\*\*Why we go for Automation?

Testing an Application with human interactions is called Mannual Testing

Problems in Mannual Testing :

No Resuability, More Resources required, Time Consuming, Budget is high, Human errors causes poor quality

1.What is Automation Testing?

Automation testing is the process of testing the software/Application using an third part automation tool to find the defects/bugs. In this process, executing the test scripts and generating the results, generating the reports are performed automatically by automation tools. Some most popular tools to do automation testing are HP QTP/UFT, Selenium WebDriver, Katalon, etc.,

Automation testing is required when huge amount of regression test cases we have. Better to automate instead of every time performing regression testing manually. If new features are there test it manually and then automate them.

2. What are the benefits of Automation Testing?

1. Saves time and money. (eg:1000 test cases by 4 mannual resources = 1 automation resource, he will completes 80% of testing remaining 20% by +manual tester, saves human effort and also salary).
2. Automation testing is faster in execution.
3. Reusability of code. Create one time and execute multiple times with less or no maintenance.
4. Easy reporting. It generates automatic reports after test execution. (eg: manual testers will execute 1000 tc’s and generate report and publish there reports and there is no guarantee, there may be chances of missing test cases or execution results, there may be human mistakes by testers)
5. Easy for compatibility testing. It enables parallel execution in the combination of different OS and browser environments. (cross browsers)
6. Low-cost maintenance. (eg: compare to 4 mannual testers, 1 automate tester is sufficient). It is cheaper compared to manual testing in a long run.
7. Automated testing is more reliable. (sometimes manual testing also reliable, than automation)(reliable – trustable)
8. Automated testing is more powerful and versatile.(eg: it supports more tools and technologies, integration with Jenkins, github, cross browser testing)
9. It is mostly used for regression testing. Supports execution of repeated test cases.
10. Minimal manual intervention. Test scripts can be run unattended. (eg: can execute at any time using Jenkins, no need of manually available there to execute, schedule in Jenkins it will run and send us report)
11. Maximum coverage. It helps to increase the test coverage. (if there 100 tc’s similar type, u can copy paste it).

3. What type of tests have you automated?

(Automation is always done with selenium functional automation, non-functionality means security, performance testing, volume testing, load testing, we have different tools for that. We do only functional testing i.e regression testing.)

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, integration testing.

4. How many test cases you have automated per day?

It depends on the project and 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. (Sometimes 15-20 it depends on project test case scenario lengthy and complexity)

5. What is a Framework?

A framework defines a set of rules which we can follow in a systematic way to achieve the desired results. (we define different components and integrate them, each and every component doing its own work, Eg: we have created testng, configuration, utility, reporting structure, test cases of testng, .. all these are components, we combine or integrate all those things called as framework)

There are different types of automation frameworks and the most common ones are:

* Data Driven Testing Framework (-driving the data from excel or different sources)
* Keyword Driven Testing Framework (- we wont prefer this, because it requires lot of effort and maintenance, keyword driven with QTP is very famous)
* Hybrid Testing Framework (-combing different approaches)

6. Have you created any Framework?

If you are a beginner:

(I haven’t created framework from the scratch, but I have contributed a lot in terms of creating libraries, creating the testcases,

or

I haven’t created framework from the scratch, my team lead and sr. Software engineer creates the libraries and utility for us on top of that I am contributed creating the test scenarios and writing testing classes , preparing the testdata for that.

If you are an experienced tester: Yes, I have created a framework.  Or I have involved or contributed in the creation of the framework.

Yes, I have worked with framework when there is no process overthere in my previous company they have only manual testers, they hired me for automation testing, we started practice, we gave the poc, demo and then we started framework after approving everything. We designed basic skeleton of framework and the foundation, on top of that we started designing test cases.

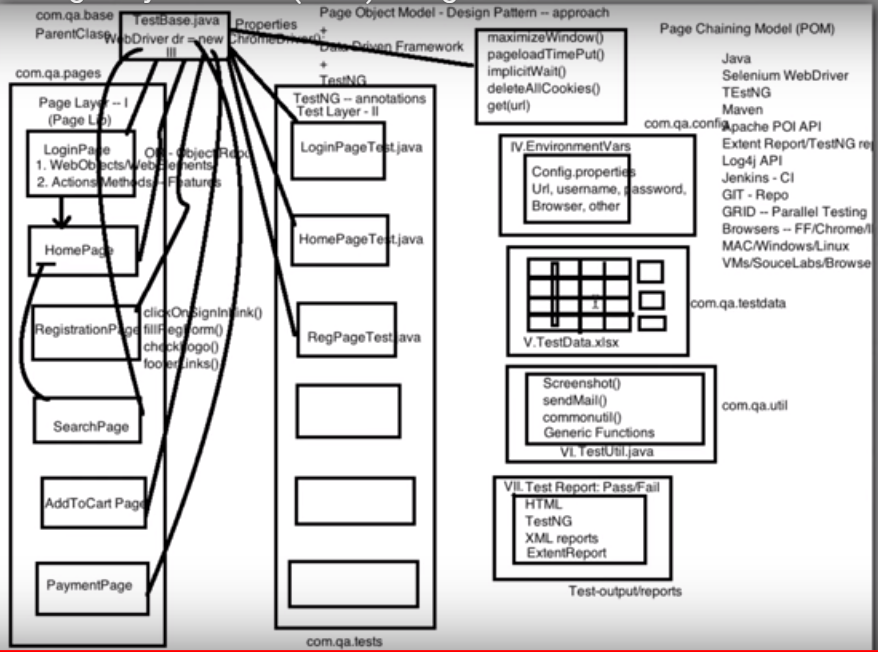
So I am having good exposure and experience , how to design framework either it is data driven, hybrid framework , page object model pattern, page factory pattern. I am very much comfortable with those things.

7. Can you explain the Framework which you have used in your Selenium Project?

Here we have clearly explained each component of Framework.

Page object model diagram

I have taken the maven project, in that we used page object model design pattern



8. Why do you prefer Selenium Automation Tool?

Primarily, Selenium was created by Jason Huggins in 2004. An engineer at ThoughtWorks, he was working on a web application that required frequent testing.

1. Free and open source – (It is free available in the market, Thoughtworks company releasing the different components, different versions of selenium automation in the market. The latest version of selenium is 3.X (3.141.59)
2. it have large user base and helping communities (To help there are lot of blogs, groups, channels, communities are available in market, youtube)
3. Cross browser compatibility (It supports different browsers like IE, Chrome, Safari, Firefox, Opera)
4. Platform compatibility (It supports different Operating systems like Windows, MAC, Unix, Linux)
5. Multiple programming languages support (different languages like Java, C#, Ruby, Perl, .net, Python, node.js)
6. It provide Integration, it can be easily integrated with TestNg, Junit, Cucumber, Jenkins, Maven, AutoIT, Sikuli for File Upload testing or Window Pop Up Testing.
7. Selenium can be easily executed at different cloud machines like BrowserStack, SauceLab…

\*\*Difference between selenium 2 & 3 versions?

