

## 鼠标操作的其他操作

```
1  需要引入
2  from selenium.webdriver.common.action_chains import ActionChains
3  格式:
4  ActionChains(浏览器).context_click(元素).perform()
5  1) 右键单击:context_click()
6  2) 鼠标悬停:move_to_element()
7  #!/usr/bin/env python
8  # -*- coding:utf-8 -*-
9  #====#====#====#====
10 #Author:
11 #CreatDate:
12 #Version:
13 #====#====#====#====
14 from selenium.webdriver.common.action_chains import ActionChains
15 from selenium import webdriver
16 import time
17
18 dr=webdriver.Firefox()
19 dr.get("https://www.baidu.com")
20 time.sleep(2)
21
22 e=dr.find_element_by_link_text('新闻')
23 #执行鼠标的动作,右击
24 ActionChains(dr).context_click(e).perform()
25 time.sleep(3)
26
27
28 e2=dr.find_element_by_id('s-usersetting-top')
29 #执行鼠标的动作,悬停
30 ActionChains(dr).move_to_element(e2).perform()
31 time.sleep(5)
32 dr.quit()
33
34 3) 双击:double_click()
35 <!DOCTYPE html>
36 <html lang="en">
37 <head>
38     <meta charset="UTF-8">
39     <title>Title</title>
40     <script language="JavaScript">
41         function test(){
42             alert("我被双击了");
43         }
44     </script>
45 </head>
46 <body>
```

```

47     <form>
48         <input type="button" value="按键" onclick="test()">
49     </form>
50 </body>
51 </html>
52
53 #!/usr/bin/env python
54 # -*- coding:utf-8 -*-
55 #====#====#====#====
56 #Author:
57 #CreatDate:
58 #Version:
59 #====#====#====#====
60 from selenium.webdriver.common.action_chains import ActionChains
61 from selenium import webdriver
62 import time
63 import os
64
65 dr=webdriver.Firefox()
66 dr.get("file:///"+os.path.abspath("双击案例.html"))
67 time.sleep(2)
68 e=dr.find_element_by_tag_name('input')
69 ActionChains(dr).double_click(e).perform()
70 time.sleep(3)
71 dr.quit()
72
73 4) 拖动:drag_and_drop()  #后面讲
74
75

```

## 下拉列表的操作

```

1  1. 下拉列表是非常常见的页面元素。是一种特殊的页面元素，定位和其他元素没区
2  别，但是其操作不其他元素不同。
3  2. 需要导入包
4  from selenium.webdriver.support.select import Select
5  3. 选择下列列表中的元素有三种方式
6  • select_by_index() #索引
7  • select_by_visible_text()#文本
8  • select_by_value()#value属性的值
9
10 <!DOCTYPE html>
11 <html lang="en">
12 <head>
13     <meta charset="UTF-8">
14     <title>Title</title>
15 </head>
16 <body>
17     <form>
18         <select name="sel">
19             <option value="sh">生蚝</option>
20             <option selected="selected">小龙虾</option>

```

```

21         <option>北京烤鸭</option>
22         <option>烤羊肉</option>
23     </select>
24 </form>
25 </body>
26 </html>
27
28
29 #!/usr/bin/env python
30 # -*- coding:utf-8 -*-
31 #====#====#====#====
32 #Author:
33 #CreatDate:
34 #Version:
35 #====#====#====#====
36 from selenium.webdriver.support.select import Select
37 from selenium import webdriver
38 import time
39 import os
40
41 dr=webdriver.Firefox()
42 dr.get("file:///"+os.path.abspath("下拉列表.html"))
43 time.sleep(2)
44 #获取下拉列表元素
45 e=dr.find_element_by_name('sel')
46 #把元素转换为Select对象
47 sel_obj=Select(e)
48 sel_obj.select_by_index(3)#选择烤羊肉
49 time.sleep(2)
50 sel_obj.select_by_visible_text("北京烤鸭")
51 time.sleep(2)
52 sel_obj.select_by_value('sh')#生蚝
53 time.sleep(3)
54 dr.quit()
55

