NEWGEN CORPORATE TRAINING CENTER

SELENIUM INTERVIEW QUIESTIONS

1. **Why should Selenium be selected as a testing tool for web applications or systems?**
    - It is free and open-source software with a large user base and supports providing community.
    - It has cross-browser compatibility and supports multiple browsers like Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera, Safari, etc.
    - It supports multiple operating systems such as Windows, Linux, macOS, etc.
    - It facilitates the usage of multiple programming languages including Scala, Ruby, Python, PHP, Perl, Java, Groovy, C#, etc.
    - It provides support for distributed testing as well.

2. **What are the disadvantages of using Selenium as a testing tool?**
    - Tests web applications only: Selenium supports the testing of only web-based applications.
    - No built-in reporting and test management facility: Selenium can generate reports only using third-party tools like TestNG or JUnit.
    - Unavailability of reliable tech support: Since Selenium is an open-source tool, no dedicated support for user issues is available.
    - May require the knowledge of programming languages: Some prior programming knowledge is required to use Selenium.

3. **What is meant by a locator and name a few different types of locators present in Selenium.**
    - ID
    - ClassName
    - Name
    - TagName
    - LinkText

- PartialLinkText
- Xpath
- CSS Selector
- DOM

QUE : **Which is the fastest Locator:**

- ANS : ID

4 . **State the major difference between “assert” and “verify” commands in Selenium.**

- Both “assert” and “verify” commands check whether the given condition is true or false and the only difference between them is that:
- **Assert(Hard Assert):** assert condition stops the execution of the testing if the given condition is false else would continue with the further tests.

Hard Assert Methods :

- assertEquals(ExpectedTitle, ActualTitle);
- assertNotEquals(ExpectedTitle, "browserstack");
- assertTrue(verifyTitleIsPresent);
- assertFalse(verifyTitleIsChanged);
- assertNotNull(verifyTitleIsPresent);
- assertNull(verifyAssertNull);

- **Verify:**
  - verify the condition doesn’t stop the flow of execution irrespective of the condition being true or false.

```
SoftAssert softAssert = new SoftAssert();

softAssert.fail("Failing first assertion");
```

5. **What is meant by X path in Selenium. Explain X path Absolute and X path Relative.**

- X path, used for locating elements in Selenium.
- In X path, data is stored in a key-value pair format similar to an HTML tag.
- **Two Types of X path.:**
  - Absolute X- path.
  - Relative X- path.

- It uses a single slash, i.e. ‘ / ’ for creating an absolute path.
- double slash, i.e. ‘ // ’ for creating a relative path for an element to be located on a webpage.

6. **Explain the difference between driver.close() and driver.quit() command in Selenium?**

- **driver.close()** command closes the currently active window on which the user is working or the window being currently accessed by the web driver.
- **driver.quit()** command, unlike the driver.close() command closes all the windows opened by the program and hence should be used with care.

7. **Explain the various navigation commands supported by Selenium?**

- **navigate().back():** This command is used for taking the user to the last webpage of the browser history.
- **navigate().forward():** This command is used for taking the user to the next web page of the browser history.
- **navigate().refresh():** This command is used for reloading the web components of a webpage by refreshing it.
- **navigate().to():** This command is used for navigating to a particular URL in a new web browser. It takes the URL to be migrated to, as a parameter.

8. **Explain the difference between findElement() and findElements() in Selenium.**

- **findElement():**

- command is used for finding a particular element on a web page, it is used to return an object of the first found element by the locator.
- It returns Web Element.
- Eg: WebElement element = driver.findElement(By.id(example));
- If Web Element not found then it throws NoSuchElementFound Exception thorws.

- **findElements():**
  - Command is used for finding all the elements in a web page specified by the locator value.
  - The return type of this command is the list of all the matching web elements.
  - It Reruns the List<WebElement>
  - If Object is not found then it returns Its Size to Zero.

