Selenium元素定位整理篇

本文档为阅读虫师自动化测试书籍以及自己探索的关于Selenium元素定位的整理文档，如有遗漏 ，请不吝补充，感谢！@邮箱：[1551577567@qq.com](mailto:1551577567@qq.com)

## 基本定位方法

单个元素

写法一

1. find\_element\_by\_name(**""**)
2. find\_element\_by\_id(**""**)
3. find\_element\_by\_class\_name(**""**)
4. find\_element\_by\_tag\_name(**""**)#标签类别
5. find\_element\_by\_link\_text(**""**)#针对link中的文本内容的完全匹配
6. find\_element\_by\_partial\_link\_text(**""**)#针对link中的文本内容的部分匹配
7. find\_element\_by\_xpath(**""**)#Xpath路径
8. find\_element\_by\_css\_selector(**""**)#css选择器

写法二

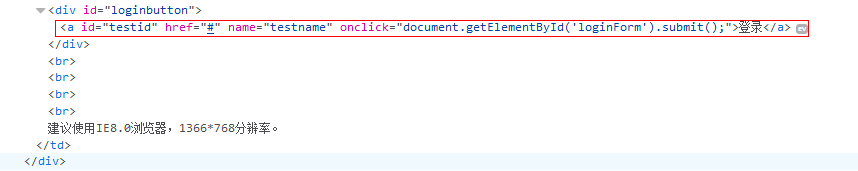
1. find\_element\_by\_name(**""**)
2. find\_element\_by\_id(**""**)
3. find\_element\_by\_class\_name(**""**)
4. find\_element\_by\_tag\_name(**""**)#标签类别
5. find\_element\_by\_link\_text(**""**)#针对link中的文本内容的完全匹配
6. find\_element\_by\_partial\_link\_text(**""**)#针对link中的文本内容的部分匹配
7. find\_element\_by\_xpath(**""**)#Xpath路径
8. find\_element\_by\_css\_selector(**""**)#css选择器

多个元素

1. find\_elements\_by\_name(**""**)
2. find\_elements\_by\_id(**""**)
3. find\_elements\_by\_class\_name(**""**)
4. find\_elements\_by\_tag\_name(**""**)#标签类别
5. find\_elements\_by\_link\_text(**""**)#针对link中的文本内容的完全匹配
6. find\_elements\_by\_partial\_link\_text(**""**)#针对link中的文本内容的部分匹配
7. find\_elements\_by\_xpath(**""**)#Xpath路径
8. find\_elements\_by\_css\_selector(**""**)#css选择器

实例运用

案例运用：对下面的登录链接按钮定位并点击（假设a标签通过文本内容”登录”，以及上级div通过id可以唯一定位）

 et：by\_name

driver.find\_element\_by\_name(**"**testname **"**).click()

et：by\_link

driver.find\_element\_by\_link\_text(**"**登录**"**).click()

et：by\_xpath(假设绝对路径是：/html/body/form/div/td/div/a)

xpath:

“/”表示根路径和转下一级,每个页面都是从根路径开始的，如下方实例”绝对路径”

“//”相对路径，当前元素向下查找子级元素，在路径最前端使用表示当前页面下查找如：**"//**div[@id=**"**loginbutton**"**]/a**"**，下方实例”相对路径”， “//”在xpath中间使用，表示以前面的路径下查找子级元素，例如：**"**/html/body**//**a **"**

“../”相对路径，表示上级目录下的路径，具体用法参见文档下方“父子兄弟相邻元素定位”，关于通过子元素定位父元素部分

[@attribute=**"**value**"**]，xpath中可以使用元素属性辅助定位，可以多属性同时使用，如下方实例”相对路径”和”属性”

绝对路径：

driver.find\_element\_by\_xpath(**"**/html/body/form/div/td/div/a **"**).click()

相对路径：

driver.find\_element\_by\_xpath(**"**//div[@id=**"**loginbutton**"**]/a**"**).click()

