

SELENIUM_COURSE_01

01 元素定位-find_element_by_元素

```
__author__ = 'Administrator'
from time import sleep
from selenium import webdriver

# 打开浏览器
browser = webdriver.Firefox()
sleep(2)
# 输入 URL
browser.get("http://127.0.0.1:8080/oa")
sleep(2)
# 输入用户名，需要先模拟眼睛的动作，找到用户名输入框，然后手才能输入
username_ele = browser.find_element_by_class_name('loginInput')
username_ele.send_keys("sup")
sleep(2)
# 输入密码，需要先模拟眼睛的动作，找到密码输入框，然后手才能输入
browser.find_element_by_name("password").send_keys("s1234567")
sleep(2)
# 点击登录按钮，需要先模拟眼睛的动作，找到登录按钮，然后手才能点击
browser.find_element_by_id("button_submit").click()
sleep(2)
# 点击新建事项，通过链接内容来定位
browser.find_element_by_link_text("新建事项").click()
sleep(2)
# 点击已发事项，通过部分链接内容来定位
browser.find_element_by_partial_link_text("已发").click()
sleep(2)

# 点击已办事项上方的图片，可以通过 css 的方式来定位
browser.find_element_by_css_selector("div.ui-accordion-content:nth-child(2) > table:nth-child(1) > tbody:nth-child(1) > tr:nth-child(8) > td:nth-child(1) > a:nth-child(1) > img:nth-child(1)").click()
sleep(2)

# 点击待发事项上方的图片，可以通过 xpath 的方式来定位
browser.find_element_by_xpath("/html/body/div[2]/div/div[1]/table/body/tr[11]/td/a/img").click()
sleep(2)

# 点击退出登录，通过标签名称来定位
```

```
browser.find_element_by_tag_name("a").click()  
sleep(2)
```

```
# 关闭浏览器  
browser.quit()
```

02 元素定位-css_combine 元素

```
__author__ = 'Administrator'
from selenium import webdriver
from time import sleep

browser = webdriver.Firefox()
sleep(2)
browser.get("http://127.0.0.1:8080/oa")
sleep(2)
# 使用 css 结合 class 属性定位用户名输入框
browser.find_element_by_css_selector(".loginInput").clear()
sleep(1)
# 使用 css 结合 name 属性定位用户名输入框
browser.find_element_by_css_selector('[name="loginId"]').send_keys("sup")
sleep(1)
# 使用 css 结合 tabindex 属性定位用户名输入框
browser.find_element_by_css_selector('[tabindex="1"]').clear()
sleep(1)
# 使用 css 结合 type 属性定位用户名输入框
browser.find_element_by_css_selector('[type="text"]').send_keys("sup")
sleep(1)
# 使用 css 结合多个属性定位用户名输入框
browser.find_element_by_css_selector('.loginInput[name="loginId"][tabindex="1"][type="text"]').clear()
browser.find_element_by_css_selector('.loginInput[name="loginId"][tabindex="1"][type="text"]').send_keys("sup")
sleep(2)

# 使用 css 结合多个属性定位密码输入框
browser.find_element_by_css_selector('.loginInput[name="password"][tabindex="2"][type="password"]').send_keys("s1234567")
sleep(2)

# 使用 css 结合多个属性定位登录按钮
browser.find_element_by_css_selector('#button_submit.loginbutton[type="button"]').click()
sleep(2)
browser.quit()
```

03 元素定位-xpath_combine 元素

```
__author__ = 'Administrator'
from time import sleep
from selenium import webdriver

browser = webdriver.Firefox()
sleep(2)
browser.get("http://127.0.0.1:8080")
sleep(2)
# 使用 xpath 结合其他属性定位用户名输入框
browser.find_element_by_xpath('//input[@class="loginInput"]').send_keys("sup")
sleep(1)
# 使用 xpath 结合 name 属性定位用户名输入框
browser.find_element_by_xpath('//input[@name="loginId"]').clear()
sleep(1)
# 使用 xpath 结合 tabindex 属性定位用户名输入框
browser.find_element_by_xpath('//input[@tabindex="1"]').send_keys("sup")
sleep(1)
# 使用 xpath 结合 type 属性定位用户名输入框
browser.find_element_by_xpath('//input[@type="text"]').clear()
sleep(1)
# 使用 xpath 结合多个属性定位用户名输入框
browser.find_element_by_xpath('//input[@class="loginInput" and @name="loginId" and @tabindex="1" and @type="text"]').send_keys("sup")
sleep(2)

# 使用 xpath 结合多个属性定位密码输入框
browser.find_element_by_xpath('//input[@class="loginInput" and @name="password"]').send_keys("s1234567")
sleep(2)

# 使用 xpath 结合多个属性定位登录按钮
browser.find_element_by_xpath('//input[@id="button_submit"]').click()
sleep(2)

# 使用 xpath 结合 text() 定位新建事项
browser.find_element_by_xpath('//a[text()="新建事项"]').click()
sleep(2)

# 使用 xpath 结合 contains 方法来定位已发事项
```

```
browser.find_element_by_xpath('//a[contains(text(),"已发")]').click()  
sleep(2)
```

```
# 使用 xpath 结合 contains 方法来定位已办事项上方的图片
```

```
browser.find_element_by_xpath('//img[contains(@src,"done.png")]').cli  
ck()  
sleep(2)
```

```
# 使用 xpath 结合其他属性定位密码修改
```

```
browser.find_element_by_xpath('//a[@class="black" and @title="密码修  
改" and text()="密码修改"]').click()  
sleep(2)
```

```
browser.quit()
```