Eg: List <WebElement> elementList = driver.findElements(By.id(example));

9. **Can you capture a screenshot using Selenium? If yes, write a simple code to illustrate the same.**

```
// Capture the screenshot
    File scrFile =
((TakeScreenshot)drv).getScreenshotAs(OutputType.FILE);
    // Code for pasting screenshot to a user-specified location
    FileUtils.copyFile(scrFile, new File("C:\\Screenshot\\Scr.jpg"))
```

10. xplain the difference between single and double slash in X-path?

Single slash '/ '

- Single slash (/) start selection from the document node

- It allows you to create 'absolute' path expressions

Double Slash '// '

- Double slash (//) start selection matching anywhere in the document
- It enables to create 'relative' path expressions

11. List out the technical challenges with Selenium?

Technical challenges with Selenium are

- 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

12. What is the difference between verify and assert commands?

- **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 terminated 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.

13. How you can use "submit" a form using Selenium ?

You can use "submit" method on element to submit form-

element.submit () ;

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

14. Mention what is the difference between Implicit wait and Explicit wait?

- **Implicit Wait:**

- The **Implicit Wait in Selenium** is used to tell the web driver to wait for a certain amount of time before it throws a "No Such Element Exception".
- The default setting is 0.
- Once we set the time, the web driver will wait for the element for that time before throwing an exception.
- Syntax :

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

- **Explicit Wait :**
  - WebDriver is directed to wait until a certain condition occurs before proceeding with executing the code.
  - Setting Explicit Wait is important in cases where there are certain elements that naturally take more time to load.
  - If one sets an implicit wait command, then the browser will wait for the same time frame before loading every web element.
  - This causes an unnecessary delay in executing the test script.

```
WebDriverWait wait = new WebDriverWait(driver,30);

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//di
v[contains(text(),'COMPOSE')]")));
```

15. xplain how you can find broken images in a page using Selenium Web driver ?

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

16. Explain how you can switch between frames?

To switch between frames webdrivers [driver.switchTo().frame()] method takes one of the three possible arguments

- A number: It selects the number by its (zero-based) index
- A name or ID: Select a frame by its name or ID
- Previously found WebElement: Using its previously located WebElement select a frame

17. Mention 5 different exceptions you had in Selenium web driver?

The 5 different exceptions you had in Selenium web drivers are

- WebDriverException
- NoAlertPresentException
- NoSuchWindowException
- NoSuchElementException
- TimeoutException

18. Explain using Webdriver how you can perform double click ?

You can perform double click by using

- Syntax- Actions act = new Actions (driver);
- act.doubleClick(webelement);

19. Which web driver implementation is fastest?

- 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

20. Explain how you can handle frames using Selenium 2.0 ?

- To bring control on HTML frame you can use “SwitchTo” frame method-

  driver.switchTo().frame(“frameName”);

- To specify a frame you can use index number

  driver.switchTo().frame(“parentFrame.4.frameName”);

This would bring control on frame named- “frameName” of the 4th sub frame names “parentFrame”

21 What is the difference between getWindowhandles() and getwindowhandle() ?

- getwindowhandles():
  - It is used to get the address of all the open browser
  - its return type is Set<String>

- getwindowhandle():
  - It is used to get the address of the current browser
  - return type is string

22 Explain how you can switch back from a frame?

- To switch back from a frame use method defaultContent()
- Syntax-driver.switchTo().defaultContent();

23. Mention what is Listeners in Selenium WebDriver?

- 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.