属性：

driver.find\_element\_by\_xpath(**"**//a[@id=’testid’ and @name=’testname’] **"**).click()

et：css选择器

|  |  |  |
| --- | --- | --- |
| 选择器 | 例子 | 描述 |
| .class | .info | 选择class=“info”的所有元素 |
| #id | #testid | 选择id=“ testid”的所有元素 |
| \* | \* | 选择所有元素 |
| element | a | 选择所有<a>标签元素 |
| element>element | div>a | 选择div标签下的a元素 |
| element+element | div+input | 选择同一级中紧接在<div>元素后的所有<input>元素 |
| [attribute=value] | a[id=’testid’] | 选择属性id=’testid’的a元素 |

css使用属性辅助定位同样可以多个属性同时使用

父子层级关系：

driver.find\_element\_by\_css\_selector(**"**div>a**"**).click()

属性：  
driver.find\_element\_by\_css\_selector(**"**[name=**'**testname**'**]**"**).click()

层级与属性结合：

driver.find\_element\_by\_css\_selector(**"**div>a#testid[name=**'**testname**'**] **"**).click()

拓展问题

1.通过任意标签元素之间的文本内容定位元素

使用1.2中的实例，我们不用link和partial link，通过文本内容”登录”定位

通过by\_xpath的text()方法完全匹配

driver.find\_element\_by\_xpath(**“//div[@id='loginbutton']/a[text()='登录']”**).click()

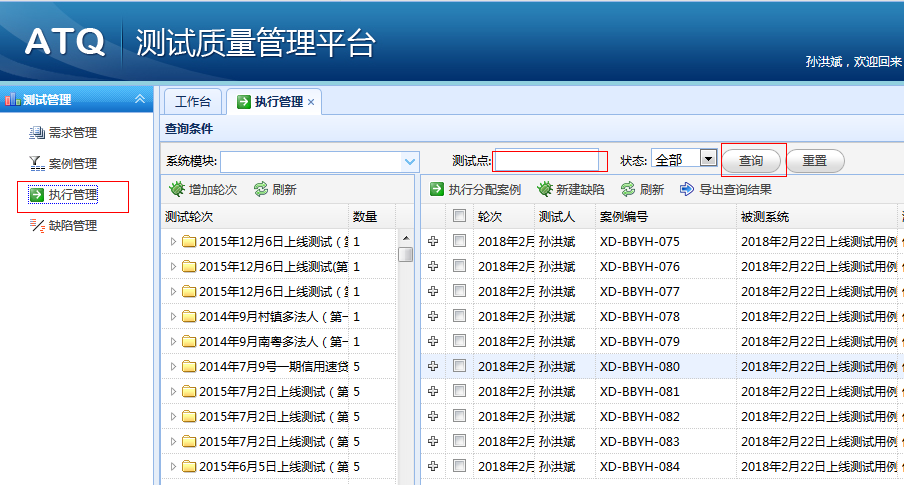
通过by\_xpath的text()方法部分匹配  
driver.find\_element\_by\_xpath(**“//div[@id='loginbutton']/a[contains(text(),'登录')]”**).click()

通过by\_css的textContent()方法完全匹配

driver.find\_element\_by\_css\_selector(**“div#loginbutton>a[textContent='登录']”**).click()*#这个是不行的，有空可以稍微研究一下*

2.页面嵌套多个frame/iframe，脚本运行无法识别对象的问题（此例只提供简单参考，后面有专门章节介绍）

如下图：selenium点击”执行管理”后，无法继续操作右下方frame中的元素，如标红：输入框、查询按钮等



经百度查询，发现问题在于selenium只能在一个页面上操作，对于嵌套的frame/iframe是不能处理的，需要借助switch\_to.frame()方法切换frame后才能操作，实现如下：

driver.find\_element\_by\_xpath(**“//a/span/span[contains(text(),'执行管理')]”**).click()  
driver.find\_element\_by\_xpath(**“/html/body/div[3]/div/div/div[2]/div[2]/div/iframe”**)  
driver.switch\_to.frame(xf)  
driver.find\_element\_by\_xpath(**“//span[contains(text(),'2015年12月6日上线测试（第一轮）')]”**).click()  
driver.find\_element\_by\_id(**“sysName”**).send\_keys(**“测试计划”**)  
driver.find\_element\_by\_xpath(**“//a[@id='']/span/span[text()='查询']”**).click()

