**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/tbody/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")]').click()

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()

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()

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/components/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/components/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/components/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 implicity\_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","s1234567")

sleep(2)

def\_user\_login.login\_by\_different\_username\_password(browser,"supppppp","s1234567")

sleep(2)

browser.quit()

1. **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="loginInput"]')

if username\_ele == True:

browser.find\_element\_by\_xpath('//input[@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('//input[@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('//input[@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assertEqual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assertEqual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assertEqual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assertIsNotNone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()

1. **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/tbody/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/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,"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\_username\_null\_password"))

testcasesuit.addTest(UserLoginTestCase("test\_05\_login\_by\_null\_username\_correct\_password"))

testcasesuit.addTest(UserLoginTestCase("test\_06\_login\_button\_clickable"))

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)

1. **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)

1. **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\\HTMLreport" + 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()