[Selenium](https://www.youtube.com/playlist?list=PLyGqUe6Oa_5ELLaJLlXAoJpEv8khLTbJ2) 3 has bug fixes from selenium 2 also it is more mobile automation focused

WebDriver or selenium 2.0 is the most widely used tool of selenium family. This API fit with the same role played by Selenium RC. In simple way can be express webdriver in following format,

[Selenium](https://www.youtube.com/playlist?list=PLyGqUe6Oa_5Elc-Dv9jPzHKDx-m2GvMOd) 1.0 (IDE) + RC = Selenium 2.0

This has come to address to reduce limitations of RC server and to overcome the IDE drawbacks (only support for Firefox browser). In the long run WebDriver goal is to provide fully object oriented supporting testing tool which will give the durable solution to many modern advanced web application testing difficulties.

Simon Stewart at ThoughtWorks developed an Automation tool for the Browser known as Webdriver and later they merged these Selenium RC with Webdriver called Selenium Webdriver(Selenium 2.0).

[Selenium](https://www.youtube.com/playlist?list=PLyGqUe6Oa_5FRM-L27FnNe8b279Z2txhJ) 3.0 was developed in this year with the new features. It is a combination of Selenium 2.0 – Selenium 1.0, which means it supports Selenium 2.0 features but doesn’t have support for Selenium 1.0 i.e. Selenium Core.

Selenium 3.0 removed Selenium Core but supports Selenium RC indirectly through back-end Webdriver.

More than 9+ versions of IE are supported in Selenium 3.0.

Now [Selenium](https://www.youtube.com/playlist?list=PLyGqUe6Oa_5EUyOz-jBZ7hiYUyQ64JHVp)3.0 has became a W3C (World wide Web consortium) standard.

Selenium 3.0 removed Selenium Core but supports Selenium RC indirectly through back-end [Webdriver.](https://www.youtube.com/playlist?list=PLyGqUe6Oa_5Et6fES_EawDMvnq_KWe1Bf)

Firefox Browser version is supported by 47+. If we want to work with the older versions i.e. less than 47 version, there is no need to install Gecko driver but working with more than 47+ either in Selenium 2.0 or Selenium 3.0 requires Gecko driver installed and configured.

\*\*About Sikuli

Integration of Sikuli with Selenium WebDriver:

Now, no need to take the headache of XPath, CSS, ID and Name.

+Sikuli is an image-based open source tool to automate the GUI and can be used on any platform like Windows/Linux/Mac/Mobile.

+Sikuli uses a technique called image recognition to identify and control GUI components. Automate using images

• No API to launch Browser and URL – need to use selenium

• Can automate flash objects – YouTube video

• Desktop application automation

Screen s = **new** Screen();

Pattern pauseimg = **new** Pattern("fullscreen.png");

s.wait(pauseimg,5000);

s.click();

s.click();

\*\* Difference between QTP & Selenium? (look in material)

QTP – It is very expensive, Paid Tool. We have to buy the licence

It supports only couple of browsers IE also started on Chrome

It is heavy tool ,around 1.5GB

It supports only one language VBScript

It supports only on Windows

It tests both web and Desktop applications

Test Scripts can be developed only in QTP

Can execute tests in parallel but by using Quality Center again a paid product

Needs to have the Application opened in the desktop, if minimized it will terminate

Cant test for cross Browser and Cross Platform

Advantages:

It has its own Data driven approach, Keyword driven approach

It has its won Object Repository, own Object spy concept, own look and feel report structures

Selenium – Selenium doesn’t have its own look and feel

It has no graphical and User Interface

Here it is purely available in API, we should know the programming

Dis:

It doesn’t support web services testing

It cant do mobile testing - APPIUM

It cant do desktop testing – SOAPUI, HTTPClient

9. What is Selenium?

Selenium is an open source (free) automated testing suite to test web applications. It supports different platforms and browsers. It has gained a lot of popularity in terms of web-based automated testing and giving a great competition to the famous commercial tool HP QTP (Quick Test Professional) AKA HP UFT (Unified Functional Testing).

Selenium is a set of different software tools. Each tool has a different approach in supporting web based automation testing.

It has four components namely,

1. Selenium IDE (Integrated Development Environment)
2. Selenium RC (Remote Control) – selenium 1
3. Selenium WebDriver – selenium 2 & 3
4. Selenium Grid

10. What is Selenium IDE?

Selenium IDE (Integrated Development Environment) is a Firefox plugin. It is the simplest framework in the Selenium Suite. It allows us to record and playback the scripts. We have to export into different programming languages. Even though we can create scripts using Selenium IDE, we need to use Selenium RC or Selenium WebDriver to write more advanced and robust test cases. We don’t prefer to use Selenium IDE.

**Advantages**

1. Selenium IDE is very easy to use.
2. It has the capability to convert the test to different programming languages such as html, java etc
3. Programming language experience is not required for Selenium IDE
4. Selenium IDE provides Logging capabilities using file login plug-in.
5. In Selenium IDE, user can debug and set breakpoints
6. Selenium IDE is flexible for the users.

**Disadvantage**

1. Selenium IDE is Firefox plugin, thus its support is limited to Firefox only
2. It will not support iteration and conditional statement
3. Selenium IDE doesn't support error handling
4. Doesn't support test script grouping
5. Selenium IDE do not support Database testing

11. What is Selenese?

Selenese is the language which is used to write test scripts in Selenium IDE.

12. Which is the only browser that supports Selenium IDE to be used?

Firefox

1. What is Selenium RC? (look in material)

To overcome the limitation of IDE (ONLY Firefox), RC was introduced(dif. Browsers).

Selenium RC AKA Selenium 1. Selenium RC was the main Selenium project for a long time before the WebDriver merge brought up Selenium 2.

Selenium 1 is still actively supported (in maintenance mode). It relies on JavaScript for automation. It supports Java, Javascript, Ruby, PHP, Python, Perl and C#. It supports almost every browser out there.

The Selenium RC supports multiple browsers like IE, Firefox, Chrome, Safari, Opera etc. It also supports multiple languages like Java, Ruby, C#, Perl, Python etc.

. Similar to Selenium IDE, the RC is also has its limitations. Before start testing, we have to start & stop the server to execute you test.