24. Mention what are the types of Listeners in TestNG?

The types of Listeners in TestNG are,

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

- ITestListener

25. code snippet to launch Firefox browser in WebDriver.

```
System.setProperty("webdriver.gecko.driver", pathToGeckoDriver +
"\\geckodriver.exe");

//Instantiating driver object and launching browser

driver = newFirefoxDriver();

//Using get() method to open a webpage

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

//Closing the browser

driver.quit();
```

26. Write a code snippet to perform right-click an element in WebDriver.

```
Actions action = newActions(driver);

WebElement element = driver.findElement(By.id("elementId"));

action.contextClick(element).perform();
```

27**. Write a code snippet to perform mouse hover in WebDriver.**

```
Actions action = newActions(driver);

WebElement element = driver.findElement(By.id("elementId"));

action.moveToElement(element).perform();
```

28. How do you perform drag and drop operation in WebDriver?

```
/WebElement on which drag and drop operation needs to be performed

WebElementfromWebElement = driver.findElement(By Locator of
fromWebElement);

//WebElement to which the above object is dropped

WebElementtoWebElement = driver.findElement(By Locator of
toWebElement);

//Creating object of Actions class to build composite actions

Actions builder = newActions(driver);

//Building a drag and drop action

Action dragAndDrop = builder.clickAndHold(fromWebElement)

        .moveToElement(toWebElement)

        .release(toWebElement)

    .build();

//Performing the drag and drop action

dragAndDrop.perform();
```

29. **How to invoke an application in WebDriver?**

- driver.get("url"); or
- driver.navigate().to("url");

30. How can we get a text of a web element?

- String Text = driver.findElement(By.id("Text")).getText();

31. How to select value in a dropdown?

- We use the WebDriver's Select class to select the value in the dropdown.

```
WebElement x = driver.findElement(By.id(“SelectID”));

Select  y = new Select(x);

y.selectByValue("greenvalue");

or

y. .selectByVisibleText("Lime");

or

y. selectByIndex(2);
```

32. How to deal with frame in WebDriver?

- An inline frame abbreviates as an iframe.
- It is used to insert another document within the current document.
- These document can be HTML document or simply web page and nested web page.
- driver.switchTo().frame("ID of the frame");

33. **How you can come back to main frame?**

- driver.switchTo().defaultContent();

34. What is POM (Page Object Model)? What are its advantages?

- Page Object Model is a design pattern for creating an Object directory for web UI elements.
- Each web page is required to have its page class.
- The page class is responsible for finding the WebElements in web pages and then perform operations on WebElements.

The benefits of using POM are as follows.

- It facilitates with separate operations and flows in the UI from Verification - improves code readability
- Multiple tests can use the same Object Repository because the Object Repository is independent of Test Cases.
- Reusability of code

35. How can you find if an element is displayed on the screen?

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

- isDisplayed()
- isSelected()
- isEnabled()

36. 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.
- Hard Assert:
    - Hard Assert throws an *AssertException* immediately when an assert statement fails and test suite continues with next @*Test*

37. What happen if you mix both implicit wait and explicit wait in a Selenium Script?

- Explicit wait will overwrite the implicit wait where ever explicit wait is applied.
- So, Explicit Wait gets first preference then Implicit Wait.

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

```
WebDriver driver = new FirefoxDriver();

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

driver.findElement(By.xpath("xpath")).sendKeys("Software Testing Material Website");
```

39. How to submit a form using Click?

```
driver.findElement(By.id("form_1")).submit();
```

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

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

41. **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
- driver.navigate.to(): To navigate to an URL and It will not wait till the whole page gets loaded

42. How can we maximize browser window in Selenium?

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

43. How to delete cookies in Selenium?

- driver.manage().deleteAllCookies()

44. 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);

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

```
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).perform();
```

46. How can we handle Web-based Pop-ups or Alerts in Selenium?

dismiss(): To click on Cancel button.
accept(): To Click on OK button.
getText(): To get the text which is present on the Alert.
sendKeys(): To enter the text into the alert box.

