**Core Concept Related Selenium Webdriver Questions.**

**Q: How many types of Webdriver APIs are available in Selenium?**

**Ans.**Below is the list of driver classes that you can use for the browser automation.

* AndroidDriver,
* ChromeDriver,
* EventFiringWebDriver,
* FirefoxDriver,
* HtmlUnitDriver,
* InternetExplorerDriver,
* IPhoneDriver,
* IPhoneSimulatorDriver,
* RemoteWebDriver.

**Q: How would you make sure that a page is loaded using Selenium and Webdriver?**

**Ans.**In Selenium, you can use the below lines of code to check for the successful loading of a web page. The best approach is by selecting an element from the page & stand by till it becomes clickable.

selenium.waitForPageToLoad("5000");

// Or

while (!(selenium.isElementPresent("any page element ")==true)) {

selenium.setSpeed("5");

Thread.sleep(5);

}

Below is the Webdriver specific code to achieve the same objective.

WebDriverWait check = new WebDriverWait(driver, 100);

check.until(ExpectedConditions.anyElement(By.id(id)));

**Q: How to launch a batch file in a Selenium Webdriver project?**

**Ans.**It's usual in a test suite to run a batch file or an executable file for setting up the pre-requisites before starting the automation. You can use the below Java code for this purpose.

Process batch = Runtime.getRuntime.exec("path of the batch file");

batch.waitFor();

**Q: How do you read a JavaScript variable in Selenium WebDriver?**

**Ans.**It's easy to access any JavaScript variable from the Selenium Webdriver test scripts. Just you need to use the below Java code snippet.

// Initialize the JS object.

JavascriptExecutor JS = (JavascriptExecutor) webdriver;

// Get the current site title.

String sitetitle = (String)JS.executeScript("return document.title");

System.out.println("My Site Title: " + sitetitle);

**Q: How to run the selenium IDE test suite from the command line?**

**Ans.**We've given the command to run the SIDE test suite in the next line.

Java -jar "C:\Selenium Webdriver Questions\selenium-server-standalone-2.33.0.jar" -htmlSuite "\*firefox" "http://www.google.com" " “C:\Selenium Webdriver Questions\SeleniumSuite.HTML"

**Q: How to run the Selenium Webdriver test from the command line?**

**Ans.**We can run any Selenium Webdriver test written in Java using the following command.

java -classpath ".;selenium-server-standalone-2.33.0.jar" SampleClass

**Q: What are the different exceptions you face in Selenium Webdriver?**

**Ans.**

* WebDriverException,
* NoAlertPresentException,
* NoSuchWindowException,
* NoSuchElementException,
* TimeoutException.

**Q: How would you automatically click a screenshot whenever any exception occurs?**

**Ans.**For this you will have to use <*EventFiringWebDriver*> class and needs to implement the <*onException*> method of the <*WebDriverEventListener*> interface. See the code example given below.

WebDriver browser = new FirefoxDriver();

EventFiringWebDriver eventDriver = new EventFiringWebDriver(browser).register(new AbstractWebDriverEventListener() {

@Override

public void onException(Throwable throwable, WebDriver browser) {

// Take the screenshot using the Webdriver.

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

// Now you can copy the screenshot somewhere on your system.

FileUtils.copyFile(screen, new File("c:\\Selenium Testing Questions\\screen.png"));

}

});

try {

eventDriver.findElement(By.id("test"));

fail("Caught the Expected exception."); // Intentionally causing the exception for demo.

} catch (NoSuchElementException e) {

// Triggering point for the <onException> event.

}

**Q: How would you select any particular text using the Selenium Webdriver?**

**Ans.**It seems an easy one at first but you need to do a little more to achieve this.

driver.get("/");

WebElement item = driver.findElement(By.xpath("//p[contains(text(),'Selenium webdriver quesions')]"));

Actions dummy = new Actions(driver);

dummy.doubleClick(item).build().perform();

**Q: Give an example to perform drag and drop action In Selenium WebDriver?**

**Ans.**Yes, we can use the Advanced User Interactions API to perform drag and drop operations in a Selenium Webdriver project.

**Code example:**

Actions act = new Actions(driver);

act.dragAndDrop(source\_locator, target\_locator).build().perform();

//Or you can use the below code style.

(new Actions(driver)).dragAndDrop(source\_locator, target\_locator).perform();

**Q: How would you fill a text field without calling the sendKeys()?**

**Ans.**It's a bit slower than the sendKeys() method but we do have means to type in a text field. See the Java code given below.

JavascriptExecutor JS = (JavascriptExecutor)webdriver;

JS.executeScript("document.getElementById(User').value='admin@testmail.com'");

JS.executeScript("document.getElementById('Pass').value='######'");

**Q: How can you check the state of a checkbox/radio button?**

**Ans.**We can call the isSelected() method to test the status of these elements.

**Example Code:**

boolean test = driver.findElement(By.xpath("checkbox/radio button XPath")).isSelected();

**Q: How would you handle the alert popups in Selenium Webdriver?**

**Ans.**First, you’ve to switch the control to the pop up then press the ok or cancel button. After that, turn back to the source page screen.

**Code Example:**

String srcPage = driver.getWindowHandle();

Alert pop = driver.switchTo().alert(); // shift control to the alert popup.

pop.accept(); // click on ok button.

pop.dismiss(); // click on cancel button.

// Move the control back to source page.

driver.switchTo().window(srcPage);  // move back to the source page.

**Q: What is the process to start the IE/Chrome browser?**

**Ans.**If you want to start a browser then, just set the system properties as mentioned below.

// For the IE web browser.

System.setProperty("webdriver.ie.driver"," iedriver.exe file path");

WebDriver driver = new InternetExplorerDriver();

// For the Chrome web browser.

System.setProperty("webdriver.chrome.driver","chrome.exe file path");

WebDriver driver = new ChromeDriver();

**Q: How would you simulate the right click operation in WebDriver?**

**Ans.**You can make use of the Actions class features.

Actions test = new Actions(driver); // Here, driver is the object of WebDriver class.

test.moveToElement(element).perform();

test.contextClick().perform();

**Q: How would you select a menu item from a drop down menu?**

**Ans.**There can be following two situations.

* If the menu has is using the <***select***> tag then you can call the <***selectByValue()>***or <**selectByIndex()>** or <**selectByVisibleText()>** methods of the Select class.
* If the menu doesn’t use the <***select***> tag then simply find the XPath of that element and perform the click action for its selection.

**Q: What is the FirefoxDriver, class or an interface? And which interface does it implement?**

***Ans. FirefoxDriver*** is a Java class, and it implements the <***WebDriver***> interface. It contains the implementations of all the methods available in the <***WebDriver***> interface.

**Q: What is the name of the super interface of the Webdriver?**

**Ans. SearchContext.**

**Q: What is the main difference between the close() and quit() methods?**

**Ans.**

**close() -**it closes the currently active browser window.

**quit()-**it will close all of the opened browser windows and the browser itself.

**Q: What is the best way to check for the highlighted text on a web page?**

**Ans.**Use the below code to verify the highlighted text for an element on the web page.

String clr = driver.findElement(By.xpath("//a[text()='Ranga']")).getCssValue("color");

String bkclr = driver.findElement(By.xpath("//a[text()='Ranga']")).getCssValue("background-color");

System.out.println(clr);

System.out.println(bkclr);

**Q: How would you use a Selenium variable say "size" from the JavaScript?**

**Ans. ${size}**

**Q: What is the Selenese command to show the value of a variable in the log file?**

**Ans. echo()**