So to overcome the all issues & increase the scope of Selenium RC, introduced new version of SE called Selenium WebDriver

14. What is Selenium WebDriver?

Selenium webdriver used to automate browsers anytype of web applications.

Selenium WebDriver AKA Selenium 2 is a browser automation framework that accepts commands and sends them to a browser. It is implemented through a browser-specific driver. It controls the browser by directly communicating with it. Selenium WebDriver supports Java, C#, PHP, Python, Perl, Ruby.

Main feature over the Selenium RC is that we don’t have to start the server in the Selenium WebDriver. One of the cool feature is that it supports the Android Testing & iPhone testing as well.

(WebDriver doesn’t require any server, it was based purely on json via protocol it completely available in API , they have given some jar files, we have to add those libraries and have to interacting with the browsers with the help of Json API.)

15. What is Selenium Grid?

Selenium Grid is a tool used together with Selenium RC to run tests on different machines against different browsers in parallel. That is, running multiple tests at the same time against different machines running different browsers and operating systems.

In simple words, it is used to distribute your test execution on multiple platforms and environments concurrently.

(Selenium Grid is used in Selenium RC and WebDriver to execute testcases on remote machines in parallel mode with different OS and browsers)

16. When do you use Selenium Grid?

Selenium Grid can be used to execute same or different test scripts on multiple platforms and browsers concurrently so as to achieve distributed test execution

(When we have to use cross browsing, when we have to save time, when we have to do parallel or concurrent testing)

17. What are the advantages of Selenium Grid?

It allows running test cases in parallel thereby saving test execution time.

It allows multi-browser testing

It allows us to execute test cases on multi-platform

18. What is a hub in Selenium Grid?

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

19. What is a node in Selenium Grid?

Node is the machine which is attached to the hub. There can be multiple nodes in Selenium Grid.

(Hub is a server, node is a client. A particular hub can be connected with multiple nodes. Hub controls the all the nodes. Hub tell the node to open particular browser and perform action)

20. What are the types of WebDriver APIs available in Selenium?

* Firefox Driver
* Gecko Driver
* InternetExplorer Driver
* Chrome Driver
* HTMLUNIT Driver // headless browser or ghost browser
* Opera Driver
* Safari Driver
* Android Driver
* iPhone Driver // not using now but present in market
* EventFiringWebDriver // to generate all event logs

\*\* Headless Browser Testing Using Chrome in Selenium:

It is also know as Ghost Driver or Headless Browser, executing program without browser.

WebDriver driver = new HTMLUnitDriver();

HTMLUNIT is not the part of selenium 3.x version, to use this we have to download HTMLUnit jar file. Before 2.5 version, it is the part of selenium.

Headless Browser using Chrome:

So if you hadn’t heard, Google has implemented a headless option for Chrome.

This will be included from version 59.

Following chrome options are required for Chrome HeadLess Browser:

WebDriver driver = null;

String url = "http://newtours.demoaut.com/";

System.setProperty("webdriver.chrome.driver", "F:\\Javaworkspace\\Practicing\_Programs\\driverFiles\\chromedriver.exe");

ChromeOptions options = new ChromeOptions();

options.addArguments("window-size=1400,800"); // without this it wont work, may get NullpointerException sometimes

options.addArguments("headless");

driver = new ChromeDriver(options);

Advantages:

1. Testing is happening behind the screen without browser launching
2. Very fast execution of testcases
3. It improves performance of the script

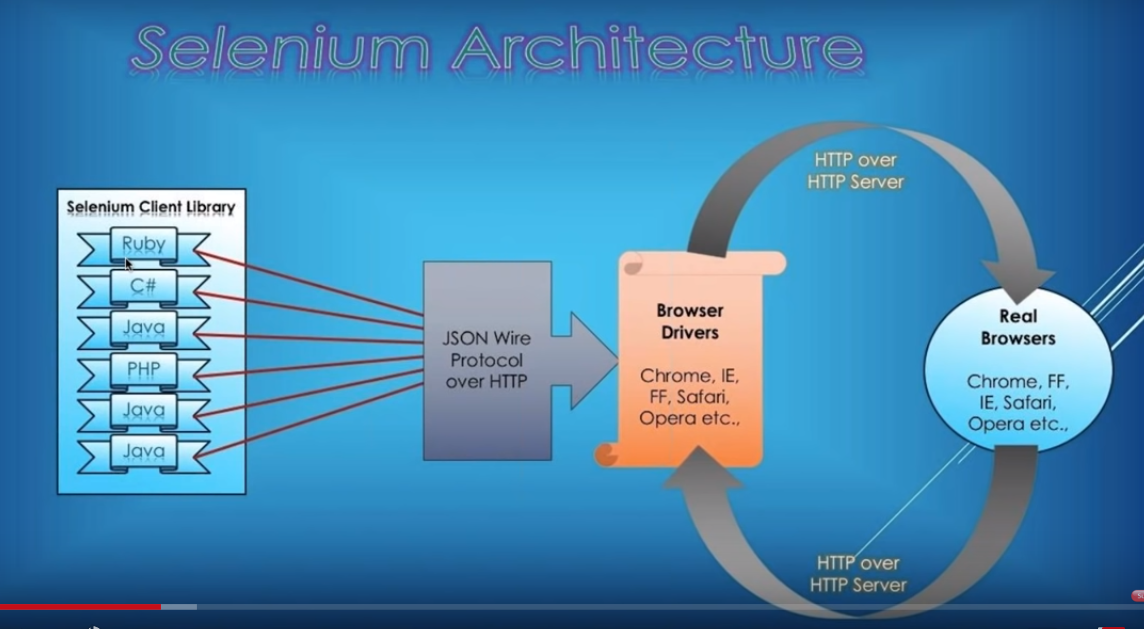
Disadvantages:

* 1. Notsuitable for Action class, user Actions, mousemovement, double click, drag&drop
  2. This is used for only small script

21. Which WebDriver implementation claims to be the fastest?

The fastest implementation of WebDriver is the HTMLUnitDriver. It is because the HTMLUnitDriver does not execute tests in the browser. It doesn’t launch browser. In selenium 3 we have to add it separately dependency or jar.

\*\*selenium Architecture



22. What are the Programming Languages supported by Selenium WebDiver?

* Java
* C#
* Python
* Ruby
* Perl
* PHP

23. What are the Operating Systems supported by Selenium WebDriver?

* Windows
* Linux
* Mac

24. What are the Open-source Frameworks supported by Selenium WebDriver?

* Junit // TDD Framework – Test Driven Development Framework
* TestNG // “”
* CUCUMBER // BDD Framework – Behavioural Driven Development Framework
* JBHEAVE // “”

25. What are the Locators available in Selenium?

We have 8 different types of locators using by. are:

1. ID –
2. ClassName –
3. Name –
4. TagName –
5. LinkText –
6. PartialLinkText –
7. XPath – address of any element or object in the page and performs some action

8 CSS Selector –

26. What is a XPath?

Xpath means extended HTML Path available in the form of XML document

XPath is used to locate the elements. Using XPath, we could navigate through elements and attributes in an XML document to locate web elements such as textbox, button, checkbox, Image etc., in a web page.