47. **How to Handle Pop ups?**

```
driver.switchTo().alert();

driver.switchTo().alert().accept()

driver.switchTo().alert().dismiss()
```

48. **How to Print all Links Names present on the URL ?**

```
// Storing the list

List <WebElement> elementList = driver.findElements(By.xpath("//a"));

// Fetching the size of the list

int listSize = elementList.size();
```

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

{

System.out.println(elementList.get(i).Text());

}
```

49. **How to assert the title of the web page?**

```
//verify the title of the web page

assertTrue("The title of the window is
incorrect.",driver.getTitle().equals("Title of the page"));
```

50. What is Page Factory?

- Page Factory gives an optimized way to implement Page Object Model. When we say it is optimized,
- it refers to the fact that the memory utilization is very good and also the implementation is done in an object oriented manner.
- Page Factory is used to initialize the elements of the Page Object or instantiate the Page Objects itself.

51. Write a code to wait for a particular element to be visible on a page. Write a code to wait for an alert to appear.

```
WebDriverWait wait=new WebDriverWait(driver, 20);

Element =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(
"<xpath")))
```

52. **What is the use of JavaScriptExecutor?**

- JavaScriptExecutor is an interface which provides a mechanism to execute Javascript through the Selenium WebDriver.

- It provides “executescript” and “executeAsyncScript” methods, to run JavaScript in the context of the currently selected frame or window. An example of that is:
- JavascriptExecutor js = (JavascriptExecutor) driver;

  js.executeScript(Script,Arguments);

53. How to scroll down a page using JavaScript in Selenium?

- We can scroll down a page by using window.scrollBy() function. Example:

  ((JavascriptExecutor) driver).executeScript("window.scrollBy(0,500)");

54. How to scroll down to a particular element?

- To scroll down to a particular element on a web page, we can use the function **scrollIntoView()**.

  ((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView();", element);

55. How to send ALT/SHIFT/CONTROL key in Selenium WebDriver?

- When we generally use ALT/SHIFT/CONTROL keys, we hold onto those keys and click other buttons to achieve the special functionality.
- So it is not enough just to specify **keys.ALT** or **keys.SHIFT** or **keys.CONTROL** functions.

56. How to set the size of browser window using Selenium?

- System.out.println(driver.manage().window().getSize());

  Dimension d = new Dimension(420,600);

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

57. How to switch to a new window (new tab) which opens up after you click on a link?

```
String handle= driver.getWindowHandle();

for (String handle : driver.getWindowHandles())

{

driver.switchTo().window(handle);

}
```

58. What are the technical challenges with Selenium?

Answer: There are several technical challenges with Selenium which includes:

1. It only supports web-based applications.
2. Bitmap comparison is not supported.
3. Third-party tools are sought for reporting purposes.
4. Vendor support is minimal as compared to other commercial tools such as HP UFT.
5. It is challenging to maintain objects in Selenium.

59. What does a single slash "/" mean in XPath?

- A single (forward) slash "/" represents the absolute path.
- In this case, the XPath engine navigates the DOM right from the first node.

60. What does a double slash "//" mean in XPath?

- A double (forward) Slash "//" represents the relative path.
- In this case, the XPath engine searches for the matching element anywhere in the DOM.

61. How do you locate an element by partially comparing its attributes in XPath?

- XPath supports the contains() method. It allows partially matching of attribute's value.
- It helps when the attributes use dynamic values while having some fixed part.

See the below example-

xPath usage => //*[contains(@category, 'tablet')]

The above expression would match all values of the category attribute having the word 'tablet' in them.

62.How do you locate elements based on the text in XPath?

- We can call the text() method. The below expression will get elements that have text nodes that equal 'Python.'
- xPath usage = //*[text()='Python']

63. How do you access the parent of a node with XPath?

We can use the double dot ("..") to point to the parent of any node using the XPath.

For example – The locator **//span[@id="current"]/..** will return the parent of the span element matching id value as 'current'.

64. What is the primary difference between the XPath and CSS selectors?

- With the XPath, we can traverse both forward and backward, whereas CSS selector only moves forward.

65. What is Web Driver Implicit wait?

- **Implicit Wait:**
  - It is a wait timeout which applies to a Webdriver instance.
  - It implies that all actions of this instance will timeout only after waiting for a duration specified by the implicit wait.