```

## 键盘操作

- 1 • keys()类提供了键盘上几乎所有按键的方法。
- 2 • send\_keys()方法可以用来模拟键盘输入。
- 3 • 还可以用它来输入键盘上的按键，甚至是组合键，如 Ctrl+A、Ctrl+C 等。
- 4 • 在使用键盘按键方法前需要先导入 keys 类。
- 5 • from selenium.webdriver.common.keys import Keys
- 6 常用的键盘操作：
- 7 • send\_keys(Keys.BACK\_SPACE) 删除键 (BackSpace)
- 8 • send\_keys(Keys.TAB) 制表键(Tab) (不常用)
- 9 • send\_keys(Keys.ESCAPE) esc键 (Esc)
- 10 • send\_keys(Keys.ENTER) 回车键 (Enter)
- 11 • send\_keys(Keys.CONTROL, 'a') 全选 (Ctrl+A)
- 12 • send\_keys(Keys.CONTROL, 'c') 复制 (Ctrl+C)

```

13 • send_keys(Keys.CONTROL, 'x') 剪切 (Ctrl+x)
14 • send_keys(Keys.CONTROL, 'v') 粘贴 (Ctrl+v)
15 #!/usr/bin/env python
16 # -*- coding:utf-8 -*-
17 #====#====#====#====
18 #Author:
19 #CreatDate:
20 #Version:
21 #====#====#====#====
22 from selenium.webdriver.common.keys import Keys
23 from selenium import webdriver
24 import time
25
26 dr=webdriver.Firefox()
27 dr.get("https://www.baidu.com")
28 time.sleep(2)
29 #在搜索框中输入软件测试,然后删除"试",然后全选,再剪切,再粘贴回去
30 e=dr.find_element_by_id('kw')
31 e.send_keys("软件测试")
32 time.sleep(2)
33 e.send_keys(Keys.BACK_SPACE)#删除"试"
34 time.sleep(2)
35 e.send_keys(Keys.CONTROL, 'a')#全选
36 time.sleep(2)
37 e.send_keys(Keys.CONTROL, 'x')#剪切
38 time.sleep(2)
39 e.send_keys(Keys.CONTROL, 'v')#粘贴
40 time.sleep(3)
41 dr.quit()
42
43
44
45
46
47 #案例:在百度搜索栏中,进行输入,删除,全选,剪切,粘贴功能
48
49

```

## 浏览器控制

```

1 • 浏览器窗口大小的控制
2 dr.set_window_size(宽,高)
3 dr.maximize_window():最大化显示
4 dr.minimize_window():最小化显示,在最小化情况下,也可以进行元素定位及操作
5
6 • 页面前进与后退
7 back()和forward()方法来模拟后退和前进按钮
8
9 #!/usr/bin/env python
10 # -*- coding:utf-8 -*-
11 #====#====#====#====
12 #Author:

```

```

13 #CreatDate:
14 #Version:
15 #====#====#====#====
16 from selenium import webdriver
17 import time
18
19 dr=webdriver.Firefox()
20 dr.get("https://www.baidu.com")
21 time.sleep(2)
22 #设置浏览器的大小
23 dr.set_window_size(400,800)
24 time.sleep(2)
25 #设置浏览器最小化
26 dr.minimize_window()
27 time.sleep(2)
28 #设置浏览器最大化
29 dr.maximize_window()
30 time.sleep(2)
31 #转换到淘宝首页
32 dr.get("https://www.taobao.com")
33 time.sleep(2)
34 #后退到百度
35 dr.back()
36 time.sleep(2)
37 #前进到淘宝
38 dr.forward()
39 time.sleep(3)
40 dr.quit()
41
42 #案例:进入淘宝,然后设置浏览器的大小我400,500,然后最大,最小,再最大,转入到百度,然后后退,再前进
43
44
45 • 页面刷新
46 driver.refresh()
47
48
49 • 针对浏览器窗口页面截图
50 get_screenshot_as_file(保存图片的位置)
51
52 • 浏览窗口的关闭和浏览器退出
53 使用quit()方法,其含义为退出相关的驱动程序和关闭所有窗口。
54 使用close()方法,关闭单个窗口,关闭最前面的窗口
55 #!/usr/bin/env python
56 # -*- coding:utf-8 -*-
57 #====#====#====#====
58 #Author:
59 #CreatDate:
60 #Version:
61 #====#====#====#====
62 from selenium import webdriver
63 import time
64
65 dr=webdriver.Firefox()

```

```

66 dr.get("https://www.baidu.com")
67 time.sleep(2)
68 #刷新
69 dr.refresh()
70 time.sleep(2)
71 #给页面进行拍照
72 dr.get_screenshot_as_file("1.png")
73 time.sleep(2)
74 dr.find_element_by_link_text('新闻').click()
75 time.sleep(2)
76 dr.close()#关闭百度首页
77 time.sleep(3)
78 dr.quit()
79
80
81
82
83 #案例:打开淘宝,刷新页面,点击天猫超时,然后截图,然后关闭淘宝首页,,最后退出浏览器
84

