

第 9 关、Selenium

1、Selenium

1-1、什么是 Selenium

- selenium 是一个 python 库，可以用几行代码，控制浏览器，做出自动打开、输入、点击等操作，就像是有一个真正的用户在操作一样。
- selenium 可以真实地打开一个浏览器，等待所有数据都加载到 Elements 中之后，再把这个网页当做静态网页爬取。
- 由于要真实地运行本地浏览器，打开浏览器以及等待网渲染完成需要一些时间，selenium 的工作不可避免地牺牲了速度和更多资源，不过，至少不会比人慢。

1-2、怎么用 Selenium

- (1) 网站：<https://localprod.pandateacher.com/python-manuscript/hello-spiderman/>；
- (2) 确定目标：模拟登录及网站操作。

1-2-1、安装 selenium

```
1 pip install selenium # Windows电脑安装selenium
2 pip3 install selenium # Mac电脑安装selenium
```

selenium 的脚本可以控制所有常见浏览器的操作，在使用之前，需要安装浏览器的驱动。（推荐使用 Chrome 浏览器：<https://localprod.pandateacher.com/python-manuscript/crawler-html/chromedriver/ChromeDriver.html>）

1-2-2、设置浏览器引擎

```
1 # 本地Chrome浏览器设置方法
2 from selenium import webdriver #从selenium库中调用webdriver模块
3 driver = webdriver.Chrome() # 设置引擎为Chrome，真实地打开一个Chrome浏览器
```

1-2-3、获取数据

```
1 # 本地Chrome浏览器设置方法
2 from selenium import webdriver #从selenium库中调用webdriver模块
3 driver = webdriver.Chrome() # 设置引擎为Chrome，真实地打开一个Chrome浏览器
4
5 driver.get('https://localprod.pandateacher.com/python-manuscript/hello-spiderman/') # 打开网页
```

```
6 time.sleep(1)
7 driver.close() # 关闭浏览器
```

- get(URL) 是 webdriver 的一个方法，它的使命是为你打开指定 URL 的网页；
- driver.close() 是关闭浏览器驱动，每次调用了 webdriver 之后，都要在用完它之后加上一行 driver.close() 用来关闭它

1-2-4、解析与提取数据

(1) 解析数据是由 driver 自动完成的，提取数据是 driver 的一个方法。

Selenium提取数据的方法

方法	作用
find_element_by_tag_name	通过元素的标签名称选择
find_element_by_class_name	通过元素的class属性选择
find_element_by_id	通过元素的id选择
find_element_by_name	通过元素的name属性选择
find_element_by_link_text	通过链接文本获取超链接
find_element_by_partial_link_text	通过链接的部分文本获取超链接

by 风变编程

```
1 # 以下方法都可以从网页中提取出 '你好，蜘蛛侠！' 这段文字
2
3 find_element_by_tag_name: 通过元素的名称选择
4 # 如<h1>你好，蜘蛛侠！</h1>
5 # 可以使用find_element_by_tag_name('h1')
6
7 find_element_by_class_name: 通过元素的class属性选择
8 # 如<h1 class="title">你好，蜘蛛侠！</h1>
9 # 可以使用find_element_by_class_name('title')
10
11 find_element_by_id: 通过元素的id选择
12 # 如<h1 id="title">你好，蜘蛛侠！</h1>
13 # 可以使用find_element_by_id('title')
14
15 find_element_by_name: 通过元素的name属性选择
16 # 如<h1 name="hello">你好，蜘蛛侠！</h1>
17 # 可以使用find_element_by_name('hello')
18
19 #以下两个方法可以提取出超链接
20
21 find_element_by_link_text: 通过链接文本获取超链接
22 # 如<a href="spidermen.html">你好，蜘蛛侠！</a>
23 # 可以使用find_element_by_link_text('你好，蜘蛛侠！')
24
```

```

25 find_element_by_partial_link_text: 通过链接的部分文本获取超链接
26 # 如<a href="https://localprod.pandateacher.com/python-manuscript/hello-
    spiderman/">你好, 蜘蛛侠! </a>
27 # 可以使用find_element_by_partial_link_text('你好')

```

(2) WebElement 类对象与 Tag 对象类似, 它也有一个方法, 可以通过属性名提取属性的值, 这个方法是 .get_attribute()