```
WebDriver driver = new ChromeDriver();

driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);
```

66. What is Web Driver Explicit wait?

- **Explicit Wait:**
  - It is an exclusive timeout method that works by adding code to delay the execution until a specific condition arises.
  - It is more customizable in terms that we can set it up to wait for any suitable situation.
  - Usually, we use a few of the pre-built Expected Conditions to wait for elements to become clickable, visible, invisible, etc.

```
WebDriver driver = new ChromeDriver();

driver.get("http://target_page_url");

WebElement dynamicElement = (new WebDriverWait(driver, 15))

 .until(ExpectedConditions.presenceOfElementLocated(By.id("dynamicElement"))
```

67. How to enter text in the HTML text box without invoking the sendKeys()?

```
// Set up the JS object

JavascriptExecutor jscript = (JavascriptExecutor)webdriver;

// Issue command to enter the text
```

jscript.executeScript("document.getElementById('textbox').value = 'Some Text';");

68. **How can you check the state of a checkbox/radio button?**

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

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

69. **How to execute the testng test suite from the command line?**

java -cp “C:\Selenium Webdriver Questions\testng \lib*;C:\Selenium Testing Questions\testng\bin” org.testng.TestNG testng.xml

70. Current Selenium version

ANS : Selenium 4

71. which version you are using

ANS : Selenium 3

72. **********Webdriver **********

Interface :

it is 100 % abstraction

All methods are default public.

We can achive multiple Inheritance.

We can not create object of interface.

**********Invoking Webdriver **********

QUE : HOw will you invoke browser

Ans :

Step1 : System.setProperty("webdriver.chrome.driver", "Path of chromedriver.exe");

Step2 : WebDriver driver=new ChromeDriver();

Step3 : driver.navigate().to("http://www.javatpoint.com/");

QUE: Why not WebDriver driver = new WebDriver()

ANS : WebDriver is a interface , so we can not create object of interface.

QUE: Why not ChromeDirver driver = new ChromeDirver()

Ans: It will work , but it will applicable to only ChromeDirver, we will not use other drivers

Because of this one.

**********Cross Browsers Testing **********

QUE : how will you perform Cross browser Testing.

ans : We will Set in TestNG parameter

<suite name="Suite">

<parameter name="Browser" value="IE"/>

<test thread-count="5" name="Test">

```
            <classes>
                <class name="newPackage.MyClass3" />
            </classes>
        </test> <!-- Test -->
</suite> <!-- Suite -->
```

```
@Test()
Login(String @Parameters("Browser") ){

Switch(@Parameters("Browser")){

        case "Chrome":
            System.setProperty("webdriver.chrome.driver", "Path of
chromedriver.exe");
            WebDriver driver = new ChromeDriver();
            driver.navigate().to("http://www.javatpoint.com/");

            break;

        case "IE":
            System.setProperty("webdriver.IE.driver", "Path of IEDriver.exe");
```

```
            WebDriver driver = new InterNetExplorere();
            driver.navigate().to("http://www.javatpoint.com/");
            break;

        Case "Mozilla":
            System.setProperty("webdriver.chrome.driver", "Path of
chromedriver.exe");
            WebDriver driver = new Mozilla();
            driver.navigate().to("http://www.javatpoint.com/");
            break;
        }
}
```

What are action perform on WebDerivers?

Driver.manage.Window().Maximize();

Driver.quit()

Driver.close()

Driver.getTitle();

Driver.getWindowHandle();

Driver.getWindowHandles();

**********Driver.manage.Window().Maximize();

```
System.setProperty("webdriver.chrome.driver", "E:\\Automation Batch\\NewProject\\WebDrivers\\chromedriver.exe");

WebDriver driver=new ChromeDriver();

driver.navigate().to("http://www.javatpoint.com/");

