需要用到的库：request,urllib,re,.....

参数代码的准备：

url = "url"

headers = {

"user-agent": "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36"

}

####发送请求并获取数据：response =requests.get(url=url,headers=headers)

print(response.text) //打印网页的html代码

#### ####筛选数据：匹配（bs4

soup = BeautifulSoup(response.content, 'html.parser')  
articles = soup.find\_all('a')  
print(articles)

-------------以上仅是爬取数据，并不能对网页进行操作-----------

Selenium库：

是最广泛使用的开源 Web UI（用户界面）自动化测试套件之一

### **打开浏览器**

以打开去 163 邮箱为例，使用 Firefox 浏览器

from selenium import webdriver  
browser =webdriver.Firefox()browser.get('https://mail.163.com/')

### **设置浏览器窗口**

**最大化显示**

browser.maximize\_window()

**最小化显示**

browser.minimize\_window()

**自定义大小**

# 宽 500，高 800

browser.set\_window\_size(500,800)

#### 浏览器刷新

在一些特殊情况下我们可能需要刷新页面来获取最新的页面数据，这时我们可以使用 refresh() 来刷新当前页面。

# 刷新页面

driver.refresh()

关于定位

##id 定位

标签的 id 具有唯一性

<input id="idname" autocomplete="off" type="text" value="" placeholder="C难在哪里？">

我们可以通过 id 定位到它

driver.find\_element\_by\_id("idname")

##name 定位

name 指定标签的名称，在页面中可以不唯一。假设有个 meta 标签如下

<meta name="keywords" content="CSDN博客,CSDN学院,CSDN论坛,CSDN直播">

我们可以使用 find\_element\_by\_name 定位到 meta 标签。

driver.find\_element\_by\_name("keywords")

##class 定位

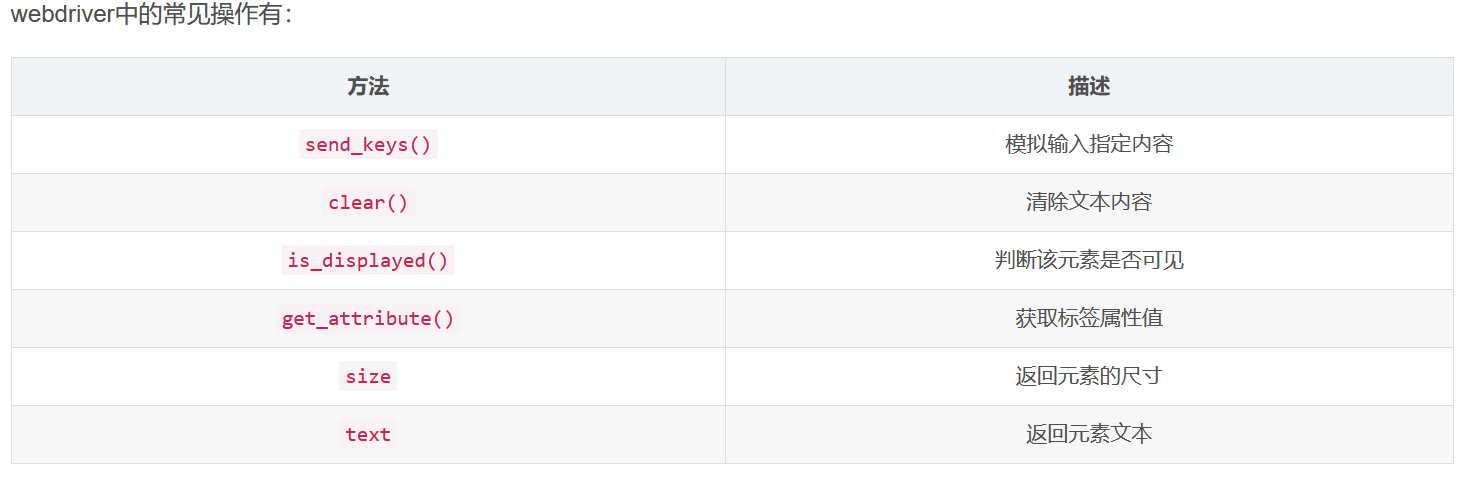
class 指定标签的类名，在页面中可以不唯一。假设有个 div 标签如下

<div class="toolbar-search-container">

我们可以使用 find\_element\_by\_class\_name 定位到 div 标签

driver.find\_element\_by\_class\_name("toolbar-search-container")

常见操作



关于一段报错：

Selenium里没有find\_element\_by\_id所以直接改用find\_element

注意要导包By

事例代码：

from selenium import webdriver  
from selenium.webdriver.common.by import By  
driver = webdriver.Firefox()  
driver.get('https://www.baidu.com/')driver.find\_element(by=By.ID,value='kw').send\_keys('test')  
driver.find\_element(by=By.ID,value='su').click()