* **What are the advantages of Selenium?**
* It supports C#, PHP, Java, Perl, Phython
* It supports different OS like Windows, [Linux](https://www.guru99.com/unix-linux-tutorial.html) and Mac OS
* It has got powerful methods to locate elements (Xpath, DOM , CSS)
* It has highly developer community supported by Google

**What are the limitations of Selenium testing?**

* Unavailability of reliable tech support: Since Selenium is an open-source tool, it does not have dedicated tech support to resolve the user queries.
* Tests web applications only: Selenium needs to be integrated with third-party tools like Appium and TestNG to test desktop and mobile applications.
* Limited support for image testing.
* No built-in reporting and test management facility: Selenium has to be integrated with tools like TestNG, or JUnit among others to facilitate test reporting and management.
* May require the knowledge of programming languages: Selenium WebDriver expects the user to have some basic knowledge about programming.

**2) How will you find an element using Selenium?**

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

**3) Mention what is the use of X-path?**

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

Refer Complete Guide on [XPath](https://www.guru99.com/xpath-selenium.html)

**4) 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

**5) 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.

**6) Why testers should opt for Selenium and not QTP?**

Selenium is more popular than [QTP](https://www.guru99.com/quick-test-professional-qtp-tutorial.html) as

* Selenium is an open source whereas QTP is a commercial tool
* Selenium is used specially for testing web based applications while QTP 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](https://www.guru99.com/python-tutorials.html) whereas QTP supports only VB script

**7) What are the four parameter you have to pass in Selenium?**

Four parameters that you have to pass in Selenium are

* Host
* Port Number
* Browser
* URL

**8) What is the difference between setSpeed() and sleep() methods?**

Both will delay the speed of execution.

Thread.sleep () :  It will stop the current (java) thread for the specified period of time.  Its done only once

* It takes a single argument in integer format

Ex: thread.sleep(2000)- It will wait for 2 seconds

* It waits only once at the command given at sleep

SetSpeed () :  For specific amount of time it will stop the execution for every selenium command.

* It takes a single argument in integer format

Ex: selenium.setSpeed(“2000”)- It will wait for 2 seconds

* Runs each command  after setSpeed delay by the number of milliseconds mentioned in set Speed

This command is useful for  demonstration purpose or if you are using a slow web application

**9)How do you identify an object using selenium?**

To identify an object using Selenium you can use

isElementPresent(String locator)

isElementPresent takes a locator as the argument and if found returns a Boolean

**10) To enter values onto text boxes what is the command that can be used?**

To enter values onto text boxes we can use command **sendkeys()**

**Explain how you can capture server side log Selenium Server?**

To capture server side log in Selenium Server, you can use command

* java –jar .jar –log selenium.log

**11) Why to use TestNG with Selenium RC ?**

If you want full automation against different server and client platforms, You need a way to invoke the tests from a command line process, reports that tells you what happened and flexibility in how you create your test suites. TestNG gives that flexibility.

**12)Explain what is the main difference between web-driver and RC ?**

The main difference between Selenium RC and Webdriver is that, selenium RC injects javascript function into browsers when the page is loaded. On the other hand, Selenium Webdriver drives the browser using browsers built in support

**13)How can you retrive the message in an alert box ?**

You can use the storeAlert command which will fetch the message of the alert pop up and store it in a variable.

**How will you handle working with multiple windows in Selenium ?**

We can use the command **selectWindow** to switch between windows. This command uses the title of Windows to identify which window to switch to.

**14)  How will you verify the specific position of an web element**

You can use verifyElementPositionLeft & verifyElementPositionTop. It does a pixel comparison of the position of the element from the Left and Top of page respectively

**14)What is regular expressions? How you can use regular expressions in Selenium ?**

A regular expression is a special text string used for describing a search pattern. In Selenium IDE regular expression can be used with the keyword- **regexp:** as a prefix to the value and patterns needs to be included for the expected values.

