

### **Question 1:**

What is Selenium?

### **Answer:**

Selenium is a robust test automation suite that is used for automating web based applications. It supports multiple browsers, programming languages and platforms.

### **Question 2:**

What are different components of selenium?

### **Answer:**

Selenium has four components-

1. *Selenium WebDriver* - Selenium WebDriver is used to automate web applications using browser's native methods.
2. *Selenium IDE* - A firefox plugin that works on record and play back principle.
3. *Selenium RC* - Selenium Remote Control (RC) is officially deprecated by selenium and it used to work on javascript to automate the web applications.
4. *Selenium Grid* - Allows selenium tests to run in parallel across multiple machines.

### **Question 3:**

What are some advantages of selenium?

### **Answer:**

Following are the advantages of selenium-

1. Selenium is open source and free to use without any licensing cost.
2. It supports multiple languages like Java, ruby, python etc.
3. It supports multi browser testing.
4. It has good amount of resources and helping community over the internet.
5. Using selenium IDE component, non-programmers can also write automation scripts
6. Using selenium grid component, distributed testing can be carried out

on remote machines possible.

### **Question 4:**

What are some limitations of selenium?

### **Answer:**

Following are the limitations of selenium-

1. We cannot test desktop application using selenium.
2. We cannot test flash based apps using selenium.
3. For creating robust scripts in selenium webdriver, programming language knowledge is required.
4. We have to rely on external libraries and tools for performing tasks like - logging(log4J), testing framework-(testNG, JUnit), reading from external files(POI for excels) etc.

### **Question 5:**

Why do you prefer Selenium Automation Tool?

**Answer:**

- Free and open source
- Have large user base and helping communities
- Cross browser compatibility
- Platform compatibility
- Multiple programming languages support

**Question 6:**

What is Automation Testing?

**Answer:**

Automation testing is the process of testing the software using an automation tools to find the defects. In this process, executing the test scripts and generating the results are performed automatically by automation tools. Some most popular tools to do automation testing are HP QTP/UFT, Selenium Web-Driver, etc.

test cases.

- Minimal manual intervention. Test scripts can be run unattended.
- Maximum coverage. It helps to increase the test coverage.

**Question 8:**

What type of tests you have automated?

**Answer:**

Our main focus is to automate test cases to do Regression testing, Smoke testing, and Sanity testing. Sometimes based on the project and the test time estimation, we do focus on End to End testing.

**Question 9:**

How many test cases you have automated per day?

**Answer:**

It depends on Test case scenario complexity and length. I did automate 2-5 test scenarios per day when the complexity is limited. Sometimes just 1 or fewer test scenarios in a day when the complexity is high.

**Question 11:**

Which all browsers are supported by selenium webdriver?

**Answer:**

Some commonly used browsers supported by selenium are-

1. Google Chrome - ChromeDriver
2. Firefox - FireFoxDriver
3. Internet Explorer - InternetExplorerDriver
4. Safari - SafariDriver
5. HtmlUnit (Headless browser) - HtmlUnitDriver
6. Android - Selendroid/Appium
7. IOS - ios-driver/Appium

**Question 12:**

What Is WebdriverIO?

**Question 7:**

What are the benefits of Automation Testing?

**Answer:**

- Saves time and money. Automation testing is faster in execution.
- Reusability of code. Create one time and execute multiple times with less or no maintenance.
- Easy reporting. It generates automatic reports after test execution.
- Easy for compatibility testing. It enables parallel execution in combination of different OS and browser environments.
- Low cost maintenance. It is cheaper compared to manual testing in a long run.
- Automated testing is more reliable.
- Automated testing is more powerful and versatile.
- It is mostly used for regression testing. Supports execution of repeated

**Question 10:**

Compare Automated Testing & Manual Testing

**Answer:**

Criteria	Automated testing	Manual testing
Speed	Excellent	Average
Initial cost	High	Low
Operation	Same operation each time	Operations are not accurate
Recommended for	Frequent testing scenario	Single testing scenario
Concurrency	Possible to test on different machines, OS and platforms simultaneously	Needs different testers to test on multiple machines & platforms

**Answer:**

WebdriverIO is a powerful framework for testing mobile and web applications. It uses JavaScript to implement the Selenium Webdriver API and the wire protocol as per the W3C standards. It is available for installation as NPM package and runs on Node.js. Hence, it is also famous with the name as "Selenium 2.0 bindings for NodeJS".

**Question 13:**

List all the features of WebdriverIO

**Answer:**

- It is a multipurpose automation tool which can fit in to test both the web applications and the native mobile Apps.
- Command execution takes place asynchronously in WebdriverIO. However, we can write them in a synchronous way so to avoid any race conditions that may occur while handling a promise.
- WebdriverIO provides a set of simple JavaScript functions to write tests which are lean, fast and easy to understand.

- It comes with an intelligent Test Runner which optimizes test execution to achieve maximum concurrency. Ideally, it can run all your tests in parallel if adequate CPU/memory resources are present.
- Test runner also adds a feature to register hooks for handling errors and altering test flow based on previous test result. For example, capturing a screenshot when any error occurs or ignoring a set of test cases if a dependent test gets failed.
- It integrates seamlessly with CI tools like Jenkins and utilizes JUnit reporter to debug issues and tracking test execution.
- Services like Sauce Labs and BrowserStack can plugin with WebdriverIO to run tests through the secure tunnel of respective environments.
- Another service for Appium enables testing for mobile devices and eliminates the need to start it manually for test execution.

The primary objective of WebdriverIO is to automate the end to end testing on a large scale. Its test runner helps in building a reliable Test Suite which is easy to understand and manage. It resolves many issues that occur with

4. className
5. tagName
6. name
7. linkText
8. partialLinkText

#### Question 16:

What is an XPath?

**Answer:**

Xpath or XML path is a query language for selecting nodes from XML documents. XPath is one of the locators supported by selenium webdriver.

#### Question 17:

What is an absolute XPath?

**Answer:**

An absolute XPath is a way of locating an element using an XML expression beginning from root node i.e. html node in case of web pages. The main disadvantage of absolute xpath is that even with slightest change in the UI or

Mention what is Selenium 3.0?

**Answer:**

Selenium 3.0 is the latest version of Selenium. It has released 2 beta versions of selenium 3.0 with few of the below changes:

A list of few new features added in Selenium 3.0:

Beta 1 Updates	Beta 2 Updates(Only for java)
Minimum requirement is java version 8+	System property webdriver.firefox.marionette now forces the server in marionette or legacy firefox driver mode, ignoring any related desired capability
It will support for Firefox via Mozilla's GeckoDriver	Grid fixes NPE's on registration when -browser not specified
It now supports safari on MacOs via Apple's	Update GeckoDriver ?port argument in all bindings

standard automation libraries. With it, we can organize the tests to run in split mode to enable concurrency. Moreover, it does provide session management and a lot of features to help in debugging runtime issues and find errors in the test cases.

#### Question 14:

What all testing types supported by Selenium WebDriver?

**Answer:**

Selenium Webdriver can be used for performing automated functional and regression testing.

#### Question 15:

What are various ways of locating an element in selenium?

**Answer:**

The different locators in selenium are-

1. Id
2. XPath
3. cssSelector

any element the whole absolute XPath fails.

Example - `html/body/div/div[2]/div/div/div/div[1]/div/input`

#### Question 18:

What is a relative XPath?

**Answer:**

A relative XPath is a way of locating an element using an XML expression beginning from anywhere in the HTML document. There are different ways of creating relative XPaths which are used for creating robust XPaths (unaffected by changes in other UI elements).

Example - `//input[@id='username']`

#### Question 19:

What is Selenium 2.0 ?

**Answer:**

Web Testing tools Selenium RC and WebDriver are consolidated in single tool in Selenium 2.0. However, latest version of Selenium is 3.0.

