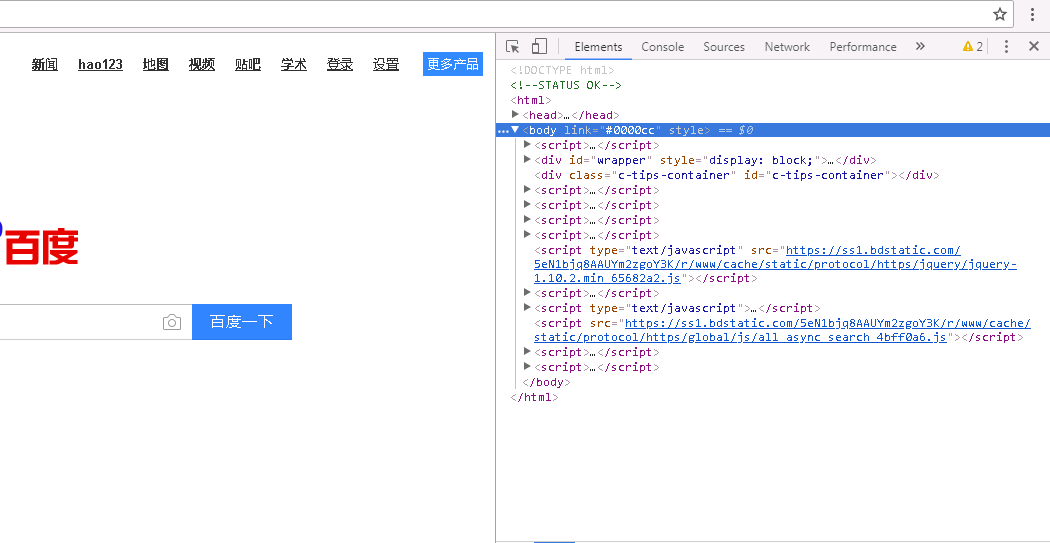
我们在做WEB自动化时，最根本的就是操作页面上的元素，首先我们要能找到这些元素，然后才能操作这些元素。工具或代码无法像我们测试人员一样用肉眼来分辨页面上的元素。那么我们怎么来定位他们呢？

在学习元素定位之前，我们最好能懂一点html的知识。

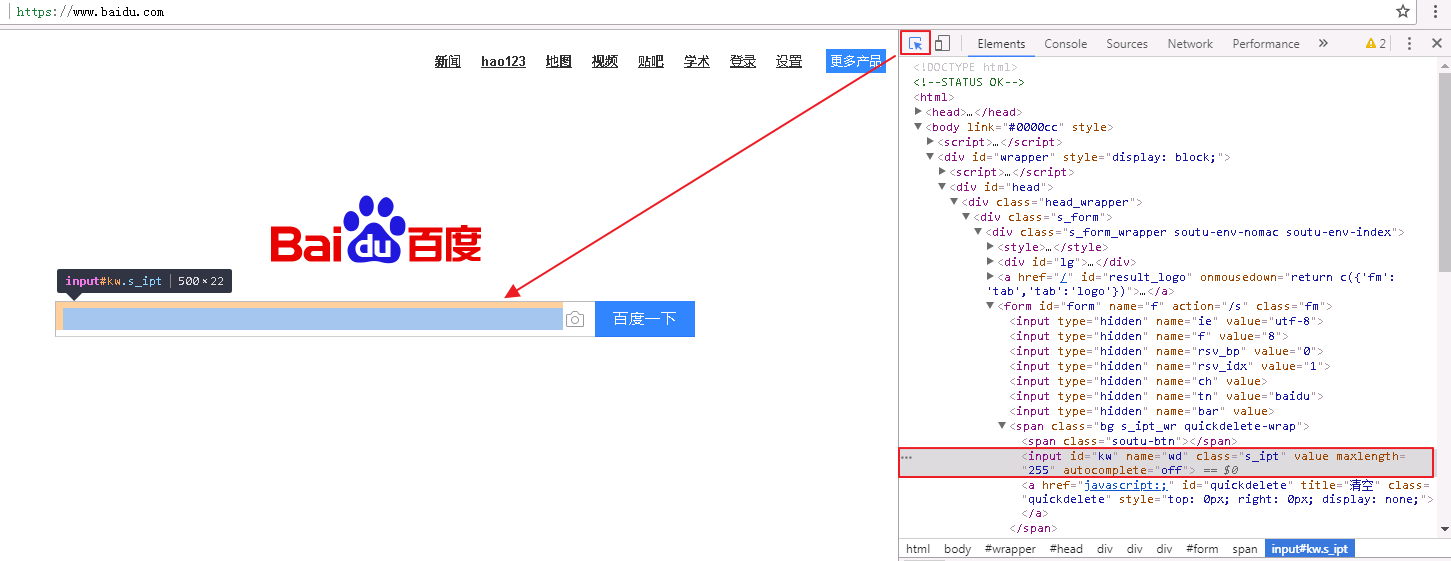
**一、查看页面元素**

用谷歌浏览器打开百度首页，点击右上角>更多工具>开发者工具，就可以看到整个页面的html代码了





点击框中左上角的箭头图标，移动鼠标到百度搜索框，就可以自动定位到百度搜索框的HTML代码了，查看到搜索框的属性，我们可以看到搜索框有id,name,class等属性。



