

一、基础题

选择题

1.[多选]通过自动化脚本获取元素信息的目的有哪些？

- ☒ A.web自动化测试是通过程序来代替手工验证系统功能是否和预期结果一致的过程，那么就需要通过自动化脚本来获取界面上面相关元素信息来做实际结果和预期结果做比对；
- ☒ B.获取元素信息可以作为其它脚本所需要的参数；
- ☐ C.获取元素

2.[多选]以下获取元素信息的常用方法说法正确的是？

- ☒ A.验证商品详情页商品图片规格大小是否符合需求规格说明书，可以使用element.size来获取图片大小；
- ☒ B.验证注册协议是否默认勾选，可以使用element.is_selected来判断元素是否是勾选状态；
- ☒ C.当在窗口上点击某功能后跳转或页面更新成新的信息，可以通过element.text获取比较个性的元素文本信息来判断跳转或功能处理是否正确；
- ☐ D.element.get_attribute("value")可以获取元素内属性值信息，方法中的value为属性值。

3.下面哪个为实例化鼠标操作的方法？

- ☐ A.action = ActionChains();
- ☒ B.aciton = ActionChains(driver);
- ☐ C.action = actionChains(driver);
- ☐ D.action = actionChains();

4.[多选]下面常用鼠标操作方法和说明正确匹配的是：

- ☒ A.action.double_click(element) -- 双击；
- ☐ B.action.context_click(element) -- 左键单击；
- ☒ C.action.move_to_element(element) -- 悬停；
- ☒ D.action.perform() # 执行鼠标操作；

5.[多选]元素对象send_keys方法用途包含：

- ☒ A.模拟键盘组合键操作element.send_keys(Key.键符,'字母键');
- ☒ B.对于上传文件控件，如是input标签，可以直接通过element.send_keys("文件完整路径")来进行上传；
- ☒ C.模拟键盘特殊按键操作element.send_keys(Key.键符);
- ☒ D.模拟文本输入element.send_keys("字符串")

6.[多选]关于元素等待下面说法错误的是：

- ☐ A.元素等待就是在定位页面元素时如果未找到，会在指定时间内一直等待的过程；
- ☐ B.元素等待包括隐式等待、显示等待、强制等待；
- ☐ C.网络速度慢、电脑配置低、服务器处理请求慢都会引起网页加载缓慢，从而导致元素定位不到；
- ☒ D.使用元素等待可以加快元素加载的速度；

7.[多选]关于隐式等待下面说法错误的是:

- ☒ A.隐式等待对其设置前的元素定位也会执行等待过程;
- ☐ B.隐式等待对该设置后所有元素定位都生效, 都会执行等待过程;
- ☐ C.隐式等待在超过最大时长后如果未找到对应元素, 则会抛出异常:NoSuchElementException;
- ☐ D.隐式等待的默认查找元素的间隔时长为0.5s,超时时长一般设置10~30秒;

8.[多选]关于显式等待下面说法错误的是:

- ☒ A.显式等待对该设置后所有元素定位都生效, 都会执行等待过程;
- ☐ B.显式等待只针对其WebDriverWait(driver, 10, 1).until(lambda x: x.find_element_by_xxx("value"))方法中所定位指定元素生效;
- ☐ C.显式等待在超过最大时长后如果未找到对应元素, 则会抛出异常: TimeoutException;
- ☐ D.执行显式等待的方法如能定位到元素, 最终返回的是元素对象;
- ☐ E.显式等待需要导包WebDriverWait;

9.[多选]:观察下面元素对象, 请选择说法正确的选项:

```
<select id='subject'>
  <option value='java'>java</option>
  <option value='c++'>c++</option>
  <option value='test'>test</option>
</select>
```

- ☒ A.select+option标签形式为html原生态下拉框选择形式, selenium提供了专门Select类来进行操作;
- ☒ B.Select(driver.find_element_by_id('subject')).select_by_index(1)可以选中c++;
- ☒ C.Select对象提供了select_by_value(value)的方法来选择选项, value表示option选项的value值;
- ☐ D.Select(driver.find_element_by_css_selector("[value='test']"))可以选中test;
- ☒ E.select标签下拉框同样也可以使用元素定位并点击的方式来实现选择操作;

10.[多选]:以下关于弹出框说法错误的是:

- ☒ A.所有界面弹出框都不能直接进行元素定位来实现对其操作;
- ☐ B.通过js实现的弹出框形式, 在触发后, 如不做处理其它元素都不能进行操作;
- ☐ C.通过js实现的弹出框实现方式常用三种形式为:alert 警告框、confirm 确认框、prompt提示框;
- ☐ D.通过js实现的弹出框形式都不能通过元素定位的方式进行定位;

11.[多选]:以下js弹出框处理方法错误的是:

- ☒ A.selenium3.0通过driver.switch_to_alert来获取弹出框对象;
- ☒ B.通过driver.switch_to.prompt来获取提示框对象;
- ☐ C.通过driver.switch_to.alert来获取弹出框对象;
- ☐ D.通过driver.switch_to.alert.accept() 或者driver.switch_to.alert.dismiss()来接受或者取消弹出框;

