**1.你如何用Selenium测试？**

**Selenium+Maven+TestNG+Jekins**

**2.如何解决问题？**

**先思考，然后百度，考虑网速、电脑配置等原因，这题主要看重解决问题的能力和思维。**

**3.你是怎么开发测试框架的？**

**Selenium+Maven+TestNG+Jekins**

**4.做自动化测试的目的？**

**a.用来回归 b.用来监控系统稳定性 c.提高工作效率**

**5.PO模型是怎么做的？**

**pageObject ：页面层，操作层，业务层集合成case**

**6.如何判断case是否通过？**

**断言，与预期结果比较**

**7.你项目中有多少case，通过率怎么样？**

**按实际回答，上千个，一般是70-80%左右，**

**8.失败截图的原理是什么？**

**利用监听，重写TestNG的失败方法进行截图。**

**9.报告哪里来，主要统计什么？**

**报告由测试用例执行情况，如TestNG生成，主要统计成功、失败率，截图和记录日志。**

**10.testNG中注解有哪些**

**@test @BeforeSuite @AfterSuite**

**@BeforeTest @AfterTest**

**@BeforeClass @AfterClass**

**@BeforeMethod @AfterMethod**

**11.TestNG有哪些优点**

**TestNG的注释非常强大和方便**

**TestNG允许并行测试**

**TestNG可以给予group方式执行测试**

**TestNG可以生成测试报表，并且报表展示多样化，可以在其基础上自行开发测试报告，插件以及测试监听器等**

**12.什么是web自动化**

**指的是从UI界面，也就是用户界面层面进行的自动化测试，测试人员通过编程自动化的程序也就是自动化脚本实现打开界面，操作界面的业务逻辑**

**13.写出selenium 中你最熟悉的接口或者类**

**WebDriver InternetExplorerDriver FirefoxDriver ChromeDriver WebElement WebDriverWait By**

**14.Xpath 是什么？**

**是web页面元素定位的一种方式，相当于元素dom中的一个路径，可以采用绝对路径和相对路径的方式进行定位，对于定位页面元素有很大的帮助，同时需谨慎使用，当页面结果发生变化，那么定位信息也可能需要变化**

**15.findElement()和findElements()方法有什么区别**

**两个都是查找元素的方式**

**findElement()查找一个页面元素，只会返回WebElement对象**

**findElements()查找页面是所有符合匹配的元素，返回元素集合**

**16.登录按钮除了click方法还有其他方式吗？**

**submit() ,前提是input元素的type 为submit**

**17.简述你知道的延迟等待方式**

**硬性等待，也叫线程等待，通过休眠的方式完成等待如等待5秒Thead.sleep(5000)**

**隐式等待，通过imlicitlyWait完成延时等待，这种事针对全局设置的等待，如设置超市10秒，使用imlicitlyWait后，如果第一次没有找到元素，会在10秒之内不断循环查找元素，如果超时间10秒还没有找到，则抛出异常**

**显式等待，智能等待，针对指定元素定位指定等待时间，指定的范围内进行元素查找，找到元素则直接返回，超时没有找到元素则抛出异常**

**18.怎么对含有验证码的功能进行自动化测试**

**图像识，难度大，效果不好不推荐**

**屏蔽验证码，邀请开发处理，在测试环境，预发和正式环境恢复**

**万能验证码，使用复杂的其他人无法猜到的验证码**

**19.如何验证复选按钮是不是被选中**

**可以使用元素的is selected()方法，如果返回的是true则说明被选中，否则表明未被选中**

**20.如何处理alert 弹窗？**

**先定位跳转到alert上，然后点击确认或取消**

**Alert alert = dirver.swichTo().alert();//切换到alert**

**alert.accept();//确定**

**alert.dismiss();//**

**21.如何用webdriver进行鼠标右键点击操作**

**使用Actions 类**

**Actions actions = new Actions(driver)**

**actions.moveToElement(element).perform();**

**actions.contextClick.perform();**

**22.如何使用webdriver 执行拖拽 的操作**

**还是使用actions类**

**Actions actions = new Actions(driver)**

**WebElement element1 = driver.findElement(by);**

**WebElement element2 = driver.findElement(by);**

**actions.dragAndDrop(element1,element2)perform();**

**23.举一个WebDriver中方法重载的例子**

**frame(string, name), frame(int index), frame(Web Element element)**

**24.下拉菜单如何选择一个菜单项？**

**如果下拉菜单是select 标签，使用方法；select By Value() 或者selectByIndex()或者select ByVisibleText()即可**

**如果这个下拉菜单不是通过select标签创建，则直接通过xpath 定位元素然后去点击选择**

**25.你如何模拟浏览器的前后移动？**

**driver.navigate().back();**

**driver.navigate().forward();**

**26.你如何获得当前页面的URL？**

**driver.getCurrentUrl();**

**27.如何在WebDriver中实现同步？**

**我们可以使用隐式等待。**

**语法 - driver.manage（）。timeouts（）。implicitlyWait（10，TimeUnit.SECONDS）;**

**如果执行驱动程序没有立即在页面中找到元素，它将等待10秒。此代码将自动附加到脚本的每一行。每次都不需要写。打开浏览器后只需编写一次。**

**28. 如何从文本框中获取打字文本？**

**通过将arg作为值传递来使用getAttribute（“value”）方法。**

**String typedText = driver.findElement(By.xpath(“xpath of box”)).getAttribute(“value”));**

**29. 你如何清除中文本框的内容？**

**使用clear（）方法。**

**driver.findElement(By.xpath(“xpath of box”)).clear();**

**30.如何在webdriver中调用应用程序？**

**driver.get(“url”); or driver.navigate().to(“url”);**

**31. 如何获取页面上的帧数？**

**List <WebElement> framesList = driver.findElements(By.xpath("//iframe"));**

**int numOfFrames = frameList.size()**

**————————————————**

**问题1：Selenium是什么，流行的版本有哪些？**

**是一个开源的web自动化测试的框架，支持多种编程语言，支持跨浏览器平台进行测试。**

**Selenium 1.0或Selenium RC**

**Selenium 2.0或Selenium Webdriver**

**Selenium 3.0**

**问题2：你如何从命令行启动Selenium RC？**

**java -jar selenium-server.jar**

**// 在浏览器中运行一套Selenese脚本**

**java -jar selenium-server.jar -htmlSuite**

**问题3：在我的机器端口4444不是免费的。我怎样才能使用另一个端口？**

**//你可以在运行selenium服务器时指定端口为 -**

**Java -jar selenium-server.jar -port 5555**

**问题4：什么是Selenium Server，它与Selenium Hub有什么不同？**

**Selenium Server是使用单个服务器作为测试节点的一个独立的应用程序。 Selenium hub代理一个或多个Selenium的节点实例。一个hub 和多个node被称为Selenium grid。运行SeleniumServer与在同一主机上用一个hub和单个节点创建de Selenium grid类似。**