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

**********Driver.close()

```
System.setProperty("webdriver.chrome.driver", "E:\\Automation Batch\\NewProject\\WebDrivers\\chromedriver.exe");

WebDriver driver=new ChromeDriver();

driver.navigate().to("http://www.javatpoint.com/");

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

driver.close();
```

**********Driver.Quit()

```
System.setProperty("webdriver.chrome.driver", "E:\\Automation Batch\\NewProject\\WebDrivers\\chromedriver.exe");

WebDriver driver=new ChromeDriver();

driver.navigate().to("http://www.javatpoint.com/");

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

driver.quit();
```

**********Difference between Driver.close() and Driver.quit()**********

Driver.close : Current Driver session closed

Driver.quit : all Drivers instances created by that dirver will close all

**********Difference between Driver.get() and Driver.navigate().to()**********

driver.get() is used to navigate particular URL(website) and wait till page load.

driver.navigate() is used to navigate to particular URL and does not wait to page load.

It maintains browser history or cookies to navigate back or forward.

**********Waits in Selenium **********

Types of Waits:

1. Implicit Wait
2. ExpliCit Wait
3. Fluent Wait.

1. Implicit Wait:
    1. It is the global Wait.
    2. It is applicable for entire test Suit.
    3. Selenium waits before throwing an exception

4. The default setting of implicit wait is zero.

It will wait till the till time mentioned. if WebElement not occured it will throw an Exception.

QUE : Which exception it will throw ?

Ans : NoSuchElementException

Syntax :

driver.manage().timeouts().implicitlyWait(TimeOut, TimeUnit.SECONDS);

QUE : Where you used Method Overloasding in your project :

ANS : Implicit wait is the Example.

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

driver.manage().timeouts().implicitlyWait(10, TimeUnit.HOURS);

driver.manage().timeouts().implicitlyWait(10, TimeUnit.MINUTES);

2. ExpliCit Wait:

1. WebDriver is directed to wait until a certain condition occurs before proceeding with executing the code.

2. Setting Explicit Wait is important in cases where there are certain elements that naturally take more time to load.

If one sets an implicit wait command, then the browser will wait for the same time frame before loading every web element.

This causes an unnecessary delay in executing the test script.

3. It is Specific to WebeElement.

Syntax :

```
WebDriverWait wait = new WebDriverWait(driver,30);

wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[contains(text(),'COMPOSE')]")));
```

In order to declare explicit wait, one has to use “ExpectedConditions”. The following Expected Conditions can be used in Explicit Wait.

alertIsPresent()

elementSelectionStateToBe()

elementToBeClickable()

elementToBeSelected()

frameToBeAvaliableAndSwitchToIt()

invisibilityOfTheElementLocated()

invisibilityOfElementWithText()

presenceOfAllElementsLocatedBy()

presenceOfElementLocated()

textToBePresentInElement()

textToBePresentInElementLocated()

textToBePresentInElementValue()

titleIs()

titleContains()

visibilityOf()

visibilityOfAllElements()

visibilityOfAllElementsLocatedBy()

visibilityOfElementLocated()

Fluent Wait:

1. The Fluent Wait command defines the maximum amount of time for Selenium WebDriver to wait for a certain condition to appear.

2. It also defines the frequency with which WebDriver will check if the condition appears before throwing the “ElementNotVisibleException”.

```
Wait wait = new FluentWait<WebDriver>(driver)
.withTimeout(50, TimeUnit.SECONDS)
.pollingevery(3, TimeUnit.SECONDS)
.ignoring(NoSuchElementException.class);
```

QUE : Difference Between Implicit and ExpliCit wait :

ANS :
https://www.google.com/search?q=difference+between+implicit+and+explicit+wait&sxsrf=ALeKk01G7_vN6yt-6r0QYdrSNxBUhgD6Iw:1618552136115&source=lnms&tbm=isch&sa=X&ved