\*\* What is HTML DOM?

\*\* StaleExpectedExpection?

\*\* Dynamic Xpath?

 **Dynamic XPath** is used to locate exact attribute or decrease the number of matching nodes/result from a webpage and following **XPath** expressions can be used for the same.

At times, XPath may change dynamically and we need to handle the elements while writing scripts. Standard way of writing xpath may not work and we need to write dynamic XPath in selenium scripts. Let’s see different way of writing dynamic XPath in Selenium with examples:

1. [Using Single Slash](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/#SINGLESLASHXPATH)
2. [Using Double Slash](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/#DOUBLESLASHXPATH)
3. [Using Single Attribute](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/#SINGLEATTRIBUTEXPATH)
4. [Using Multiple Attribute](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/#MULTIPLEATTRIBUTEXPATH)
5. [Using AND](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/#ANDXPATH)
6. [Using OR](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/#ORXPATH)
7. [Using contains()](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/#CONTAINSXPATH)
8. [Using starts\_with()](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/#STARTSWITHXPATH)
9. [Using text()](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/#TEXTXPATH)
10. [Using last()](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/#LASTXPATH)
11. [Using position()](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/#POSITIONXPATH)
12. [Using index()](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/#INDEXXPATH)
13. [Using following xpath axes](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/#FOLLOWINGXPATH)
14. [Using preceding xpath axes](https://www.softwaretestingmaterial.com/dynamic-xpath-in-selenium/#PRECEDINGXPATH)

\*\*Relative Xpath?

\*\* Absolute Xpath?

27. What is the difference between “/” and “//”

Single Slash “/” – Single slash is used to create XPath with absolute path i.e. the XPath would be created to start selection from the document node/start node.

Eg: /html/body/div[3]/div[1]/table/tbody/tr[1]/td/input/

Double Slash “//” – Double slash is used to create XPath with relative path i.e. the XPath would be created to start selection from anywhere within the document.

Eg: //input[@id=”hello”]

28. What is the difference between Absolute Path and Relative Path?

Absolute XPath starts from the root node and ends with desired descendant element’s node. It starts with top HTML node and ends with input node. It starts with a single forward slash(/) as shown below.

/html/body/div[3]/div[1]/form/table/tbody/tr[1]/td/input

Relative XPath starts from any node in between the HTML page to the current element’s node(last node of the element). It starts with a double forward slash(//) as shown below.

//input[@id='email']

29. What is the difference between Assert and Verify in Selenium?

Assert: In simple words, if the assert condition is true then the program control will execute the next test step but if the condition is false, the execution will stop and further test step will not be executed.

Verify: In simple words, there won’t be any halt in the test execution even though the verify condition is true or false. So everytime it will be passed. It will failed only when unexpected exceptions are raised like JavaNullPointerException.

Verification means if else condition

30. What are Soft Assert and Hard Assert in Selenium?

Soft Assert: Soft Assert collects errors during *@Test,* Soft Assert does not throw an exception when an assert fails and would continue with the next step after the assert statement. It will give report after completion of execution.

Hard Assert: Hard Assert throws an AssertException immediately when an assert statement fails and test suite continues with next *@Test*

31. What are the verification points available in Selenium?

In Selenium IDE, we use Selenese Verify and Assert Commands as Verification points

In Selenium WebDriver, there is no built-in features for verification points. It totally depends on our coding style. (If your using Testng, u have to use Assertions). some of the Verification points are

To check for page title

To check for certain text

To check for certain element (text box, button, drop down, etc.)

32. How to launch a browser using Selenium WebDriver?

WebDriver is an Interface. We create Object of a WebDriver Interface.

Less than 2.53 version – there is no geckodriver

3.x – geckodriver for FF

System.setProperty(“webdriver.gecko.driver”,”path of geckoDriver”);

To launch Firefox Driver: WebDriver driver = new FirefoxDriver();

To launch Chrome Driver: WebDriver driver = new ChromeDriver();

To launch Internet Explorer Driver: WebDriver driver = new InternetExplorerDriver();

WebDriver = it is a interface

driver = it is a object reference variable or interface reference variable

new = keyword used to create object

InternetExplorerDriver() = it is a class

i.e InternetExplorerDriver() implementing WebDriver interface

new InternetExplorerDriver() = it is an object , who is referring , ‘driver’ is referring this object

This is know as Dynamic Polymorphism or Dynamic Binding or Runtime Polymorphism

\*\*\* How to launch Safari browser?

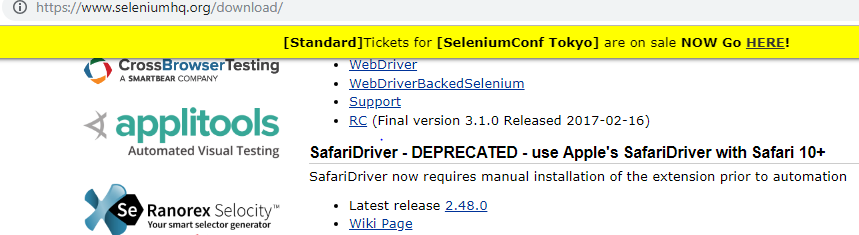
1. Download safari browser on your system
2. For safari browser no need to write system.setProperty can directly write driver = new SafariDriver();

WebDriver driver = **new** SafariDriver();

driver.get("http://google.com");

System.***out***.println(driver.getTitle());

1. There is no safari driver .exe file to download like other browsers
2. Now go to seleniumhq.org and download safariDriver as shown below



1. Click on add extension or Trust
2. Goto preferences check there whether selenium webdriver is added or not
3. Goto preferences -- > advanced enable the checkbox (Show Develop menu in menu bar)
4. Then go to develop tab and click on Allow Remote automation

33. Is the FirefoxDriver a Class or an Interface?

FirefoxDriver is a Java class, and it implements the WebDriver interface.

WebDriver interface

Chrome Firefox Safari IE

This concept is called Dynamic Polymorphism

34. What is the super interface of WebDriver?

SearchContext.

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

‘WebDriver‘ is an interface and we are creating an object reference of type WebDriver instantiating an object of FirefoxDriver class.