**问题5：你如何从Selenium连接到数据库？**

**Selenium是一个Web UI自动化工具。它不提供任何API来建立数据库连接。这取决于你使用Selenium进行自动化的编程语言。在下面的例子中，我们假设正在使用Java。**

**一个Connection对象表示与数据库的连接。当我们使用连接方法连接到一个数据库时，我们创建了一个连接对象，它代表了与数据库的连接。单个数据库可能有一个连接或多个连接，还可能有多个连接到不同的数据库上。**

**我们可以使用Connection对象来做以下事情：**

**创建用于执行SQL语句的Statement，PreparedStatement和CallableStatement对象。**

**可以帮助我们提交或回滚一个JDBC事务。**

**如果你想知道连接到的数据库或数据源信息，Connection对象通过使用DatabaseMetaData就可以收集有关数据库或数据源的信息。**

**可以帮助我们关闭数据源。Connection.isClosed() 方法只有在调用了Connection.close()时才返回true 。此方法用于关闭所有连接。**

**首先我们需要通过使用DriverManager.getConnection()方法，建立与数据库的连接。这个方法接受一个包含URL的字符串。DriverManager类尝试查找可以连接到由字符串URL表示的数据库的驱动程序。每当调用getConnection()方法时，DriverManager类都会检查可以连接到URL中指定的数据库的所有已注册的Driver类的列表。**

**句法：**

**String url = “jdbc: odbc: makeConnection”;**

**Connection con = DriverManager.getConnection(url, “userID”, “password”);**

**问题6：Selenium RC中定位器方法有哪些？**

**ID Name CSS XPATH Dom**

**问题7：你如何验证多个页面上存在的一个对象？**

**可以使用下面的Selenium命令来检查：**

**assertTrue(selenium.isElementPresent(locator));**

**问题8：XPath中使用单斜杠和双斜杠有什么区别？**

**如果XPath是从文档节点开始，它将允许创建“绝对”路径表达式。**

**例如 “/ html / body / p”匹配所有的段落元素。**

**如果XPath在文档中的任意位置开始进行选择匹配，那么它将允许创建“相对”路径表达式。**

**例如 “// p”匹配所有的段落元素。**

**问题9：如何编写Selenium IDE / RC的用户扩展？**

**用户扩展（UX）存储在Selenium IDE或Selenium RC用来激活扩展的单独文件中。它包含用JavaScript编写的函数定义。**

**因为Selenium的核心是用JavaScript开发的，所以要符合原语言的标准规则来创建扩展。要创建一个扩展，我们必须用下面的设计格式来编写函数。**

**// 样例**

**Selenium.prototype.doFunctionName = function(){}**

**函数名称前面的“do”告诉Selenium这个函数可以被调用为一个步骤命令，而不是作为内部函数或私有函数被调用。**

**问题10：如何在页面加载成功后验证元素的存在？**

**它可以通过下面的代码行来实现。**

**只需一点时间（以秒为单位）来检查元素，如下所示：**

**public void waitForElementPresent(String element, int timeout) throws Exception {**

**for (int second = 0;; second++) {**

**if (second >= timeout)**

**fail(“Timeout. Unable to find the Specified element” + element);**

**try {**

**if (selenium.isElementPresent(element))**

**break;**

**} catch (Exception e) {**

**}**

**Thread.sleep(1000);**

**}}**

**问题11：你对Selenium Grid有什么了解？它提供了什么功能？**

**Selenium Grid是一款利用现有计算基础架构大幅加速Web应用程序功能测试的工具。允许测试者轻松地在多台机器上并行运行多个测试，并且可以在异构环境中运行。**

**基于优秀的Selenium Web测试工具，Selenium Grid允许测试者并行运行多个Selenium Remote Control实例。更好的是，它集成显示所有Selenium远程控制，所以不必担心实际的基础设施。Selenium Grid将运行Selenium测试套件所需的时间，缩短到Selenium实例的单个实例运行时间的一小点。**

**问题12：如何从你的Java Class启动Selenium服务器？**

**try {**

**seleniumServer = new SeleniumServer();**

**seleniumServer.start();**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**问题13：Selenium中有哪些验证点？**

**Selenium主要有三种验证点 -**

**检查页面标题**

**检查某些文字**

**检查某些元素（文本框，下拉菜单，表等）**

**问题14：什么是XPath？什么时候应该在Selenium中使用XPath？**

**XPath是一种在HTML / XML文档中定位的方法，可用于识别网页中的元素。如果没有与页面上的元素相关联的名称/ ID，或者名称/ ID的一部分是常量，则必须使用XPath。**

**绝对路径用 - / 单斜杠**

**相对路径用 - // 双斜杠**

**ID，类，名称也可以用于XPath ：**

**//input[@name=’q’]**

**//input[@id=’lst-ib’]**

**//input[@class=’ lst’]**

**如果id / name / class的一部分是常量 ：**

**//input[contains(@id,’lst-ib’)**

**问题15：什么是Selenium的CSS定位器策略？用例子来解释。**

**CSS位置策略可以与Selenium一起使用来定位元素，它使用CSS定位方法，其中 -**

**绝对路径用 - （空格符号）**

**相对路径用 - >表示**

**ID，类，名称也可以用于XPath：**

**css=input[name=’q’]**

**css=input[id=’lst-ib’] or input#lst-ib**

**css=input[class=’lst’] or input.lst**

**如果id / name / class只有一部分是常量：**

**css=input[id\*=’lst-ib’)]**

**使用内部文本的元素位置策略：**

**css = a:contains(‘log out’)**

**问题17：在处理多个弹出窗口的机制是什么？**

**可以使用命令getWindowHandles()来处理多个弹出窗口。**

**然后将所有窗口名称存储到Set变量中并将其转换为数组。**

**接下来，通过使用数组索引，导航到特定的窗口。**

**driver.switchTo().window(ArrayIndex);**

**问题18：你如何处理使用Selenium的Ajax控件？**

**来看一个例子。假如一个文本框是一个Ajax控件，当我们输入一些文本时，它会显示自动建议的值。**

**处理这样的控件，需要在文本框中输入值之后，捕获字符串中的所有建议值；然后，分割字符串，取值就好了。**

**问题19：Selenium Webdriver优于Selenium RC的优点是什么？**

**Selenium RC的架构相当复杂，WebDriver的架构比Selenium RC简单些。**

**Selenium RC比较慢，因为它使用了另外一个名为Selenium Core的JavaScript程序。相反，WebDriver比Selenium RC更快，因为它直接与浏览器对话，并使用浏览器自己的引擎来进行控制。**

**像其他JavaScript代码一样，Selenium Core可以访问禁用的元素。Webdriver以更现实的方式与页面元素进行交互。**

**Selenium RC的API集已经有所改进，但是仍有经常让人困惑的冗余部分。WebDriver API更简单，不包含任何冗余或混淆的命令。**

**Selenium RC无法支持无头HtmlUnit浏览器。它需要一个真正的、可见的浏览器来操作。Web Driver可以支持无头HtmlUnit浏览器。**

**Selenium RC内置了测试结果生成器，并自动生成测试结果的HTML文件。Web驱动程序没有自动生成测试结果文件的内置命令。**

**问题20：“GET”和“NAVIGATE”方法的主要区别是什么？**

**Get方法能获得一个页面进行加载、或获取页面源代码、或获取文本，就这三。而Navigate将通过刷新，回退，前进的方式导航。**

**Partial Link Text.部分链接文本这可以通过调用< navigate()>方法来实现。**

**driver.get() 方法会等到整个页面被加载后才可以，而driver.navigate()只是重定向到该网页，并不会等待。**

**问题21：隐式等待与显式等待有什么不同？**

**隐式等待是设置的全局等待，分为1、页面加载超时等待 ；2、页面元素加载超时；3、异步脚本超时。如果是页面元素超时，设置等待时间，是对页面中的所有元素设置加载时间。隐式等待是其实可以理解成在规定的时间范围内，浏览器在不停的刷新页面，直到找到相关元素或者时间结束。**