**15)  Explain how you can insert a break point in Selenium IDE ?**

In [Selenium IDE](https://www.guru99.com/introduction-selenuim-ide.html) to insert a break point

* Select “Toggle break point” by right click on the command in Selenium IDE
* Press “B” on the keyboard and select the command in Selenium IDE
* Multiple break points can be set in Selenium IDE

**16)   Explain how you can use recovery scenario with Selenium?**

Recovery scenarios depends upon the programming language you use.  If you are using Java then you can use exception handling to overcome same.  By using “Try Catch Block” within your Selenium WebDriver Java tests

**17)List out different types of locators?**

18)Different types of locators are

* By.id()
* By.name()
* By.tagName()
* By.className()
* By.linkText()
* By.partialLinkText()
* By.xpath
* By.cssSelector()

**19) In selenium IDE what are the element locators that can be used to locate elements on web page?**

In selenium there are mainly 4 locators that are used

* X-path locators
* Css locators
* Html id
* Html name

**20)In Selenium IDE how you can generate random numbers and dates for test data ?**

In Selenium IDE you can generate random numbers by using Java Script

type

css=input#s

javascript{Math.random()}

And for

type

css=input#s

javascript{new Date()}

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

TestNG is a testing framework based on JUnit and NUnit to simplify a broad range of testing needs, from [Unit Testing](https://www.guru99.com/unit-testing-guide.html) to [Integration Testing](https://www.guru99.com/integration-testing.html). 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

**22) Mention what is the difference between Implicit wait and Explicit wait?**

Implicit Wait: Sets a timeout for all successive Web Element searches. For the specified amount of time it will try looking for element again and again before throwing a NoSuchElementException.  It waits for elements to show up.

Explicit Wait :  It is a one-timer, used for a particular search.

**23) Explain what are the JUnits annotation linked with Selenium?**

The JUnits annotation linked with Selenium are

* @Before public void method() – It will perform the method () before each test, this method can prepare the test
* @Test public void method() – Annotations @Test identifies that this method is a test method environment
* @After public void method()- To execute a method before this annotation is used, test method must start with test@Before

**24) Explain what is Datadriven framework and Keyword driven?**

**Datadriven framework:**  In this framework, the test data is separated and kept outside the Test Scripts, while [Test Case](https://www.guru99.com/test-case.html) logic resides in Test Scripts.  Test data is read from the external files ( Excel Files) and are loaded into the variables inside the Test Script.  Variables are used for both for input values and for verification values.

**Keyworddriven framework:** The keyword driven frameworks requires the development of data tables and keywords, independent of the test automation.  In a keyword driven test, the functionality of the application under test is documented in a table as well as step by step instructions for each test.

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

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.

**26) Mention what is IntelliJ?**

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.

**27) Mention in what ways you can customize TestNG report?**

You can customize TestNG report in two ways,

* Using ITestListener Interface
* Using IReporter Interface

28) Explain the different exceptions in Selenium WebDriver.

Exceptions in Selenium are similar to exceptions in other programming languages. The most common exceptions in Selenium are:

* **TimeoutException:** This exception is thrown when a command performing an operation does not complete in the stipulated time
* **NoSuchElementException:** This exception is thrown when an element with given attributes is not found on the web page
* **ElementNotVisibleException:** This exception is thrown when the element is present in DOM (Document Object Model), but not visible on the web page
* **StaleElementException:** This exception is thrown when the element is either deleted or no longer attached to the DOM

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

[*Page Object Model*](https://www.edureka.co/blog/page-object-model-in-selenium/) is a design pattern for creating an Object Repository for web UI elements. Each web page in the application is required to have it’s own corresponding page class. The page class is thus responsible for finding the WebElements in that page and then perform operations on those WebElements.

The advantages of using POM are:

* Allows us to separate operations and flows in the UI from Verification – improves code readability
* Since the Object Repository is independent of Test Cases, multiple tests can use the same Object Repository
* Reusability of code

30. 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.

