参考代码
2 import scrapy
3 class DangdangItem(scrapy.Item):
4     name = scrapy.Field()
5     author = scrapy.Field()
6     price = scrapy.Field()
```

```
1 # bestsellers.py文件里spider参考代码
2 import scrapy
3 import bs4
4 from ..items import DangdangItem
5
6 class DangdangSpider(scrapy.Spider):
7     name = 'dangdang'
8     allowed_domains = ['http://bang.dangdang.com']
9     start_urls = []
10    for x in range(1, 4):
11        url = 'http://bang.dangdang.com/books/bestsellers/01.00.00.00.00.00-
12        year-2018-0-1-' + str(x)
13        start_urls.append(url)
```

```

13     def parse(self, response):
14         soup = bs4.BeautifulSoup(response.text, 'html.parser')
15         elements = soup.find('ul', class_="bang_list clearfix
16             bang_list_mode").find_all('li')
17         for element in elements:
18             item = DangdangItem()
19             item['name'] = element.find('div', class_="name").find('a')['title']
20             item['author'] = element.find('div', class_="publisher_info").text
21             item['price'] = element.find('div', class_="price").find('span',
22                 class_="price_n").text
23             yield item

```

```

1 # settings.py文件参考代码 (添加请求头, 修改ROBOTSTXT_OBEY的值)
2 USER_AGENT = 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/71.0.3578.98 Safari/537.36'
3 # Obey robots.txt rules
4 ROBOTSTXT_OBEY = True

```

```

1 # main.py文件参考代码
2 from scrapy import cmdline
3 cmdline.execute(['scrapy', 'crawl', 'dangdang'])

```

第14关 Scrapy实操

课后练习：豆瓣图书短评

- 题目解析：
 - 分析网址：翻页查看网址规律，前面已经分析过了，规律就是，只有一个数字不同，可以使用for循环控制变量来给网址赋值该数字；
 - 定位标签：借助小箭头进行定位，查找短评所有数据都在标签<div class="comment">下，详细数据标签如下：
 - 发表人在<div class="comment">下的第1个（索引值）<a>标签
 - 内容在
- 文件编写：
 - 在items.py文件里定义item；
 - 在comments.py文件里编写spider；

- 在settings.py文件里修改设置；
 - 在pipelines.py文件里写入存储成Excel的代码；
 - 在main.py文件里写入运行scrapy的代码，并点击运行。
- 注意：
 - 如果你的代码成功运行，在【文件】里的root文件夹下的douban文件夹里，你就可以找到你创建的存储数据的Excel文件，将其下载到本地，你就能用Excel打开查看。
 - scrapy的项目和spider的文件，已经在练习系统里提前为你创建好了。
- 参考代码：

```

1 # items.py文件参考代码
2 import scrapy
3 class DoubantopItem(scrapy.Item):
4     book_name = scrapy.Field()
5     ID_name = scrapy.Field()
6     comment = scrapy.Field()

```

```

1 # comments.py文件里编写spider
2 import scrapy,bs4
3 from ..items import DoubantopItem
4
5 class DoubantopSpider(scrapy.Spider):
6     name = 'doubantop'
7     allowed_domains = ['https://book.douban.com']
8     start_urls = []
9     for x in range(2):
10         url = 'https://book.douban.com/top250?start=' + str(x * 25)
11         start_urls.append(url)
12
13     def parse(self, response):
14         soup = bs4.BeautifulSoup(response.text,'html.parser')
15         datas = soup.find_all('tr',class_='item')
16         for data in datas:
17             book_url = data.find_all('a')[1]['href']
18             comment_url = book_url+'comments/'
19             yield scrapy.Request(comment_url,callback=self.parse_comment)
20
21     def parse_comment(self,response):
22         soup = bs4.BeautifulSoup(response.text,'html.parser')
23         book_name = soup.find('div',id='content').text.split()[0]
24         datas = soup.find_all('div',class_='comment')
25         for data in datas:

```

```
26         item = DoubantopItem()
27
28         item['book_name'] = book_name
29
30         item['ID_name'] = data.find_all('a')[1].text
31         item['comment'] = data.find('span', class_='short').text
32
33         yield item
```

```
1 # settings.py文件参考代码 (添加请求头, 修改ROBOTSTXT_OBEY的值, 取消6~8行的注释)
2 USER_AGENT = 'Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/71.0.3578.98 Safari/537.36'
3 # Obey robots.txt rules
4 ROBOTSTXT_OBEY = True
5 ITEM_PIPELINES = {
6     'doubantop.pipelines.DoubantopPipeline': 300,
7 }
```

```
1 # pipelines.py文件参考代码
2 import openpyxl
3 class DoubantopPipeline(object):
4     def __init__(self):
5         self.wb = openpyxl.Workbook()
6         self.ws = self.wb.active
7         self.ws.append(['书名', 'ID名', '短评'])
8
9     def process_item(self, item, spider):
10        line = [item['book_name'], item['ID_name'], item['comment']]
11        self.ws.append(line)
12        return item
13
14    def close_spider(self, spider):
15        self.wb.save('book.xlsx')
16        self.wb.close()
```

```
1 # main.py文件参考代码
2 from scrapy import cmdline
3 cmdline.execute(['scrapy', 'crawl', 'doubantop'])
```

使用腾讯文档创作和发布 多人协作 / 远程办公 / 信息收集

我也要用 >