**显式等待只是用于特定搜索的一个计时器。它的可扩展性更强，你可以设置它来等待任何条件。通常情况下，可以使用一些预先构建的条件来等待元素变得可点击，可见，不可见等，或者只是编写适合需求的条件。**

**问题22：如何解决IE中的SSL认证问题？**

**// 打开浏览器后添加下面的命令**

**driver.navigate().to(“javascript:document.getElementById(‘overridelink’).click()”);**

**问题23：如何处理WebDriver中的AJAX控件？**

**AJAX代表异步JavaScript和XML。它不依赖于创建有效的XML所需的打开和关闭标签的额外开销。大部分时间WebDriver自动处理Ajax控件和调用。如果不能处理的话，可以按照下面的方式来处理。**

**//Waiting for Ajax Control**

**WebElement AjaxElement = (new WebDriverWait(driver,**

**10)).until(ExpectedConditions.presenceOfElementLocated(By.("")));**

**问题24：如何在标题菜单的子菜单项上执行鼠标移动操作？**

**应该首先移动菜单标题，然后移至弹出菜单项并单击它。不要忘记在最后调用actions.perform() 。**

**问题25：什么是数据驱动框架？它与关键字驱动框架有什么不同？**

**数据驱动框架。**

**在这个框架中，测试用例逻辑驻留在测试脚本中。测试数据被分离并保存在测试脚本之外。测试数据是从外部文件（Excel文件）中读取的，并被加载到测试脚本中的变量中。变量用于输入值和验证值。**

**关键字驱动。**

**关键字/表驱动框架需要开发数据表和关键字。它们独立于执行它们的测试自动化工具。可以使用或不使用应用程序来设计测试。在关键字驱动的测试中，被测试的应用程序的功能记录在一个表格中，以及每个测试的分步说明。**

**问题26：解释使用TestNG而不是JUnit框架的好处？**

**TestNG相较于Junit的优势：**

**在JUnit中，我们必须声明@BeforeClass和@AfterClass，这是JUnit中的一个约束，而在TestNG中没有像这样的约束。**

**TestNG提供了更多的setUp / tearDown级别。1.@ Before/AfterSuite 2.@Before/AfterTest 3.@Before/AfterGroup**

**TestNG中不需要扩展任何类。**

**TestNG中没有方法名称约束，就像JUnit一样。**

**在TestNG中，我们可以告诉测试一个方法依赖于另一个方法，而在JUnit中这是不可能的。**

**测试用例的分组在TestNG中可用，而JUnit中则不可用。执行可以基于组完成。例如，如果你已经定义了许多案例，并通过将2个组分别定义为“离职“与”回归”隔离。如果你只是想执行“理智”的情况，那就告诉TestNG执行“理智”。TestNG将自动执行属于“离职”组的案例。**

**另外，TestNG支持并行测试用例执行。**

**问题27：与@Test注释相关的TestNG参数的目的是什么？**

**在TestNG中，参数是修改注释功能的关键字。**

**问题28：可以使用TestNG运行一组测试用例吗？**

**是的，TestNG框架支持在测试组的帮助下执行多个测试用例。**

**它提供了以下选项来运行特定组中的测试用例。**

**如果想基于回归测试或冒烟测试等其中一个组来执行测试用例，那么：**

**@Test(groups = {“regression-tests”, “smoke-tests”})**

**————————————————**

1.你如何用Selenium测试？

Selenium+Maven+TestNG+Jekins

2.如何解决问题？

先思考，然后百度，考虑网速、电脑配置等原因，这题主要看重解决问题的能力和思维。

3.你是怎么开发测试框架的？

Selenium+Maven+TestNG+Jekins

4.做自动化测试的目的？

a.用来回归 b.用来监控系统稳定性 c.提高工作效率

5.PO模型是怎么做的？

pageObject ：页面层，操作层，业务层集合成case

6.如何判断case是否通过？

断言，与预期结果比较

7.你项目中有多少case，通过率怎么样？

按实际回答，上千个，一般是70-80%左右，

8.失败截图的原理是什么？

利用监听，重写TestNG的失败方法进行截图。

9.报告哪里来，主要统计什么？

报告由测试用例执行情况，如TestNG生成，主要统计成功、失败率，截图和记录日志。

10.testNG中注解有哪些

@test @BeforeSuite @AfterSuite

@BeforeTest @AfterTest

@BeforeClass @AfterClass

@BeforeMethod @AfterMethod

11.TestNG有哪些优点

TestNG的注释非常强大和方便

TestNG允许并行测试

TestNG可以给予group方式执行测试

TestNG可以生成测试报表，并且报表展示多样化，可以在其基础上自行开发测试报告，插件以及测试监听器等

12.什么是web自动化

指的是从UI界面，也就是用户界面层面进行的自动化测试，测试人员通过编程自动化的程序也就是自动化脚本实现打开界面，操作界面的业务逻辑

13.写出selenium 中你最熟悉的接口或者类

WebDriver InternetExplorerDriver FirefoxDriver ChromeDriver WebElement WebDriverWait By

14.Xpath 是什么？

是web页面元素定位的一种方式，相当于元素dom中的一个路径，可以采用绝对路径和相对路径的方式进行定位，对于定位页面元素有很大的帮助，同时需谨慎使用，当页面结果发生变化，那么定位信息也可能需要变化

15.findElement()和findElements()方法有什么区别

两个都是查找元素的方式

findElement()查找一个页面元素，只会返回WebElement对象

findElements()查找页面是所有符合匹配的元素，返回元素集合

16.登录按钮除了click方法还有其他方式吗？

submit() ,前提是input元素的type 为submit

17.简述你知道的延迟等待方式

硬性等待，也叫线程等待，通过休眠的方式完成等待如等待5秒Thead.sleep(5000)

隐式等待，通过imlicitlyWait完成延时等待，这种事针对全局设置的等待，如设置超市10秒，使用imlicitlyWait后，如果第一次没有找到元素，会在10秒之内不断循环查找元素，如果超时间10秒还没有找到，则抛出异常