We can write a code such that we specify the [*XPath*](https://www.edureka.co/blog/xpath-in-selenium/) of the web element that needs to be visible on the page and then ask the WebDriver to wait for a specified time. Look at the sample piece of code below:

|  |  |
| --- | --- |
| 1  2 | WebDriverWait wait=**new** WebDriverWait(driver, 20);  Element = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath( “<xpath”))); |

31. Which files can be used as data source for different frameworks?

Some of the file types of the dataset can be: excel, xml, text, csv, etc.

32. How can you fetch an attribute from an element? How to retrieve typed text from a textbox?

We can fetch the attribute of an element by using the **getAttribute()** method. Sample code:

|  |  |
| --- | --- |
| 1  2 | WebElement eLogin = driver.findElement(By.name(“Login”);  String LoginClassName = eLogin.getAttribute("classname"); |

33) What are different types of frameworks?

The different types of [*frameworks*](https://www.edureka.co/blog/selenium-framework-data-keyword-hybrid-frameworks) are:

* **Data Driven Framework**:-  
  When the entire test data is generated from some external files like Excel, CSV, XML or some database table, then it is called Data Driven framework.
* **Keyword Driven Framework**:-  
  When only the instructions and operations are written in a different file like an Excel worksheet, it is called Keyword Driven framework.
* **Hybrid Framework**:-  
  A combination of both the Data Driven framework and the Keyword Driven framework is called Hybrid framework.

Here, I am finding the web page’s login button named ‘Login’. Once that element is found, getAttribute() can be used to retrieve any attribute value of that element and it can be stored it in string format. In my example, I have retrieved ‘classname’ attribute and stored it in LoginClassName.

Similarly, to retrieve some text from any textbox, we can use getText() method. In the below piece of code I have retrieved the text typed in the ‘Login’ element.

|  |  |
| --- | --- |
| 1  2 | WebElement eLogin = driver.findElement(By.name(“Login”);  String LoginText = Login.getText (); |

Question 34: How would you take a screenshot with Selenium Webdriver?

Yet another one of the Selenium interview questions - you are more than likely to get quite a few of these.

To take a screenshot, you would have to use the **TakeScreenshot** function. After that, you can save the screenshot by using the getScreenshotAs() command.

Example:

from selenium import webdriver browser = webdriver.Firefox() browser.get('http://www.bitdegree.org/') browser.save\_screenshot('courses.png') browser.quit()

Question 35: How would you upload a file via Selenium Webdriver?

An example of how you could do it is provided below. Keep in mind that you have to include **input** and **file** tags.

element = driver.find\_element\_by\_id(”uploaded\_file")  
element.send\_keys("C:\pictures.png")

36) How to handle an Authentication Pop-up using Selenium WebDriver?

As IT is rapidly changing, likewise, applications must be more secure. So, many companies have their own proxy settings for applications. If you open their server in a browser, it will ask you to enter the credentials. Even Selenium needs to handle the same auth pop-up before accessing the server.

Here are 3 ways how you can do that.

**– Via the URL**

Here, one is passing the username and password via the URL.  
The syntax is: [http://username:password@URL](http://username:password@url/)   
For example,

Username: rohit  
Password: P@ssword  
URL:[www.myurl.com](http://www.myurl.com/)

*String URL = “http://” + rohit + ”:” + P@ssword + “@” +*[www.myurl.com](http://www.myurl.com/)*;*  
*driver.get(URL);*  
*Alert alert = driver.switchTo().alert();*  
*alert.accept();*

– **Using AutoIT**

Sample AutoIT script,

*WinWaitActivate(“Authentication Required”,””)*  
*Send(“rohit{TAB}P@ssword{ENTER}”)*  
  
The AutoIT script would be passed within the Java code

– **With Alerts**  
One can handle auth pop-ups with alerts with,  
*driver.switchTo().alert();*

//WebDriver Java Code for entering Username and Password

*driver.findElement(By.id(“userID”)).sendKeys(“userName”);*  
*driver.findElement(By.id(“password”)).sendKeys(“myPassword”);*  
*driver.switchTo().alert().accept();*  
*driver.switchTo().defaultContent();*

In case passing by URL does not work, one can also obtain credentials by Chrome extensions and driver.