36. We do create a reference variable ‘driver’ of type WebDriver

WebDriver driver = new FirefoxDriver();

37. instead of creating

|  |  |
| --- | --- |
| 1  2  3  4  5 | WebDriver driver = new FirefoxDriver();    instead of creating    FirefoxDriver driver = new FirefoxDriver(); |

What is the purpose of doing this way?

If we create a reference variable driver of type WebDriver then we could use the same driver variable to work with any browser of our choice such as IEDriver, SafariDriver etc.,

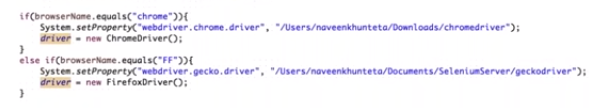
//FirefoxDriver driver = new FirefoxDriver(); // it is only to use single browser

ChromeDriver driver = new ChromeDriver(); // it is only to use single browser

driver.get(“http://www.google.com”);

WebDriver driver = new FirefoxDriver(); // with this variable you can use any browser

 // initializing once



// can use same driver variable for any number of times, U can write maximise, timeout with using this driver only

ChromeDriver driver = new ChromeDriver(); // it is not recommended

If you use this, u have to comment it and have to write another related browser code

37. What are the different exceptions you have faced in Selenium WebDriver?

1. WebDriverException
2. TimeoutException
3. NoAlertPresentException
4. NoSuchWindowException
5. NoSuchElementException
6. StaleElementReferenceException
7. IllegalStateException

NoSuchElementException:

If xpath is wrong, so if element not found then we get this exception

StaleElementReferenceException:

If HTML DOM is changed or if DOM is not loaded properly

TimeoutException:

If element is not found in the given time implicit or explicit

WebDriverException:

If your chrome is not launching properly may be because of version, using old version of selenium and latest version of chrome

IllegalStateException:

If we don’t specify property of Chrome or Firefox or IE then we will get this

For eg: if u executing program directly writing

Driver = new ChromeDriver(); without writing System.setProperty(“ ”,” ”);

Then you will get this exception

38. How To Login Into Any Site If It Is Showing Any Authentication Pop-Up For Username And Password?

Some sites may not have login screens directly, one pop up will come to enter username & password without login screen. i.e on alert page u r getting username & pwd screen where u can’t perform inspect element and write driver.findelement

To do this we pass username and password with the URL

Syntax:

http://username:password@url

e.g. <http://admin:admin123@xyz.com>

39. What are the types of waits available in Selenium WebDriver?

In Selenium we could see three types of waits such as Implicit Waits, Explicit Waits and Fluent Waits.

* Implicit Waits –
* Explicit Waits –
* Fluent Waits –
* PageLoadTimeOut
* Thread.sleep() – static wait

(First of all in selenium we have 2 waits 1. Static 2. Dynamic

Static wait is very straight forward which is Thread.sleep(), it is used in emergency cases in debugging purpose

In Dynamic wait we have Implicit Waits, Explicit Waits and Fluent Waits.

Implicit wait – it is a global wait which is applicable in the webdriver with driver instance.

Explicit wait – on the basis of condition, on the specific element we have to provide on the basis of expected condition

Fluent Wait – It is the mechanism of polling)

… Check the implicit\_explicit concept

40. What is Implicit Wait In Selenium WebDriver?

Implicit waits tell to the WebDriver to wait for a certain amount of time before it throws an exception. Once we set the time, WebDriver will wait for the element based on the time we set before it throws an exception. The default setting is 0 (zero). We need to set some wait time to make WebDriver to wait for the required time.

(It is global wait , waits for all elements)

41. What is WebDriver (explicit) Wait In Selenium WebDriver?

WebDriverWait is applied on a certain element with defined expected condition and time. This wait is only applied to the specified element. This wait can also throw an exception when an element is not found.

42. What is Fluent Wait In Selenium WebDriver?

FluentWait can define the maximum amount of time to wait for a specific condition and frequency with which to check the condition before throwing an “ElementNotVisibleException” exception.

43. How to input text in the text box using Selenium WebDriver?

By using sendKeys() method

WebDriver driver = new FirefoxDriver();

driver.get("https://www.gmail.com");

driver.findElement(By.xpath("xpath")).sendKeys("test");

44. How to input text in the text box without calling the sendKeys()?

// To initialize js object

JavascriptExecutor JS = ((JavascriptExecutor)driver);

// To enter username

JS.executeScript("document.getElementById(‘User').value=+’test.com'+");

It can be used when sendkeys are doesn’t work

, or xpath doesn’t work

45. How to clear the text in the text box using Selenium WebDriver?

By using clear() method

WebDriver driver = new FirefoxDriver();

driver.get("https://www.gmail.com");

driver.findElement(By.xpath("xpath\_of\_element1")).sendKeys("Software Testing Material Website");

driver.findElement(By.xpath("xpath\_of\_element1")).clear();

46. How to get a text of a web element?

By using getText() method

47. How to get an attribute value using Selenium WebDriver?

By using getAttribute(value);

Eg: driver.findElement(by.id(“hello”)).getAttribute(“type”);

48. How to click on a hyperlink using Selenium WebDriver?

We use click() method in Selenium to click on the hyperlink

1. linkText
2. PartialLinkText

driver.findElement(By.linkText(“Software Testing Material Website”)).click();

|  |  |
| --- | --- |
| 1 | driver.findElement(By.linkText(“Software Testing Material Website”)).click(); |

49. How to submit a form using Selenium WebDriver?

We use “submit” method on element to submit a form

driver.findElement(By.id("form\_1")).submit();

|  |
| --- |
| 1 |

Alternatively, you can use click method on the element which does form submission

1. Click()
2. Submit()

50. How to press ENTER key on text box In Selenium WebDriver?

To press ENTER key using Selenium WebDriver, We need to use Selenium Enum Keys with its constant ENTER.

driver.findElement(By.xpath("xpath")).sendKeys(Keys.ENTER);

For TAB

driver.findElement(By.xpath("xpath")).sendKeys(Keys.TAB);

For some extra combinations

driver.findElement(By.xpath("xpath")).sendKeys(Keys.Ctrl+F);

(When you write sendKeys(Keys. …..) U will find available options there)

51. How to pause a test execution for 5 seconds at a specific point?

By using java.lang.Thread.sleep(long milliseconds) method we could pause the execution for a specific time. To pause 5 seconds, we need to pass parameter as 5000 (5 seconds)

Thread.sleep(5000)

52. Is Selenium Server is needed to run Selenium WebDriver Scripts?

When we are distributing our Selenium WebDriver scripts to execute using Selenium Grid, we need to use Selenium Server.

(Yes, if you run on remote machines or virtual machines, here selenium server is responsible, it will creates a hub and controls the nodes

No, if you are running in local machines.)