显式等待，智能等待，针对指定元素定位指定等待时间，指定的范围内进行元素查找，找到元素则直接返回，超时没有找到元素则抛出异常

18.怎么对含有验证码的功能进行自动化测试

图像识，难度大，效果不好不推荐

屏蔽验证码，邀请开发处理，在测试环境，预发和正式环境恢复

万能验证码，使用复杂的其他人无法猜到的验证码

19.如何验证复选按钮是不是被选中

可以使用元素的is selected()方法，如果返回的是true则说明被选中，否则表明未被选中

20.如何处理alert 弹窗？

先定位跳转到alert上，然后点击确认或取消

Alert alert = dirver.swichTo().alert();//切换到alert

alert.accept();//确定

alert.dismiss();//

21.如何用webdriver进行鼠标右键点击操作

使用Actions 类

Actions actions = new Actions(driver)

actions.moveToElement(element).perform();

actions.contextClick.perform();

22.如何使用webdriver 执行拖拽 的操作

还是使用actions类

Actions actions = new Actions(driver)

WebElement element1 = driver.findElement(by);

WebElement element2 = driver.findElement(by);

actions.dragAndDrop(element1,element2)perform();

23.举一个WebDriver中方法重载的例子

frame(string, name), frame(int index), frame(Web Element element)

24.下拉菜单如何选择一个菜单项？

如果下拉菜单是select 标签，使用方法；select By Value() 或者selectByIndex()或者select ByVisibleText()即可

如果这个下拉菜单不是通过select标签创建，则直接通过xpath 定位元素然后去点击选择

25.你如何模拟浏览器的前后移动？

driver.navigate().back();

driver.navigate().forward();

26.你如何获得当前页面的URL？

driver.getCurrentUrl();

27.如何在WebDriver中实现同步？

我们可以使用隐式等待。

语法 - driver.manage（）。timeouts（）。implicitlyWait（10，TimeUnit.SECONDS）;

如果执行驱动程序没有立即在页面中找到元素，它将等待10秒。此代码将自动附加到脚本的每一行。每次都不需要写。打开浏览器后只需编写一次。

28. 如何从文本框中获取打字文本？

通过将arg作为值传递来使用getAttribute（“value”）方法。

String typedText = driver.findElement(By.xpath(“xpath of box”)).getAttribute(“value”));

29. 你如何清除中文本框的内容？

使用clear（）方法。

driver.findElement(By.xpath(“xpath of box”)).clear();

30.如何在webdriver中调用应用程序？

driver.get(“url”); or driver.navigate().to(“url”);

31. 如何获取页面上的帧数？

List <WebElement> framesList = driver.findElements(By.xpath("//iframe"));

int numOfFrames = frameList.size();

————————————————

1.Selenium是否支持桌面应用软件的自动化测试。

Selenium不支持桌面软件的自动化测试，Selenium是根据网页元素的属性才定位元素，而其他桌面软件自动化测试工具是根据桌面元素的位置来定位元素，当然现在也有根据桌面元素的属性来定位的。

2.Selenium是否支持用例的执行的引擎。

引擎好比就是一个发动机。Selenium是没有关于测试用例和测试套件管理和执行的模块。我们需要借助第三方单元测试框架来实现用例管理和用例的执行。例如Java中有Junit或者testNG,Python中有unittest单元测试框架。

3.Seleinum是否有读取excel文件的库

没有，这里需要用到第三方工具。例如Apache POI插件。

4.Selenium有哪些组件？

最早的有Selenium IDE,IDE只支持安装在fiefox上一个插件，支持录制自动化脚本。还有

remote RC,和Grid 和webdriver。我们一般最重要的就是使用webdriver。

5.Selenium有什么限制或者缺陷

除了基于web的软件和mobile的程序，selenium不支持桌面软件自动化测试。软件测试报告，和用例管理只能依赖第三方插件，例如Junit/TestNG和unittest。由于它是免费的软件，所以没有供应商去提供支持和服务，有问题，只能求助selenium社区。还有一个就是，selenium入门门槛可能有点高，需要具备一定编程语言基础的才能玩转。

加入我们。群，642830685，领取最新软件测试资料大厂面试和Python自动化、接口、框架搭建学习资料！同行一起交流，一起学习进步！

6.在selenium中，有哪些不同定位元素方法

ID/className/Name/LinkText/PartialLinkText/Xpath/CSS selector

7.什么是imlicitlyWait(隐式等待)

webdriver 会在指定的超时时间范围内不断的查找元素，直到找到元素或超时

8.什么是expliciteWait(显式等待)

通常是我们自定义的一段代码，这段代码用来等待某种条件发生后，再继续执行后续的代码

9.什么是线程等待(硬式等待)

如Thread.sleep(2000),webdriver线程强制休眠2秒钟，2秒过后，再执行后续的代码。

10.什么是pollingEvery

这个是设置个一段时间就去做一件事，例如下面设置隔一秒就去查找元素一次。

WebDriverWait wait = new WebDriverWait(driver,30);

wait.pollingEvery(1, TimeUnit.SECONDS);

driver.findElement(By.xpath(“xxxx”));

11.你写的测试脚本能在不同浏览器上运行吗，支持跨浏览器平台吗

是的，我写的测试用例能在IE，火狐和谷歌这三种浏览器上运行。主要是在windows平台上运行脚本，所以mac的safari浏览器暂时没有写过。主要实现这个跨浏览器的思想就是，把浏览器类型写到配置文件，代码里写if语句去判断配置文件的浏览器的类型，来决定用什么浏览器去执行测试用例。

12.什么是POM，为什么要使用它

POM是Page Object Model的简称，它是一种设计思想，而不是框架。大概的意思是，把一个一个页面，当做一个对象，页面的元素和元素之间操作方法就是页面对象的属性和行为，所以自然而然就用了类的思想来组织我们的页面。一般一个页面写一个类文件，这个类文件包含该页面的元素定位和业务操作方法

为了我们测试用例写的简单，清晰，我们很多时候在页面对象会封装很多业务操作方法，测试脚本只需要调用相关方法就可以。

还有一个可能和这个问题相关的面试题，如果页面元素经常发生需求变化，你是如何做，答案就是采用POM思想。好处就是只要该一个页面，我就去修改这个页面对象的元素定位和相关方法，脚本不需要修改。

13.在你做自动化过程中，遇到了什么问题吗？举例下

这个问题，不管是自动化还是任何工作，都会被问到。主要想知道你是如何解决问题的，从而推断你问题分析和解决的能力。

当然有遇到问题和挑战，主要有以下几点：

频繁地变更UI，经常要修改页面对象里面代码

运行用例报错和处理，例如元素不可见，元素找不到这样异常

测试脚本复用，尽可能多代码复用

一些新框架产生的页面元素定位问题，例如ck编辑器，动态表格等

14.举例一下你遇到过那些异常，在selenium自动化测试过程中

通过这个问题，大概知道你写过多少脚本。写脚本过程最常见的异常就是，这个元素无法找到。

常见的selenium有以下这些：

1.ElementNotSelectableException ：元素不能选择异常

2.ElementNotVisibleException ：元素不可见异常

3.NoSuchAttributeException ：没有这样属性异常

4.NoSuchElementException：没有该元素异常

5.NoSuchFrameException ：没有该frame异常

6.TimeoutException ： 超时异常

7.Element not visible at this point ：在当前点元素不可见

15 如何处理alert弹窗

我们常见的alert弹窗有两种：基于windows弹窗和基于web页面弹窗

我们知道，webdriver是能够处理alert弹窗的，Selenium提供了Alert这个接口。相关操作代码如下：

// 切换到Alert

Alert alert = driver.switchTo().alert();