WebElement与Tag的用法对比		
WebElement	Tag	作用
WebElement.text	Tag.text	提取文字
WebElement.get_attribute()	Tag[]	输入参数: 属性名, 可以提取属性值

by 风变编程

【selenium 解析与提取数据的过程】

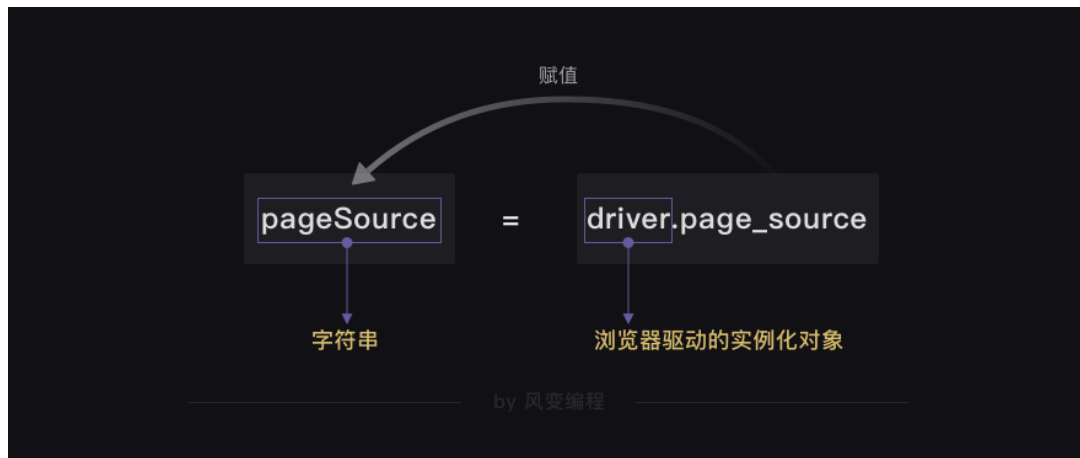


(3) selenium 获取渲染完整的网页源代码, 使用 driver 的一个方法: page_source, 获取到的网页源代码, 本身已经是字符串了。

```

1 HTML源代码字符串 = driver.page_source

```



1-2-5、自动操作浏览器

页面模拟操作点击的方法：

```
1 .send_keys() # 模拟按键输入，自动填写表单
2 .click() # 点击元素
```

```
1 # 本地Chrome浏览器设置方法
2 from selenium import webdriver # 从selenium库中调用webdriver模块
3 import time # 调用time模块
4 driver = webdriver.Chrome() # 设置引擎为Chrome，真实地打开一个Chrome浏览器
5
6 driver.get('https://localprod.pandateacher.com/python-manuscript/hello-
7 spiderman/') # 访问页面
8
9 teacher = driver.find_element_by_id('teacher') # 找到【请输入你喜欢的老师】下
10 面的输入框位置
11 teacher.send_keys('必须是吴枫呀') # 输入文字
12
13 assistant = driver.find_element_by_name('assistant') # 找到【请输入你喜欢的助
14 教】下面的输入框位置
15 assistant.send_keys('都喜欢') # 输入文字
16
17 button = driver.find_element_by_class_name('sub') # 找到【提交】按钮
18 button.click() # 点击【提交】按钮
19 time.sleep(1)
20 driver.close() # 关闭浏览器
```

(4) Selenium 操作元素的常用方法

Selenium操作元素的常用方法

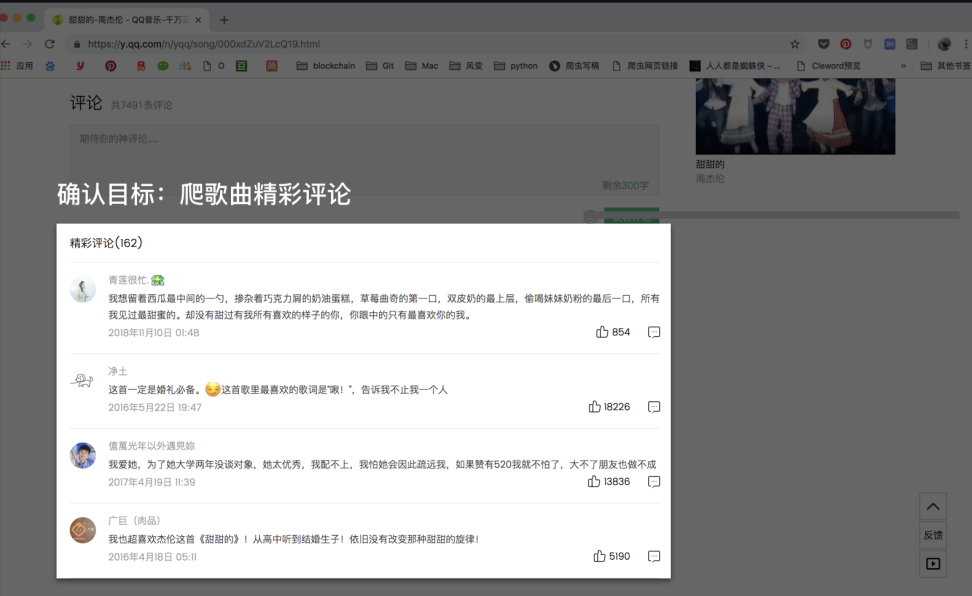
方法	作用
<code>.clear()</code>	清除元素的内容
<code>.send_keys()</code>	模拟按键输入，自动填写表单
<code>.click()</code>	点击元素

