

一、基础题

选择题

1.[多选]以下场景需要通过控制滚动条才能操作有场景有哪些:

- ☒ A.今日头条首页新闻翻页加载;
- ☒ B.淘宝首页商品信息异步加载;
- ☒ C.Tpshop商城首页楼层切换按钮;
- ☒ D.微信/QQ翻看当前聊天窗口历史记录;

2.[多选]以下web自动化实现控制滚动说法正确的有哪些:

- ☒ A.selenium本身不能直接控制滚动条,但能通过执行js脚本来控制滚动条操作;
- ☒ B.webdriver提供了driver.execute_script(js)来执行js脚本,js表示javaScript脚本字符串;
- ☒ C.JavaScript中提供了window.scrollTo(x,y)来控制滚动条;
- ☒ D.需要将滚动条移动到最低部或最左侧只需要将scrollTo对应的x, y设置到足够大即可;

3.[多选]针对下面的元素请选择下面frame切换正确做法:

```
<body>
  <iframe id="iframe_a" name="login_frame" height="100%" scrolling="no"
width="100%" frameborder="0"></iframe>
  <html>
    <body>
      <div id='testA'>
        <iframe id="iframe_b" name="login_frame" height="50%" scrolling="no"
width="50%" frameborder="0"></iframe>
        <html>
          <body>
            <div id='testB'>
              </body>
            </html>
          </body>
        </html>
      <iframe id="iframe_c" src="c_iframe.htm" width="200" height="200"></iframe>
    </body>
  </html>
```

- ☒ A.从默认界面切换到id='iframe_a'的
frame:driver.swtich_to.frame(driver.find_element_by_id('iframe_a'));
- ☒ B.从默认界面切换到id='iframe_b'的
frame:driver.swtich_to.frame(driver.find_element_by_id('iframe_b'));
- ☒ C.从iframe_a切换到iframe_c:driver.swtich_to.frame(driver.find_element_by_id('iframe_c'));
- ☒ D.从iframe_b切换到默认页面: driver.switch_to.default_content()

4.[多选]针对下面的元素请选择下面frame切换正确做法:



```
<body>
  <iframe id="login_frame" name="login_frame" height="100%" scrolling="no"
width="100%" frameborder="0"></iframe>
    <html>
      <html>...</html>
      <body>
        <div class="bottom hide" id="bottom_qlogin" style="display:
block;">
          <a class="link" id="switcher_plogin"
href="javascript:void(0);" tabindex="8">帐号密码登录</a>
          <span class="dotted" id="docs_dotted">|</span>
          <a href="https://qzs.qq.com/qzone/v6/reg/index.html"
class="link" target="_blank">注册新帐号</a>
          <span class="dotted">|</span>
          <a class="link" id="feedback_qlogin"
href="https://support.qq.com/products/14800" target="_blank">意见反馈</a>
        </div>
      </body>
    </html>
</body>
```



点击账户密码登陆

```
driver.switch_to.frame(driver.find_element_by_id('login_frame'))
driver.find_element_by_id("switcher_plogin").click
```



点击注册新账户

```
driver.find_element_by_xpath("//*[text()='注册新账号']")
```



点击账户密码登陆

```
driver.switch_to.frame(driver.find_element(By.NAME, 'login_frame'))
driver.find_element_by_css_selector("#switcher_plogin").click
```



点击意见反馈

```
driver.switch_to_frame(driver.find_element_by_css_selector("#login_frame"))
driver.find_element_by_id("feedback_qlogin").click
```

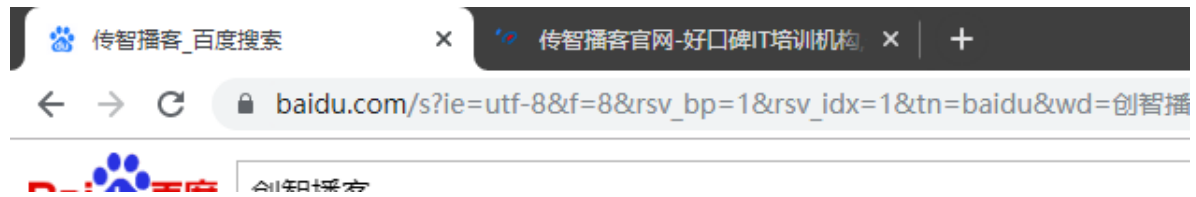
5.[多选]下面对于多窗口描述正确的是:



A.使用自动化脚本打开浏览器A页面后,对A页面业务功能按钮执行了点击事件,触发了新的页面窗口打开,不做窗口切换,可以直接对新页面窗口的元素进行操作;

- ☒ B.selenium通过句柄来实现driver传递;
- ☒ C.句柄:handler, 是浏览器窗口的唯一标识码; 每个句柄可以对应一个页面对象。
- ☐ D.使用自动化脚本打开浏览器A页面后, 对页面业务功能按钮执行了点击事件, 触发了页面刷新, 也需要做窗口切换。

6.[多选]下面界面有两个窗口, 对于窗口切换方式使用错误是:



- ☒ A.需要切换到传智播客官网, 需要先获取到所有窗口句柄信息:driver.window_handles(), 返回一个列表数据
- ☐ B.需要切换到传智播客官网, 完成切换操作:driver.switch_to.window(driver.window_handles[-1])
- ☐ C.需要切换到传智播客官网, 完成切换操作:driver.switch_to.window(driver.window_handles[1])
- ☒ D.需要切换到传智播客官网, selenium3完成切换操作:driver.switch_to_window(driver.window_handles[1])

7.[多选]关于窗口截图说法正确的是:

- ☒ A.在执行出错的时候对当前窗口截图保存, 那么通过图片就可以非常直观地看到出错的原因;
- ☒ B.selenium通过driver.get_screenshot_as_file("文件的完整路径名")来实现窗口截图;
- ☒ C.selenium实现窗口截图其截图文件的存放路径必须提前手工先创建好;
- ☒ D.可以通过随机取值、获取当前系统时间的方式来参数化文件名, 解决文件重名的问题;

8.[多选]下面对验证码作用描述正确是:

- ☒ A.防止恶意注册;
- ☒ B.防止某个黑客对某一个特定注册用户用特定程序暴力破解验证码进行不断的登录尝试;
- ☐ C.为了防止正常用户正常登陆;
- ☒ D.增加应用安全性, 降低服务器处理垃圾请求的压力;

9.[多选]web项目验证码常见处理方式有:

- ☒ A.去掉验证码, 通常在测试环境应用;
- ☒ B.万能验证码;
- ☒ C.验证码识别技术;
- ☒ D.cookie技术;

10.[多选]下面说法正确是:

- ☐ A.单元测试一定要使用单元测试框架;
- ☒ B.使用单元测试框架来组织自动化脚本是为了更好的用例组织和执行;
- ☒ C.单元测试框架提供了丰富的断言方法, 且能更方便的生成测试报告;
- ☒ D.框架是一个半成品, 已经对基础的代码进行了封装并提供相应的API, 开发者在使用框架是直接调用封装好的api可以省去很多代码编写, 从而提高工作效率和开发速度;

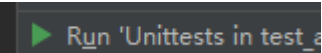
11.[多选]下面对于unittest核心要素说法正确的是:

- ☒ A.TestCase表示测试用例;
- ☒ B.TestSuite表示测试测试套件, 即多个测试用例的集合, 能通过组织测试套件来方便批量执行多个测试用例;
- ☒ C.TextTestRunner表示测试运行器, 能用来运行测试套件;
- ☒ D.TestSuite和TextTestRunner需要配套使用才能执行测试用例;

12.对于组织TestCase说法有误的是:

- ☐ A.定义测试用例类类名需要使用驼峰命名法;
- ☐ B.定义测试用例类名称必须继承unittest.TestCase类;
- ☐ C.测试类中的测试方法其方法名必须以小写的test开头;
- ☒ D.测试类中的测试方法可以重名;

13.对于运行TestCase错误的是:

- ☐ A.使用unittest组织的测试用例类在运行时选择unittest框架运行 
- ☒ B.可以使用python解释器直接运行"Run 文件名";
- ☐ C.使用unittest组织的测试用例可以通过光标锁定到指定的测试方法名上, 单独运行该测试方法;
- ☐ D.当测试类中包含多个测试方法且未使用ASCII码进行排序时, 运行测试类, 测试方法的执行顺序可能是乱序的;