// 点击弹窗上确定按钮

alert.accept();

// 点击弹窗的取消按钮

alert.dismiss()

// 获取弹窗上线上的文本文字内容

alert.getText();

// 有些弹窗还支持文本输入，这个可以把要输入字符通过sendkeys方法输入

alert.sendkeys();

16 在selenium中如何处理多窗口？

这个多窗口之间跳转处理，在实际selenium自动化测试经常遇到。就是，你点击一个链接，这个链接会在一个新的tab打开，然后你接下来要查找元素在新tab打开的页面，所以这里需要用到swithTo方法。

需要获取当前浏览器多窗口句柄，然后根据判断跳转新句柄还是旧句柄

17 你查找元素遇到过在Frame里面吗?你是如何处理Frame里面元素定位的？

有时候我们知道元素定位表达式没有问题，但是还是提示no such element，那么我们就需要考虑这个元素是否在frame中。如果在，我们就需要从topwindow，通过swithcTo.Frame()方法来切换到目标frame中，可以通过frame的name和id和索引三种方法来定位frame。

18 怎么验证勾选框是enable/disabled/ checked/Unchecked/ displayed/ not displayed？

通过以下方法来验证元素是enable 还是disable

boolean enabled = driver.findElement(By.xpath(“元素定位表达式”)).isEnabled();

通过以下方法来验证元素是select/check

boolean checked = driver.findElement(By.xpath(“元素定位表达式”)).isSelected();

通过以下方法来验证元素是dispalyed还是not display

boolean displayed = driver.findElement(By.xpath(“元素定位表达式”)).isDisplayed();

19 如何处理下拉菜单？

通常我们也可以通过Click方法来点击下拉菜单里面的元素，还有一种方法，在Selenium中有一个类叫Select，支持这种下拉菜单交互的操作。

基本使用语法是这样的：

Select Se=new Select(element);

Se.selectByIndex(index);

Se.selectByvalue(value);

Se.selectByVisibleText(text);

20 在日历这种web 表单你是如何处理的?

首先要分析当前网页试用日历插件的前端代码，看看能不能通过元素定位，点击日期实现，如果不能，可能需要借助javascript。还有些日历控件一个文本输入框，可以直接sendKeys()方法来实现传入一个时间的数据。

21关闭浏览器中quit和close的区别

简单来说，两个都可以实现退出浏览器session功能，close是关闭你当前聚焦的tab页面，而quit是关闭全部浏览器tab页面，并退出浏览器session。知道这两个区别，我们就知道quit一般用在结束测试之前的操作，close用在执行用例过程中关闭某一个页面的操作。

22 什么是页面加载超时

Selenium中有一个 Page Load wait的方法，有时候，我们执行脚本的速度太快，但是网页程序还有一部分页面没有完全加载出来，就会遇到元素不可见或者元素找不到的异常。为了解决问题，让脚本流畅的运行，我们可以通过设置页面加载超时时间。具体代码是这个：driver.manage().timeouts().pageLoadTimeout(10,TimeUnit.SECONDS);

这行作用就是，如果页面加载超过10秒还没有完成，就抛出页面加载超时的异常。

23.什么是JavaScript Executor，你什么时候会用到这个？

JavaScript Executor是一个接口，给driver对象提供一个执行javaScript并访问和修改前端元素属性和值。

还是有比较多的场景，我们可能或者需要借助javaScript来实现：

1.元素通过现有定位表达式不能够实现点击

2.前端页面试用了ck-editor这个插件

3.处理时间日期插件(可能)

4.生成一个alert弹窗

5.拖拽滚动条

基本语法：

JavascriptExecutor js =(JavascriptExecutor) driver;

js.executeScript(Script,Arguments);

24.在Selenium中如何实现截图，如何实现用例执行失败才截图

在Selenium中提供了一个TakeScreenShot这么一个接口，这个接口提供了一个getScreenshotAs()方法可以实现全屏截图。然后我们通过java中的FileUtils来实现把这个截图拷贝到保存截图的路径。

代码举例：

File src=((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

try {

// 拷贝到我们实际保存图片的路径

FileUtils.copyFile(src,new File(“C:/selenium/error.png”));

}

catch (IOException e)

{

System.out.println(e.getMessage());

}

如果要实现执行用例发现失败就自动截图，那么我们需要把这个截图方法进行封装。然后在测试代码中的catch代码块去调用这个截图方法。这个我们在POM的框架中一般是把截图方法封装到BasePage这个文件中。

25.在Selenium中如何实现拖拽滚动条？

在Selenium中通过元素定位会自动帮你拖拽到对应位置，所以是没有自带的scoll方法。但是这个是有限制，例如当前页面高度太长，默认是页上半部分，你定位的元素在页尾，这个时候可能就会报元素不可见的异常。我们就需要利用javaScript来实现拖拽页面滚动条。

我们一般可以两个方法去拖拽，一个是根据拖拽的坐标(像素单位)，另外一个是根据拖拽到一个参考元素附件。

代码举例(根据元素坐标拖拽)：

JavascriptExecutor jse= (JavascriptExecutor)driver;

jse.executeScript(“window.scrollBy(0,250)”, “”);

26.如何实现文件上传？

我们在web页面实现文件上传过程中，可以直接把文件在磁盘完整路径，通过sendKeys方法实现上传。如果这种方法不能实现上传，我们就可能需要借助第三方工具，我用过一个第三方工具叫autoIT.

还有一个方法是利用robot类

27.如何处理“不受信任的证书”的问题？

例如，在登录12306网站的时候，如果你没有下载和安装过这个网站的根证书，那么你就会遇到打开12306网站提示证书不受信任的拦截页面。

下面举例火狐和谷歌上处理这个问题的基本代码

火狐：

// 创建firefoxprofile

FirefoxProfile profile=new FirefoxProfile();

// 点击继续浏览不安全的网站

profile.setAcceptUntrustedCertificates(true);

// 使用带条件的profile去创建一个driver对象

WebDriver driver=new FirefoxDriver(profile);

Chrome：

// 创建类DesiredCapabilities的对象

DesiredCapabilities cap=DesiredCapabilities.chrome();

// 设置ACCEPT\_SSL\_CERTS 变量值为true

cap.setCapability(CapabilityType.ACCEPT\_SSL\_CERTS, true);

// 新建一个带capability的chromedriver对象

WebDriver driver=new ChromeDriver(cap);

28.什么是Firefox Profile?