#### Question 20:

own safari driver

#### Question 21:

How will you find an element using Selenium?

**Answer:**

In Selenium every object or control in a web page is referred as an elements, there are different ways to find an element in a web page they are

- ID
- Name
- Tag
- Attribute
- CSS
- Linktext
- PartialLink Text
- Xpath etc

#### Question 22:

List out the test types that are supported by Selenium?

**Answer:**

For web based application testing selenium can be used

The test types can be supported are:

- a) Functional
- b) Regression

**Question 23:**

What are the different network protocols that selenium supports?

**Answer:**

HTTP and HTTPS are the two most commonly used network protocols.

And in Selenium, there are multiple ways to handle both of these protocols.

**Question 24:**

How Will You Handle Different Network Protocols In Selenium?

**Answer:**

**Website Not Using Authentication:**

If a website is only using the HTTP and does not require any

HTTPS is the secure version of HTTP. And it encrypts all the communication exchanged between the web server and the browser. It makes use of SSL certificates which get downloaded into the browser upon initiating the first request. The certificate contains a public key for encrypting the data flowing from client to the server. Similarly, the server reserves a private key corresponding to the certificate and uses it to decrypt the data received. These public/private keys are none other than the famous RSA key-pair and available in different key lengths such as 512, 1024, 2048, and 4096 bits.

So in reality, we need to handle the SSL certificates via Selenium. Also, we've to follow different approaches for managing the SSL in Firefox, Chrome, and IE browsers.

**Question 25:**

Mention what is the use of X-path?

**Answer:**

X-Path is used to find the WebElement in web pages. It is also useful in identifying the dynamic elements.

can be used for testing client server application also

- Selenium supports Firefox, IE, Opera, Safari on operating systems like Windows, Mac, Linux etc. however QTP is limited to Internet Explorer on Windows.
- Selenium supports many programming languages like Ruby, Perl, Python whereas QTP supports only VB script

**Question 26:**

What is the difference between Selenium IDE, Selenium RC and WebDriver?

**Answer:**

Feature	Selenium IDE	Selenium RC	WebDriver
Browser Compatibility	Selenium IDE comes as a Firefox plugin, thus it supports	Selenium RC supports a varied range of versions of Mozilla Firefox,	WebDriver supports a varied range of versions of Mozilla Firefox,

authentication, then it's very easy to handle in Selenium.

```
WebDriver driver = new ChromeDriver();
```

```
driver.get("http://destination-url/");
```

**Website Using Basic Authentication:**

There are many websites which apply basic authentication scheme before allowing access to their home page. Here are three most prominent ways to do it in Selenium.

Below code is using WebDriverWait and Alert classes to implement basic authentication for HTTP.

```
WebDriverWait testwait = new WebDriverWait(driver, 10);
```

```
Alert testalert = testwait.until(ExpectedConditions.alertIsPresent());
```

```
testalert.authenticateUsing(new UserAndPassword(**user**, **pass**));
```

Alternatively, we can also pass the user/pass pair within the HTTP URL as a parameter to the Webdriver's Get method.

```
String target = "http://user:pass@host";
```

**Handle HTTPS Protocol:**

**Question 26:**

List out the technical challenges with Selenium?

**Answer:**

- Selenium supports only web based applications
- It does not support the Bitmap comparison
- For any reporting related capabilities have to depend on third party tools
- No vendor support for tool compared to commercial tools like HP UFT
- As there is no object repository concept in Selenium, maintainability of objects becomes difficult

**Question 27:**

Why testers should opt for Selenium and not QTP?

**Answer:**

- Selenium is an open source whereas QTP is a commercial tool
- Selenium is used specially for testing web based applications while QTP

	only Firefox	Google Chrome, Internet Explorer and Opera	Google Chrome, Internet Explorer and Opera	Google Chrome, Internet Explorer and Opera. Also supports HtmlUnitDriver which is a GUI less or headless browser.
Record and Playback	Selenium IDE supports record and playback feature	Selenium RC doesn't support record and playback feature	Selenium RC doesn't support record and playback feature	WebDriver doesn't support record and playback feature
Server Requirement	Selenium IDE doesn't require any server to be started before executing the test scripts	Selenium RC requires server to be started before executing the test scripts	WebDriver doesn't require any server to be started before executing the test	WebDriver doesn't require any server to be started before executing the test

	scripts	scripts				
Architecture	Selenium IDE is a Javascript based framework	Selenium RC is a JavaScript based Framework	WebDriver uses the browser's native compatibility to automation	navigations, dropdowns	navigations, dropdowns	in handling alerts, navigations, and dropdowns efficiently and effectively.
Object Oriented	Selenium IDE is not an object oriented tool	Selenium RC is semi object oriented tool	WebDriver is a purely object oriented tool			WebDriver is designed in a way to efficiently support testing of iPhone/Android applications.
Dynamic Finders (for locating web elements on a webpage)	Selenium IDE doesn't support dynamic finders	Selenium RC doesn't support dynamic finders	WebDriver supports dynamic finders	WAP (iPhone/Android) Testing	Selenium IDE doesn't support testing of iPhone/Andriod applications	Selenium RC doesn't support testing of iPhone/Andriod applications
Handling Alerts, Navigations, Dropdowns	Selenium IDE doesn't explicitly provides aids to handle alerts,	Selenium RC doesn't explicitly provides aids to handle alerts,	WebDriver offers a wide range of utilities and classes that helps			WebDriver comes with a large range of drivers for WAP based testing. For example, AndroidDriver,
		iPhoneDriver				with the browser.
Listener Support	Selenium IDE doesn't support listeners	Selenium RC doesn't support listeners	WebDriver supports the implementation of Listeners			
Speed	Selenium IDE is fast as it is plugged in with the web-browser that launches the test. Thus, the IDE and browser communicates directly	Selenium RC is slower than WebDriver as it doesn't communicate directly with the browser; rather it sends selenium commands over to Selenium Core which in turn communicates				

specifically for inspecting web-elements in order to use their attributes like id, class, name etc. in different locators.

#### Question 31:

What is the difference between POI and jxl jar?

#### Answer:

#	JXL jar	POI jar
1	JXL supports ".xls" format i.e. binary based format. JXL doesn't support Excel 2007 and ".xlsx" format i.e. XML based format	POI jar supports all of these formats
2	JXL API was last updated in the year 2009	POI is regularly updated and released
3	The JXL documentation is not as comprehensive as that of POI	POI has a well prepared and highly comprehensive documentation
4	JXL API doesn't support rich text	POI API supports rich text

#### Question 29:

What is the difference between single slash(/) and double slash(//) in XPath?

#### Answer:

In XPath, a single slash is used for creating XPaths with absolute paths beginning from root node whereas double slash is used for creating relative XPaths.

#### Question 30:

How can we inspect the web element attributes in order to use them in different locators?

#### Answer:

Using Firebug or developer tools we can inspect the specific web elements.

Firebug is a plugin of firefox that provides various development tools for debugging applications. From automation perspective, firebug is used

formatting	formatting
5 JXL API is faster than POI API	POI API is slower than JXL API
<b>Question 32:</b>	What is the difference between Selenium and QTP?
<b>Answer:</b>	
<b>Feature</b>	<b>Selenium</b>
Browser Compatibility	Selenium supports almost all the popular browsers like Firefox, Chrome, Safari, Internet Explorer, Opera etc
Distribution	Selenium is distributed as an open source tool and is freely available
Application under	Selenium supports testing
	<b>Quick Test Professional (QTP)</b>
	QTP supports Internet Explorer, Firefox and Chrome. QTP only supports Windows Operating System
	QTP is distributed as a licensed tool and is commercialized
	QTP supports testing of

Test	of only web based applications	both the web based application and windows based application
Object Repository	Object Repository needs to be created as a separate entity	QTP automatically creates and maintains Object Repository
Language Support	Selenium supports multiple programming languages like Java, C#, Ruby, Python, Perl etc	QTP supports only VB Script
Vendor Support	As Selenium is a free tool, user would not get the vendor's support in troubleshooting issues	Users can easily get the vendor's support in case of any issue

#### Question 33:

How can we locate an element by only partially matching its attributes value in Xpath?