=2ahUKEwiVjv7IiILwAhV2wzgGHRDHBD0Q_AUoAXoECAEQAw&biw=1024&bih=657#imgrc=oWhgsBZF_rIoLM

**********Screen Shot **********

QUE : HOw will you capture/ take Sceenshot

ANS :

Step 1) Convert web driver object to TakeScreenshot

TakesScreenshot scrShot =((TakesScreenshot)webdriver);

Step 2) Call getScreenshotAs method to create image file

File SrcFile=scrShot.getScreenshotAs(OutputType.FILE);

Step 3) Copy file to Desired Location

File DestFile=new File(fileWithPath);

//Copy file at destination

FileUtils.copyFile(SrcFile, DestFile);

**********Handling Multiple Windows **********

QUE : How will you handle multiple windows.

ANS : by using driver.getWindowHandles().

Code :

Step1 : Get and store current window ID

```
String currentHandle= driver.getWindowHandle();
```

Step2 : Get All opened Driver ID in to the Set

```
Set<String> handles=driver.getWindowHandles();
```

Step3 : Itereate Set so switch to Another Tab.

```
for(String actual: handles)
{
```

Step4 : condition check , parent and child name not equals

```
if(!actual.equalsIgnoreCase(currentHandle))
{
    //switching to the opened tab
    driver.switchTo().window(actual);
}
}
```

Step5 : Once your task finished , they moved back to original Window.

```
driver.switchTo().window(currentHandle);
}
```

QUE: What is the return type of driver.getWindowHandles();

ANS : Set<String>

Que:What is the return type of driver.getWindowHandle();

ANS : String

QUE : Print All opeend Child Windows Names:

ANS : Step1 : Get and store current window ID

```
String currentHandle= driver.getWindowHandle();
```

Step2 : Get All opened Driver ID in to the Set

```
Set<String> handles=driver.getWindowHandles();
```

Step3 : Itereate Set so switch to Another Tab.

```
for(String actual: handles)
{
```

Step4 : condition check , parent and child name not equals

```
//switching to the opened tab
driver.switchTo().window(actual);

}
```

Step5 : Once your task finished , they moved back to original Window.

```
driver.switchTo().window(currentHandle);
}
```

********** Handling Multiple Frames **********

https://www.guru99.com/handling-iframes-selenium.html

QUE : How to switch over the elements in iframes using Web Driver commands:

ANS :

By Index

driver.switchTo().frame(0);

driver.switchTo().frame(1);

By Name or Id

driver.switchTo().frame("iframe1");

driver.switchTo().frame("id of the element");

By Web Element

QUE : How to switch back to the Main Frame ?

ANS : driver.switchTo().parentFrame();

driver.switchTo().defaultContent();

********** Action Class:

https://www.browserstack.com/guide/action-class-in-selenium

Action Class in Selenium is a built-in feature provided by the selenium for handling keyboard and mouse events.

It includes various operations such as multiple events clicking by control key, drag and drop events and many more.

Mouse Actions:

doubleClick(): Performs double click on the element

clickAndHold(): Performs long click on the mouse without releasing it

dragAndDrop(): Drags the element from one point and drops to another

moveToElement(): Shifts the mouse pointer to the center of the element

contextClick(): Performs right-click on the mouse

Keyboard Actions:

sendKeys(): Sends a series of keys to the element

keyUp(): Performs key release

keyDown(): Performs keypress without release

Syntax :

1. Click :

```
driver.navigate().to("https://www.browserstack.com/");
//driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
WebElement element = driver.findElement(By.xpath("//a[@id='signupModalButton']"));
Actions action = new Actions(driver);
action.moveToElement(element).click().perform();
```

2. Perform Mouse Hover Action on the Web Element

Actions action = new Actions(driver);

WebElement live= driver.findElement(By.
cssSelector("div.product-cards-wrapper--click a[title='Live']"));

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

3. QUE : HOw will you Scroll Down A Page.

ANS : By Using Java Script Executor we can Scroll doown a page

driver.get("http://demo.guru99.com/test/guru99home/");

//To maximize the window. This code may not work with Selenium 3 jars. If script fails you can remove the line below

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

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("window.scrollBy(0,1000)"); //Scroll vertically down by 1000 pixels

Or

By Using SendKeys "Page Down"

3. QUE : HOw will you Scroll Up A Page.

https://www.guru99.com/scroll-up-down-selenium-webdriver.html

ANS : By Using Java Script Executor we can Scroll doown a page.