Profile是一组文件，主要用来记录用户在火狐浏览器上的私人信息，例如书签，密码，用户首选项，下载文件夹保存路径等。简单来说，你打开火狐浏览器输入about://config，这个页面有些设置选项是可以通过profile来实现修改的。

29.如何实现鼠标悬停，键盘事件和拖拽动作？

在Webdriver中，处理键盘事件和鼠标事件，一般使用Actions类提供的方法，包括鼠标悬停，拖拽和组合键输入。

这里介绍几个方法

方法：clickAndHold()

使用场景：找到一个元素，点击鼠标左键，不放手。自己可以点击鼠标不松开试试这个场景。

方法：contentClick()

使用场景：模拟鼠标右键点击，一般右键会带出菜单来。

方法：doubelClick()

使用场景：模拟鼠标双击

方法：dragAndDrop(source,target)

使用场景：模拟从source这个位置，拖拽一个元素到target位置

键盘事件方法:keyDown(keys.ALT), keyUp(keys.SHIFT)

使用场景：点击键盘事件，分为两个动作，一个点击键盘，第二个动作是释放点击(松开)

30.在执行脚本过程，如何实现当前元素高亮显示？

这个其实就是利用javaScript去修改当前元素的背景颜色来到达高亮显示的效果，

31.如何获取页面标题，悬浮文本和错误文本，并验证？

标题，我们可以通过driver.getTitle()方法来得到一个字符串，然后使用字符串的containts方法或者equals方法去进行断言。

悬浮文本(tooltip)，一般是利用Actions类，然后鼠标悬停方法，然后通过getText()方法来得到这个tooltip字符串。

错误信息，直接把这个错误字段先进行定位，然后通过getText()方法拿到错误文本，主要的断言有包含，相等，不相等，不包含，以什么开头等。

32.在selenium自动化测试中，你一般完成什么类型的测试？

主要是冒烟测试和回归测试。回归测试主要写一些功能稳定，容易实现的场景，通过自动化手段去实现，节约测试时间。

33.你是如何管理你的测试用例并执行？

写用例和管理并执行用例，我们都需要借助单元测试框架来实现，如果是Java语言一般有junit和TestNG，如果是python，常见的有unittest。

就你实际情况，说一下。例如我使用TestNG比较多，需要配置testng.xml文件来实现测试用例的执行。有时候需要配置多个testng.xml去实现不同的任务场景。再展开，可能问你一下testng框架的知识点。例如，方法依赖，用例执行优先级，数据源驱动等。

34关于自动化测试报告生成？

我个人一般用TestNG原生的测试报告，也有第三方叫reportNG的插件，不过我没有实际使用过。

————————————————

问题1：Selenium是什么，流行的版本有哪些？

是一个开源的web自动化测试的框架，支持多种编程语言，支持跨浏览器平台进行测试。

Selenium 1.0或Selenium RC

Selenium 2.0或Selenium Webdriver

Selenium 3.0

问题2：你如何从命令行启动Selenium RC？

java -jar selenium-server.jar

// 在浏览器中运行一套Selenese脚本

java -jar selenium-server.jar -htmlSuite

问题3：在我的机器端口4444不是免费的。我怎样才能使用另一个端口？

//你可以在运行selenium服务器时指定端口为 -

Java -jar selenium-server.jar -port 5555

问题4：什么是Selenium Server，它与Selenium Hub有什么不同？

Selenium Server是使用单个服务器作为测试节点的一个独立的应用程序。 Selenium hub代理一个或多个Selenium的节点实例。一个hub 和多个node被称为Selenium grid。运行SeleniumServer与在同一主机上用一个hub和单个节点创建de Selenium grid类似。

问题5：你如何从Selenium连接到数据库？

Selenium是一个Web UI自动化工具。它不提供任何API来建立数据库连接。这取决于你使用Selenium进行自动化的编程语言。在下面的例子中，我们假设正在使用Java。

一个Connection对象表示与数据库的连接。当我们使用连接方法连接到一个数据库时，我们创建了一个连接对象，它代表了与数据库的连接。单个数据库可能有一个连接或多个连接，还可能有多个连接到不同的数据库上。

我们可以使用Connection对象来做以下事情：

创建用于执行SQL语句的Statement，PreparedStatement和CallableStatement对象。

可以帮助我们提交或回滚一个JDBC事务。

如果你想知道连接到的数据库或数据源信息，Connection对象通过使用DatabaseMetaData就可以收集有关数据库或数据源的信息。

可以帮助我们关闭数据源。Connection.isClosed() 方法只有在调用了Connection.close()时才返回true 。此方法用于关闭所有连接。

首先我们需要通过使用DriverManager.getConnection()方法，建立与数据库的连接。这个方法接受一个包含URL的字符串。DriverManager类尝试查找可以连接到由字符串URL表示的数据库的驱动程序。每当调用getConnection()方法时，DriverManager类都会检查可以连接到URL中指定的数据库的所有已注册的Driver类的列表。

句法：

String url = “jdbc: odbc: makeConnection”;

Connection con = DriverManager.getConnection(url, “userID”, “password”);

问题6：Selenium RC中定位器方法有哪些？

ID Name CSS XPATH Dom

问题7：你如何验证多个页面上存在的一个对象？

可以使用下面的Selenium命令来检查：

assertTrue(selenium.isElementPresent(locator));

问题8：XPath中使用单斜杠和双斜杠有什么区别？

如果XPath是从文档节点开始，它将允许创建“绝对”路径表达式。

例如 “/ html / body / p”匹配所有的段落元素。

如果XPath在文档中的任意位置开始进行选择匹配，那么它将允许创建“相对”路径表达式。

例如 “// p”匹配所有的段落元素。

问题9：如何编写Selenium IDE / RC的用户扩展？

用户扩展（UX）存储在Selenium IDE或Selenium RC用来激活扩展的单独文件中。它包含用JavaScript编写的函数定义。

因为Selenium的核心是用JavaScript开发的，所以要符合原语言的标准规则来创建扩展。要创建一个扩展，我们必须用下面的设计格式来编写函数。

// 样例

Selenium.prototype.doFunctionName = function(){}

函数名称前面的“do”告诉Selenium这个函数可以被调用为一个步骤命令，而不是作为内部函数或私有函数被调用。

问题10：如何在页面加载成功后验证元素的存在？

它可以通过下面的代码行来实现。

只需一点时间（以秒为单位）来检查元素，如下所示：

public void waitForElementPresent(String element, int timeout) throws Exception {

for (int second = 0;; second++) {

if (second >= timeout)

fail(“Timeout. Unable to find the Specified element” + element);

try {

if (selenium.isElementPresent(element))

break;

} catch (Exception e) {

}

Thread.sleep(1000);

}}

问题11：你对Selenium Grid有什么了解？它提供了什么功能？

Selenium Grid是一款利用现有计算基础架构大幅加速Web应用程序功能测试的工具。允许测试者轻松地在多台机器上并行运行多个测试，并且可以在异构环境中运行。