53. What happens if I run this command. driver.get(“www.softwaretestingmaterial.com”) ;

An exception is thrown. We need to pass HTTP protocol within driver.get() method.

driver.get("http://www.softwaretestingmaterial.com");

|  |  |
| --- | --- |
| 1 | driver.get("http://www.softwaretestingmaterial.com"); |

54. What is the alternative to driver.get() method to open an URL using Selenium WebDriver?

Alternative method to driver.get(“url”) method is driver.navigate.to(“url”)

55. What is the difference between driver.get() and driver.navigate.to(“url”)?

driver.get(): To open an URL and it will wait till the whole page gets loaded

Selenium wont perform any other actions until page is loaded

driver.navigate.to(): To navigate to an URL and It will not wait till the whole page gets loaded

56. Can I navigate back and forth in a browser in Selenium WebDriver?

We use Navigate interface to do navigate back and forth in a browser. It has methods to move back, forward as well as to refresh a page.

driver.navigate().forward(); – to navigate to the next web page with reference to the browser’s history

driver.navigate().back(); – takes back to the previous webpage with reference to the browser’s history

driver.navigate().refresh(); – to refresh the current web page thereby reloading all the web elements (don’t use this, if you use this, DOM will again loaded, u will get StaleElementException)

driver.navigate().to(“url”); – to launch a new web browser window and navigate to the specified URL (but it wont wait until page is fully loaded)

57. What are the different types of navigation commands?

Refer above question (Can I navigate back and forth in a browser)

58. How to fetch the current page URL in Selenium?

To fetch the current page URL, we use getCurrentURL()

driver.getCurrentUrl();

59. How can we maximize browser window in Selenium?

To maximize browser window in selenium we use maximize() method. This method maximizes the current window if it is not already maximized

driver.manage().window().maximize();

60. How to delete cookies in Selenium?

To delete cookies we use deleteAllCookies() method

driver.manage().deleteAllCookies();

61. What are the ways to refresh a browser using Selenium WebDriver?

There are multiple ways to refresh a page in selenium

* Using driver.navigate().refresh() command as mentioned in the question 45
* Using driver.get(“URL”) on the current URL or using driver.getCurrentUrl()
* Using driver.navigate().to(“URL”) on the current URL or driver.navigate().to(driver.getCurrentUrl());
* Using sendKeys(Keys.F5) on any textbox on the webpage

62. What is the difference between driver.getWindowHandle() and driver.getWindowHandles() in Selenium WebDriver?

The purpose of getWindowHandle() is to handle browser window popup.

It will return unique id of that page. There are 2 methods

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

driver.getWindowHandles() – It returns a set of handles of the all the pages available. (returns parent and child id’s)

* Return type getWindowHandle() is Set which stores in <String>
* How will you switch to to parent to child : driver.switchTo().window(childwindow)

**for**(String childwindow : child)

{

**if**(!childwindow.equals(parent))

{

driver.switchTo().window(childwindow);

System.***out***.println("child id is " +childwindow);

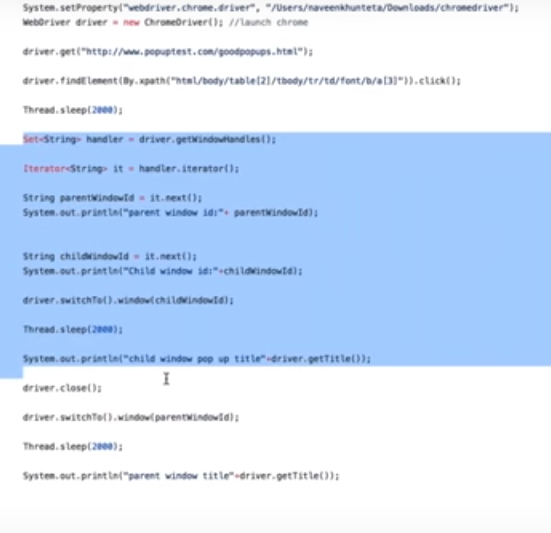
* How will you switch to child to parent :

driver.switchTo().window(parentwindow);

1st Method



2nd Method



63. What is the difference between driver.close() and driver.quit() methods?

Purpose of these two methods (driver.close and driver.quit) is almost same. Both allow us to close a browser but still, there is a difference.

driver.close(): To close current WebDriver instance

driver.quit(): To close all the opened WebDriver instances

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

The difference between driver.findElement() - (used for single WebElement)

and driver.findElements() - (used for multiple elements in single WebElement of same type )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()-
* 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 0 elements.

Eg: How many links in this page…..

List<WebElement> list = driver.findElements(By.tagName(“a”));

Sop(list.size()); ==40

Returntype of findElement is ‘WebElement’

Returntype of findElements is ‘List’, here we wont perform any actions like .click(); we have to do it individually

65. How to find whether an element is displayed or availability on the web page?

WebDriver facilitates the user with the following methods to check the visibility of the web elements. These web elements can be buttons, drop boxes, checkboxes, radio buttons, labels etc.

1. isDisplayed()

particular element is available or not in this page (or) visible or not

1. boolean elePresent = driver.findElement(By.xpath("xpath")).isDisplayed();

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 1 | 1. boolean elePresent = driver.findElement(By.xpath("xpath")).isDisplayed(); |  |  |

1. isSelected()

It is for List object

1. boolean eleSelected= driver.findElement(By.xpath("xpath")).isSelected();

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 1 | 1. boolean eleSelected= driver.findElement(By.xpath("xpath")).isSelected(); |  |  |

1. isEnabled()

element is available and visible on the page, and it is selected

1. boolean eleEnabled= driver.findElement(By.xpath("xpath")).isEnabled();

|  |  |  |  |
| --- | --- | --- | --- |
| 1. 1 | 1. boolean eleEnabled= driver.findElement(By.xpath("xpath")).isEnabled(); |  |  |

Difference between isDisplayed(), isEnabled() and isSelected() Methods in Selenium WebDriver:

1. isDisplayed() is the method used to verify the presence of a web element within the web page. The method returns a “true” value if the specified web element is present on the web page and a “false” value if the web element is not present on the webpage.

2. isDisplayed() is capable to check for the presence of all kinds of web elements available.

3. isEnabled() is the method used to verify if the web element is enabled or disabled within the web page.

4. isEnabled() is primarily used with buttons.

5. isSelected() is the method used to verify if the web element is selected or not.

6. isSelected() method is predominantly used with radio buttons, dropdowns and checkboxes.