04 元素定位-By

```
__author__ = 'Administrator'
from selenium import webdriver
from time import sleep
from selenium.webdriver.common.by import By

browser = webdriver.Firefox()
sleep(2)
browser.get("http://127.0.0.1:8080/oa")
sleep(2)
browser.find_element(By.CLASS_NAME, "loginInput").send_keys("sup")
sleep(2)
browser.find_element(By.NAME, "password").send_keys("s1234567")
sleep(2)
browser.find_element(By.ID, "button_submit").click()
sleep(2)
browser.find_element(By.LINK_TEXT, "新建事项").click()
sleep(2)
browser.find_element(By.PARTIAL_LINK_TEXT, "已发").click()
sleep(2)
browser.find_element(By.CSS_SELECTOR, '[src="/oa/common/images/leftmenu/collaborate/done.png"]').click()
sleep(2)
browser.find_element(By.XPATH, '//a[text()="待发事项"]').click()
sleep(2)
browser.find_element(By.TAG_NAME, "a").click()
sleep(2)
browser.quit()
```

05 元素定位 find_elements_by_元素

```
__author__ = 'Administrator'
from selenium import webdriver
from time import sleep

from selenium.webdriver.common.by import By

browser = webdriver.Firefox()
sleep(2)
browser.get("http://127.0.0.1:8080/oa")
sleep(2)
elements = browser.find_elements_by_class_name('loginInput')
print(type(elements), len(elements))

for i in elements:
    i.clear()

elements[0].send_keys("sup")
sleep(2)

elements[1].send_keys("s1234567")
sleep(2)

elements1 = browser.find_elements(By.ID, "button_submit")
print(type(elements1), len(elements1))
elements1[0].click()
sleep(2)

browser.quit()
```

不同浏览器

```
__author__ = 'Administrator'  
# coding = utf-8
```

```
import time  
from selenium import webdriver
```

```
browser = webdriver.Chrome()  
time.sleep(2)  
browser.get("http://127.0.0.1:8080/oa")  
browser.find_element_by_name("loginId").send_keys("sup")  
browser.find_element_by_name("password").send_keys("s1234567")  
browser.find_element_by_id("button_submit").click()  
time.sleep(3)  
browser.quit()
```

```
__author__ = 'Administrator'  
# coding = utf-8
```

```
import time  
from selenium import webdriver
```

```
browser = webdriver.Firefox()  
time.sleep(2)  
browser.get("http://127.0.0.1:8080/oa")  
browser.find_element_by_name("loginId").send_keys("sup")  
browser.find_element_by_name("password").send_keys("s1234567")  
browser.find_element_by_id("button_submit").click()  
time.sleep(3)  
browser.quit()
```

```
__author__ = 'Administrator'  
# coding = utf-8  
import time  
from selenium import webdriver  
from selenium.webdriver.common.desired_capabilities import  
DesiredCapabilities
```

```
DesiredCapabilities.INTERNETEXPLORER["ignoreProtectedModeSettings"] =  
True  
browser = webdriver.Ie()  
browser.get("http://127.0.0.1:8080/oa")  
browser.find_element_by_name("loginId").clear()  
browser.find_element_by_name("loginId").send_keys("sup")
```



```
browser.find_element_by_name("password").send_keys("s1234567")
browser.find_element_by_id("button_submit").click()
time.sleep(5)
browser.quit()
```

SELENIUM_COURSE_02

01 frame_change

```
__author__ = 'Administrator'
from selenium import webdriver
from time import sleep

browser = webdriver.Firefox()
sleep(2)
browser.get("http://127.0.0.1:8080/oa")
sleep(2)
# 使用 sup 的账号登录
browser.find_element_by_xpath('//input[@class="loginInput"]').send_keys("sup")
sleep(2)
browser.find_element_by_xpath('//input[@name="password"]').send_keys("s1234567")
sleep(2)
browser.find_element_by_xpath('//input[@id="button_submit"]').click()
sleep(2)
# 点击新建事项
browser.find_element_by_xpath('//a[text()="新建事项"]').click()
sleep(2)
# 输入标题，需要先切入框架
browser.switch_to.frame("iframe_main")
browser.find_element_by_xpath('//input[@id="subject"]').send_keys("今天是 selenium 第二天，快要放假了！")
sleep(2)
# 点击流程，已经在框架中，不需要切框架
browser.find_element_by_xpath('//input[@id="trackInput"]').click()
sleep(2)
# 点击集团职能部，需先从之前的框架中切出，然后还需要切入另外一个框架
browser.switch_to.default_content()

liuchengl =
browser.find_element_by_xpath('//iframe[contains(@src,"/oa/common/components/selectemp/selectempforcollaborate.jsp?uuid=")]')
browser.switch_to.frame(liuchengl)
browser.find_element_by_xpath('//span[text()="集团职能部"]').click()
sleep(2)
# 点击总经办
browser.find_element_by_xpath('//span[text()="总经办"]').click()
sleep(2)
# 点击添加全部按钮
```

```
browser.find_element_by_xpath('//button[@class="button" and @title="
添加全部"]').click()
sleep(2)
# 点击确定按钮
browser.find_element_by_xpath('//button[contains(text(),"确定
")]').click()
sleep(2)

# 点击流程 2 中的确定按钮，需要从原来的框架中切出，然后再切入第二个框
架
browser.switch_to.default_content()
liucheng2 =
browser.find_element_by_xpath('//iframe[contains(@src,"/oa/common/com
ponents/flowchart/newflowchart_svg.jsp?uuid=")]')
browser.switch_to.frame(liucheng2)
browser.find_element_by_xpath('//button[contains(text(),"确定
")]').click()
sleep(2)
# 输入正文，需要先切出原来的框架，然后再切入正文所在的框架
browser.switch_to.default_content()
browser.switch_to.frame("iframe_main")
browser.switch_to.frame("baidu_editor_0")
browser.find_element_by_xpath('/html/body').send_keys("今天主要讲框架
的切换、网页的操作、鼠标的操作、警告窗口的操作！")
sleep(2)

# 点击立即发送按钮，需要切出到上一层框架
browser.switch_to.parent_frame()
browser.find_element_by_xpath('//button[@id="button_send"]').click()
sleep(2)

# 点击确定按钮，需要切换到默认框架
browser.switch_to.default_content()
browser.find_element_by_xpath('//input[@value="确定" and @class="d-
button"]').click()
sleep(2)

browser.quit()
```