基于优秀的Selenium Web测试工具，Selenium Grid允许测试者并行运行多个Selenium Remote Control实例。更好的是，它集成显示所有Selenium远程控制，所以不必担心实际的基础设施。Selenium Grid将运行Selenium测试套件所需的时间，缩短到Selenium实例的单个实例运行时间的一小点。

问题12：如何从你的Java Class启动Selenium服务器？

try {

seleniumServer = new SeleniumServer();

seleniumServer.start();

} catch (Exception e) {

e.printStackTrace();

}

问题13：Selenium中有哪些验证点？

Selenium主要有三种验证点 -

检查页面标题

检查某些文字

检查某些元素（文本框，下拉菜单，表等）

问题14：什么是XPath？什么时候应该在Selenium中使用XPath？

XPath是一种在HTML / XML文档中定位的方法，可用于识别网页中的元素。如果没有与页面上的元素相关联的名称/ ID，或者名称/ ID的一部分是常量，则必须使用XPath。

绝对路径用 - / 单斜杠

相对路径用 - // 双斜杠

ID，类，名称也可以用于XPath ：

//input[@name=’q’]

//input[@id=’lst-ib’]

//input[@class=’ lst’]

如果id / name / class的一部分是常量 ：

//input[contains(@id,’lst-ib’)

问题15：什么是Selenium的CSS定位器策略？用例子来解释。

CSS位置策略可以与Selenium一起使用来定位元素，它使用CSS定位方法，其中 -

绝对路径用 - （空格符号）

相对路径用 - >表示

ID，类，名称也可以用于XPath：

css=input[name=’q’]

css=input[id=’lst-ib’] or input#lst-ib

css=input[class=’lst’] or input.lst

如果id / name / class只有一部分是常量：

css=input[id\*=’lst-ib’)]

使用内部文本的元素位置策略：

css = a:contains(‘log out’)

问题17：在处理多个弹出窗口的机制是什么？

可以使用命令getWindowHandles()来处理多个弹出窗口。

然后将所有窗口名称存储到Set变量中并将其转换为数组。

接下来，通过使用数组索引，导航到特定的窗口。

driver.switchTo().window(ArrayIndex);

问题18：你如何处理使用Selenium的Ajax控件？

来看一个例子。假如一个文本框是一个Ajax控件，当我们输入一些文本时，它会显示自动建议的值。

处理这样的控件，需要在文本框中输入值之后，捕获字符串中的所有建议值；然后，分割字符串，取值就好了。

问题19：Selenium Webdriver优于Selenium RC的优点是什么？

Selenium RC的架构相当复杂，WebDriver的架构比Selenium RC简单些。

Selenium RC比较慢，因为它使用了另外一个名为Selenium Core的JavaScript程序。相反，WebDriver比Selenium RC更快，因为它直接与浏览器对话，并使用浏览器自己的引擎来进行控制。

像其他JavaScript代码一样，Selenium Core可以访问禁用的元素。Webdriver以更现实的方式与页面元素进行交互。

Selenium RC的API集已经有所改进，但是仍有经常让人困惑的冗余部分。WebDriver API更简单，不包含任何冗余或混淆的命令。

Selenium RC无法支持无头HtmlUnit浏览器。它需要一个真正的、可见的浏览器来操作。Web Driver可以支持无头HtmlUnit浏览器。

Selenium RC内置了测试结果生成器，并自动生成测试结果的HTML文件。Web驱动程序没有自动生成测试结果文件的内置命令。

问题20：“GET”和“NAVIGATE”方法的主要区别是什么？

Get方法能获得一个页面进行加载、或获取页面源代码、或获取文本，就这三。而Navigate将通过刷新，回退，前进的方式导航。

Partial Link Text.部分链接文本这可以通过调用< navigate()>方法来实现。

driver.get() 方法会等到整个页面被加载后才可以，而driver.navigate()只是重定向到该网页，并不会等待。

问题21：隐式等待与显式等待有什么不同？

隐式等待是设置的全局等待，分为1、页面加载超时等待 ；2、页面元素加载超时；3、异步脚本超时。如果是页面元素超时，设置等待时间，是对页面中的所有元素设置加载时间。隐式等待是其实可以理解成在规定的时间范围内，浏览器在不停的刷新页面，直到找到相关元素或者时间结束。

显式等待只是用于特定搜索的一个计时器。它的可扩展性更强，你可以设置它来等待任何条件。通常情况下，可以使用一些预先构建的条件来等待元素变得可点击，可见，不可见等，或者只是编写适合需求的条件。

问题22：如何解决IE中的SSL认证问题？

// 打开浏览器后添加下面的命令

driver.navigate().to(“javascript:document.getElementById(‘overridelink’).click()”);

问题23：如何处理WebDriver中的AJAX控件？

AJAX代表异步JavaScript和XML。它不依赖于创建有效的XML所需的打开和关闭标签的额外开销。大部分时间WebDriver自动处理Ajax控件和调用。如果不能处理的话，可以按照下面的方式来处理。

//Waiting for Ajax Control

WebElement AjaxElement = (new WebDriverWait(driver,

10)).until(ExpectedConditions.presenceOfElementLocated(By.("")));

问题24：如何在标题菜单的子菜单项上执行鼠标移动操作？

应该首先移动菜单标题，然后移至弹出菜单项并单击它。不要忘记在最后调用actions.perform() 。

问题25：什么是数据驱动框架？它与关键字驱动框架有什么不同？

数据驱动框架。

在这个框架中，测试用例逻辑驻留在测试脚本中。测试数据被分离并保存在测试脚本之外。测试数据是从外部文件（Excel文件）中读取的，并被加载到测试脚本中的变量中。变量用于输入值和验证值。

关键字驱动。

关键字/表驱动框架需要开发数据表和关键字。它们独立于执行它们的测试自动化工具。可以使用或不使用应用程序来设计测试。在关键字驱动的测试中，被测试的应用程序的功能记录在一个表格中，以及每个测试的分步说明。

问题26：解释使用TestNG而不是JUnit框架的好处？

TestNG相较于Junit的优势：

在JUnit中，我们必须声明@BeforeClass和@AfterClass，这是JUnit中的一个约束，而在TestNG中没有像这样的约束。

TestNG提供了更多的setUp / tearDown级别。1.@ Before/AfterSuite 2.@Before/AfterTest 3.@Before/AfterGroup

TestNG中不需要扩展任何类。

TestNG中没有方法名称约束，就像JUnit一样。

在TestNG中，我们可以告诉测试一个方法依赖于另一个方法，而在JUnit中这是不可能的。

测试用例的分组在TestNG中可用，而JUnit中则不可用。执行可以基于组完成。例如，如果你已经定义了许多案例，并通过将2个组分别定义为“离职“与”回归”隔离。如果你只是想执行“理智”的情况，那就告诉TestNG执行“理智”。TestNG将自动执行属于“离职”组的案例。