#### Question 36:

How can we move to nth child element using XPath?

##### Answer:

There are two ways of navigating to the nth element using XPath-

- Using square brackets with index position-  
Example - div[2] will find the second div element.
- Using position().  
Example - div[position()=3] will find the third div element.

#### Question 37:

How can we select elements by their attribute value using CSS Selector?

##### Answer:

Using [attribute=value], we can select all the element belonging to a particular class e.g. '[type=small]' will select the element having attribute type of value 'small'.

#### Question 38:

Both `driver.get("URL")` and `driver.navigate().to("URL")` commands are used to navigate to a URL passed as parameter.

There is no difference between the two commands.

#### Question 41:

Explain the difference between close and quit command.

##### Answer:

`driver.close()` - Used to close the current browser having focus  
`driver.quit()` - Used to close all the browser instances

#### Question 42:

How to switch between multiple windows in selenium?

##### Answer:

Selenium has `driver.getWindowHandles()` and `driver.switchTo().window("{windowHandleName}")` commands to work with multiple windows. The `getWindowHandles()` command returns a list of ids corresponding to each window and on passing a particular window handle to `driver.switchTo().window ("{windowHandleName}")` command we can switch

#### Answer:

Using contains() method we can locate an element by partially matching its attribute's value. This is particularly helpful in the scenarios where the attributes have dynamic values with certain constant part.

**XPath expression = //\*[contains(@name,'user')]**

The above statement will match the all the values of name attribute containing the word 'user' in them.

#### Question 34:

How can we locate elements using their text in XPath?

##### Answer:

Using xpath inbuilt function i.e. `text()`-

**xPathExpression = //\*[text()='username']**

#### Question 35:

How can we move to parent of an element using XPath?

##### Answer:

Using '..' expression in XPath we can move to parent of an element.

What is fundamental difference between XPath and css selector?

##### Answer:

The fundamental difference between XPath and css selector is using XPaths we traverse up in the document i.e. we can move to parent elements. Whereas using CSS selector we can only move downwards in the document.

#### Question 39:

How can we launch different browsers in selenium webdriver?

##### Answer:

By creating an instance of driver of a particular browser-

`WebDriver driver = new FirefoxDriver();`

#### Question 40:

What is the use of `driver.get("URL")` and `driver.navigate().to("URL")` commands? Is there any difference between the two?

##### Answer:

control/focus to that particular window.

```
for (String windowHandle : driver.getWindowHandles()) {  
    driver.switchTo().window(handle);
```

#### Question 43:

What is the difference between `driver.getWindowHandle()` and `driver.getWindowHandles()` in selenium?

##### Answer:

`driver.getWindowHandle()` returns a handle of the current page (a unique identifier)

Whereas `driver.getWindowHandles()` returns a set of handles of all the pages available.

#### Question 43:

How can we move to a particular frame in selenium?

##### Answer:

The `driver.switchTo()` commands can be used for switching to frames.  
`driver.switchTo().frame("{frameIndex/frameId/frameName}");`

For locating a frame we can either use the index (starting from 0), its name or Id.

**Question 44:**

Can we move back and forward in browser using selenium?

**Answer:**

Yes,

using `driver.navigate().back()` and `driver.navigate().forward()` commands we can move backward and forward in a browser.

**Question 45:**

Is there a way to refresh browser using selenium?

**Answer:**

There are multiple ways to refresh a page in selenium-

- Using `driver.navigate().refresh()` command
- Using `sendKeys(Keys.F5)` on any textbox on the webpage
- Using `driver.get("URL")` on the current URL or using `driver.getCurrentUrl()`
- Using `driver.navigate().to("URL")` on the current URL

**Answer:**

Using `getAttribute("{attributeName}")` method, we can find the value of different attributes of an element e.g.-

```
String valueAttribute =  
driver.findElement(By.id("elementLocator")).getAttribute("value");
```

**Question 49:**

How to delete cookies in selenium?

**Answer:**

Using `deleteAllCookies()` method-  
`driver.manage().deleteAllCookies();`

**Question 50:**

What is an implicit wait in selenium?

**Answer:**

An implicit wait is a type of wait which waits for a specified time while locating an element before throwing `NoSuchElementException`. As by default selenium tries to find elements immediately when required without any wait.

- `elementToBeClickable(WebElement element or By locator)`
- `stalenessOf(WebElement element)`
- `visibilityOf(WebElement element)`
- `visibilityOfElementLocated(By locator)`
- `invisibilityOfElementLocated(By locator)`
- `attributeContains(WebElement element, String attribute, String value)`
- `alertIsPresent()`
- `titleContains(String title)`
- `titleIs(String title)`
- `textToBePresentInElementLocated(By, String)`

**Question 53:**

What is the difference between implicit wait and explicit wait in selenium webdriver?

**Answer:**

**Implicit Wait:**

In Implicit wait, if WebDriver cannot find an element in the Document

```
or driver.navigate().to(driver.getCurrentUrl());
```

**Question 46:**

How can we maximize browser window in selenium?

**Answer:**

We can maximize browser window in selenium using following command-  
`driver.manage().window().maximize();`

**Question 47:**

How can we fetch a text written over an element?

**Answer:**

Using `getText()` method we can fetch the text over an element.  
String text = driver.findElement("elementLocator").getText();

**Question 48:**

How can we find the value of different attributes like name, class, value of an element?

So, it is good to use implicit wait. This wait is applied to all the elements of the current driver instance.

```
driver.manage().timeouts().implicitlyWait(5, TimeUnit.SECONDS);
```

**Question 51:**

What is an explicit wait in selenium?

**Answer:**

An explicit wait is a type of wait which is applied to a particular web element until the expected condition specified is met.

```
WebDriverWait wait = new WebDriverWait(driver, 10);  
WebElement element = wait.until(ExpectedConditions.elementToBeClickable(By.id("elementId")));
```

**Question 52:**

What are some expected conditions that can be used in Explicit waits?

**Answer:**

Some of the commonly used expected conditions of an element that can be used with explicit waits are-

Object Model (DOM), it will wait for a defined amount of time for the element to appear in the DOM. The Implicit wait may slow down your tests, because once set, the implicit wait is set for the life of the WebDriver object's instance.

An example of Implicit Wait:

```
1. driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

**Explicit Wait:**

Explicit waits are better than implicit wait. Unlike an implicit wait, you can write custom code or conditions for wait before proceeding further in the code.

An explicit wait can be used where synchronization is needed, for example the page is loaded but we are still waiting for a call to complete and an element to appear.

An example of Explicit Wait:

```
1. WebDriverWait wait = new WebDriverWait(driver, 10);  
2. wait.until(ExpectedConditions.titleContains("selenium"));
```

}

;

**Question 55:**

What are the different keyboard operations that can be performed in selenium?

**Answer:**

The different keyboard operations that can be performed in selenium are

1. .sendKeys("sequence of characters") - Used for passing character sequence to an input or textbox element.
2. .pressKey("non-text keys") - Used for keys like control, function keys etc that are non text.
3. .releaseKey("non-text keys") - Used in conjunction with keypress event to simulate releasing a key from keyboard event.

**Question 56:**

What are the different mouse actions that can be performed?

action.doubleClick(element).perform();

**Question 58:**

Write the code to right click an element in selenium?

**Answer:**

Code to right click an element in selenium-

```
Actions action = new Actions(driver);
WebElement element=driver.findElement(By.id("elementId"));
action.contextClick(element).perform();
```

**Question 59:**

How to mouse hover an element in selenium?

**Answer:**

Code to mouse hover over an element in selenium-

```
Actions action = new Actions(driver);
WebElement element=driver.findElement(By.id("elementId"));
action.moveToElement(element).perform();
```

**Question 60:**

How to locate a link using its text in selenium?

**Answer:**

Using linkText() and partialLinkText() we can locate a link. The difference between the two is linkText matches the complete string passed as parameter to the link texts. Whereas partialLinkText matches the string parameter partially with the link texts.