**二、元素定位**

1、id定位： find\_element\_by\_id()

从上面定位到的搜索框属性中，有个id="kw"的属性，我们可以通过这个id定位到这个搜索框

 代码：

# coding = utf-8

from time import sleep

from selenium import webdriver

# 驱动文件路径

driverfile\_path = r'D:\coship\Test\_Framework\drivers\chromedriver.exe'

# 启动浏览器

driver = webdriver.Chrome(executable\_path=driverfile\_path)

# 打开百度首页

driver.get(r'https://www.baidu.com/')

# 通过id定位搜索框，并输入selenium

driver.find\_element\_by\_id('kw').send\_keys('selenium')

# 等待5秒

sleep(5)

# 退出

driver.quit()

2、name定位： find\_element\_by\_name()

从上面定位到的搜索框属性中，有个name="wd"的属性，我们可以通过这个name定位到这个搜索框

代码：

# coding = utf-8

from time import sleep

from selenium import webdriver

# 驱动文件路径

driverfile\_path = r'D:\coship\Test\_Framework\drivers\chromedriver.exe'

# 启动浏览器

driver = webdriver.Chrome(executable\_path=driverfile\_path

)# 打开百度首页

driver.get(r'https://www.baidu.com/')

# 通过name定位搜索框，并输入selenium

driver.find\_element\_by\_name('wd').send\_keys('selenium')

# 等待5秒

sleep(5)

# 退出

driver.quit()

3、class定位：find\_element\_by\_class\_name()

从上面定位到的搜索框属性中，有个class="s\_ipt"的属性，我们可以通过这个class定位到这个搜索框

代码：

# coding = utf-8

from time import sleep

from selenium import webdriver

# 驱动文件路径

driverfile\_path = r'D:\coship\Test\_Framework\drivers\chromedriver.exe'

# 启动浏览器

driver = webdriver.Chrome(executable\_path=driverfile\_path)

# 打开百度首页

driver.get(r'https://www.baidu.com/')

# 通过class定位搜索框，并输入selenium

driver.find\_element\_by\_class\_name('s\_ipt').send\_keys('selenium')

# 等待5秒

sleep(5)

# 退出

driver.quit()

4、tag定位：find\_element\_by\_tag\_name()

如果懂HTML知识，我们就知道HTML是通过tag来定义功能的,比如input是输入，table是表格，等等...。每个元素其实就是一个tag，一个tag往往用来定义一类功能，我们查看百度首页的html代码，可以看到有很多div,input,a等tag,所以很难通过tag去区分不同的元素。基本上在我们工作中用不到这种定义方法，仅了解就行。下面代码仅做参考，运行时必定报错

代码：

# coding = utf-8

from time import sleep

from selenium import webdriver

# 驱动文件路径

driverfile\_path = r'D:\coship\Test\_Framework\drivers\chromedriver.exe'

# 启动浏览器

driver = webdriver.Chrome(executable\_path=driverfile\_path)

# 打开百度首页

driver.get(r'https://www.baidu.com/')