3.弹出窗口怎么办（此例只提供简单参考，后面有专门章节介绍）

**JS定位及操作**

实例：

如图基本定位中的实例，在实际操作中遇到一个现象，定位元素click()操作，firefox上运行能够正常点击，但是ie上始终无法点击（selenium ie click失效），通过高亮显示在ie上又确实定位到了此元素，解决历程如下：

1. 尝试使用各个版本的Iedriverserver.exe;
2. 导入ActionChains，通过此类鼠标移动到对象双击

**from** selenium.webdriver.common.action\_chains **import** ActionChains ActionChains(driver).move\_to\_element(element).double\_click()

1. 通过JS操作执行点击

js= ’’’document.getElementById(**"**testid**"**).click()’’’

driver.execute\_script(js)

1. 降低浏览器版本（我用的是11）
2. 通过action和Keys发送Tab键和Enter键

上述方法，1、2不行，3解决问题，4、5因为一些原因没有完成尝试，猜测原因可能是浏览器版本过高或者iedriver及selenium存在一些缺陷，但是代码阅读能力较弱，暂不分析，基于上面解决问题的JS定位及操作，简单研究了一下JS的定位操作

JS中的document节点中的定位相关：

1. html和body

1>获取html：document.documentElement

2>获取body：document.body

1. 一般元素的获取

1> document.getElementById();

2> document.getElementsByName();

3> document.getElementsByTagName();

4> document.getElementsByClassName();

5> document.querySelector();获取单个元素，()中可以是css选择器的标准形式，和driver.find\_element\_by\_css\_selector()相类似的定位

6> document.querySelectorAll();获取单个元素集合，()中可以是css选择器的标准形式

附件：后面得空在此添加一个关于document的附件

JS中$x使用XPATH定位：

$x:浏览器控制台使用此方法可以根据XPATH定位元素，写法如下

$x("//div[@id='djDiag']/../following-sibling::div[4]/div/div[2]")

JS中$使用CSS定位：

$:浏览器控制台使用此方法可以根据CSS定位元素，写法如下

$("body")

JS对元素的click操作和赋值：

1. click()点击

js= ’’’document.getElementById(**"**testid**"**).click()’’’

driver.execute\_script(js)

1. value赋值

js= ’’’document.getElementById(**"sysName"**).value=“123”’’’

driver.execute\_script(js)

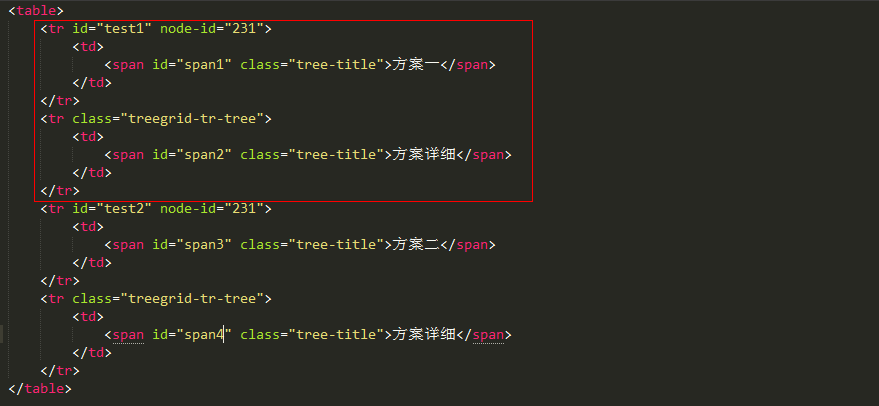
可以把JS的这两种对对象的操作方法及JS元素定位结合封装成一个JS操作元素的方法

附件：请编辑并附上封装后的JS操作方法文档

**父子兄弟相邻元素定位**

实例：

工作中遇到一个问题，存在多组类似标签代码如下图。我们试图通过方案名称，点击方案详细。如果通过find\_element\_by\_xpath(//span[text()=**"**方案详细**"**])定位元素，存在几个问题：存在多个text为”方案详细”的span元素、不知道对应的方案名称是哪个，而如果使用绝对的xpath路径，因为需要分别测试多个方案，实现脚本太多，而且不方便。故而需要如下顺序定位：**"方案名称"的span**>>**反定位上级tr**>>**相邻关系下方tr**>>**"方案详细"的span**