```
WebElement link1 = driver.findElement(By.linkText("artOfTesting"));
WebElement link2 = driver.findElement(By.partialLinkText("artOf"));
```

**Question 64:**

What are DesiredCapabilities in selenium webdriver?

**Answer:**

Desired capabilities are a set of key-value pairs that are used for storing or configuring browser specific properties like its version, platform etc in the browser instances.

**Question 65:**

How can we find all the links on a web page?

**Question 54:**

What is fluent wait in selenium?

**Answer:**

A fluent wait is a type of wait in which we can also specify polling interval(intervals after which driver will try to find the element) along with the maximum timeout value.

```
Wait wait = new FluentWait(driver)
.withTimeout(20, SECONDS)
.pollingEvery(5, SECONDS)
.ignoring(NoSuchElementException.class);

WebElement textBox = wait.until(new Function<webdriver,webElement>()
{
    public WebElement apply(WebDriver driver) {
        return driver.findElement(By.id("textBoxId"));
    }
});
```

**Answer:**

The different mouse events supported in selenium are

1. click(WebElement element)
2. doubleClick(WebElement element)
3. contextClick(WebElement element)
4. mouseDown(WebElement element)
5. mouseUp(WebElement element)
6. mouseMove(WebElement element)
7. mouseMove(WebElement element, long xOffset, long yOffset)

**Question 57:**

Write the code to double click an element in selenium?

**Answer:**

Code to double click an element in selenium-

```
Actions action = new Actions(driver);
WebElement element=driver.findElement(By.id("elementId"));
```

How can we fetch title of the page in selenium?

**Answer:**

Using driver.getTitle(); we can fetch the page title in selenium. This method returns a string containing the title of the webpage.

**Question 61:**

How can we fetch the page source in selenium?

**Answer:**

Using driver.getPageSource(); we can fetch the page source in selenium.

This method returns a string containing the page source.

**Question 62:**

How to verify tooltip text using selenium?

**Answer:**

Tooltips webelements have an attribute of type 'title'. By fetching the value of 'title' attribute we can verify the tooltip text in selenium.

```
String toolTipText = element.getAttribute("title");
```

**Question 63:**

**Answer:**

All the links are of anchor tag 'a'. So by locating elements of tagName 'a' we can find all the links on a webpage.

```
List<WebElement> links = driver.findElements(By.tagName("a"));
```

**Question 66:**

What are some commonly encountered exceptions in selenium?

**Answer:**

Some of the commonly seen exception in selenium are-

- NoSuchElementException - When no element could be located from the locator provided.
- ElementNotVisibleException - When element is present in the dom but is not visible.
- NoAlertPresentException - When we try to switch to an alert but the targeted alert is not present.
- NoSuchFrameException - When we try to switch to a frame but the targeted frame is not present.

```
File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
FileUtils.copyFile(scrFile, new File("D:\\testScreenShot.jpg"));
```

**Question 68:**

How to handle dropdowns in selenium?

**Answer:**

Using Select class-

```
Select countriesDropDown = new Select(driver.findElement(By.id("countries")));
dropdown.selectByVisibleText("India");
//or using index of the option starting from 0
dropdown.selectByIndex(1);
//or using its value attribute
dropdown.selectByValue("Ind");
```

**Question 69:**

How to check which option in the dropdown is selected?

How can we check if an element is enabled for interaction on a web page?

**Answer:**

Using isEnabled method we can check if an element is enabled or not.

```
driver.findElement(By locator).isEnabled();
```

**Question 72:**

What is the difference between driver.findElement() and driver.findElements() commands?

**Answer:**

The difference between driver.findElement() and driver.findElements() commands is-

- findElement() returns a single WebElement (found first) based on the locator passed as parameter. Whereas findElements() returns a list of WebElements, all satisfying the locator value passed.
- Syntax of findElement()-

```
WebElement textbox = driver.findElement(By.id("textBoxLocator"));
```

Syntax of findElements()

- NoSuchElementException - When we try to switch to a window but the targeted window is not present.
- UnexpectedAlertPresentException - When an unexpected alert blocks normal interaction of the driver.
- TimeoutException - When a command execution gets timeout.
- InvalidElementStateException - When the state of an element is not appropriate for the desired action.
- NoSuchAttributeException - When we are trying to fetch an attribute's value but the attribute is not correct
- WebDriverException - When there is some issue with driver instance preventing it from getting launched.

**Question 67:**

How can we capture screenshots in selenium?

**Answer:**

Using getScreenshotAs method of TakesScreenshot interface we can take the screenshots in selenium.

**Answer:**

Using isSelected() method we can check the state of a dropdown's option.

```
Select countriesDropDown = new Select(driver.findElement(By.id("countries")));
dropdown.selectByVisibleText("India");
//returns true or false value
System.out.println(driver.findElement(By.id("India")).isSelected());
```

---

**Question 70:**

How can we check if an element is getting displayed on a web page?

**Answer:**

Using isDisplayed method we can check if an element is getting displayed on a web page.

```
driver.findElement(By locator).isDisplayed();
```

**Question 71:**

List <WebElement> elements = element.findElements(By.id("value"));

- Another difference between the two is- if no element is found then findElement() throws NoSuchElementException whereas findElements() returns a list of o elements.

**Question 73:**

What is soft assertion in Selenium? How can you mark a test case as failed by using soft assertion?

**Answer:**

Soft Assertions are customized error handlers provided by TestNG. Soft Assertions do not throw exceptions when assertion fails, and they simply continue to the next test step. They are commonly used when we want to perform multiple assertions.

To mark a test as failed with soft assertions, call assertAll() method at the end of the test.

**Question 74:**

How can we handle window UI elements and window POP ups using

selenium)?

**Answer:**

Selenium is used for automating Web based application only(or browsers only). For handling window GUI elements we can use AutoIT. AutoIT is a free-ware used for automating window GUI. The Autolt scripts follow simple BASIC language like syntax and can be easily integrated with selenium tests.

Following are the four methods that can be used along with the Alert interface:

- void dismiss() – The accept() method clicks on the “Cancel” button as and when the pop up window appears.
- void accept() – The accept() method clicks on the “Ok” button the moment the pop up window appears.
- String getText() – The getText() method returns the text displayed on the alert box.
- void sendKeys(String stringToSend) – The sendKeys() method enters the specified string pattern into the alert box.

1. Using element.sendKeys("path of file") on the webElement of input tag and type file i.e. the elements should be like -

```
<input type="file" name="fileUpload">
```

1. Using Robot API.
2. Using AutoIT API.

**Question 77:**

Can captcha be automated?

**Answer:**

No, captcha and bar code reader cannot be automated.

**Question 78:**

What Types Of Data Have You Handled In Selenium For Automating Web Applications?

**Answer:**

Data is probably a very critical and important piece of information in

columns (delimited by commas) stored in a text file with CSV as the file extension. Each row corresponds to a record whereas every column maps to a field.

It is trivial to access CSV data as most programming languages like Java and Python come with standard libraries to read/write text files. However, Excel does support this type of data and can handle it the same way as it does its own.

**XML Data.**

XML is an acronym for Extensible Markup Language. It generates documents in a format which is readable by both human and machine. XML is a web standard and tends to be faster than Excel when it comes to read/write data. And there are either built-in or third-party XML parsers available in Java, Python, and others to manage the data.

Sometimes both Excel and XML get used together in Selenium automation projects. For example, we can write the test cases in Excel and create a macro to convert them in XML format. The automation suite then reads the test cases from XML and executes them.

**Syntax:**

```
// accepting javascript alert  
Alert alert = driver.switchTo().alert();  
alert.accept();
```

**Question 75:**

What is Robot API?

**Answer:**

Robot API is used for handling Keyboard or mouse events. It is generally used to upload files to the server in selenium automation.