66. How to select a value in a dropdown?

By using ‘Select’ class

WebElement mySelectElement = driver.findElement(By.name("dropdown"));

Select dropdown = new Select(mySelectElement);

dropdown.selectByVisibleText(Text);

dropdown.selectByIndex(Index);

dropdown.selectByValue(Value);

Bootstrop Dropdown:

Eg:

**import** java.util.List;

**import** java.util.concurrent.TimeUnit;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.chrome.ChromeDriver;

**public** **class** BootstrapDropDown {

**public** **static** **void** main(String[] args)

{

WebDriver driver = **null**;

String url = "https://www.jquery-az.com/boots/demo.php?ex=63.0\_2";

System.*setProperty*("webdriver.chrome.driver", "F:\\Javaworkspace\\Practicing\_Programs\\driverFiles\\chromedriver.exe");

driver = **new** ChromeDriver();

//driver = new HtmlUnitDriver();

driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(60, TimeUnit.***SECONDS***);

driver.manage().timeouts().pageLoadTimeout(60, TimeUnit.***SECONDS***);

driver.manage().deleteAllCookies();

driver.get(url);

driver.findElement(By.*xpath*("//button[@title='HTML, CSS']")).click();

List<WebElement> ListElements= driver.findElements(By.*xpath*("//ul[@class='multiselect-container dropdown-menu']/li"));

**int** size1= ListElements.size();

**for**(**int** i=1;i<size1;i++)

{

System.***out***.println(ListElements.get(i).getText());

//ul[@class='multiselect-container dropdown-menu']/li/a/descendant::label//input[@value='jQuery']

}

**for**(**int** i=1;i<size1;i++)

{

**if**(ListElements.get(i).getText().contains("MySQL"))

{

//driver.findElement(By.xpath("//ul[@class='multiselect-container dropdown-menu']/li/a/descendant::label//input[@value='jQuery']")).click();

ListElements.get(i).click();

**break**;

}

}

**for**(**int** i=1;i<size1;i++)

{

**try**

{

**if**(ListElements.get(i).isEnabled()!=**false**)

{

ListElements.get(i).click();

}

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

} // PSVM

}//CLASS

67. How to capture Screenshot in Selenium WebDriver?

By using TakesScreenshot Interface

In Selenium 3, we may face few issues while capturing Screenshots. To overcome we use aShot utility. Click on below links to see posts related to the normal way of capturing a screenshot and capturing a screenshot using aShot utility.

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

FileUtils.*copyFile*(src, **new** File ("F:\\Javaworkspace\\ScreenshotsForFailureTCs\\Screenshots\\" + testMethod+ "\_" + ".png"));

Or

File scrFile = ((TakesScreenshot) *driver*).getScreenshotAs(OutputType.***FILE***);

String currentDir = System.*getProperty*("user.dir");

FileUtils.*copyFile*(scrFile, **new** File(currentDir + "/screenshots/" +System.*currentTimeMillis*() + ".png"));

68. How to mouse hover on a web element using WebDriver?

By using Actions class

WebElement ele = driver.findElement(By.xpath("xpath"));

//Create object 'action' of an Actions class

Actions action = new Actions(driver);

//Mouseover on an element

action.moveToElement(ele).build().perform();

69. How can we handle web based pop-up or browser based pop-up?

To handle alerts popups we need to do switch to the alert window and call Selenium WebDriver Alert API methods.

Eg:

**public** **class** WindowHandlerAPI {

**public** **static** **void** main(String[] args)

{

// **TODO** Auto-generated method stub

WebDriver driver = **null**;

String url = "http://www.popuptest.com/goodpopups.html";

System.*setProperty*("webdriver.chrome.driver", "F:\\Javaworkspace\\Practicing\_Programs\\driverFiles\\chromedriver.exe");

driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.manage().deleteAllCookies();

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

driver.manage().timeouts().pageLoadTimeout(20, TimeUnit.***SECONDS***);

driver.get(url);

// <a class="black" href="http://www.popuptest.com/popup3.html" onclick="NewWindow(this.href,'rank','450','450','yes','center');return false" onfocus="this.blur()">Good PopUp #3</a>

String parentwindowid = driver.getWindowHandle();

System.***out***.println("parent id is :"+parentwindowid);

driver.findElement(By.*linkText*("Good PopUp #3")).click();

Set<String> childwindowid = driver.getWindowHandles();

System.***out***.println("child id is :" +childwindowid);

**for**(String handle : childwindowid)

{

**if**(handle!=parentwindowid)

{

System.***out***.println("\*\*\*\*\*\*\*\*\*\*\*\*");

driver.switchTo().window(handle);

System.***out***.println("title of child window is : "+driver.getTitle());

//driver.close();

}

}

/\*

Iterator<String> handle = childwindowid.iterator();

String child = handle.next();

System.out.println(child);

driver.switchTo().window(child);

driver.close();

\*/

}

}

70. How can we handle windows based pop up or desktop based popup?

Selenium doesn’t support windows based applications. It is an automation testing tool which supports only web application testing. We can send file path in sendkeys if it is browse file element and its type attribute should be ‘FILE’

Eg:

driver.findElement(By.*id*("photofile")).sendKeys("C:\\Users\\ANUSHA\\Downloads\\flower.png");

We could handle windows based popups in Selenium using some third party tools such as AutoIT, SIKULI, Robot class etc.

71. How to handle hidden elements in Selenium WebDriver?

It is one of the most important selenium interview questions.

We can handle hidden elements by using javaScript executor

(JavascriptExecutor(driver)).executeScript("document.getElementsByClassName(ElementLocator).click();");

Because JS will hit directly to the HTML DOM with the help of JavascriptExecutor

…. [Js concept refer all syntaxes](JavaScriptExecutor.docx)

72. How can you find Broken Links in a page using Selenium WebDriver?

[…](http://www.softwaretestingmaterial.com/broken-links-using-selenium/) refer broken links concept

73. How to find more than one web element in the list?

// To store the list

List <WebElement> eleList = driver.findElements(By.xpath("xpath"));

// To fetch the size of the list

int listSize = eleList.size();

//for loop

for (int i=0; i<listSize; i++)

{

// Clicking on each link

links.get(i).click();

// Navigating back to the previous page that stores the links

driver.navigate().back();

}

74. How to read a JavaScript variable in Selenium WebDriver?

By using JavascriptExecutor

// To initialize the JS object.

JavascriptExecutor JS = (JavascriptExecutor) webdriver;

// To get the site title.

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

System.out.println("Title of the webpage : " + title);

75. How do you read test data from excels?

Test data can efficiently be read from excel using JXL or Apache POI API. POI API has many advantages than JXL. So we use Apache POI API. (And we have using XLS\_Reader utility for this )

…refer datadriven concept

76. Is it possible to automate the captcha using Selenium?

No, It’s not possible to automate captcha and bar code reader.

