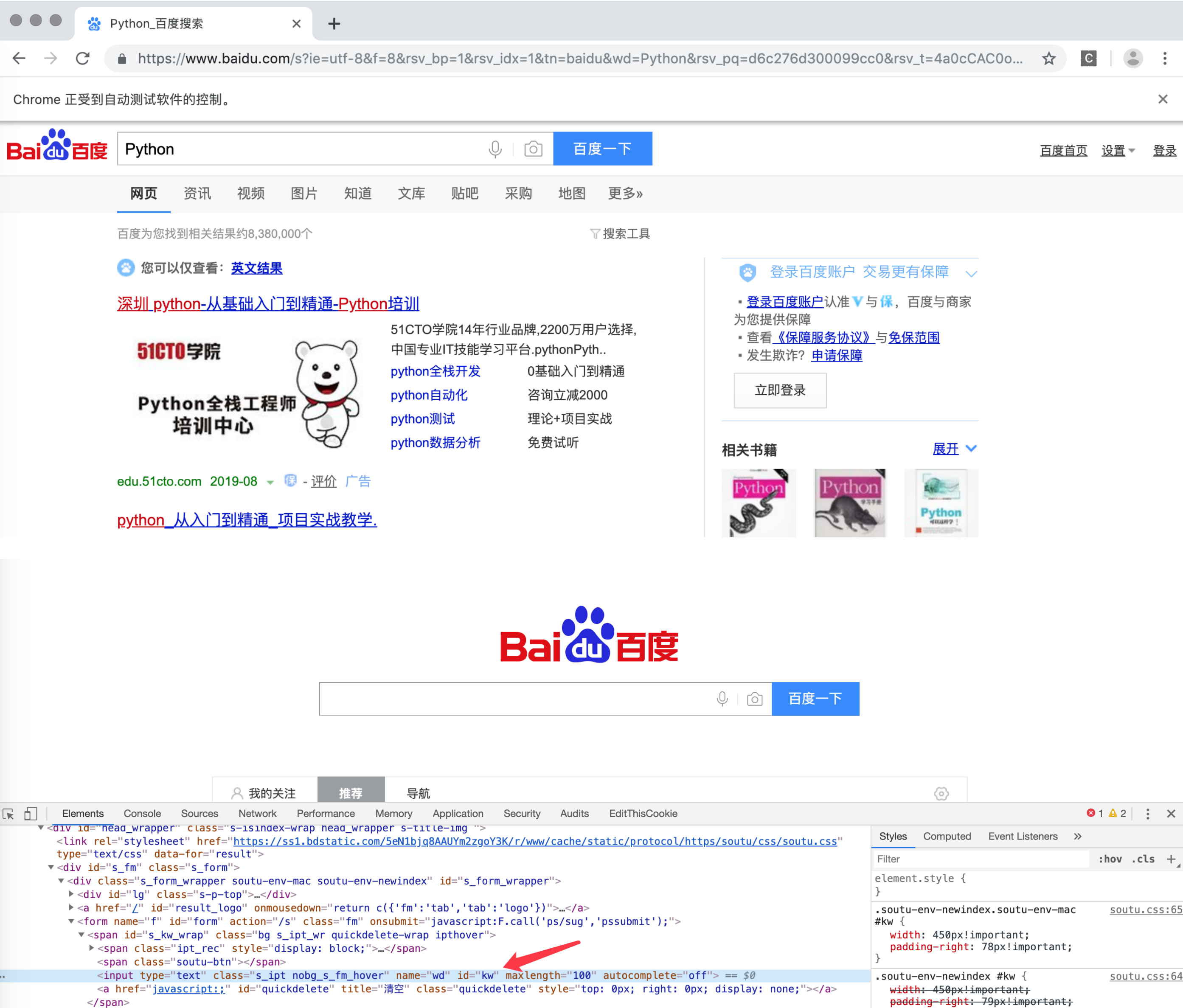


## # Selenium操作网页元素

一个简单的例子，通过selenium打开chrome浏览器，并输入关键字进行信息查询。

```
In [ ]: from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.support.wait import WebDriverWait
import time
browser=webdriver.Chrome()
try:
    browser.get("https://www.baidu.com") #打开百度首页
    input=browser.find_element_by_id("kw") #在首页中寻找一个id为kw的元素
    input.send_keys("Python") #在kw元素的文本框中输入Python
    input.send_keys(Keys.ENTER) #模拟点击回车按钮进行WEB请求
    wait=WebDriverWait(browser,10) #等待10秒时间
    wait.until(EC.presence_of_element_located((By.ID,"content_left"))) #等待页面中出现一个ID叫做content_left的元素，此时
    #搜索结果加载完成。
    print(browser.current_url)
    print(browser.get_cookies())
    print(browser.page_source)
    time.sleep(10)
finally:
    browser.close()
```

效果:



The image is a screenshot of a web browser window. At the top, the Baidu search engine is visible with the search term 'Python'. Below the search bar, there are navigation links like '网页', '资讯', '视频', etc. The search results show a link to 'Welcome to Python.org' with a '官网' (Official Site) tag and a '翻译此页' (Translate this page) button. The browser's developer tools are open at the bottom, showing the HTML structure. A red arrow points to the 'div#content\_left' element in the HTML. The right sidebar of the browser shows '相关书籍' (Related Books) with several Python-related book covers and titles. The overall layout is a typical search results page with a focus on the 'div#content\_left' element.

### 筛选单个元素：

```
In [ ]: from selenium import webdriver
from selenium.webdriver.common.by import By
browser=webdriver.Chrome()
browser.get("http://www.taobao.com")
input_first=browser.find_element_by_id("q")    ##淘宝搜索框的input标签ID配q
input_second=browser.find_element_by_css_selector("#q")    ##选择ID等于q的所有元素
input_third=browser.find_element(By.ID, "q")    ##选择ID为的所有元素
print(input_first)
print(input_second)
print(input_third)
browser.close()