02 web_change

```
__author__ = 'Administrator'
from selenium import webdriver
from time import sleep

browser = webdriver.Firefox()
sleep(2)
browser.get("http://127.0.0.1:8080/oa")
sleep(2)
# 打开万年历和计算器
browser.find_element_by_xpath('//a[text()="万年历"]').click()
sleep(2)
browser.find_element_by_xpath('//a[text()="计算器"]').click()
sleep(2)
# 获取打开网页的句柄
windows = browser.window_handles
print(type(windows), len(windows))

# 切换到万年历的页面中，点击下一月和上一年
browser.switch_to.window(windows[1])
browser.find_element_by_xpath('//a[@id="yuejia"]').click()
sleep(2)
browser.find_element_by_xpath('//a[@id="nianjian"]').click()
sleep(2)

# 切换到计算器的页面中，计算 4+9=
browser.switch_to.window(windows[2])
# 点击 4 的按钮
browser.find_element_by_xpath('//input[@name="k4"]').click()
sleep(2)
# 点击+的按钮
browser.find_element_by_xpath('//input[@value=" +"]').click()
sleep(2)
# 点击 9 的按钮
browser.find_element_by_xpath('//input[@name="k9"]').click()
sleep(2)
# 点击=的按钮
browser.find_element_by_xpath('//input[@value=" ="]').click()
sleep(2)

# 关闭计算器的窗口
browser.close()
sleep(2)
```

```
# 关闭万年历的窗口，还需要切换到万年历的窗口中再关闭
browser.switch_to.window(windows[1])
browser.close()
sleep(2)

# 登录 OA，需要切换到登录的页面中
browser.switch_to.window(windows[0])
# 使用 sup 的账号登录
browser.find_element_by_xpath('//input[@class="loginInput"]').send_keys("sup")
sleep(2)
browser.find_element_by_xpath('//input[@name="password"]').send_keys("s1234567")
sleep(2)
browser.find_element_by_xpath('//input[@id="button_submit"]').click()
sleep(2)

browser.quit()
```

03 windows_operation

```
__author__ = 'Administrator'
from selenium import webdriver
from time import sleep, localtime, strftime

browser = webdriver.Firefox()
sleep(2)
browser.get("http://127.0.0.1:8080/oa")
sleep(2)

# 设置网页最小化
browser.minimize_window()
sleep(2)

# 设置网页最大化
browser.maximize_window()
sleep(2)

# 设置网页的大小
browser.set_window_size(300, 100)
sleep(2)

# 设置网页最大化
browser.maximize_window()
sleep(2)

# 刷新网页
browser.refresh()
sleep(2)

# 网页前进后退需要先登录
# 使用 sup 的账号登录
browser.find_element_by_xpath('//input[@class="loginInput"]').send_keys("sup")
sleep(2)
browser.find_element_by_xpath('//input[@name="password"]').send_keys("s1234567")
sleep(2)
browser.find_element_by_xpath('//input[@id="button_submit"]').click()
sleep(2)

# 获取网页的 URL 和标签
URL = browser.current_url
print(URL)
```

```
title = browser.title
print(title)

# 获取当前登录用户的名字，需要先定位显示名字的元素信息，然后用 text 来
获取
user_ele =
browser.find_element_by_xpath('/html/body/div[1]/table/tbody/tr/td[2]
/table/tbody/tr[3]/td[1]/label')
username = user_ele.text
print(username)

# 截图，截登录后的首页
file_name =
"D:\\zhaojia\\selenium\\selenium_105\\selenium_study\\picture\\首页截
图.png"
browser.get_screenshot_as_file(file_name)

# 使用时间戳的方式截图
file_path =
"D:\\zhaojia\\selenium\\selenium_105\\selenium_study\\picture\\"
file_name1 = "首页截图" + strftime("%Y%m%d%H%M%S", localtime()) +
'.png'
browser.get_screenshot_as_file(file_path+file_name1)

# 网页的后退
browser.back()
sleep(2)

# 网页的前进
browser.forward()
sleep(2)

browser.quit()
```

04 alarm

```
__author__ = 'Administrator'
from selenium import webdriver
from time import sleep
from selenium.webdriver.common.action_chains import ActionChains

browser = webdriver.Firefox()
sleep(2)
browser.get("http://127.0.0.1:8080/oa")
sleep(2)
# 使用 sup 的账号登录
browser.find_element_by_xpath('//input[@class="loginInput"]').send_keys("sup")
sleep(2)
browser.find_element_by_xpath('//input[@name="password"]').send_keys("s1234567")
sleep(2)
browser.find_element_by_xpath('//input[@id="button_submit"]').click()
sleep(2)
# 点击新建事项
browser.find_element_by_xpath('//a[text()="新建事项"]').click()
sleep(2)
browser.switch_to.frame("iframe_main")
browser.switch_to.frame("baidu_editor_0")
# 首先导入 actionchains 模块，将驱动好的浏览器作为参数传入
# 调用 context_click 方法，需要将元素信息传入，最后调用 perform 来执行
zhengwen = browser.find_element_by_xpath('/html/body')
zhengwen.send_keys("今天心情还不错!")
ActionChains(browser).context_click(zhengwen).perform()
sleep(2)
# 点击清空文档，需要先切出框架到上一层框架，然后定位元素
browser.switch_to.parent_frame()
browser.find_element_by_xpath('//div[text()="清空文档"]').click()
sleep(2)
# 切入到警告窗口
alert = browser.switch_to.alert
# 获取警告窗口的提示信息
tips = alert.text
print(tips)
# 点击取消按钮
alert.dismiss()
sleep(2)
# 点击确定按钮
browser.quit()
```