实现如下：

driver.find\_element\_by\_xpath(**"**//span[text()=’方案一’]/../../fllowing::tr[1]/td/span**"**)

**由父节点定位子节点**

上方实例，通过id=**"**test1**"**的tr定位id=**"**span1**"**的span元素

1. 串联查找

# firefox上尝试，这种串联查找不是以第一个查询为父单位，查找不正确，还有待研究，如果要解决父元素路径下使用相对路径查询子元素，xpath中间也能使用“//”

driver.find\_element\_by\_id(**"test1"**).find\_element\_by\_xpath(**"//span[text()='方案一']"**)

1. xpath父子关系

driver.find\_element\_by\_xpath(**"//tr[@id='test1']/td/span"**)

1. css selector父子关系

driver.find\_element\_by\_css\_selector(**"tr#test1>td>span"**)

1. css selector nth-child

# **span:nth-child(1)** 表示上级td元素下的第1个元素，元素名称为span

driver.find\_element\_by\_css\_selector(**"tr#test1 td:nth-child(1) span:nth-child(1) "**)

1. css selector nth-of-type

# **span:nth-of-type(1)** 表示上级td元素下的第一个span元素

driver.find\_element\_by\_css\_selector(**"tr#test1 td:nth-of-type(1) span:nth-of-type(1) "**)

1. xpath轴 child

#child是xpath默认的轴，可以忽略

driver.find\_element\_by\_xpath(**"//tr[@id='test1']/td /child::span"**)

第1到第三个是我们比较熟悉的方法，4、5的区别在于一个是子节点的第n个元素，一个是同一标签的元素的第n个元素，第6个，xpath都有使用，不用写每个节点默认child：：

当然css中还有一些选择器可以选择父子关系，如**last-child**、**nth-last-child**等，感兴趣的可以百度一下

**由子节点定位父节点**

上方实例，通过方案一的span元素反定位id=**"**test1**"**的祖父元素

1. xpath:”.”代表当前节点；”..”代表父节点

find\_element\_by\_xpath(**"//span[text()='方案一']/../.."**)

1. xpath轴 parent

driver.find\_element\_by\_xpath(**"//span[text()='方案一']/parent::\*/parent::\*"**)

目前只有上述两种方法定位父节点，css暂时无法定位父节点

**由弟弟节点定位哥哥节点**

上方实例，通过id=**"**test2**"**的tr元素，定位id=**"**test1**"**的哥哥tr元素

1. 通过父元素定位

driver.find\_element\_by\_xpath(**"//tr[@id='test2']/../tr[1]"**)

1. xpath轴 preceding-sibling

# **preceding-sibling::tr[n]** 表示同级节点向上（哥哥节点）的第n个节点元素

driver.find\_element\_by\_xpath(**"//tr[@id='test2']/preceding-sibling::tr[2]"**)

**xpath轴preceding-sibling**，能够获取当前节点的所有同级哥哥节点，注意标号[]中的数字，1表示最近的一个哥哥节点，数字越大表示离当前节点越远。**xpath轴preceding**也能使用但是较为麻烦，不研究。

**由哥哥节点定位弟弟节点**

上方实例，通过id=**"**test1**"**的tr元素，定位id=**"**test2**"**的弟弟tr元素

1. 通过父元素定位

driver.find\_element\_by\_xpath(**"//tr[@id='test1']/../tr[3]"**)

1. xpath轴 following-sibling

#**following-sibling::tr[n]** 表示同级节点向下（弟弟节点）的第n个节点元素

driver.find\_element\_by\_xpath(**"//tr[@id='test1']/following-sibling::tr[2]"**)

**xpath轴following -sibling**和**xpath轴preceding-sibling**用法相似。其他如**css selector +、~**也能使用写法如：

driver.find\_element\_by\_css\_selector(**"tr#test1 + tr "**)表示紧跟当前节点的tr

driver.find\_element\_by\_css\_selector(**"tr#test1 ~ tr "**)表示当前节点后的tr，如果使用find\_elements表示一组tr元素

**xpath轴following**同样可以定位，暂时不研究。

**弹出窗口元素定位**