```

### 筛选多个元素：

```
In [ ]: from selenium import webdriver
from selenium.webdriver.common.by import By
browser=webdriver.Chrome()
browser.get("http://www.taobao.com")
lis=browser.find_element_by_css_selector("li")
lis_c=browser.find_element(By.CSS_SELECTOR, "li")
print(lis)
print(lis_c)
browser.close()
```

In [ ]: 元素的交互操作:

```
In [ ]: from selenium import webdriver
from selenium.webdriver import ActionChains
import time
from selenium.webdriver.common.alert import Alert
browser=webdriver.Chrome()
url="https://www.runoob.com/try/try.php?filename=jqueryui-api-droppable"
browser.get(url)
#切换到目标元素所在的frame
browser.switch_to.frame("iframeResult")
#确定拖拽目标的起点
source=browser.find_element_by_id("draggable")
#确定拖拽目标的终点
target=browser.find_element_by_id("droppable")
#形成动作链
actions=ActionChains(browser)
actions.drag_and_drop(source,target)
#执行
actions.perform()
```

In [ ]: 执行javascript, 将滚动条拖至底部, 并执行一句JS代码.

```
In [ ]: from selenium import webdriver
browser=webdriver.Chrome()
browser.get("https://www.zhihu.com/explore")
browser.execute_script("window.scrollTo(0,document.body.scrollHeight)")
browser.execute_script("alert('To Button')")
browser.close()
```

In [ ]: Cookie处理

```
In [ ]: from selenium import webdriver
import time

browser=webdriver.Chrome()
browser.get("https://www.zhihu.com/explore")
print(browser.get_cookies())
browser.add_cookie({"name": "name", "domain": "www.zhihu.com", "value": "germey"})
print(browser.get_cookies())
browser.delete_all_cookies()
print(browser.get_cookies())
browser.close()
```

异常处理:

```
In [ ]: from selenium import webdriver
from selenium.common.exceptions import TimeoutException, NoSuchElementException
browser=webdriver.Chrome()
try:
    browser.get("https://www.zhihu.com/explore")
except TimeoutException:
    print("Time out")
try:
    browser.find_element_by_id("hello")
except NoSuchElementException:
    print("No Element")
finally:
    browser.close()
```