```
Robot robot = new Robot();  
//Simulate enter key action  
robot.keyPress(KeyEvent.VK_ENTER);
```

**Question 76:**

How to do file upload in selenium?

**Answer:**

File upload action can be performed in multiple ways-

ensuring that Automation can run successfully. And it can be in a variety of form factors to serve different purposes. For example, in automation, we use input data that includes test cases, and it could be test data that feeds into test cases as parameters.

Here are some of the commonly used mechanisms to handle data in automation:

**Excel Data**

It is the most popular data format used widely by automation testers. We can not only use it to store test cases but to keep test data as well.

This format is very flexible and supports CRUD (Create /Delete /Update /Delete) operations also. And almost all languages that integrate with Selenium like Java, Python, Ruby have ready-made libraries available to access and manipulate Excel data.

**CSV Data**

CSV stands for comma-separated values. It is the most simplistic way of organizing test data in tabular format. It represents a sequence of rows and

Also, XML comes with another unique feature to search specific elements called as XPath. It is a known Selenium term which we usually use to traverse the DOM of a web page. However, it works the same with any XML data or document.

**JSON Data.**

JSON is an acronym for JavaScript Object Notation. It supports a light-weight data interchange format. Since JSON data is compact and fast, it has become a preferred choice for supplying input in web services.

And we can also use it to represent the test data replacing Excel and XML formats. If the size of data is invariably large, then JSON would surely deliver better performance.

**YAML Data.**

YAML is another human-readable language for data serialization. It stands for YAML Ain't Markup Language. This type of data format is ideal for holding configurations. However, it could be useful for various other purposes. It is a superset of JSON language and can store both JSON and XML data within

itself.

Unlike other data formats, it allows referencing other elements inside YAML using anchors.

#### **SQL Data.**

Another type of data which doesn't get used often is SQL data. It is highly customizable and enables fast data access. It can store both the test cases and any configuration or settings as required.

However, there is a separate database software like MySQL/MongoDB/SQL/Ora-cle needed to make it work. It is an additional overhead for the automation engineers to install the database software to prepare a new setup.

#### **Question 79:**

How to handle HTTPS website in selenium? or How to accept the SSL untrusted connection?

#### **Answer:**

Using profiles in firefox we can handle accept the SSL untrusted connection certificate. Profiles are basically set of user preferences stored in a file.

How to execute javascript in selenium?

#### **Answer:**

JavaScript can be executed in selenium using JavaScriptExecutor. Sample code for javascript execution-

```
WebDriver driver = new FirefoxDriver();
if (driver instanceof JavascriptExecutor) {
    ((JavascriptExecutor)driver).executeScript("{'JavaScript Code'}");
}
```

#### **Question 82:**

What is HtmlUnitDriver?

#### **Answer:**

HtmlUnitDriver is the fastest WebDriver. Unlike other drivers (FirefoxDriver, ChromeDriver etc), the HtmlUnitDriver is non-GUI, while running no browser gets launched.

#### **Question 83:**

How to handle hidden elements in Selenium webDriver?

What are the advantages of POM?

#### **Answer:**

The advantages are POM are-

1. Using POM we can create an Object Repository, a set of web elements in separate files along with their associated functions. Thereby keeping code clean.
2. For any change in UI(or web elements) only page object files are required to be updated leaving test files unchanged.
3. It makes code reusable and maintainable.

#### **Question 86:**

What is Page Factory?

#### **Answer:**

Page factory is an implementation of Page Object Model in selenium. It provides @FindBy annotation to find web elements and PageFactory.initElements() method to initialize all web elements defined with @FindBy

```
FirefoxProfile profile = new FirefoxProfile();
profile.setAcceptUntrustedCertificates(true);
profile.setAssumeUntrustedCertificateIssuer(false);
WebDriver driver = new FirefoxDriver(profile);
```

#### **Question 80:**

How to do drag and drop in selenium?

#### **Answer:**

Using Action class, drag and drop can be performed in selenium. Sample code-

```
Actions builder = new Actions(driver);
Action dragAndDrop = builder.clickAndHold(SourceElement)
    .moveToElement(TargetElement)
    .release(TargetElement)
    .build();
dragAndDrop.perform();
```

#### **Question 81:**

#### **Answer:**

Using javaScript executor we can handle hidden elements-  
((JavascriptExecutor(driver)).executeScript("document.getElementsByClassName(ElementLocator).click();");

#### **Question 84:**

What is Page Object Model or POM?

#### **Answer:**

Page Object Model(POM) is a design pattern in selenium. A design pattern is a solution or a set of standards that are used for solving commonly occurring software problems.

POM helps to create a framework for maintaining selenium scripts. In POM for each page of the application a class is created having the web elements belonging to the page and methods handling the events in that page. The test scripts are maintained in separate files and the methods of the page object files are called from the test scripts file.

#### **Question 85:**

annotation.

```
public class SamplePage {
    WebDriver driver;
    @FindBy(id="search")
    WebElement searchTextBox;
    @FindBy(name="searchBtn")
    WebElement searchButton;
    //Constructor
    public samplePage(WebDriver driver){
        this.driver = driver;
        //initElements method to initialize all elements
        PageFactory.initElements(driver, this);
    }
    //Sample method
    public void search(String searchTerm){
        searchTextBox.sendKeys(searchTerm);
    }
}
```