另外，TestNG支持并行测试用例执行。

问题27：与@Test注释相关的TestNG参数的目的是什么？

在TestNG中，参数是修改注释功能的关键字。

问题28：可以使用TestNG运行一组测试用例吗？

是的，TestNG框架支持在测试组的帮助下执行多个测试用例。

它提供了以下选项来运行特定组中的测试用例。

如果想基于回归测试或冒烟测试等其中一个组来执行测试用例，那么：

@Test(groups = {“regression-tests”, “smoke-tests”})

1、selenium中如何判断元素是否存在？

没有提供原生的方法判断元素是否存在，一般我们可以通过定位元素+异常捕获的方式判断

2、selenium中hidden或者是display ＝ none的元素是否可以定位到？

不可以，想点击的话，可以用js去掉dispalay=none的属性

3、selenium中如何保证操作元素的成功率？也就是说如何保证我点击的元素一定是可以点击的？

1.首先通过封装find方法，实现wait\_for\_element\_ispresent（WebDriverWait）

2.在对页面进行click之前，先滚动到该元素（通过Js封装），避免在页面未加载完成前或是在下拉之后才能显示。

3.不同方式进行定位，与expected\_conditions判断方法封装，循环判断页面元素出现后再操作；

4.开发人员规范开发习惯，如给页面元素加上唯一的name,id等。

4、如何提高selenium脚本的执行速度？

1.优化测试用例。设置等待时间的时候，少用sleep，尽量不用implicitly\_wait，多用显式等待方法；

2.减少不必要的操作步骤。如经过三四步才能打开要测试的页面的话，可以直接通过网址来打开；

3.中断页面加载。如果加载的内容不影响我们测试，就设置超时时间，中断页面加载；

4.使用Selenium grid，通过testNG实现并发执行。 在编写测试用例的时候，实现松耦合，然后再服务器允许的情况下，尽量设置多线程实现并发运行。

5、用例在运行过程中经常会出现不稳定的情况，也就是说这次可以通过，下次就没办法通过了，如何去提升用例的稳定性？

1.在经常检测失败的元素前尽量加上显式等待时间，等要操作的元素出现之后再执行下面的操作；

2.多线程的时候，减少测试用例耦合度，因为多线程的执行顺序是不受控制的；

3.多用 try 捕捉，处理异常；

4.尽量使用测试专用环境，避免其他类型的测试同时进行，对数据造成干扰。

6、如何设计高质量自动化脚本

1.使用四层结构实现业务逻辑、脚本、数据分离。

2.使用PO设计模式，将一个页面用到的元素和操作步骤封装在一个页面类中。如果一个元素定位发生了改变，我们只用修改这个页面的元素属性

3.对于页面类的方法，我们尽量从客户的正向逻辑去分析，方法中是一个独立场景，例如：登录到退出，而且不要想着把所有的步骤都封装在一个方法中。

4 测试用例设计中，减少测试用例之间的耦合度。

7、你的自动化用例的执行策略是什么？

1.自动化测试用例是用来监控的。集成到jenkins，创建定时任务定时执行；

2.有些用例在产品上线前必须回归。jenkins上将任务绑定到开发的build任务上，触发执行；

3.有些用例不需要经常执行。jenkins创建一个任务，需要执行的时候人工构建即可。

8、什么是持续集成？

频繁的将代码集成到主干，持续性的进行项目的构架，以便能能够快速发现错误，防止分支大幅度偏离主干

9、自动化测试的时候是不是需要连接数据库做数据校验？

UI自动化不需要

接口测试会需要

10、Selenium有几种定位方式？你最偏爱哪一种，为什么？

Selenium有八种定位方式：

与name有关的有三种：name、class\_name、tag\_name

与link相关的有两种：link\_text、partitial\_link\_text

与id有关：id

全能选手：xpath、css\_selector

如果存在id，我一定使用Id，因为简单方便，定位最快。其次是Xpath，因为很多情况下html标签的属性不够规范，无法唯一定位。Xpath是通过相对位置定位

如果没有，那么CSS定位器应该被优先考虑，因为在大多数现代浏览器中，它们的评估速度比XPath更快。

11、如何去定位页面上动态加载的元素？

首先触发动态事件，然后再定位。如果是动态菜单，则需要层级定位。——JS实现（对动态事件封装）

12、如何去定位属性动态变化的元素？

先去找该元素不变的属性，要是都变，那就找不变的父元素，用层级定位(以不变应万变)

属性动态变化也就是指该元素没有固定的属性值，可以通过：

JS实现，

通过相对位置来定位，比如xpath的轴，paren/following-sibling/percent-sibling

http://www.cnblogs.com/zhaozhan/archive/2009/09/10/1564332.html

13、点击链接以后，selenium是否会自动等待该页面加载完毕？

不会的。所以有的时候，当selenium并未加载完一个页面时再请求页面资源，则会误报不存在此元素。所以首先我们应该考虑判断，selenium是否加载完此页面。其次再通过函数查找该元素。

14、webdriver client的原理是什么？

在selenium启动以后，driver充当了服务器的角色，跟client和浏览器通信，client根据webdriver协议发送请求给driver。driver解析请求，并在浏览器上执行相应的操作，并把执行结果返回给client.

15、webdriver的协议是什么？

WebDrive协议本身是http协议，数据传输使用json

16、启动浏览器的时候用到的是哪个webdriver协议？

-http

17、什么是page object设计模式？

1.通俗来讲，把每个页面当成一个页面对象，页面层写定位元素方法和页面操作方法

2.用例层从页面层调用操作方法，写成用例

3.可以做到定位元素与脚本的分离

18、怎样去选择一个下拉框中的value＝xx的option？

1.select类里面提供的方法：select\_by\_value（“xxx”）

2.xpath的语法也可以定位到

19、如何在定位元素后高亮元素（以调试为目的）？

-重置元素属性，给定位的元素加背景、边框

20、什么是断言和验证？

断言（assert）：测试将会在检查失败时停止，并不运行后续的检查

优点：可以直截了当的看到检查是否通过

缺点：检查失败后，后续检查不会执行，无法收集那些检查结果状态

验证（vertify）：将不会终止测试

缺点：你必须做更多的工作来检查测试结果：查看日志——>耗时多，所以更偏向于断言

21、page object设置模式中，是否需要在page里定位的方法中加上断言？

不需要，page页只做元素抓取和操作方法

22、page object设计模式中，如何实现页面的跳转？

初始化driver参数，Page类传driver参数

23、自动化测试用例从哪里来？

-手工用例中抽取

24、你觉得自动化测试最大的缺陷是什么？

1.不稳定

2.可靠性不强

3.不易维护

4.成本与收益

25、什么是分层测试？

1.数据层

2.接口层’

3.UI层

26、webdriver可以用来做接口测试吗？

不可以，webdriver是专门做web的UI自动化参数

27、get和post 的区别？

（1）GET请求：请求的数据会附加在URL之后，以?分割URL和传输数据，多个参数用&连接。

POST请求：POST请求会把请求的数据放置在HTTP请求包的包体中。

（2）传输数据的大小

使用GET请求时，传输数据会受到URL长度的限制。

对于POST，理论上是不会受限制的

（3）安全性。POST的安全性比GET的高

————————————————