1. What are the Selenium suite components?

Selenium IDE

It is a Firefox/Chrome plug-in that was developed to speed up the creation of automation scripts. It records the user actions on the web browser and exports them as a reusable script.

Selenium Remote Control (RC)

RC is a server that allows users to write application tests in various programming languages. The commands from the test script are accepted by this server and are sent to the browser as Selenium core JavaScript commands. The browser then behaves accordingly.

Selenium WebDriver

WebDriver is a programming interface that helps create and run test cases. It makes provision to act on web elements. Unlike RC, WebDriver does not require an additional server and interacts natively with the browser applications.

Selenium Grid

The grid was designed to distribute commands to different machines simultaneously. It allows the parallel execution of tests on different browsers and different operating systems. It is exceptionally flexible and is integrated with other suite components for simultaneous execution.

2. What are the limitations of Selenium testing?

1. Unavailability of reliable tech support: Since Selenium is an open-source tool, it does not have dedicated tech support to resolve the user queries.
2. Tests web applications only: Selenium needs to be integrated with third-party tools like Appium and TestNG to test desktop and mobile applications.
3. Limited support for image testing.
4. 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.
5. May require the knowledge of programming languages: Selenium WebDriver expects the user to have some basic knowledge about programming.

3. What are the testing types supported by Selenium?

Selenium supports Regression testing and Functional testing.

Regression testing - It is a full or partial selection of already executed test cases that are re-executed to ensure existing functionalities work fine.

The steps involved are -

1. Re-testing: All tests in the existing test suite are executed. It proves to be very expensive and time-consuming.
2. Regression test selection: Tests are classified as feature tests, integration tests,  and the end to end tests. In this step, some of the tests are selected.
3. Prioritization of test cases: The selected test cases are prioritized based on business impact and critical functionalities.

Functional testing - Functional Testing involves the verification of every function of the application with the required specification.

The following are the steps involved:

1. Identify test input
2. Compute test outcome
3. Execute test
4. Compare the test outcome with the actual outcome

4. What is the difference between Selenium 2.0 and Selenium 3.0?

Selenium 2.0 is a tool that makes the development of automated tests for web applications easier. It represents the merger of the original Selenium project with the WebDriver project. Selenium RC got deprecated since the merge, however, was used for backward compatibility

https://www.simplilearn.com/ice9/free_resources_article_thumb/selenium2.JPG

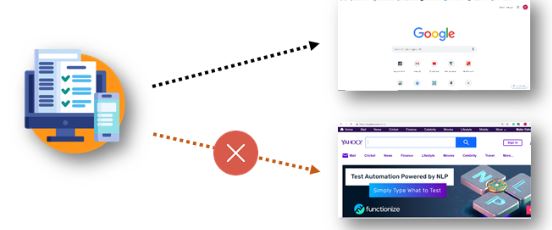
Selenium 3.0 is the extended version of Selenium 2.0. It is inherently backward compatible and does not involve Selenium RC. The new version came along with several bug fixes and increased stability.

selenium3.

5. What is the same-origin policy and how is it handled?

Same Origin policy is a feature adopted for security purposes. According to this policy, a web browser allows scripts from one webpage to access the contents of another webpage provided both the pages have the same origin. The origin refers to a combination of the URL scheme, hostname, and port number.

The same Origin Policy prevents a malicious script on one page to access sensitive data on another webpage.

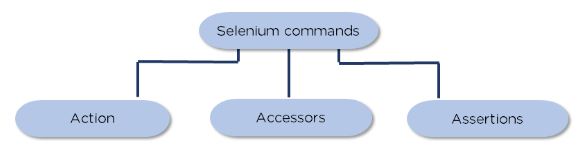


Consider a JavaScript program used by google.com. This test application can access all Google domain pages like google.com/login, google.com/mail, etc. However, it cannot access pages from other domains like yahoo.com

Selenium RC was introduced to address this. The server acts as a client configured HTTP proxy and "tricks" the browser into believing that Selenium Core and the web application being tested come from the same origin.

### 6. What is Selenese? How is it classified?

Selenese is the set of Selenium commands which are used to test your web application. The tester can test the broken links, the existence of some object on the UI, Ajax functionality, alerts, window, list options, and a lot more using Selenese.



**Action:** Commands which interact directly with the application

**Accessors:** Allow the user to store certain values to a user-defined variable

**Assertions:** Verifies the current state of the application with an expected state

### 7. 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”));

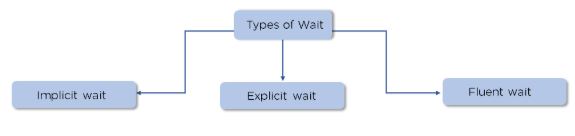
**Locator by XPath:** It takes a parameter of String which is a XPATHEXPRESSION and it returns an object to findElement() method.