# 通过tag定位搜索框，并输入selenium, 此处必报错driver.find\_element\_by\_tag\_name('input').send\_keys('selenium')

# 等待5秒

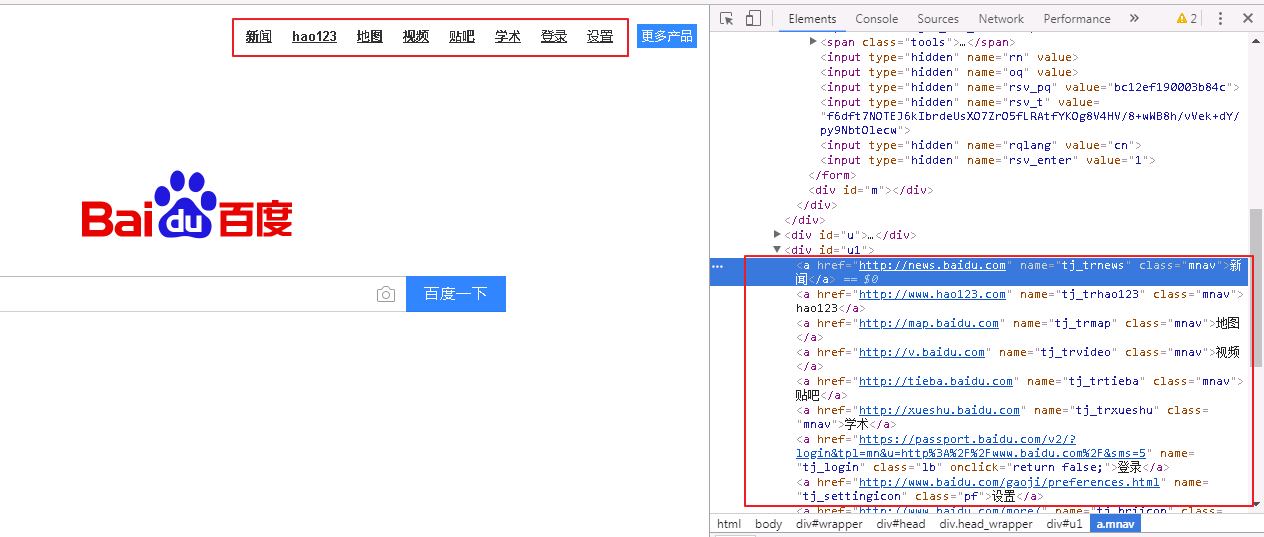
sleep(5)

# 退出

driver.quit()

5、link定位：find\_element\_by\_link\_text()

此种方法是专门用来定位文本链接的，比如百度首页右上角有“新闻”，“hao123”，“地图”等链接



我们来定位“新闻”这个链接元素

代码：

# coding = utf-8

from time import sleep

from selenium import webdriver

# 驱动文件路径

driverfile\_path = r'D:\coship\Test\_Framework\drivers\chromedriver.exe'

# 启动浏览器

driver = webdriver.Chrome(executable\_path=driverfile\_path)

# 打开百度首页

driver.get(r'https://www.baidu.com/')

# 通过link定位"新闻"这个链接并点击

driver.find\_element\_by\_link\_text('新闻').click()

# 等待5秒

sleep(5)

# 退出

driver.quit()

6、partial\_link定位：find\_element\_by\_partial\_link\_text()

有时候一个超链接的文本很长很长，我们如果全部输入，既麻烦，又显得代码很不美观，这时候我们就可以只截取一部分字符串，用这种方法模糊匹配了。

我们用这种方法来定位百度首页的“新闻”超链接

# coding = utf-8

from time import sleep

from selenium import webdriver

# 驱动文件路径

driverfile\_path = r'D:\coship\Test\_Framework\drivers\chromedriver.exe'

# 启动浏览器

driver = webdriver.Chrome(executable\_path=driverfile\_path)

# 打开百度首页

driver.get(r'https://www.baidu.com/')

# 通过partial\_link定位"新闻"这个链接并点击driver.find\_element\_by\_partial\_link\_text('闻').click()

# 等待5秒

sleep(5)

# 退出

driver.quit()

7、xpath定位：find\_element\_by\_xpath()

前面介绍的几种定位方法都是在理想状态下，有一定使用范围的，那就是：在当前页面中，每个元素都有一个唯一的id或name或class或超链接文本的属性，那么我们就可以通过这个唯一的属性值来定位他们。

但是在实际工作中并非有这么美好，有时候我们要定位的元素并没有id,name,class属性，或者多个元素的这些属性值都相同，又或者刷新页面，这些属性值都会变化。那么这个时候我们就只能通过xpath或者CSS来定位了。

代码：

# coding = utf-8

from time import sleep

from selenium import webdriver

# 驱动文件路径

driverfile\_path = r'D:\coship\Test\_Framework\drivers\chromedriver.exe'

# 启动浏览器

driver = webdriver.Chrome(executable\_path=driverfile\_path)

# 打开百度首页

driver.get(r'https://www.baidu.com/')

# 通过xpath定位搜索框，并输入selenium

driver.find\_element\_by\_xpath("//\*[@id='kw']").send\_keys('selenium')

# 等待5秒

sleep(5)

# 退出driver.quit()

8、CSS定位：find\_element\_by\_css\_selector()

这种方法相对xpath要简洁些，定位速度也要快些，但是学习起来会比较难理解，这里只做下简单的介绍。

CSS定位百度搜索框

# coding = utf-8

from time import sleep

from selenium import webdriver

# 驱动文件路径

driverfile\_path = r'D:\coship\Test\_Framework\drivers\chromedriver.exe'

# 启动浏览器

driver = webdriver.Chrome(executable\_path=driverfile\_path)

# 打开百度首页

driver.get(r'https://www.baidu.com/')

# 通过CSS定位搜索框，并输入selenium

driver.find\_element\_by\_css\_selector('#kw').send\_keys('selenium')

# 等待5秒

sleep(5)

# 退出

driver.quit()