77. List some scenarios which we cannot automate using Selenium WebDriver?

1. Bitmap comparison Is not possible using Selenium WebDriver (Image comparisions, pdf comparision)

2. Automating Captcha is not possible using Selenium WebDriver

3. We can not read bar code using Selenium WebDriver

4. windows OS based pop ups

5. third party calendars/element

6. Image

7. Word/PDF

We can do it using third party tools

(Selenium cant automate WebServices also)

78. What is Object Repository in Selenium WebDriver?

Object Repository is used to store element locator values in a centralized location instead of hard coding them within the scripts. We do create a property file (.properties) to store all the element locators and these property files act as an object repository in Selenium WebDriver.

(In selenium, there is no default Object Repository available and it not providing any feature of API or utility to create Object Repository. In QTP, we have Global Object Repository, Local Object Repository interfaces available)

U can use Object Repository in 2 ways

* 1. U can define in config.properties file
  2. U can use @FindBy in POM ie in pagefactory (pagewise)

…. Refer how to use properties file

79. How can you use the Recovery Scenario in Selenium WebDriver (Runtime exceptions )?

By using “Try Catch Block” within Selenium WebDriver Java tests.

try {

driver.get("www.xyz.com");

}catch(Exception e){

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

}

80. How to Upload a file in Selenium WebDriver?

There are two cases which are majorly used to upload a file in Selenium WebDriver such as using SendKeys Method and using AutoIT Script.

Browser Button – type =“file”

SendKeys (c:\\test\\naveen.jpg);

81. How to Download a file in Selenium WebDriver?

By using AutoIT script, we could download a file in Selenium WebDriver.

82. How to run Selenium WebDriver Test from the command line?

Class A{

}

cd c

c: javac A.java //compile – by compiler

c: java A.java // run – by interpreter

how to run testing:

java org.testng.TestNG C:\Users \Desktop\ \workspace\testing\testng.xml

83. How to switch between frames in Selenium?

By using the following code, we could switch between frames.

driver.switchTo().frame(“frameid or index”);

84. How to connect a Database in selenium?

As we all know Selenium WebDriver is a tool to automate User Interface. We could only interact with Browser using Selenium WebDriver.

We use JDBC Driver to connect the Database in Selenium (While using Java Programming Language).

85. How To Resize Browser Window Using Selenium WebDriver?

To resize the browser window to particular dimensions, we use ‘Dimension’ class to resize the browser window.

//Create object of Dimensions class

        Dimension d = new Dimension(480,620); (x,y coordinates)

        //Resize the current window to the given dimension

        driver.manage().window().setSize(d);

86. How To Scroll Web Page Down Or UP Using Selenium WebDriver?

JavaScript scrollBy() method scrolls the document by the specified number of pixels.

Eg:

**public** **static** **void** scrollpageDown(WebDriver driver)

{

JavascriptExecutor js = ((JavascriptExecutor)driver);

js.executeScript("window.scrollTo(0,document.body.scrollHeight)");

}

**public** **static** **void** scrollpageIntoView(WebElement element,WebDriver driver)

{

JavascriptExecutor js = ((JavascriptExecutor)driver);

js.executeScript("arguments[0].scrollIntoView(true);",element);

}

87. How To Perform Right Click Action (Context Click) In Selenium WebDriver?

We use Actions class in Selenium WebDriver to do Right-Click (Context Click) action.

action.contextClick(driver.findElement(By.xpsjht()).build().perform();

88. How To Perform Double Click Action In Selenium WebDriver?

We use Actions class to do Double click action in selenium.

action.doubleClick(driver.findElement(By.xpsjht()).build().perform();

89. How To Perform Drag And Drop Action in Selenium WebDriver?

We use Actions class to do Drag And Drop Action

1st Method:

WebElement draggable = driver.findElement(By.*id*("draggable"));

// droppable

WebElement droppable = driver.findElement(By.*id*("droppable"));

Actions act = **new** Actions(driver);

act.dragAndDrop(draggable, droppable).build().perform();

2nd Method

Actions action = new Actions(driver);

action.clickAndHold(driver.findElement(By.xpath("//\*[@id='draggable']")))

.moveToElement(driver.findElement(By.xpath("//\*[@id='droppable']")))

.release()

.build()

.perform();

90. How To Highlight Element Using Selenium WebDriver?

There is no Highlight method available WebDriver

By using JavascriptExecutor interface, we could highlight the specified element

Eg:

**public** **static** **void** flash(WebElement element,WebDriver driver)

{

JavascriptExecutor js = ((JavascriptExecutor)driver);

String bgcolor = element.getCssValue("backgroundColor");

**for**(**int** i=0;i<10;i++)

{

*changecolor*("rgb(0,200,0)", element, driver);

*changecolor*(bgcolor, element, driver);

}

}

**public** **static** **void** changecolor(String color,WebElement element, WebDriver driver)

{

JavascriptExecutor js = ((JavascriptExecutor)driver);

js.executeScript("arguments[0].style.backgroundColor = '"+color+"'",element);

**try**

{

Thread.*sleep*(20);

}

**catch**(InterruptedException e)

{

}

}

91. How to go back from frame to driver?

driver.swithchTo().defaultContent();

92. Return type of split

String array[]

Eg:

String xyz="Naveen teach selenium,core java and live framework excellent ";

String rev[]=xyz.split(" ");

for(int i=rev.length-1;i>=0;i--) {

System.out.print(rev[i]+" ");

}

# 93. length vs length() in Java

**array.length :**length is a final variable applicable for[arrays](https://www.geeksforgeeks.org/arrays-in-java/). With the help of length variable, we can obtain the size of the array.  
**string.length() :** length() method is a final variable which is applicable for string objects. length() method returns the number of characters presents in the string.

Size() :

length is constant which is used to find out the array storing capacity not the number of elements in the array

Example:

int a[] = new int[5]

a.length always returns 5 which is called the capacity of an array, so length always returns the CAPACITY. but

"number of elements in the array is called size"

Example:

int a[] = new int[5]

a[0] = 10

Will result in a.size = 1 and a.length = 5.

size() works with collection, length works with arrays in java.

94) Null :

String s1 = ""; means that the empty String is assigned to s1. In this case, s1.length() is the same as "".length(), which will yield 0 as expected.

String s2 = **null**; means that (**null**) or "no value at all" is assigned to s2. So this one, s2.length() is the same as **null**.length(), which will yield a NullPointerException as you can't call methods on **null** variables (pointers, sort of) in Java.

95) Challenges that faced till now

1. As it is a open source, no vendor support is available but helping blogs and communities are there to help