简答题

1.如何实现输入悬停到下面所描述元素对象上?

```
<div id='testA' name='testA' />
```

```
# 1. 导包
from selenium import webdriver
from selenium.webdriver import ActionChains
# 2. 实例化浏览器对象
driver = webdriver.Chrome()
# 3. 打开测试网址
driver.get('https://www.baidu.com/')
# 4. 业务操作
# 实例化鼠标对象
action = ActionChains(driver)
# 执行鼠标操作
action.move_to_element(driver.find_element_by_id('testA'))
# 执行鼠标操作
action.perform()
# 5. 关闭浏览器
driver.quit()
```

2. Select下拉框操作脚本步骤?

```
# 1. 导包
from selenium import webdriver
from selenium.webdriver.support.select import Select
# 2. 实例化浏览器对象
driver = webdriver.Chrome()
# 3. 打开测试网址
driver.get('url')
# 4. 业务操作
# 实例化下拉框对象
select = Select(driver)
# 执行下拉框操作
select.select_by_value('option选项的value属性值')
select.select_by_index('option选项的属性，起始值为0')
select.select_by_visible_text('option选项的文本信息')
# 5. 关闭浏览器
driver.quit()
```

3. 弹出框操作脚本实现步骤?

```
# 1. 导包
from selenium import webdriver
# 2. 实例化浏览器对象
driver = webdriver.Chrome()
# 3. 打开测试网址
driver.get('url')
# 4. 业务操作
# 获取弹出框对象
alert = driver.switch_to.alert
# 取消弹出框
alert.dismiss()
# 确定弹出框
alert.accept()
# 5. 关闭浏览器
driver.quit()
```

1.按以下要求完成操作流程(定位方式不限):

"""

- 1.进入tpshop首页并登陆
- 2.循环打印会员首页的会员折扣、可用积分、账户余额、优惠券信息
- 3.在会员首页顶部的搜索引擎中输入小米，点击搜索
- 4.将第一个搜索到的数据结果加入到购物车,点击继续购物车
- 5.到购物车的中获取商品数量以及总金额，需要用到判断条件，判断是否为空。

"""

```
import time
from selenium import webdriver
from selenium.webdriver import ActionChains

driver = webdriver.Chrome()
driver.implicitly_wait(30)
driver.maximize_window()
driver.get('http://localhost')

# 1.登陆
driver.find_element_by_css_selector(".red").click()
driver.find_element_by_id("username").send_keys("15800000003")
driver.find_element_by_id("password").send_keys("123456")
driver.find_element_by_id("verify_code").send_keys("8888")
driver.find_element_by_name("sbtbutton").click()
time.sleep(4)

# 2.循环打印会员首页的会员折扣、可用积分、账户余额、优惠券信息
accout_info = driver.find_elements_by_css_selector(".mu-afte span")
accout_num = driver.find_elements_by_css_selector(".mu-num")
accout_unit = driver.find_elements_by_css_selector(".mu-unit")

for i in range(len(accout_info)):
    print(accout_info[i].text, accout_num[i].text, accout_unit[i].text)

# 3.在会员首页顶部的搜索引擎中输入小米，点击搜索
driver.find_element_by_id("q").send_keys("小米")
driver.find_element_by_css_selector(".search_usercenter_btn").click()

# 4.将第一个搜索到的数据结果加入到购物车,点击继续购物车
driver.find_element_by_css_selector('[onclick*="AjaxAddCart(104)"]').click()
driver.find_element_by_id("join_cart").click()
driver.switch_to.frame(driver.find_element_by_css_selector("[id*='layui-layer-iframe']"))
driver.find_element_by_css_selector(".go-shopping").click()

# 5.到购物车的中获取商品数量以及总金额，需要用到判断条件，判断是否为空。
action = ActionChains(driver)
action.move_to_element(driver.find_element_by_xpath("//*[text()='我的购物车']"))
action.perform()

try:
    print(driver.find_element_by_css_selector("#total_pay").text,
          driver.find_element_by_css_selector("#total_qty").text)
except Exception as e:
```

```
print(driver.find_element_by_css_selector(".ma").text)

driver.quit()
```

二、提高题

代码题

1.完成下面所描述操作:

```
#1.进入tpshop首页并登陆
#2.循环打印会员首页的会员折扣、可用积分、账户余额、优惠券信息
#  会员折扣 - **折
#  可用积分 - **分
```

```
import time

from selenium import webdriver

driver = webdriver.Chrome()
driver.implicitly_wait(30)
driver.maximize_window()
driver.get('http://localhost')

# 1.登陆
driver.find_element_by_css_selector(".red").click()
driver.find_element_by_id("username").send_keys("15800000003")
driver.find_element_by_id("password").send_keys("123456")
driver.find_element_by_id("verify_code").send_keys("8888")
driver.find_element_by_name("sbtbutton").click()
time.sleep(4)

# 2.循环打印会员首页的会员折扣、可用积分、账户余额、优惠券信息
accout_info = driver.find_elements_by_css_selector(".mu-afte span")
accout_num = driver.find_elements_by_css_selector(".mu-num")
accout_unit = driver.find_elements_by_css_selector(".mu-unit")

for i in range(len(accout_info)):
    print(accout_info[i].text, accout_num[i].text, accout_unit[i].text)

driver.quit()
```