```

## 多窗口处理

```

1  多窗口的意思是一个浏览器中打开了多个窗口
2
3  • 浏览器标签页的切换
4      • 浏览器的标签页在应用中一般称为页面句柄 ( handle )。
5      • 获取当前窗口句柄 : driver.current_window_handle
6      • 获取浏览器所有句柄 : driver.window_handles
7      • 切换到指定的浏览器窗口 : driver.switch_to.window(handle)
8
9  注意:
10  关闭页面需要切换到页面中,获取操控页面的元素也需要切换到该页面
11  关闭其中某个页面后,需要重新获取当前浏览器所有的句柄
12
13  #!/usr/bin/env python
14  # -*- coding:utf-8 -*-
15  #====#====#====#====
16  #Author:
17  #CreatDate:
18  #Version:
19  #====#====#====#====
20  from selenium import webdriver
21  import time
22
23  def mytest01():
24      dr=webdriver.Firefox()
25      dr.get("https://www.baidu.com")
26      time.sleep(2)
27      dr.find_element_by_link_text('新闻').click()
28      time.sleep(2)
29      dr.find_element_by_link_text('hao123').click()
30      time.sleep(2)

```

```
31     #获取浏览器中所有窗口的句柄
32     mylist=dr.window_handles
33     print(mylist)
34     #['1478f47f-4d7a-4db3-9335-87ee8631656a', 'd88dba90-3af1-4e4e-8ccb-
44a96619badc', '45a8db54-5bdf-443d-9580-fbd85c2ad210']
35     #列表第一个元素是百度首页的句柄,第二个元素是hao123窗口的句柄,第三个元素是新闻的句柄
36     dr.switch_to.window(mylist[1])#切换到hao123窗口
37     time.sleep(2)
38     dr.close()#关闭窗口
39     mylist = dr.window_handles
40     print(mylist)
41     #['84f322f5-bd2a-46c0-89af-05efee8e4738', '3f12b771-90b0-4ee6-8655-
d7cf78883d2c']
42
43     time.sleep(3)
44     dr.quit()
45
46
47 # mytest01()
48
49 #进入到百度首页,然后打开新闻,hao123,地图,视频,然后关闭的顺序是地图,新闻,视频,hao123,最后退出浏
览器
50 def mytest02():
51     dr=webdriver.Firefox()
52     dr.get("https://www.baidu.com")
53     time.sleep(2)
54
55     dr.find_element_by_link_text('新闻').click()
56     time.sleep(2)
57     dr.find_element_by_link_text('hao123').click()
58     time.sleep(2)
59     dr.find_element_by_link_text('地图').click()
60     time.sleep(2)
61     dr.find_element_by_link_text('视频').click()
62     mylist=dr.window_handles
63     print(mylist)
64     # [百度首页,视频,地图,hao123,新闻]
65     #切换到地图窗口
66     dr.switch_to.window(mylist[2])
67     dr.close()#关闭地图
68     time.sleep(2)
69     mylist = dr.window_handles
70     print(mylist)
71     # [百度首页,视频,hao123,新闻]
72     #切换到新闻窗口
73     dr.switch_to.window(mylist[3])
74     dr.close()#关闭新闻
75     time.sleep(2)
76     mylist = dr.window_handles
77     print(mylist)
78     # [百度首页,视频,hao123]
79     # 切换到视频窗口
80     dr.switch_to.window(mylist[1])
```

```

81     dr.close() # 关闭视频
82     time.sleep(2)
83     mylist = dr.window_handles
84     print(mylist)
85     # [百度首页,hao123]
86     # 切换到hao123窗口
87     dr.switch_to.window(mylist[1])
88     dr.close() # 关闭hao123
89     time.sleep(3)
90     dr.quit()
91
92 # mytest02()
93
94 #把百度首页的百度搜索的第一条新闻标题,粘贴到新闻窗口的搜索框中
95 def mytest03():
96     dr=webdriver.Firefox()
97     dr.get("https://www.baidu.com")
98     time.sleep(2)
99     e=dr.find_element_by_partial_link_text("湖南遇特")
100    msg=e.text#获取标签对之间的内容
101    dr.find_element_by_link_text('新闻').click()
102    time.sleep(2)
103    #切换到新闻窗口
104    mylist=dr.window_handles
105    dr.switch_to.window(mylist[1])
106    e2=dr.find_element_by_id('ww')
107    e2.send_keys(msg)
108    time.sleep(3)
109    dr.quit()
110
111 mytest03()
112