```
    searchButton.click();
}
}
```

**Question 87:**

What is an Object repository?

**Answer:**

An object repository is centralized location of all the object or WebElements of the test scripts. In selenium we can create object repository using Page Object Model and Page Factory design patterns.

**Question 88:**

What is a data driven framework?

**Answer:**

A data driven framework is one in which the test data is put in external files like csv, excel etc separated from test logic written in test script files. The test data drives the test cases, i.e. the test methods run for each set of test data values. TestNG provides inherent support for data driven testing using

it is associated with combination of data driven and keyword driven frameworks where both the test data and test actions are kept in external files(in the form of table).

**Question 91:**

What are the different Selenium components?

**Answer:**

The suite package constitutes of the following sets of tools :

- **Selenium Integrated Development Environment (IDE)**– It is a Firefox plug-in which is easily installable along with other plug-ins. Since it has a simple framework, Selenium IDE should be used as prototyping tool. For advanced use cases, Selenium RC or WebDriver would be suitable.
- **Selenium Remote Control (RC)** –It is a testing framework that allows the developers to write the code in any programming language. Usually it supports a wide spectrum of languages like Java, PHP, Python, Perl, C#, etc.
- **Selenium WebDriver** –This tool has more advanced features than that of

What is selenium Grid?

**Answer:**

Selenium grid is a tool that helps in distributed running of test scripts across different machines having different browsers, browser version, platforms etc in parallel. In selenium grid there is hub that is a central server managing all the distributed machines known as nodes.

**Question 94:**

When do we use Selenium Grid?

**Answer:**

When want to run same or different test scripts on multiple platforms at a time, we use selenium grid. This helps us to test under various environments and also the loss of time consumed for testing is reduced.

**Question 95:**

What are some advantages of selenium grid?

**Answer:**

The advantages of selenium grid are-

@dataProvider annotation.

**Question 89:**

What is a keyword driven framework?

**Answer:**

A keyword driven framework is one in which the actions are associated with keywords and kept in external files e.g. an action of launching a browser will be associated with keyword - launchBrowser(), action to write in a textbox with keyword - writeInTextBox(webElement, textToWrite) etc. The code to perform the action based on a keyword specified in external file is implemented in the framework itself.

In this way the test steps can be written in a file by even a person of non-programming background once all the identified actions are implemented.

**Question 90:**

What is a hybrid framework?

**Answer:**

A hybrid framework is a combination of one or more frameworks. Normally

RC and IDE. It applies better approach to automate the browser activities. It does not typically rely upon JavaScript, it interacts with browser.

- **Selenium Grid** –It works along with Selenium RC and runs tests on different nodes using different browsers simultaneously.

**Question 92:**

What programming languages are best for writing Selenium tests?

**Answer:**

Selenium is a portable testing technology that comes with its own testing language called 'Selenese'. However it also supports other programming language such as Java, Ruby, Python, C#, etc. When the tests are written in these programming languages, the codes communicate with the help of Selenium Client API.

---

**Question 93:**

1. It allows running test cases in parallel thereby saving test execution time.
2. Multi browser testing is possible using selenium grid by running the test on machines having different browsers.
3. It is allows multi-platform testing by configuring nodes having different operating systems.

**Question 96:**

What is a hub in selenium grid?

**Answer:**

A hub is server or a central point in selenium grid that controls the test executions on the different machines.

**Question 97:**

What is a node in selenium grid?

**Answer:**

Nodes are the machines which are attached to the selenium grid hub and have selenium instances running the test scripts. Unlike hub there can be multiple nodes in selenium grid.

**Question 98:**

Explain the line of code Webdriver driver = new FirefoxDriver();

**Answer:**

In the line of code **Webdriver driver = new FirefoxDriver();** 'WebDriver' is an interface and we are creating an object of type WebDriver instantiating an object of FirefoxDriver class.

**Question 99:**

What is the purpose of creating a reference variable- 'driver' of type WebDriver instead of directly creating a FireFoxDriver object or any other driver's reference in the statement Webdriver driver = new FirefoxDriver();?

**Answer:**

By creating a reference variable of type WebDriver we can use the same variable to work with multiple browsers like ChromeDriver, IEDriver etc.

**Question 100:**

What Are Various Build And Deploy Tools Apart From Maven Used In The Industry?

It is a perfect light-weight alternative of Maven.

**Gradle.**

Gradle is an open source build tool. It automates the build process taking over Ant and Maven tools. Unlike its predecessors, Gradle makes use of a domain-specific language (DSL) instead of the XML templates for defining the project configuration. This DSL follows the groovy syntax.

It proposes a polyglot build system which integrates projects with different technologies and programming languages. It makes the dependency version management easier than Maven with the help of dynamic versioning.

**Question 101:**

What is TestNG?

**Answer:**

TestNG(NG for Next Generation) is a testing framework that can be integrated with selenium or any other automation tool to provide multiple capabilities like assertions, reporting, parallel test execution etc.

**Question 102:**

**Answer:**

testng.xml file is used for configuring the whole test suite. In testng.xml file we can create test suite, create test groups, mark tests for parallel execution, add listeners and pass parameters to test scripts. Later this testng.xml file can be used for triggering the test suite.

**Question 104:**

How can we pass parameter to test script using testNG?

**Answer:**

Using @Parameter annotation and 'parameter' tag in testng.xml we can pass parameters to test scripts.

Sample testng.xml -

```
<suite name="sampleTestSuite">
<test name="sampleTest">
<parameter name="sampleParamName" value="sampleParamValue"/>
<classes>
<class name="TestFile" />
```

**Answer:**

Use of Maven is wide-spread not just because it builds code or manages dependencies on the fly but also due to its ability for supporting deployments. So it eases up a lot of post development tasks. However, we can't ignore the challenges that show up while working with complex commands and XML configuration.

Hence, here are a bunch of other tools that are worth trying:

**Ant.**

ANT is an acronym for Another Neat Tool. It is a command-line tool developed by James Duncan Davidson and written in Java. It makes use of XML configuration to build modules, define targets, and specify dependencies.

**IV.**

Ivy is another build tool from Apache and brought as an extension to Ant. It helps in managing the external dependencies during the development, provides a way of adding them to the classpath, and packaging them with the application.

What are some advantages of testNG?

**Answer:**

Following are the advantages of testNG-

1. TestNG provides different assertions that helps in checking the expected and actual results.
2. It provides parallel execution of test methods.
3. We can define dependency of one test method over other in TestNG.
4. We can assign priority to test methods in selenium.
5. It allows grouping of test methods into test groups.
6. It allows data driven testing using @DataProvider annotation.
7. It has inherent support for reporting.
8. It has support for parameterizing test cases using @Parameters annotation.

**Question 103:**

What is the use of testng.xml file?

```
</classes>
</test>
</suite>
Sample test script-
public class TestFile {
    @Test
    @Parameters("sampleParamName")
    public void parameterTest(String paramValue) {
        System.out.println("Value of sampleParamName is - " + sampleParamName);
    }
}
```

**Question 105:**

How can we create data driven framework using testNG?

**Answer:**

Using @DataProvider we can create a data driven framework in which data is passed to the associated test method and multiple iteration of the test runs

for the different test data values passed from the @DataProvider method. The method annotated with @DataProvider annotation return a 2D array of object.

```
//Data provider returning 2D array of 3*2 matrix
@DataProvider(name = "dataProvider1")
public Object[][] dataProviderMethod1() {
    return new Object[][] {{ {"kuldeep", "rana"}, {"k1", "r1"}, {"k2", "r2"} }};
}
//This method is bound to the above data provider returning 2D array of
3*2 matrix
//The test case will run 3 times with different set of values
@Test(dataProvider = "dataProvider1")
public void sampleTest(String s1, String s2) {
    System.out.println(s1 + " " + s2);
}
```

**Question 106:**  
How to run a group of test cases using TestNG?

What is the use of @Listener annotation in TestNG?

**Answer:**

TestNG provides us different kind of listeners using which we can perform some action in case an event has triggered. Usually testNG listeners are used for configuring reports and logging. One of the most widely used listener in testNG is ITestListener interface. It has methods like onTestSuccess, onTestFailure, onTestSkipped etc. We need to implement this interface creating a listener class of our own. After that using the *@Listener annotation we can use specify that for a particular test class our customized listener class should be used.*

```
@Listeners(PackageName.CustomizedListenerClassName.class)
public class TestClass {
    WebDriver driver= new FirefoxDriver();@Test
    public void testMethod(){
        //test logic
    }
}

@Test
public void TestMethod() {
    System.out.println(str);
}

public class TestFactory{
    //The test methods in class TestClass will run twice with data "k1" and "k2"
    @Factory
    public Object[] factoryMethod() {
        return new Object[] { new TestClass("K1"), new TestClass("k2") };
    }
}
```

**Question 109:**  
What is difference between @Factory and @DataProvider annotation?

**Answer:**  
@Factory method creates instances of test class and run all the test

**Answer:**

TestNG allows you to perform sophisticated groupings of test methods. Not only can you declare that methods belong to groups, but you can also specify groups that contain other groups. Then TestNG can be invoked and asked to include a certain set of groups (or regular expressions) while excluding another set. This gives you maximum flexibility in how you partition your tests and doesn't require you to recompile anything if you want to run two different sets of tests back to back.

Groups are specified in your testng.xml file and can be found either under the <test> or <suite> tag. Groups specified in the <suite> tag apply to all the <test> tags underneath.