05 mouse

```
__author__ = 'Administrator'
from selenium import webdriver
from time import sleep
from selenium.webdriver.common.action_chains import ActionChains

browser = webdriver.Firefox()
sleep(2)
browser.get("http://127.0.0.1:8080/oa")
sleep(2)
# 使用 sup 的账号登录
browser.find_element_by_xpath('//input[@class="loginInput"]').send_keys("sup")
sleep(2)
browser.find_element_by_xpath('//input[@name="password"]').send_keys("s1234567")
sleep(2)
browser.find_element_by_xpath('//input[@id="button_submit"]').click()
sleep(2)

# 在个人设置后面的三角形上悬停
# 首先导入 actionchains 模块，将驱动好的浏览器作为参数传入
# 调用 move_to_element 方法，需要将元素信息传入，最后调用 perform 来执行
personal_setting = browser.find_element_by_xpath('//span[@class="m-btn-downarrow"]')
ActionChains(browser).move_to_element(personal_setting).perform()
sleep(2)
# 点击常用语设置
browser.find_element_by_xpath('//div[text()="常用语设置"]').click()
sleep(2)
# 关闭常用语设置
browser.find_element_by_xpath('//a[@class="d-close"]').click()
sleep(2)
# 点击新建事项
browser.find_element_by_xpath('//a[text()="新建事项"]').click()
sleep(2)
# 点击流程，需要切换框架
browser.switch_to.frame("iframe_main")
browser.find_element_by_xpath('//input[@id="trackInput"]').click()
sleep(2)
# 点击集团职能部，需要切回默认框架，再切入框架
```

```

browser.switch_to.default_content()
liucheng1 =
browser.find_element_by_xpath('//iframe[contains(@src,"/oa/common/com
ponents/selectemp/selectempforcollaborate.jsp?uuid=")]')
browser.switch_to.frame(liucheng1)
browser.find_element_by_xpath('//span[text()="集团职能部"]').click()
sleep(2)
browser.find_element_by_xpath('//span[text()="总经办"]').click()
sleep(2)
# 双击邢佳军
# 首先导入 actionchains 模块，将驱动好的浏览器作为参数传入
# 调用 double_click 方法，需要将元素信息传入，最后调用 perform 来执行
leader =
browser.find_element_by_xpath('/html/body/div[2]/select/option[1]')
ActionChains(browser).double_click(leader).perform()
sleep(2)
# 点击确定按钮
browser.find_element_by_xpath('//button[contains(text(),"确定
")]').click()
sleep(2)
# 切换到流程 2 的框架中，点击确定按钮
browser.switch_to.default_content()
liucheng2 =
browser.find_element_by_xpath('//iframe[contains(@src,"/oa/common/com
ponents/flowchart/newflowchart_svg.jsp?uuid=")]')
browser.switch_to.frame(liucheng2)
# 点击确定按钮
browser.find_element_by_xpath('//button[contains(text(),"确定
")]').click()
sleep(2)

# 右击正文输入框，需要切换框架
browser.switch_to.default_content()
browser.switch_to.frame("iframe_main")
browser.switch_to.frame("baidu_editor_0")
# 首先导入 actionchains 模块，将驱动好的浏览器作为参数传入
# 调用 context_click 方法，需要将元素信息传入，最后调用 perform 来执行
zhengwen = browser.find_element_by_xpath('/html/body')
ActionChains(browser).context_click(zhengwen).perform()
sleep(2)
browser.quit()

```

SELENIUM_COURSE_02

01 keyboard

```
__author__ = 'Administrator'
from selenium import webdriver
from time import sleep

from selenium.webdriver.common.keys import Keys

browser = webdriver.Firefox()
sleep(2)
browser.get("http://127.0.0.1:8080/oa")
sleep(2)
# 登录
browser.find_element_by_xpath('//input[@class="loginInput"]').send_keys("sup")
sleep(2)
browser.find_element_by_xpath('//input[@name="password"]').send_keys("s1234567")
sleep(2)
browser.find_element_by_xpath('//input[@id="button_submit"]').click()
sleep(2)

# 点击新建事项
browser.find_element_by_xpath('//a[contains(text(),"新建")]').click()
sleep(2)

# 输入正文需要切换框架
browser.switch_to.frame("iframe_main")
browser.switch_to.frame("baidu_editor_0")

# 定位正文输入框
zhengwen = browser.find_element_by_xpath('/html/body')
zhengwen.send_keys("今天是收假后第一天，有点不在状态，需要喝一个脉动。。。。")
sleep(2)

# 在正文中换行 5 次
zhengwen.send_keys(Keys.ENTER*5)
sleep(2)

# 在正文中 TAB 键 3 次
zhengwen.send_keys(Keys.TAB*3)
sleep(2)
```

```
# 在正文中删除 10 次
zhengwen.send_keys(Keys.BACK_SPACE*10)
sleep(2)

# 在正文中全选、复制、粘贴 10 次
zhengwen.send_keys(Keys.CONTROL, 'a')
sleep(2)
zhengwen.send_keys(Keys.CONTROL, 'c')
sleep(2)
zhengwen.send_keys(Keys.CONTROL, 'v'*10)
sleep(2)

browser.quit()
```

