**为什么要学习获取元素信息的方法？**

**主要为了**获取相关的信息进行**断言**，判断自动化用例最终的执行结果

**获取元素信息的常用方法**

size 获取元素的大小，返回的是一个字典，里面包含元素高度和宽度的值

text 获取元素的文本

get\_attribute(“xxx”) 获取属性值，传递的参数为元素的属性名

is\_displayed() 判断是否可见（返回结果为true或者false）

is\_enabled() 判断元素是否可见(返回值为true或者false)

is\_selected() 判断元素是否选中，用来检查复选框或单选按钮是否被选中

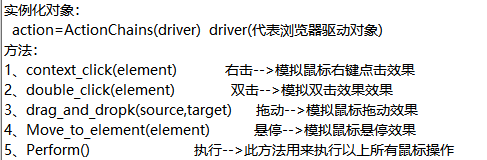
提示：

Size、text：为属性，调用时无括号：如xxx.size

**鼠标操作和键盘操作**

selenium提供鼠标操作的方法及步骤

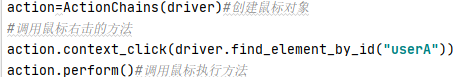
1. 通过ActionChains实例化鼠标对象 action=ActionChains(driver) driver表示的是浏览器驱动对象
2. 调用鼠标的事件方法
3. 调用鼠标的执行方法 action.perform()



**鼠标右击操作**

右击操作的实现步骤：

1. 创建鼠标对象
2. 调用右击事件方法
3. 调用鼠标执行方法



**鼠标双击操作**

鼠标双击的实现步骤

1. 创建鼠标对象 action=ActionChains(driver)
2. 调用鼠标双击事件方法action.double\_click(element)#element表示的是元素对象
3. 调用鼠标的执行方法 action.perform()



**鼠标拖动操作**

鼠标拖动的实现步骤：

1. 创建鼠标对象 action=ActionChains(driver)
2. 调用鼠标拖动事件方法 action.drag\_and\_drop(source,target)#source源元素，被拖动的元素,target表示的是目标源，也就是要拖动到哪个元素上
3. 调用鼠标执行方法 action.perform()



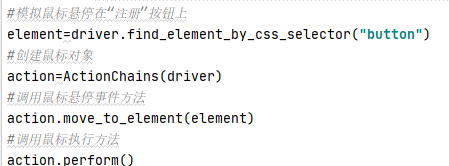
**鼠标悬停操作**

鼠标悬停实现步骤：

1、实例化鼠标对象 action=ActionChains(driver)

2、调用鼠标悬停事件方法 action.move.to\_element()#element表示的是元素对象

3、调用鼠标执行方法 action.perform()

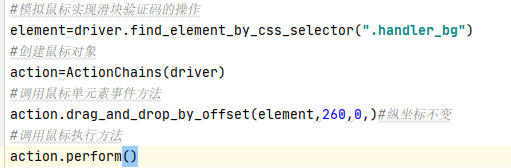


**鼠标单元素拖动操作**

鼠标单元素拖动实现步骤

1. 实例化鼠标对象 action=ActionChains(driver)
2. 调用鼠标单元素拖动事件方法 action.drag\_and\_drop\_by\_offset(element,x,y) x,y 表示的是元素拖动时横向和纵向移动的距离，单位为像素,element表示的是元素对象,移动的像素最终要比在web页面中看到的移动像素值要大，最好大于5个像素或者10像素
3. 调用鼠标执行方法 action.perform()





**弹出框处理方法**

1. 获取弹出框对象

driver.switch\_to.alert

1. 处理弹出框

alert.text获取弹出框提示信息

alert,accept()确认弹出框

alert.dismiss()取消弹出框