```

## 多Frame框处理

```

1  在一个页面中可以嵌套另外一个页面，如frame/iframe技术，这是现在很多web应用中使用的一种方式，
   webdriver对象只能在一个页面（外层是默认的）中定位元素，需要一种方式将driver对象从外层切换给内层使用
   才能对内层的对象进行处理。
2  webdriver中提供API：driver.switch_to.frame()实现
3
4  1) 切入Frame
5  第一种方式,默认是可以给ID或者name的
6
7      driver.switch_to.frame("login_frame")
8
9  第二种方式,可以传参iframe的元素对象
10
11      iframeObj = driver.find_element_by_xpath('//*[@id="login_frame"]')
12      driver.switch_to.frame(iframeObj)
13  # 切换到目标元素所在的frame
14  dr.switch_to.frame("iframeResult")
15

```



```

16 2) 从Frame切入到主体
17 dr.switch_to.default_content()
18
19 #!/usr/bin/env python
20 # -*- coding:utf-8 -*-
21 #====#====#====#====
22 #Author:
23 #CreatDate:
24 #Version:
25 #====#====#====#====
26 from selenium import webdriver
27 import time
28
29 #往QQ邮箱登录页面的用户名输入框输入内容
30 def mytest():
31     dr=webdriver.Firefox()
32     dr.get("https://mail.qq.com/")
33     time.sleep(3)
34     #切换到小的html,小括号中的值是iframe标签的id或name的值
35     #获取第一个内层的iframe元素
36     e=dr.find_element_by_class_name("QQMailSdkTool_login_loginBox_qq_iframe")
37     dr.switch_to.frame(e)
38
39     #切换到第二内层
40     dr.switch_to.frame("ptlogin_iframe")
41     dr.find_element_by_id('switcher_plogin').click()#点击密码登录
42     time.sleep(2)
43     e=dr.find_element_by_id('u')#获取输入框
44     e.send_keys("123456789")
45     time.sleep(3)
46     #切换回大的html
47     dr.switch_to.default_content()
48     #点击基本版链接
49     dr.find_element_by_link_text('基本版').click()
50     time.sleep(3)
51     dr.quit()
52
53 mytest()
54
55
56
57
58 https://www.runoob.com/try/try.php?filename=jqueryui-api-droppable
59 4) 拖动:drag_and_drop()
60
61 #!/usr/bin/env python
62 # -*- coding:utf-8 -*-
63 #====#====#====#====
64 #Author:
65 #CreatDate:
66 #Version:
67 #====#====#====#====
68

```

```

69 from selenium.webdriver.common.action_chains import ActionChains
70 from selenium import webdriver
71 import time
72
73 dr=webdriver.Firefox()
74 dr.get("https://www.runoob.com/try/try.php?filename=jqueryui-api-droppable")
75 time.sleep(2)
76 #切换到小的html
77 dr.switch_to.frame("iframeResult")
78 e=dr.find_element_by_id('draggable')
79 e2=dr.find_element_by_id('droppable')
80 ActionChains(dr).drag_and_drop(e,e2).perform()
81 time.sleep(3)
82 dr.quit()
83
84
85

```

## 警告框处理

```

1  警告框-alert，是一个模式框，该怎么使用代码的方式去操作他？
2  1、driver对象不在alert上，并且我们没办法去定位这个窗口的元素
3  2、driver.switch_to.alert:暂时将浏览器对象driver交给alert用
4  3、可以对alert警告框作什么事情：
5      text：返回（获取） alert/confirm/prompt 中的文字信息。
6      accept()：接受现有警告框，就是点他的确定按钮
7      dismiss()：放弃现有警告框，取消
8      send_keys(keysToSend)：发送文本至警告框。
9
10 #!/usr/bin/env python
11 # -*- coding:utf-8 -*-
12 #====#====#====#====
13 #Author:
14 #CreatDate:
15 #Version:
16 #====#====#====#====
17 import time
18 from selenium import webdriver
19 from selenium.webdriver.common.action_chains import ActionChains
20
21 dr=webdriver.Firefox()
22 dr.get("https://www.baidu.com")
23 time.sleep(2)
24 e=dr.find_element_by_id('s-usersetting-top')#设置元素
25 ActionChains(dr).move_to_element(e).perform()
26 time.sleep(2)
27 dr.find_element_by_partial_link_text('搜索设置').click()
28 time.sleep(2)
29 dr.find_element_by_id('nr_2').click()
30 time.sleep(2)

```

```
31 dr.find_element_by_link_text('保存设置').click()
32 time.sleep(2)
33 #这时弹出警告框
34 at=dr.switch_to.alert
35
36 print(at.text)#获取警告框上的文字信息
37 at.accept()#确认警告框
38
39 time.sleep(3)
40 dr.quit()
41
42
43
```