02 implicit_wait

```
__author__ = 'Administrator'
from selenium import webdriver
from time import sleep

browser = webdriver.Firefox()
browser.get("http://127.0.0.1:8080/oa")
browser.implicitly_wait(10)
# 使用 sup 的账号登录
browser.find_element_by_xpath('//input[@class="loginInput"]').send_keys("sup")
browser.find_element_by_xpath('//input[@name="password"]').send_keys("s1234567")
browser.find_element_by_xpath('//input[@id="button_submit"]').click()
# 点击新建事项
browser.find_element_by_xpath('//a[text()="新建事项"]').click()
# 输入标题, 需要先切入框架
browser.switch_to.frame("iframe_main")
browser.find_element_by_xpath('//input[@id="subject"]').send_keys("今天是 selenium 第二天, 快要放假了!")
# 点击流程, 已经在框架中, 不需要切框架
browser.find_element_by_xpath('//input[@id="trackInput"]').click()
sleep(2)
# 点击集团职能部, 需要先从之前的框架中切出, 然后还需要切入另外一个框架
browser.switch_to.default_content()
# 切入流程 1 的框架, 需要先定位这个框架, 得到对象以后再切入
liucheng1 =
browser.find_element_by_xpath('//iframe[contains(@src,"/oa/common/components/selectemp/selectempforcollaborate.jsp?uuid=")]')
browser.switch_to.frame(liucheng1)
browser.find_element_by_xpath('//span[text()="集团职能部"]').click()
# 点击总经办
browser.find_element_by_xpath('//span[text()="总经办"]').click()
# 点击添加全部按钮
browser.find_element_by_xpath('//button[@class="button" and @title="添加全部"]').click()
# 点击确定按钮
browser.find_element_by_xpath('//button[contains(text(),"确定")]').click()
sleep(2)
```

```
# 点击流程 2 中的确定按钮，需要从原来的框架中切出，然后再切入第二个框架
browser.switch_to.default_content()
liucheng2 =
browser.find_element_by_xpath('//iframe[contains(@src,"/oa/common/components/flowchart/newflowchart_svg.jsp?uuid=")]')
browser.switch_to.frame(liucheng2)
browser.find_element_by_xpath('//button[contains(text(),"确定")]').click()
# 输入正文，需要先切出原来的框架，然后再切入正文所在的框架
browser.switch_to.default_content()
browser.switch_to.frame("iframe_main")
browser.switch_to.frame("baidu_editor_0")
browser.find_element_by_xpath('/html/body').send_keys("今天主要讲框架的切换、网页的操作、鼠标的操作、警告窗口的操作！")

# 点击立即发送按钮，需要切出到上一层框架
browser.switch_to.parent_frame()
browser.find_element_by_xpath('//button[@id="button_send"]').click()

# 点击确定按钮，需要切换到默认框架
browser.switch_to.default_content()
browser.find_element_by_xpath('//input[@value="确定" and @class="d-button"]').click()

browser.quit()
```

03 webdriver_wait

```
__author__ = 'Administrator'
from selenium import webdriver

from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as ec

from selenium.webdriver.common.by import By
from selenium.webdriver.common.action_chains import ActionChains

browser = webdriver.Firefox()
browser.get("http://127.0.0.1:8080/oa")

# 设置动态等待，判断所有符合 class 属性为 logininput 的元素全部加载完成，然后再执行后续脚本
# 用到的方法：presence_of_all_elements_located
WebDriverWait(browser, 10).until(ec.presence_of_all_elements_located((By.CLASS_NAME, "loginInput"))))
# 输入用户名
browser.find_element_by_xpath('//input[@class="loginInput"]').send_keys("sup")
# 输入密码
browser.find_element_by_xpath('//input[@name="password"]').send_keys("s1234567")

# 设置动态等待，判断登录按钮的 value 值是否出现预期的登录两个字
# 用到的方法：text_to_be_present_in_element_value
WebDriverWait(browser, 10).until(ec.text_to_be_present_in_element_value((By.ID, "button_submit"), "登录"))
# 点击登录按钮
browser.find_element_by_xpath('//input[@id="button_submit"]').click()

# 点击新建事项之前，设置动态等待，判断新建事项是否可以点击
# 用到的方法：element_to_be_clickable
WebDriverWait(browser, 10).until(ec.element_to_be_clickable((By.LINK_TEXT, "新建事项"))))
# 点击新建事项
browser.find_element_by_xpath('//a[text()="新建事项"]').click()

# 输入正文，需要切入框架，设置动态等待，判断框架是否加载出来
# 用到的方法：frame_to_be_available_and_switch_to_it
WebDriverWait(browser, 10).until(ec.frame_to_be_available_and_switch_to_it("iframe_main"))
```

```
WebDriverWait(browser, 10).until(ec.frame_to_be_available_and_switch_to_it("baidu_editor_0"))
```

```
# 设置动态等待，判断正文输入框是否出现
```

```
# 用到的方法：presence_of_element_located
```

```
WebDriverWait(browser, 10).until(ec.presence_of_element_located((By.XPATH, "/html/body")))
```

```
# 定位正文输入框
```

```
zhengwen = browser.find_element_by_xpath('/html/body')
```

```
zhengwen.send_keys("今天天气有点凉")
```

```
# 点击清空文档，需要先对正文进行右击，然后切出框架，再设置动态等待，判断元素的文本值是否为期望的“清空文档”
```

```
# 用到的方法：text_to_be_present_in_element
```

```
ActionChains(browser).context_click(zhengwen).perform()
```

```
browser.switch_to.parent_frame()
```

```
WebDriverWait(browser, 10).until(ec.text_to_be_present_in_element((By.XPATH, "/html/body/div[4]/div[5]/div/div[2]/div/div[2]/div/div[3]"), "清空文档"))
```

```
# 点击清空文档
```

```
browser.find_element_by_xpath('/html/body/div[4]/div[5]/div/div[2]/div/div[2]/div/div[3]').click()
```

```
# 对警告窗口进行操作，需要设置的动态等待，判断警告窗口是否出现
```

```
# 用到的方法：alert_is_present
```

```
alert = WebDriverWait(browser, 10).until(ec.alert_is_present())
```

```
# 获取警告窗口中的文本，然后点击确定按钮
```

```
text = alert.text
```

```
print(text)
```

```
alert.accept()
```

```
browser.quit()
```