14.[多选]下面unittest组织的代码错误是:

- ☐

```
"""组织一个测试用例"""
import unittest
class TestDemo(unittest.TestCase):
    def test_demo(self):
        pass
```
- ☒

```
"""实例化测试套件"""
import unittest
import ClassName#导入测试类名
suite = unittest.TestSuite
suite.addTest(unittest.makeSuite(ClassName))
```
- ☒

```
"""组织一个测试用例"""
import unittest
class TextDemo(unittest.TestCase):
    def text_demo(self):
        print("第一个测试用例")
```
- ☒

```
"""实例化测试套件"""
"""定义本文件时文件名为:unittest"""
import unittest
from filename import ClassName#导入测试类名
suite = unittest.TestSuite()
suite.addTest(unittest.makeSuite(ClassName))
```

简答题

1.使用测试套件和测试运行器组织多个测试用例执行的实现步骤：

```
# 1.导包
import unittest
# 2.实例化测试套件
suite = unittest.TestSuite()
# 3.添加测试用例，测试类需要执行导包
suite.addTest(TestDivide("test_1_divide_01")) # 添加单个测试方法
suite.addTest(unittest.makeSuite(TestDivide01)) # 添加指定测试类
# 4.实例化运行器
runner = unittest.TextTestRunner()
# 5.运行测试套件
runner.run(suite)
```

2.如何定义测试用例并制定测试方法执行顺序：

```
#1.导包
import unittest
#2.定义测试类
class TestDemo(unittest.TestCase):
    #3.定义测试方法
    def test_01_demoA(self):
        print("测试方法一")
    def test_02_demoB(self):
        print("测试方法二")
```

二、提高题

1.按以下要求完成新增地址操作：

```
"""
1.要求使用unittest组织测试用例
2.进入tpshop首页并登陆
3.进入地址管理界面
4.获取当前地址条数信息
5.新增一条地址
6.获取新增成功后地址条数信息，判断新增前后地址条数是否相等如不相等则打印"新增地址成功"否则打印"新增地址失败"
"""
```

```
import time
from selenium import webdriver
from selenium.webdriver.support.select import Select
import unittest

# 定义测试类
class TestAddAddress(unittest.TestCase):

    # 定义新增地址方法
    def test_add_address(self):
        # 实例化浏览器驱动
```

```
driver = webdriver.Chrome()
# 设置隐式等待
driver.implicitly_wait(30)
# 窗口最大化
driver.maximize_window()
driver.get('http://localhost')
# 执行登陆操作
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)
# 在会员中心点击地址管理
driver.find_element_by_xpath("//*[text()='地址管理']").click()
# 获取新增地址前地址条数
num_old = driver.find_elements_by_css_selector(".gp_num2 .red")[0].text
# 点击新增地址
driver.find_element_by_css_selector("a .co_blue").click()
# frame切换
driver.switch_to.frame(driver.find_element_by_css_selector("[id*='layui-
layer-iframe']"))
# 输入联系人信息
driver.find_element_by_name("consignee").send_keys("test-customer001")
# 选择省份
province = Select(driver.find_element_by_id("province"))
province.select_by_value("1")
# 选择城市
city = Select(driver.find_element_by_id("city"))
city.select_by_value("2")
# 选择区
district = Select(driver.find_element_by_id('district'))
district.select_by_value("3")
# 选择城镇
twon = Select(driver.find_element_by_id("twon"))
twon.select_by_value("4")
# 输入其它信息
driver.find_element_by_id('address').send_keys("address")
driver.find_element_by_name('mobile').send_keys("15800000093")
driver.find_element_by_css_selector("[type='submit']").click()
# 获取新增后的地址
num_new = driver.find_elements_by_css_selector(".gp_num2 .red")[0].text
print("新增{}条地址".format(int(num_new) - int(num_old)))

# 判断新增前后地址条数是否相等如不相等则打印"新增地址成功"否则打印"新增地址失败"
if num_old == num_new:
    print("新增地址失败")
else:
    print("新增地址成功")
# 关闭浏览器驱动
driver.quit()
```