```
@Test (groups = { "smokeTest", "functionalTest" })
public void loginTest(){
    System.out.println("Logged in successfully");
}
```

**Question 107:**

**Question 108:**

What is the use of @Factory annotation in TestNG?

**Answer:**

@Factory annotation helps in dynamic execution of test cases. Using @Factory annotation we can pass parameters to the whole test class at run time. The parameters passed can be used by one or more test methods of that class.

Example - there are two classes TestClass and the TestFactory class. Because of the @Factory annotation the test methods in class TestClass will run twice with the data "k1" and "k2"

```
public class TestClass{
    private String str;
    //Constructor
    public TestClass(String str) {
        this.str = str;
    }
}
```

methods in that class with different set of data.

Whereas, @DataProvider is bound to individual test methods and run the specific methods multiple times.

**Question 110:**

Explain what does @Test(invocationCount=?) and @Test(threadPoolSize=?) indicate.

**Answer:**

@Test(invocationCount=?) is a parameter that indicates the number of times this method should be invoked.

@Test(threadPoolSize=?) is used for executing suites in parallel. Each suite can be run in a separate thread.

To specify how many times @Test method should be invoked from different threads, you can use the attribute threadPoolSize along with invocationCount. Example:

```
@Test(threadPoolSize = 3, invocationCount = 10)
public void testCase1() {
}
```

**Question 111:**

How can we make one test method dependent on other using TestNG?

**Answer:**

Using dependsOnMethods parameter inside @Test annotation in testNG we can make one test method run only after successful execution of dependent test method.

```
@Test(dependsOnMethods = { "preTests" })
```

**Question 112:**

How can we set priority of test cases in TestNG?

**Answer:**

Using priority parameter in @Test annotation in TestNG we can define priority of test cases. The default priority of test when not specified is integer value 0. Example-

methods in the current class have been run.

- @BeforeTest-The annotated method will run before any test method belonging to the classes inside the <test> tag is run.
- @AfterTest-The annotated method will run after all the test methods belonging to the classes inside the <test> tag have run.

**Question 114:**

What are some common assertions provided by testNG?

**Answer:**

Some of the common assertions provided by testNG are-

1. assertEquals(String actual, String expected, String message) - (and other overloaded data type in parameters)
2. assertNotEquals(double data1, double data2, String message) - (and other overloaded data type in parameters)
3. assertFalse(boolean condition, String message)
4. assertTrue(boolean condition, String message)

**Answer:**

Log4j is an open source API widely used for logging in Java. It supports multiple levels of logging like - ALL, DEBUG, INFO, WARN, ERROR, TRACE and FATAL.

**Question 117:**

What is the use of logging in automation?

**Answer:**

Logging helps in debugging the tests when required and also provides a storage of test's runtime behaviour.

**Question 118:**

What is the difference between "type" and "typeAndWait" command?

**Answer:**

"type" command is used to type keyboard key values into text box of software web application. It can also be used for selecting values of combo box whereas "typeAndWait" command is used when your typing is completed and software web page start reloading. This command will wait for software

```
@Test(priority=1)
```

**Question 113:**

What are commonly used TestNG annotations?

**Answer:**

The commonly used TestNG annotations are-

- @Test- @Test annotation marks a method as Test method.
- @BeforeSuite- The annotated method will run only once before all tests in this suite have run.
- @AfterSuite-The annotated method will run only once after all tests in this suite have run.
- @BeforeClass-The annotated method will run only once before the first test method in the current class is invoked.
- @AfterClass-The annotated method will run only once after all the test

5. assertNotNull(Object object)
6. fail(boolean condition, String message)
7. true(String message)

**Question 115:**

How can we run test cases in parallel using TestNG?

**Answer:**

In order to run the tests in parallel just add these two key value pairs in suite-

- parallel="{methods/tests/classes}"
- thread-count="{number of threads you want to run simultaneously}".

```
<suite name="ArtOfTestingTestSuite" parallel="methods" thread-count="5">
```

**Question 116:**

Name an API used for logging in Java.

application page to reload. If there is not page reload event on typing, you have to use simple "type" command.

**Question 119:**

What is the difference between verify and assert commands?

**Answer:**

Assert: Assert allows to check whether an element is on the page or not. The test will stop on the step failed, if the asserted element is not available. In other words, the test will terminate at the point where check fails.

Verify: Verify command will check whether the element is on the page, if it is not then the test will carry on executing. In verification, all the commands are going to run guaranteed even if any of test fails.

**Question 120:**

What is JUnit Annotations and what are different types of annotations which are useful ?

**Answer:**

In JAVA a special form of syntactic meta-data can be added to Java source

code, this is known as Annotations. Variables, parameters, packages, methods and classes are annotated some of the Junit annotations which can be useful are

- **@Test:**Annotation lets the system know that the method annotated as @Test is a test method. There can be multiple test methods in a single test script.
- **@Before:**Method annotated as @Before lets the system know that this method shall be executed every time before each of the test methods.
- **@After:**Method annotated as @After lets the system know that this method shall be executed every time after each of the test methods.
- **@BeforeClass:**Method annotated as @BeforeClass lets the system know that this method shall be executed once before any of the test methods.
- **@AfterClass:**Method annotated as @AfterClass lets the system know that this method shall be executed once after any of the test methods.
- **@Ignore:**Method annotated as @Ignore lets the system know that this method shall not be executed.

difference is that the browsers are launched in a special mode called heightened privileges. By using these browser mode, Selenium core can open the AUT directly and also read/write its content without passing the whole AUT through the Selenium RC server.

**Question 123:**

What are the features of TestNG and list some of the functionality in TestNG which makes it more effective?

**Answer:**

TestNG is a testing framework based on JUnit and NUnit to simplify a broad range of testing needs, from unit testing to integration testing. And the functionality which makes it efficient testing framework are

- Support for annotations
- Support for data-driven testing
- Flexible test configuration
- Ability to re-execute failed test cases

**Question 126:**

Explain how to assert text of webpage using selenium 3.0 ?

**Answer:**