37)The Selenium script runs in Chrome but not in IE. What can be done?

HTML DOM rendering and CSSOM construction differs from one browser to the next. So, there may be multiple reasons for a script not working on IE. One can attempt to troubleshoot the problem by:

1. Using the updated Selenium IE Driver

2. Verifying that the IE driver and working environment are compatible

3. Configuring the IE driver with the setProperty method and by importing dependencies

4. Setting the same value for the ‘Enable Protected Mode’ option for all zones from the Security tab

5. Turning off the internet security settings in IE when running the script

6.Using CSS Selectors to minimise exceptions

7. Setting a registry entry

8. Avoiding declaring the driver instance as static for running scripts on browsers parallelly

9. Using the latest Selenium jars

10. Enabling Javascript on the IE browser

11. Using JavaScriptExecutor instead of native click when clicking elements

38)What is the difference between ChromeOptions and DesiredCapabilities?

ChromeOptions is a class that can be used to manipulate capabilities specific to ChromeDriver. For instance, you can disable Chrome extensions with:

ChromeOptions options = new ChromeOptions()

options.addArgument(“disable-extensions”);

ChromeDriver driver = new ChromeDriver(options);

DesiredCapabilities can also be used to manipulate a ChromeDriver session. To change individual web driver properties, DesiredCapabilities class provides key-value pairs.

But, ChromeOptions supports limited clients whereas DesiredCapabilities supports a vast number of clients. ChromeOptions is supported by Java and other languages. DesiredCapabilities is available in Java, but its use is deprecated. When working with a client library, like Selenium Ruby Client, the ChromeOption class will not be available and DesiredCapabilities will have to be used.

DesiredCapabilities are commonly used with Selenium Grid, for parallel execution of test cases on different browsers. However, one can use both DesiredCapabilities and ChromeOptions using merge:

DesiredCapabilities capabilities = new DesiredCapabilities();

options = new ChromeOptions();

options.merge(capabilities);00000

driver = new ChromeDriver(options);

39. Is it possible to do responsive web design testing using Selenium?

Responsive web design testing is important, since many users use smartphones or tabs, and not just laptops, to access applications. Manually testing the application across all the platforms would entail a cumbersome and time-consuming task, which could potentially delay releases. To automate RWD testing one can use the Galen Framework. This add-on runs in Selenium Grid and with it one can run parallel tests as well as have the tests run in a cloud.

For visual validations, one can do RWD testing with Ocular library. Similarly, WAVE Evaluator can be used for check compliance with accessibility standards automatically.

40. What is the need of database automation testing? Is it possible to perform database testing with Selenium?

Currently, the amount of data being generated and used is increasing by the day. Moreover, with advanced front ends, back ends need to be more robust and detailed. Databases play a major role in storing and sequencing data.

– Defect-free data processing

– ACID properties validation (ACID means Atomicity, Consistency, Isolation and Durability)

– Accurate storage and retrieval of values in and from the database

– Data integrity and proper data mapping

– Bugs that cannot be found in front-end testing are brought to light

Yes, it’s possible to do database testing with Selenium. First you need to make a connection between the server and the database. To do so you need a JDBC connection. JDBC is an SQL-based Java API that allows for connectivity between the Java programming language and various databases. With JDBC it is possible to connect with the database and execute queries. Once these two steps are done, one can process the results. Key components of JDBC are:

1. Driver manager

2. Driver

3. Connection

The syntax to connect with the database is:

DriverManager.getConnection(URL, “userid”, “password” )

Similarly, the code to load the JDBC driver is:

Class.forName(“com.mysql.jdbc.Driver”);

To send queries to the database one can use the Statement object. Similarly, to process the data one can use a getXXX() method.

41. Why are assertions important in Selenium? What are different types of assertions?

Assertions are important for validation. They clearly state whether or not a test case is behaving as expected. As a regression suite or sanity suite, for instance, runs for a long duration, it is not always possible to sit in front of the system and look at the execution. Assertions help testers mark test cases as passed or failed.