04 函数调用

1) User_login

```
__author__ = 'Administrator'
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as ec
from selenium.webdriver.common.by import By
from selenium_day3.shijianchuo.def_shijianchuo import jietu
# 定义函数，实现用户登录的过程
def login_sup(browser):
    try:
        browser.get("http://127.0.0.1:8080/oa")
        # 设置动态等待，判断所有符合 class 属性为 logininput 的元素全部
        # 加载完成，然后再执行后续脚本
        # 用到的方法：presence_of_all_elements_located

        WebDriverWait(browser, 10).until(ec.presence_of_all_elements_located((
            By.CLASS_NAME, "loginInput")))
        # 输入用户名

        browser.find_element_by_xpath('//input[@class="loginInput"]').send_keys("sup")
        # 输入密码

        browser.find_element_by_xpath('//input[@name="password"]').send_keys("s1234567")
        # 设置动态等待，判断登录按钮的 value 值是否出现预期的登录两个字
        # 用到的方法：text_to_be_present_in_element_value

        WebDriverWait(browser, 10).until(ec.text_to_be_present_in_element_value(
            (By.ID, "button_submit"), "登录"))
        # 点击登录按钮

        browser.find_element_by_xpath('//input[@id="button_submit"]').click()
        except Exception as e:
            jietu(browser, "登录报错")
            raise e

# 定义函数，实现使用不同的用户名密码登录的过程
def login_by_different_username_password(browser, username, password):
    try:
        browser.get("http://127.0.0.1:8080/oa")
        # 设置动态等待，判断所有符合 class 属性为 logininput 的元素全部
        # 加载完成，然后再执行后续脚本
```

```

        # 用到的方法: presence_of_all_elements_located

WebDriverWait(browser, 10).until(ec.presence_of_all_elements_located((
    By.CLASS_NAME, "loginInput")))
    # 输入用户名

browser.find_element_by_xpath('//input[@class="loginInput"]').clear()

browser.find_element_by_xpath('//input[@class="loginInput"]').send_keys(
    username)
    # 输入密码

browser.find_element_by_xpath('//input[@name="password"]').send_keys(
    password)
    # 设置动态等待, 判断登录按钮的 value 值是否出现预期的登录两个字

    # 用到的方法: text_to_be_present_in_element_value

WebDriverWait(browser, 10).until(ec.text_to_be_present_in_element_value(
    (By.ID, "button_submit"), "登录"))
    # 点击登录按钮

browser.find_element_by_xpath('//input[@id="button_submit"]').click()
    except Exception as e:
        jietu(browser, "登录报错")
        raise e

__author__ = 'Administrator'
from selenium import webdriver
from selenium_day3.user_login import def_user_login
from time import sleep

browser = webdriver.Firefox()
# def_user_login.login_sup(browser)
def_user_login.login_by_different_username_password(browser, "sup", "s1
234567")
sleep(2)
def_user_login.login_by_different_username_password(browser, "suppppppp
", "s1234567")
sleep(2)
browser.quit()

```

2) time screen_shot

```
__author__ = 'Administrator'
from time import localtime, strftime
# 定义函数，时间生成当前时间的时间戳
def shijianchuo():
    # 获取当前时间
    current_time = localtime()
    # 将获取到的时间转换为标准的输出格式
    change_time = strftime("%Y%m%d%H%M%S", current_time)
    return change_time

# 定义函数，实现截图
def jietu(browser, text):
    # 确定截图的保存路径、图片的名字、图片的格式
    file_name =
"D:\\zhaojia\\selenium\\selenium_105\\selenium_study\\picture\\"+text
+shijianchuo()+".png"
    # 截图
    browser.get_screenshot_as_file(file_name)

__author__ = 'Administrator'
from selenium import webdriver

from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as ec

from selenium.webdriver.common.by import By

# from selenium_day3.shijianchuo import def_shijianchuo
from selenium_day3.shijianchuo.def_shijianchuo import jietu

browser = webdriver.Firefox()
browser.get("http://127.0.0.1:8080/oa")

WebDriverWait(browser, 10).until(ec.presence_of_all_elements_located((
By.CLASS_NAME, "loginInput"))))
# 输入用户名
browser.find_element_by_xpath('//input[@class="loginInput"]').send_keys("sup")
# 输入密码
browser.find_element_by_xpath('//input[@name="password"]').send_keys("s1234567")
```

```
# 设置动态等待，判断登录按钮的 value 值是否出现预期的登录两个字
# 用到的方法：text_to_be_present_in_element_value
WebDriverWait(browser, 10).until(ec.text_to_be_present_in_element_value((By.ID, "button_submit"), "登录"))
# 点击登录按钮
browser.find_element_by_xpath('//input[@id="button_submit"]').click()

# 点击新建事项之前，设置动态等待，判断新建事项是否可以点击
# 用到的方法：element_to_be_clickable
WebDriverWait(browser, 10).until(ec.element_to_be_clickable((By.LINK_TEXT, "新建事项")))
# def_shijianchuo.jietu(browser, "首页截图")
jietu(browser, "首页截图")
browser.quit()
```

4) element_exit

```
__author__ = 'Administrator'
from selenium_day3.shijianchuo.def_shijianchuo import jietu
from time import sleep
# 定义函数，通过 xpath 来判断元素是否存在，如果存在则返回 true，否则返回 False
def element_exit_by_xpath(browser, elexpath):
    if browser.find_element_by_xpath(elexpath):
        return True
    else:
        return False

# 定义函数，结合 try 和 except 来判断元素是否存在，通过 link_text 来判断
def element_exit_by_link_text(browser, text):
    try:
        browser.find_element_by_link_text(text)
        return True
    except:
        return False

# 定义函数，实现判断链接是否存在，如果存在则截图，否则输出“链接未找到”
def text_click_jietu(browser, text):
    result = element_exit_by_link_text(browser, text)
    if result==True:
        browser.find_element_by_link_text(text).click()
        sleep(2)
        jietu(browser, text)
    else:
        print("链接未找到。。。。")

__author__ = 'Administrator'
from selenium import webdriver
from time import sleep

from selenium_day3.element_exit import def_element_exit

browser = webdriver.Firefox()
browser.get("http://127.0.0.1:8080/oa")
sleep(2)
# 调用函数，判断用户名输入框是否存在
username_ele =
def_element_exit.element_exit_by_xpath(browser, '//input[@class="login
Input"]')
```

```

if username_ele == True:

browser.find_element_by_xpath('//*[@class="loginInput"]').send_keys("sup")
else:
    print("用户名输入框未找到。")

# 调用函数，判断密码输入框是否存在
password_ele =
def_element_exit.element_exit_by_xpath(browser,'//input[@name="password"]')
if password_ele==True:

browser.find_element_by_xpath('//*[@name="password"]').send_keys("s1234567")
else:
    print("密码输入框未找到。。。。")

# 调用函数，判断登录按钮是否存在
login_button =
def_element_exit.element_exit_by_xpath(browser,'//input[@id="button_submit"]')
if login_button == True:

browser.find_element_by_xpath('//*[@id="button_submit"]').click()
else:
    print("登录按钮未找到。。。。")

sleep(2)

# def_element_exit.text_click_jietu(browser,"新建事项")
# def_element_exit.text_click_jietu(browser,"已发事项")
# def_element_exit.text_click_jietu(browser,"已办事项")
# def_element_exit.text_click_jietu(browser,"待发事项")
# def_element_exit.text_click_jietu(browser,"待办事项")
# def_element_exit.text_click_jietu(browser,"综合查询")
# def_element_exit.text_click_jietu(browser,"退出登录")

list1 = ["新建事","已发事项","已办事项","待发事项","待办事项","综合查询","退出登录"]
for i in list1:
    def_element_exit.text_click_jietu(browser,i)
    sleep(2)

