

JavaScript 是一种脚本语言，有的场景需要使用 js 脚本注入辅助我们完成 Selenium 无法做到的事情。

当 webdriver 遇到无法完成的操作时，可以使用 JavaScript 来完成，webdriver 提供了 `execute_script()` 方法来调用 js 代码。

执行 js 有两种场景：

- 在页面上直接执行 js
- 在某个已经定位的元素上执行 js

执行js

Selenium 可以通过 `execute_script()` 来执行 JavaScript 脚本。

- `driver.execute_script()`：同步执行 JavaScript 在当前的窗口框架下
- js 脚本可以在浏览器的开发者工具 -> Console 中进行调试



js的返回结果

- 获取元素控件中的属性值



- 与 Selenium 结合，在代码中返回 js 结果
- Python 版本

```
1 # 获取网页性能的响应时间，js脚本中使用return代表返回获取的结果js = "return  
JSON.stringify(performance.timing);"driver.execute_script(js)
```

- Java 版本

```
1 // 获取网页性能的响应时间，js脚本中使用return代表返回获取的结果String js = "return  
JSON.stringify(performance.timing);"JavascriptExecutor j = (JavascriptExecutor)  
driver;j.executeScript(js);
```

arguments传参

执行 JavaScript 也可以通过传参的方式传入元素信息，还可以通过下面的方法点击被遮挡的元素。

例如：某个元素在实际的操作过程中被其他的元素遮挡，就可以使用 js 点击的方式

- Python 版本

```
1 element = driver.find_element(by, locator)#arguments[0]代表所传值element的第一个参数
  #click()代表js中的点击动作driver.execute_script("arguments[0].click();",element)
```

- Java 版本

```
1 WebElement element = driver.findElement(By);// arguments[0]代表所传值element的第一个参
  数// click()代表js中的点击动作JavascriptExecutor j = (JavascriptExecutor)
  driver;j.executeScript("arguments[0].click();", element);
```

自动化过程中的报错如下：

```
1 Other element would receive the click: <label class="custom-control-label"
  for="user_remember_me">...</label>
```

如果出现上面的错误，可以使用 js 的方式进行点击。

实战演示

以企业微信为例，使用 js 点击添加图片

<< 返回

发消息

已发送

素材库

文字

图文

图片

语音

视频

文件

+ 添加图片 共 0 条

搜索



Python 代码

```
1 #导入依赖from selenium import webdriverfrom selenium.webdriver.common.by import Byclass
TestWework:    def setup(self):        self.driver = webdriver.Chrome()        #隐式等
待            self.driver.implicitly_wait(10)    def test_upload(self):        #元素定位
            element_add = self.driver.find_element\                (By.CSS_SELECTOR,
            ".js_upload_file_selector")        #执行js代码        self.driver.execute_script\
            ("arguments[0].click();", element_add)
self.driver.find_element_by_id('js_upload_input').\
send_keys('D:\project\demo1\demo.png')    assert
len(self.driver.find_elements(By.CSS_SELECTOR,\                '.material_pic_list_item'))
== 1    def teardown(self):        self.driver.quit()
```