

## 实验流程

### 实验软件版本

- Pycharm 2021.2.3
- python 3.9
- selenium 3.9.0
- chromedriver.exe 106.0.5249.67
- chrome浏览器 106.0.5249.119

### 实验流程

录屏详见：

链接：[https://pan.baidu.com/s/1Jfu1\\_7nTv0RTseR9i6bQLw](https://pan.baidu.com/s/1Jfu1_7nTv0RTseR9i6bQLw)

提取码：cjej

### 相关知识点

```
1  # 定义驱动对象
2  driver = webdriver.Chrome()
3
4  # 打开网页
5  url = 'https://nj.zu.anjuke.com/'
6  driver.get(url)
7
8  # 界面最大化
9  driver.maximize_window()
10
11 # 关闭驱动对象
12 driver.quit()
```

**Selenium的定位方式:**

定位一个元素	定位多个元素	含义
find_element_by_id	find_elements_by_id	通过元素id定位
find_element_by_name	find_elements_by_name	通过元素name定位
find_element_by_xpath	find_elements_by_xpath	通过xpath表达式定位
find_element_by_link_text	find_elements_by_link_text	通过完整超链接定位
find_element_by_partial_link_text	find_elements_by_partial_link_text	通过部分链接定位
find_element_by_tag_name	find_elements_by_tag_name	通过标签定位
find_element_by_class_name	find_elements_by_class_name	通过类名进行定位
find_elements_by_css_selector	find_elements_by_css_selector	通过css选择器进行定位

通过id定位:

```
1 driver.find_element_by_id("kw")
```

通过name定位:

```
1 driver.find_element_by_name("wd")
```

通过class name定位:

```
1 driver.find_element_by_class_name("s_ippt")
```

通过tag name定位:

```
1 driver.find_element_by_tag_name("input")
```

通过xpath定位, 几个常用写法:

```
1 dr.find_element_by_xpath("//*[@id='kw']")
```

过css定位, 几个常用写法:

```
1 dr.find_element_by_css_selector("#kw")
```

通过link text定位:

```
1 dr.find_element_by_link_text("新闻")
```

通过partial link text定位:

```
1 | dr.find_element_by_partial_link_text("新")
```

### 控制浏览器操作方法:

- set\_window\_size() 设置浏览器的大小
- back() 控制浏览器后退
- forward() 控制浏览器前进
- refresh() 刷新当前页面
- clear() 清除文本
- send\_keys(value) 模拟按键输入
- click() 单击元素
- submit() 用于提交表单
- get\_attribute(name) 获取元素属性值
- is\_displayed() 设置该元素是否用户可见
- size 返回元素的尺寸
- text 获取元素的文本

### 鼠标事件

在 WebDriver 中, 将这些关于鼠标操作的方法封装在 ActionChains 类提供。

- ActionChains(driver) 构造ActionChains对象
- context\_click() 执行鼠标悬停操作
- move\_to\_element(above) 右击
- double\_click() 双击
- drag\_and\_drop() 拖动
- move\_to\_element(above) 执行鼠标悬停操作
- context\_click() 用于模拟鼠标右键操作, 在调用时需要指定元素定位
- perform() 执行所有 ActionChains 中存储的行为, 可以理解成是对整个操作的提交动作

```
1 | ActionChains(driver).move_to_element(above).perform()
```

### 鼠标拖放操作

drag\_and\_drop(source,target) 在源元素上按住鼠标左键, 然后移动到目标元素上释放。

- source 鼠标拖动的源元素
- target鼠标释放的目标元素

```
1 | #定位元素的初始位置
2 | source = driver.find_element_by_css_selector("")
3 | #定位元素的要移动到的位置
4 | target = driver.find_element_by_name("")
5 | #执行元素的拖放操作
6 | ActionChains(driver).drag_and_drop(source,target).perform()
```

## 实验代码整理：

```
1 from selenium import webdriver
2 from selenium.webdriver import ActionChains
3 import time
4
5 driver = webdriver.Chrome()
6 url = 'https://nj.zu.anjuke.com/'
7 driver.get(url)
8
9 driver.maximize_window()
10
11 driver.find_element_by_partial_link_text('租房').click()
12 time.sleep(1)
13 above = driver.find_element_by_id('switch_apf_id_5')
14 ActionChains(driver).move_to_element(above).perform()
15 driver.find_element_by_partial_link_text('南京').click()
16 time.sleep(1)
17 driver.find_element_by_partial_link_text('地铁找房').click()
18 time.sleep(1)
19 driver.find_element_by_partial_link_text('2号线').click()
20 time.sleep(1)
21 driver.find_element_by_partial_link_text('马群').click()
22 time.sleep(1)
23 driver.find_element_by_id('from-price').send_keys("5000")
24 time.sleep(1)
25 driver.find_element_by_id('to-price').send_keys("8000")
26 time.sleep(1)
27 driver.find_element_by_id('pricerange_search').click()
28 time.sleep(1)
29
30 driver.find_element_by_partial_link_text('整租').click()
31 time.sleep(1)
32
33 above = driver.find_element_by_id('condhouseage_txt_id')
34 ActionChains(driver).move_to_element(above).perform()
35 driver.find_element_by_partial_link_text('普通住宅').click()
36 time.sleep(1)
37
38 driver.find_element_by_id('search-rent').send_keys('经天路')
39 driver.find_element_by_id('search-button').click()
40 time.sleep(1)
41
42 driver.find_element_by_class_name('lastTab').click()
43 time.sleep(1)
44
45 driver.find_element_by_class_name('icon-arrup').click()
46 time.sleep(1)
47
48 driver.find_element_by_class_name('icon-arrdown').click()
49 time.sleep(1)
50 driver.find_element_by_class_name('zu-itemmod').click()
51 time.sleep(1)
```

# 实验感悟

## 学习心得

由于之前已经了一定的python基础，此次web实验实验中基本没有遇到问题，在学习了Python使用Selenium的webdriver的方法之后，在写代码阶段基本没有遇到问题，很快就能完成任务。我认为主要的难度在于环境的配置，首先是chrome浏览器的版本和chromedriver的版本要适配。其次Selenium的版本也要注意，不同版本的Selenium，其各种方法的名称和功能也不尽相同，在写代码时可能会遇到问题。

## 代码建议

- 每一次点击过后，最好使用 `time.sleep()` 休眠一段时间，防止当前页面还没有完全加载出来，就进行了下一次点击，可能程序会报错。
- 当网页经过点击后打开了一个新的界面时，会发现在新的界面里 `click()` 方法失效，是由于程序依然在原来的界面中进行点击，需要转换网页，再进行下一步操作。

2022/10/24