

1.常用简单操作

1.1 click() # (鼠标左键)点击页面按钮

1.2 clear() # 清空输入框

1.3 send_keys() # 输入字符串

```
driver.find_element_by_id('kw').send_keys(u'中文输入')
```

windows系统输入中文需要在输入内容前添加‘u’,例： u’中文输入’。(windows系统是GBK编码，测试脚本是utf-8，所以需要将输入内容转换为Unicode编码后才能识别。

1.4 submit() # 模拟回车键输入，如下代码模拟百度搜索框输入内容后点击回车搜索。

```
ele = driver.find_element_by_id('kw').send_keys('吃货的终极目标');
ele.submit()
```

2.模拟键盘操作

selenium提供了模拟键盘操作的支持。模拟键盘操作需要先导入Keys模块, 具体的参数可查看Keys类了解。

2.1 导入Keys模块后，通过Keys.方式调用。

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys # 导入Keys模块

driver = webdriver.Chrome()
driver.get('http://www.baidu.com')
ele = driver.find_element(By.ID, value='kw')
ele.send_keys('如何成为吃货')
ele.send_keys(Keys.ENTER) #模拟回车键输入
```

2.2 常用的键盘操作

```
send_keys(Keys.F1) # 模拟键盘F1到F12
send_keys(Keys.CONTROL,'c') # 复制Ctrl+C
send_keys(Keys.CONTROL,'v') # 粘贴Ctrl+V
send_keys(Keys.CONTROL,'a') # 全选Ctrl+A
send_keys(Keys.CONTROL,'x') #剪切Ctrl+X
send_keys(Keys.TAB) # 制表键Tab
```

3.模拟鼠标操作

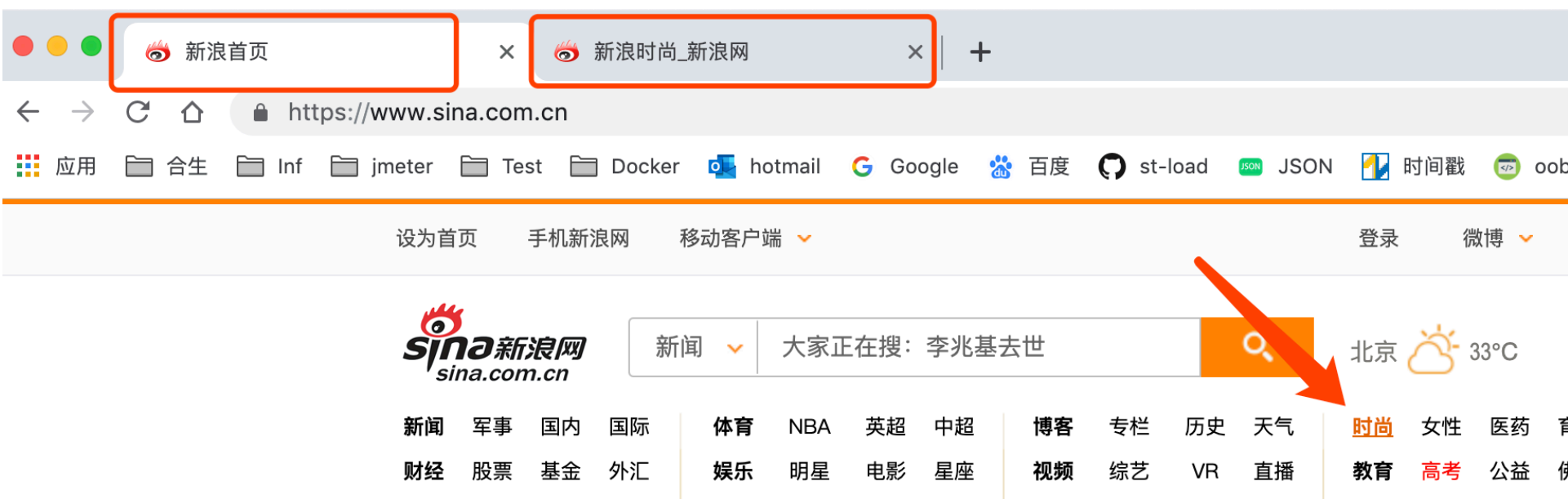
常用的鼠标事件除click()外，还有如悬停、右击、按住拖动等。上述操作都可以通过导入ActionChains模块来实现。perfrom()方法调用后会执行所有ActionChains中定义的行为。我们以百度首页的’设置’为例，设置选项需要鼠标悬停才会显示。

```
from selenium import webdriver
from selenium.webdriver.common.action_chains import ActionChains

driver = webdriver.Chrome()
driver.get('http://www.baidu.com')
button_setting = driver.find_element_by_link_text('设置')
ActionChains(driver).move_to_element(button_setting).perform() # 实现鼠标悬停操作
ActionChains(driver).context_click().perform() # 实现点击鼠标右键
ActionChains(driver).double_click() # 实现鼠标双击
```

4.句柄(handel)、多窗口操作

有些网页内的链接点击后，会重新打开一个新的窗口，如新浪。如果想要操作新页面内的元素，就需要先切换窗口后才能执行操作了。浏览器的每一个窗口都有一个唯一的句柄标识，selenium可以通过切换窗口句柄来实现窗口切换的目的。



4.1 通过driver.current_windwo_handle获取窗口句柄

执行如下代码断，打开新浪首页后点击时尚页，发现新打开了一个页面。如图所示，此时包含了两个窗口句柄。

```
from selenium import webdriver
from time import sleep
|
driver = webdriver.Chrome()
driver.get('http://www.sina.com')
button_fashion = driver.find_element_by_css_selector('body > div.main > '
                                                    'div.main-nav > div:nth-child(4) > '
                                                    'ul:nth-child(1) > li:nth-child(1) > a > b')

button_fashion.click()
cur_handle = driver.current_window_handle # 获取当前窗口句柄
all_handels = driver.window_handles # 获取所有窗口句柄
print(cur_handle)
print(all_handels)
```

```
/Users/rasule/PvcharmProjects/live-server/venv/bin/python /Users/rasule/PvcharmProjects/webtest/cases/test1.py
CDwindow-0C4418E7F3A594020C0BFDE500E20034
['CDwindow-0C4418E7F3A594020C0BFDE500E20034', 'CDwindow-0665466723C430A2C57F249CA64CCE2B']
```

4.2 句柄操作

获取当前浏览器的窗口句柄后，我们可以执行切换窗口、关闭窗口等操作。

```
all_handels = driver.windwo_hanles # 返回当前所有的窗口句柄
driver.switch_to.window(all_handels[0]) # 通过切换句柄进入第1个窗口
driver.switch_to.window(all_handels[1]) # 通过切换句柄回到第2个窗口
driver.close() # 关闭当前窗口
driver.quit() # 退个整个进程（关闭所有打开的窗口）
```