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("window.scrollBy(0,0)"); //Scroll vertically down by 0 pixels

or

By Using SendKeys "Page Up"

4. How will you perform Double Click

https://www.guru99.com/double-click-and-right-click-selenium.html

By Using Action Class and its Method as doubleClick

Syntax :

Actions a = new Actions(driver);

WebElement trialaction = driver.findElement(By.xpath("//a[@id='free-trial-link-anchor']"));

a.doubleClick(trialaction).perform();

5. How will you perform Right Click

https://www.guru99.com/double-click-and-right-click-selenium.html

By Using Action Class and its Method as contextClick

Syntax :

Actions a = new Actions(driver);

WebElement trialaction = driver.findElement(By.xpath("//a[@id='free-trial-link-anchor']"));

a.contextClick(trialaction).perform();

6. HOw will you perform Drag and Drop

https://www.guru99.com/drag-drop-selenium.html

By Using Action class we can perform Drag and Drop.

driver.get("http://demo.guru99.com/test/drag_drop.html");

Actions a = new Actions(driver);

WebElement From=driver.findElement(By.xpath("//*[@id='credit2']/a"));

WebElement To=driver.findElement(By.xpath("//*[@id='bank']/li"));

a.dragAndDrop(From, To).build().perform();

********** Pop Up **********

QUE : HOw will you handle Pop ups

http://demo.guru99.com/test/delete_customer.php

By using driver.switchTo().alert()

Methods for Pop Ups :

1) void dismiss() // To click on the 'Cancel' button of the alert.

driver.switchTo().alert().dismiss();

2) void accept() // To click on the 'OK' button of the alert.

driver.switchTo().alert().accept();

3) String getText() // To capture the alert message.

driver.switchTo().alert().getText();

4) void sendKeys(String stringToSend) // To send some data to alert box.

driver.switchTo().alert().sendKeys("Text");

**********Execption in Selenium **********

1. An exception is an error that happens at the time of execution of a program.

2. However, while running a program, programming languages generates an exception that should be handled to avoid your program to crash.

Common Exceptions in Selenium Web driver

ElementNotVisibleException :

This type of Selenium exception occurs when an existing element in DOM has a feature set as hidden.

NoSuchElementException : (VIMP)

1. Mostly This Exception Occurs

2. When we trying to locate Element but Selenium not able to find that Element on DOM, at that time it throw

NoSuchElementException found

NoSuchFrameException :This Exception occurs if the frame target to be switched to does not exist.

NoAlertPresentException :This Exception occurs when you switch to no presented alert.

NoSuchWindowException :This Exception occurs if the window target to be switch does not exist.

StaleElementReferenceException : (VIMP)

This Selenium exception occurs happens when the web element is detached/vanished from the current DOM.

TimeoutException :Thrown when there is not enough time for a command to be completed. For Example, the element searched wasn't found in the specified time.

NoSuchAttributeException :This kind of Exception occurs when the attribute of an element could not be found.

MoveTargetOutOfBoundsException :It takes place if the target provided to the ActionChains move() methodology is not valid. For Example, out of the document.

**********How will you handle the Execption **********

By Using Try Catch block we Handle the Exceptions.

For each and every Step should include in Method

```
public boolean x (){
        try{
                WebElement y = driver.findElement(By.xpath("//sdfsfdsdfds"));
                y.Click
        return true;
        }
        catch(Exception e){
                return false;
        }
}
```

********** Assertion **********