```
WebElement el = driver.findElement(By.id("ElementID"))
//get test from element and stored in text variable
String text = el.getText();
//assert text from expected
Assert.assertEquals("Element Text", text);
```

**Question 127:**

Explain what is the difference between Borland Silk and Selenium?

**Answer:**

- Borland Silk test is not a free testing tool
- Selenium is completely free test automation tool
- Silk test supports only Internet Explorer and Firefox
- Selenium supports many browsers like Internet Explorer, Firefox, Safari, Opera and so on

**Question 121:**

What is same origin policy? How you can avoid same origin policy?

**Answer:**

The "Same Origin Policy" is introduced for security reason, and it ensures that content of your site will never be accessible by a script from another site. As per the policy, any code loaded within the browser can only operate within that website's domain.

To avoid "Same Origin Policy" proxy injection method is used, in proxy injection mode the Selenium Server acts as a client configured HTTP proxy , which sits between the browser and application under test and then masks the AUT under a fictional URL

**Question 122:**

What is the purpose behind heightened privileges browsers?

**Answer:**

The purpose of heightened privileges is similar to Proxy Injection, allows websites to do something that are not commonly permitted. The key

**Question 124:**

Which attribute you should consider throughout the script in frame for "if no frame Id as well as no frame name"?

**Answer:**

You can use driver.findElements(By.xpath("//iframe"))

This will return list of frames.

You will need to switch to each and every frame and search for locator which we want.

Then break the loop

**Question 125:**

Explain how you can login into any site if it's showing any authentication popup for password and username?

**Answer:**

Pass the username and password with url

Syntax: http://username:password@url

ex- http://creyate:tom@www.gmail.com

- Silk test uses test scripting language
- Selenium suite has the flexibility to use many languages like Java, Ruby, Perl and so on
- Silk test can be used for client server applications
- Selenium can be used for only web application

**Question 128:**

Can we use Selenium grid for performance testing?

**Answer:**

Yes but not as effectively as a dedicated performance testing tool like Loadrunner.

**Question 129:**

While injecting capabilities in webdriver to perform tests on a browser which is not supported by a webdriver what is the limitation that one can come across?

**Answer:**

Major limitation of injecting capabilities is that "findElement" command

may not work as expected.

**Question 130:**

Explain how you can find broken images in a page using Selenium Web driver?

**Answer:**

To find the broken images in a page using Selenium web driver is

- Get Xpath and get all the links in the page using tag name
- In the page click on each and every link
- Look for 404/500 in the target page title

**Question 131:**

Explain how you can handle colors in web driver?

**Answer:**

To handle colors in web driver you can use

Use `getCssValue(argo)` function to get the colors by sending 'color' string as an argument

**Question 134:**

Explain what are the limitations of Selenium IDE?

**Answer:**

The limitations of Selenium IDE

- Exceptional handling is not present
- Selenium IDE uses only HTML languages
- External databases reading is not possible with IDE
- Reading from the external files like .txt, .xls is not possible
- Conditional or branching statements execution like if,else, select statements is not possible

**Question 135:**

What are regular expressions? How you can use regular expressions in Selenium?

**Answer:**

A regular expression is a special text string used for describing a search

Mention why to choose Python over Java in Selenium?

**Answer:**

Few points that favor Python over Java to use with Selenium is,

- Java programs tend to run slower compared to Python programs.
- Java uses traditional braces to start and ends blocks, while Python uses indentation.
- Java employs static typing, while Python is dynamically typed.
- Python is simpler and more compact compared to Java.

**Question 139:**

Mention what is IntelliJ?

**Answer:**

IntelliJ is an IDE that helps you to write better and faster code for Selenium. IntelliJ can be used in the option to Java bean and Eclipse.

**Question 140:**

Mention in what ways you can customize TestNG report?

**Question 132:**

Which web driver implementation is fastest?

**Answer:**

HTMLUnit Driver implementation is fastest, HTMLUnitDriver does not execute tests on browser but plain http request, which is far quick than launching a browser and executing tests

**Question 133:**

What is Selenese and what are the types of Selenese ?

**Answer:**

Selenese is a selenium set of command which are used for running the test.

There are three types of Selenese

- Actions: It is used for performing the operations and interactions with the target elements
- Assertions: It is used as a check points
- Accessors: It is used for storing the values in a variable

pattern. In Selenium IDE regular expression can be used with the keyword- `reg-exp`: as a prefix to the value and patterns needs to be included for the expected values.

**Question 136:**

What are core extension ?

**Answer:**

If you want to "extend" the default functionality provided by Selenium Function Library , you can create a Core Extension. They are also called "User Extension". You can even download ready-made Core Extension created by other Selenium enthusiasts.

**Question 137:**

Using Selenium how can you handle network latency?

**Answer:**

To handle network latency you can use `driver.manage.pageLoadingTime` for network latency

**Question 138:**

**Answer:**

You can customize TestNG report in two ways,

- Using ITestListener Interface
- Using IReporter Interface

**Question 141:**

To generate pdf reports mention what Java API is required?

**Answer:**

To generate pdf reports, you need Java API IText.

**Question 142:**

Mention what is Listeners in Selenium WebDriver?

**Answer:**

In Selenium WebDriver, Listeners "listen" to the event defined in the selenium script and behave accordingly. It allows customizing TestNG reports or logs. There are two main listeners i.e. WebDriver Listeners and TestNG Listeners.

**Question 143:**

Mention what are the types of Listeners in TestNG?

**Answer:**

The types of Listeners in TestNG are,

- IAnnotationTransformer
- IAnnotationTransformer2
- IConfigurable
- IConfigurationListener
- IExecutionListener
- IHookable
- IInvokedMethodListener
- IInvokedMethodListener2
- IMethodInterceptor
- IReporter
- ISuiteListener

Mention when to use AutoIT?

**Answer:**

Selenium is designed to automate web-based applications on different browsers. But to handle window GUI and non-HTML popups in the application you need AutoIT. know more about How to use AutoIT with Selenium

---

**Question 147:**

Mention why do you need Session Handling while working with Selenium?

**Answer:**

While working with Selenium, you need Session Handling. This is because, during test execution, the Selenium WebDriver has to interact with the browser all the time to execute given commands. At the time of execution, it is also possible that, before current execution completes, someone else starts execution of another script, in the same machine and in the same type of browser. So to avoid such situation you need Session Handling.

**Question 144:**

Mention what is desired capability? How is it useful in terms of Selenium?

**Answer:**

The desired capability is a series of key/value pairs that stores the browser properties like browser name, browser version, the path of the browser driver in the system, etc. to determine the behavior of the browser at run time.

For Selenium, it can be used to configure the driver instance of Selenium WebDriver.

When you want to run the test cases on a different browser with different operating systems and versions.

**Question 145:**

For Database Testing in Selenium Webdriver what API is required?

**Answer:**

For Database Testing in Selenium Webdriver, you need JDBC (Java Database Connectivity) API. It allows you to execute SQL statements.

**Question 146:****Question 148:**

Mention what are the advantages of Using Git Hub For Selenium?

**Answer:**

The advantages of Using Git Hub for Selenium are

- Multiple people when they work on the same project they can update project details and inform other team members simultaneously.
- Jenkins can help you to build the project from the remote repository regularly. This helps you to keep track of failed builds.

**Question 149:**

What Is Continuous Integration (CI), What Are Its Benefits?

**Answer:**

Continuous Integration is a software engineering technique which regulates the development process by automating the build and testing of code for every commit made to the version control.

It greatly helps in making sure that new changes didn't have any adverse

effect on the build and existing functionality is intact. Even in a failed scenario, everyone including the one responsible for the changes would know about the failure without any delay. With such a timely reporting, the developer not only finds fair time to fix the issue but also gets the prompt feedback.

It promotes best practices amongst the developers by making them merge their changes with other's code before checking into a shared repository. Timely code integration eliminates chances of conflicts, duplication of functionality, and diverging design tactics.

CI proposes to use a version control system like Git which supports Git workflows for intelligent branching. It suggests having a main branch which keeps the stable code and can be used to deliver a build at any point in time. On the other hand, the developer may create transient feature branches to begin his work without interrupting others. He then needs to sync his code with the Main Branch at regular intervals. Once the feature is complete, he should merge the code into the main after due verification. The feature branch is subject to deletion when no longer it is required.

Some other CI tools are TeamCity, Travis CI, and Circle CI.

#### **Question 150:**

What Are Different CI Tools Available In The Industry?

#### **Answer:**

There are many CI tools available which can assist in implementing the continuous integration process. Here we are listing a few of the popular ones.

##### **1. Jenkins.**

Jenkins is undoubtedly the most popular CI tool used across the IT industry. It is open source, written in Java and created by Kohsuke Kawaguchi. It offers easy management via commands and GUI. It supports a rich set of plugins to customize both build and automation jobs.

##### **2. Bamboo.**

Bamboo is another great CI tool but a paid one. It's from the Atlassian, the company which provides helping tools like JIRA, BitBucket for software development. Unlike Jenkins which uses Build/Automation jobs terminology, Bamboo has introduced the concepts of projects and plans. It targets to bring more user-friendly features so that the user can implement CI with ease.

#### **Official sites**

- Home page: <http://www.seleniumhq.org> and <http://code.google.com/p/selenium>
- Manual and documentation: <http://seleniumhq.org/docs/>
- Wiki: <http://code.google.com/p/selenium/w/list>
- Blog: <http://aroraglobalservices.blogspot.com/>

#### **Blogs**

- The blog of Adam Goucher at <http://element34.ca/?cat=blog> & <http://adam.goucher.ca/?p=1331>
- The blog of David Burns at <http://www.theautomatedtester.co.uk/>
- Alister Scott at <http://watirmelon.com/>