```

```
browser.quit()
```

SELENIUM_04

01 TESTCASE

1) login_testcase

```
__author__ = 'Administrator'
import unittest
from selenium import webdriver
from time import sleep
from selenium_day3.user_login.def_user_login import
login_by_different_username_password
from selenium_day3.shijianchuo.def_shijianchuo import jietu

# 定义一个类，这个类必须继承自 unittest.TestCase 这个类
class UserLoginTestCase(unittest.TestCase):
    # 定义 setUp 这个前置方法，执行用例之前的公共操作
    def setUp(self):
        # 驱动浏览器，设置最大化
        self.browser = webdriver.Firefox()
        self.browser.maximize_window()

    # 定义 tearDown 这个后置方法，执行用例之后的公共操作
    def tearDown(self):
        # 关闭浏览器
        self.browser.quit()

    # 定义方法，方法名字以 test_ 开头，实现正常场景登录，登录完成后使用
    # assertEquals 进行断言
    def test_01_login_by_correct_username_password(self):
        try:
            login_by_different_username_password(self.browser, "sup", "s1234567")
            sleep(2)
            # 使用登录后的用户名进行断言
            username_ele =
self.browser.find_element_by_xpath('/html/body/div[1]/table/tbody/tr/
td[2]/table/tbody/tr[3]/td[1]/label')
            username = username_ele.text
            self.assertEqual(username, "苏鹏")
        except Exception as e:
            jietu(self.browser, "登录异常")
            raise e
```


定义方法，使用正确的用户名和错误的密码登录，登录完成后提示错误信息使用 assertEquals 进行断言

```
def test_02_login_by_correct_username_wrong_password(self):
    try:

login_by_different_username_password(self.browser, "sup", "s1234567888"
)

        sleep(2)
        # 使用错误提示信息的内容进行断言
        tips_ele =
self.browser.find_element_by_xpath('/html/body/div[1]/div/table/tbody
/tr[2]/td[2]/div/table/tbody/tr[2]/td/div/table/tbody/tr/td[2]')
        tips = tips_ele.text
        self.assertEqual("登录名或密码错误，请重试！", tips)
    except Exception as e:
        jietu(self.browser, "登录异常")
        raise e
```

定义方法，使用错误的用户名和正确的密码登录，登录完成后提示错误信息使用 assertEquals 进行断言

```
def test_03_login_by_wrong_username_correct_password(self):
    try:

login_by_different_username_password(self.browser, "supppp", "s1234567"
)

        sleep(2)
        # 使用错误提示信息的内容进行断言
        tips_ele =
self.browser.find_element_by_xpath('/html/body/div[1]/div/table/tbody
/tr[2]/td[2]/div/table/tbody/tr[2]/td/div/table/tbody/tr/td[2]')
        tips = tips_ele.text
        self.assertEqual("登录名或密码错误，请重试！", tips)
    except Exception as e:
        jietu(self.browser, "登录异常")
        raise e
```

定义方法，使用正确的用户名空的密码进行登录，完成后使用是否存在提示信息 assertTrue 进行断言

```
def test_04_login_by_correct_username_null_password(self):
    try:

login_by_different_username_password(self.browser, "sup", "")
        sleep(2)
```

```

        tips_ele =
self.browser.find_element_by_xpath('/html/body/div[1]/div/table/tbody
/tr[2]/td[2]/div/table/tbody/tr[2]/td/div/table/tbody/tr/td[2]')
        self.assertIsNotNone(tips_ele)
    except Exception as e:
        jietu(self.browser, "登录异常")
        raise e

```

定义方法，使用空的用户名正确的密码进行登录，完成后使用是否存在提示信息 `assertIsNotNone` 进行断言

```

def test_05_login_by_null_username_correct_password(self):
    try:

login_by_different_username_password(self.browser, "", "s1234567")
        sleep(2)
        tips_ele =
self.browser.find_element_by_xpath('/html/body/div[1]/div/table/tbody
/tr[2]/td[2]/div/table/tbody/tr[2]/td/div/table/tbody/tr/td[2]')
        self.assertIsNotNone(tips_ele)
    except Exception as e:
        jietu(self.browser, "登录异常")
        raise e

```

定义方法，假定登录按钮不可点击，输入用户名密码后获取登录按钮的点击状态，使用 `assertTrue` 进行断言

```

def test_06_login_button_clickable(self):
    try:
        self.browser.get("http://172.21.5.51:8080/oa")
        sleep(2)
        # 输入用户名

self.browser.find_element_by_xpath('//input[@class="loginInput"]').clear()

self.browser.find_element_by_xpath('//input[@class="loginInput"]').send_keys("sup")
        # 输入密码

self.browser.find_element_by_xpath('//input[@name="password"]').send_keys("s1234567")
        # 获取登录按钮的状态
        button_status =
self.browser.find_element_by_xpath('//input[@id="button_submit"]').is_enabled()