There are 2 types of assertions:

1. Soft assert

This type of assertion will just verify the condition and give the result, but it won’t abort the test case execution when the test case fails.

2. Hard Assert

This type of assertion checks for the expected result, and if the condition fails to match, it will abort execution and throw the “java.lang.AssertionError” exception.

The different types of hard assertions in Selenium are:

assertEquals()

assertNotEquals()

assertNull()

assertNotNull()

assertTrue()

assertFalse()

42) Mention the types of Web locators.

Locator is a command that tells Selenium IDE which GUI elements ( say Text Box, Buttons, Check Boxes, etc) it needs to operate on. Locators specify the area of action.

Locator by ID: It takes a string parameter which is a value of the ID attribute which returns the object to findElement() method.

  driver.findElement(By.id(“user”));

Locator by the link: If your targeted element is a link text then you can use the by.linkText locator to locate that element.

  driver.findElement(By.linkText(“Today’s deals”)).click();

Locator by Partial link: The target link can be located using a portion of text in a link text element.

  driver.findElement(By.linkText(“Service”)).click();

Locator by Name: The first element with the name attribute value matching the location will be returned.

  driver.findElement(By.name(“books”).click());

Locator by TagName: Locates all the elements with the matching tag name

  driver.findElement(By.tagName(“button”).click());

Locator by classname: This finds elements based on the value of the CLASS attribute. If an element has many classes then this will match against each of them.

  driver.findElement(By.className(“inputtext”));

Locator by XPath: It takes a parameter of String which is a XPATHEXPRESSION and it returns an object to findElement() method.

  driver.findElement(By.xpath(“//span[contains(text(),’an account’)]”)).getText();

Locator by CSS Selector: Locates elements based on the driver’s underlying CSS selector engine.

  driver.findElement(By.cssSelector(“input#email”)).sendKeys(“myemail@email.com”);

**43) What are the types of waits supported by WebDriver?**



Implicit wait - Implicit wait commands Selenium to wait for a certain amount of time before throwing a “No such element” exception.

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

Explicit wait - Explicit wait is used to tell the Web Driver to wait for certain conditions before throwing an "ElementNotVisibleException" exception.

WebDriverWait wait = new WebDriverWait(WebDriver Reference, TimeOut);

Fluent wait - It is used to tell the web driver to wait for a condition, as well as the frequency with which we want to check the condition before throwing an "ElementNotVisibleException" exception.

Wait wait = new FluentWait(WebDriver reference).withTimeout(timeout, SECONDS).pollingEvery(timeout, SECONDS).ignoring(Exception.class);

44) Mention the types of navigation commands

driver.navigate().to("https://www.ebay.in/"); - Navigates to the provided URL

driver.navigate().refresh(); - This method refreshes the current page

driver.navigate().forward(); - This method does the same operation as clicking on the Forward Button of any browser. It neither accepts nor returns anything.

driver.navigate().back(); - This method does the same operation as clicking on the Back Button of any browser. It neither accepts nor returns anything.

30. What is the major difference between driver.close() and driver.quit()?

driver.close()

This command closes the browser’s current window. If multiple windows are open, the current window of focus will be closed.

driver.quit()

When quit() is called on the driver instance and there are one or more browser windows open, it closes all the open browser windows.

31) How to click on a hyperlink in Selenium?

driver.findElement(By.linkText(“Today’s deals”)).click();

The command finds the element using link text and then clicks on that element, where after the user would be redirected to the corresponding page.

driver.findElement(By.partialLinkText(“Service”)).click();

The above command finds the element based on the substring of the link provided in the parenthesis and thus partialLinkText() finds the web element.

Reference :

[Top 30 Selenium Interview Questions and Answers in [2020] (simplilearn.com)](https://www.simplilearn.com/tutorials/selenium-tutorial/selenium-interview-questions-and-answers?source=sl_frs_nav_playlist_video_clicked)