by 风变编程

2、项目实操

2-1、确定目标

- (1) 目标网站：<https://y.qq.com/n/yqq/song/000xdZuV2LcQ19.html>;
- (2) 项目目标：爬取歌曲《甜甜的》的歌曲评论。



评论 共7491条评论

期待你的神评论...

确认目标：爬歌曲精彩评论

剩余300字

甜甜的
周杰伦

精彩评论 (162)

青莲很忙
我想留着西瓜最中间的一勺，掺杂着巧克力屑的奶油蛋糕，草莓曲奇的第一口，双皮奶的最上层，偷喝妹妹奶粉的最后一口，所有我见过最甜蜜的，却没有甜过有我所有喜欢的样子的你，你眼中的只有最喜欢你的我。
2018年11月10日 01:48 854

净土
这首一定是婚礼必备。😂这首歌里最喜欢的歌词是“嘿！”，告诉我不止我一个人
2016年5月22日 19:47 18226

億萬光年以外遇見妳
我爱她，为了她大学两年没谈对象，她太优秀，我配不上，我怕她会因此疏远我，如果赞有520我就不怕了，大不了朋友也做不成
2017年4月19日 11:39 13836

广巨 (肉品)
我也超喜欢杰伦这首《甜甜的》！从高中听到结婚生子！依旧没有改变那种甜甜的旋律！
2016年4月18日 05:11 5190

by 风变编程

2-2、过程分析

直接使用 selenium 控制浏览器点击【点击加载更多】的按钮，让评论数据都加载到 elements 中。

- (1) 所有评论信息的共同标签是 `class="c_tx_normal comment__text js_hot_text"`;



(2) 【点击加载更多】按钮的标签是 class="comment__show_all_link c_tx_thin js_get_more_hot"。



2-3、代码实现

```

1 # 本地Chrome浏览器设置方法
2 from selenium import webdriver #从selenium库中调用webdriver模块
3 from bs4 import BeautifulSoup
4 import time
5
6 driver = webdriver.Chrome() # 设置引擎为Chrome，真实地打开一个Chrome浏览器
7 driver.get('https://y.qq.com/n/yyq/song/000xdZuV2LcQ19.html') # 访问页面
8 time.sleep(2)
9
10 button = driver.find_element_by_class_name('js_get_more_hot') # 根据类名找到【点击加载更多】
11 button.click() # 点击
12 time.sleep(2) # 等待两秒
13
14 pageSource = driver.page_source # 获取Elements中渲染完成的网页源代码
15 soup = BeautifulSoup(pageSource,'html.parser') # 使用bs解析网页
16 comments =
17     soup.find('ul',class_='js_hot_list').find_all('li',class_='js_cmt_li') #
18     使用bs提取元素
19 print(len(comments)) # 打印comments的数量
20
21 for comment in comments: # 循环
22     sweet = comment.find('p') # 提取评论
23     print ('评论: %s\n ---\n'%sweet.text) # 打印评论

```

3、Selenium 静默模式

```
1 # 本地Chrome浏览器的静默默模式设置:
2 from selenium import webdriver #从selenium库中调用webdriver模块
3 from selenium.webdriver.chrome.options import Options # 从options模块中调用
  Options类
4
5 chrome_options = Options() # 实例化Option对象
6 chrome_options.add_argument('--headless') # 把Chrome浏览器设置为静默模式
7 driver = webdriver.Chrome(options = chrome_options) # 设置引擎为Chrome, 在后
  台默默运行
```

与上面浏览器的可视设置相比，3、5、6行代码是新增的，首先调用了一个新的类——Options，然后通过它的方法和属性，给浏览器输入了一个参数——headless。第7行代码中，把刚才所做的浏览器设置传给了 Chrome 浏览器。