```

```

        self.assertTrue(button_status)
    except Exception as e:
        jietu(self.browser, "登录异常")
        raise e

if __name__=="__main__":
    unittest.TextTestRunner()

```

2) change_password

```

__author__ = 'Administrator'
import unittest
from selenium import webdriver
from time import sleep
from selenium_day3.user_login.def_user_login import
login_by_different_username_password
from selenium_day3.shijianchuo.def_shijianchuo import jietu
from selenium_day3.change_password.def_change_password import
change_password

class ChangePasswordTestCase(unittest.TestCase):
    def setUp(self):
        self.browser = webdriver.Firefox()
        self.browser.maximize_window()

login_by_different_username_password(self.browser, "sup", "s1234567"
)
        sleep(2)

    def tearDown(self):
        self.browser.quit()

    def test_01_wrong_old_password(self):
        try:

change_password(self.browser, "s123123123", "s1111111", "s1111111")
        sleep(2)
        # 断言

```

```

        tips =
self.browser.find_element_by_xpath('/html/body/div[1]/div/table/tb
ody/tr[2]/td[2]/div/table/tbody/tr[2]/td/div/table/tbody/tr/td[2]'
).text
        self.assertEqual("旧密码输入错误!", tips)
    except Exception as e:
        jietu(self.browser, "修改密码报错")
        raise e

def test_02_different_new_password(self):
    try:

change_password(self.browser, "s1234567", "s1111111", "s2222222")
        sleep(2)
        # 断言
        tips_ele =
self.browser.find_element_by_xpath('/html/body/div[1]/div/table/tb
ody/tr[2]/td[2]/div/table/tbody/tr[2]/td/div/table/tbody/tr/td[2]'
)
        self.assertIsNotNone(tips_ele)
    except Exception as e:
        jietu(self.browser, "修改密码报错")
        raise e

unittest.TextTestRunner()

```

02_testsuit

1) addtest

```
__author__ = 'Administrator'
import unittest
from selenium_day4.testcase.login_testcase import UserLoginTestCase

def TestCase_to_suit_by_addtest():
    # 实例化 unittest 下的 testsuit 类的对象
    testcasesuit = unittest.TestSuite()
    # 使用 testcasesuit 对象调用 addtest 的方法，将类中的用例一个一个添加到套件中

    testcasesuit.addTest(UserLoginTestCase("test_01_login_by_correct_username_password"))

    testcasesuit.addTest(UserLoginTestCase("test_02_login_by_correct_username_wrong_password"))

    testcasesuit.addTest(UserLoginTestCase("test_03_login_by_wrong_username_correct_password"))
```

```

testcasesuit.addTest(UserLoginTestCase("test_04_login_by_correct_user
name_null_password"))

testcasesuit.addTest(UserLoginTestCase("test_05_login_by_null_usernam
e_correct_password"))

testcasesuit.addTest(UserLoginTestCase("test_06_login_button_clickabl
e"))

return testcasesuit

if __name__=="__main__":
    # 实例化 unittest 下的 TextTestRunner 类的对象
    test_runner = unittest.TextTestRunner()
    # 获取测试套件
    test_suit = TestCase_to_suit_by_addtest()
    # 通过 test_runner 调用 run 的方法来执行套件
    test_runner.run(test_suit)

```

2) makesuit

```

__author__ = 'Administrator'
import unittest
from selenium_day4.testcase.login_testcase import UserLoginTestCase

def TestCase_to_suit_by_makesuit():
    # 调用 unittest 下的 makesuit 的方法将类中的用例打包
    class_testcase = unittest.makeSuite(UserLoginTestCase)
    # 调用 unittest 下的 testsuit 的方法获取套件并将打包好的用例添加到
    套件中
    test_suit = unittest.TestSuite(class_testcase)
    return test_suit

if __name__ == "__main__":
    test_runner = unittest.TextTestRunner()
    test_suit = TestCase_to_suit_by_makesuit()
    test_runner.run(test_suit)

```

3) discover

```
__author__ = 'Administrator'
import unittest

def TestCase_to_suit_by_discover():
    # 确定用例存放的路径
    testcase_path =
"D:\\zhaojia\\selenium\\selenium_105\\selenium_study\\selenium_day4\\
testcase\\"
    # 调用 unittest 下的 defaultTestLoader 方法下的 discover 方法，来获取
    测试用例并添加到测试套件中
    test_suit =
unittest.defaultTestLoader.discover(testcase_path, pattern="*.py")
    return test_suit

if __name__ == "__main__":
    test_runner = unittest.TextTestRunner()
    test_suit = TestCase_to_suit_by_discover()
```

```
test_runner.run(test_suit)
```

3) report

```
__author__ = 'Administrator'
import sys
sys.path.append("D:\\zhaojia\\selenium\\selenium_105\\selenium_study")

import HTMLTestRunner
from selenium_day4.testsuit.discover import
TestCase_to_suit_by_discover
from selenium_day3.shijianchuo.def_shijianchuo import shijianchuo

# 确定测试报告存放的路径、名字、格式
HTMLfile =
"D:\\zhaojia\\selenium\\selenium_105\\selenium_study\\report\\HTMLrep
ort" + shijianchuo() + ".html"
```



```
# 打开文件并准备写入
htmlreport_file = open(HTMLfile, "wb") # wb 表示二进制的方式打开并写入

# 获取测试套件，获取执行的用例
test_suit = TestCase_to_suit_by_discover()

# 实例化 HTMLTestRunner 下的 HTMLTestRunner 对象，需要配置参数，写清楚
# 报告的标题、描述内容
htmlrunner =
HTMLTestRunner.HTMLTestRunner(stream=htmlreport_file, title="OA 测试报
告", description="共计 8 条用例，执行结果如下：")

# 使用 runner 的对象来调用 run 的方法，执行测试套件
htmlrunner.run(test_suit)

# 关闭文件
htmlreport_file.close()
```