  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”);

### 8. 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);

### 9. 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.

### 10. 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.

### 11. What makes Selenium such a widely used testing tool? Give reasons.

1. Selenium is easy to use since it’s essentially developed in JavaScript.
2. Selenium can test web applications against browsers like Firefox, Opera, Chrome, and Safari, to name a few.
3. The test code can be written in various programming languages like Java, Perl, Python, and PHP.
4. Selenium is platform-independent, and can be deployed on different Operating systems like Windows, Linux, and Macintosh.
5. Selenium can be integrated with third-party tools like JUnit and TestNG for test management.

## **Intermediate Level Selenium Interview Questions**

### 12. How to type text in an input box using Selenium?

sendKeys() is the method used to type text in input boxes

Consider the following example -

  WebElement email = driver.findElement(By.id(“email”)); - Finds the “email” text using the ID locator

  email.sendKeys(“[abcd.efgh@gmail.com](mailto:abcd.efgh@gmail.com)”);  - Enters text into the URL field

  WebElement password = driver.findElement(By.id(“Password”)); - Finds the “password” text using the ID locator

  password.sendKeys(“abcdefgh123”); - Enters text into the password field

### 13. 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.

### 14. How to scroll down a page using JavaScript?

scrollBy() method is used to scroll down the webpage

**General syntax:**

executeScript("window.scrollBy(x-pixels,y-pixels)");

First, create a JavaScript object

   JavascriptExecutor js = (JavascriptExecutor) driver;

Launch the desired application

   driver.get(“https://www.amazon.com”);

Scroll down to the desired location

   js.executeScript("window.scrollBy(0,1000)");

The window is not scrolled vertically by 1000 pixels

### 15. How to assert the title of a webpage?

Get the title of the webpage and store in a variable

    String actualTitle = driver.getTitle();

Type in the expected title

   String expectedTitle = “abcdefgh";

Verify if both of them are equal

   if(actualTitle.equalsIgnoreCase(expectedTitle))

   System.out.println("Title Matched");

  else

  System.out.println("Title didn't match");

Alternatively,

   Assert.assertEquals(actualTitle, expectedTitle);

### 16. How to mouse hover over a web element?

Actions class utility is used to hover over a web element in Selenium WebDriver

Instantiate Actions class.

    Actions action = new Actions(driver);

In this scenario, we hover over search box of a website

  actions.moveToElement(driver.findElement(By.id("id of the searchbox"))).perform();

Master important testing concepts such as TestNG, Selenium IDE, Selenium Grid, Selenium WebDriver. Check out [Selenium Certification Training](https://www.simplilearn.com/selenium-certification-training). Enroll now!

### 17. How to retrieve CSS properties of an element?

getCssValue() method is used to retrieve CSS properties of any web element

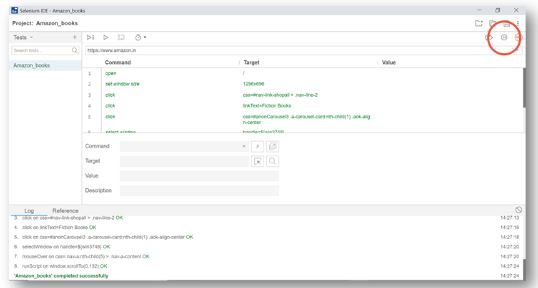
**General Syntax:**

  driver.findElement(By.id(“id“)).getCssValue(“name of css attribute”);

**Example:**

   driver.findElement(By.id(“email“)).getCssValue(“font-size”);

### 18. What is POM (Page Object Model)?



Every webpage of the application has a corresponding page class that is responsible for locating the web elements and performing actions on them. Page Object Model is a design pattern that helps create object repositories for the web elements. POM improves code reusability and readability. Multiple test cases can be run on the object repository.

### 19. Can Captcha be automated?

No, Selenium cannot automate Captcha. Well, the whole concept of Captcha is to ensure that bots and automated programs don’t access sensitive information - which is why, Selenium cannot automate it. The automation test engineer has to manually type the captcha while other fields can be filled automatically.

### 20. How does Selenium handle Windows-based pop-ups?

Selenium was designed to handle web applications. Windows-based features are not natively supported by Selenium. However, third-party tools like AutoIT, Robot, etc can be integrated with Selenium to handle pop-ups and other Windows-based features.

### 21. How to take screenshots in WebDriver?

TakeScreenshot interface can be used to take screenshots in WebDriver.

getScreenshotAs() method can be used to save the screenshot

File scrFile = ((TakeScreenshot)driver).getScreenshotAs(outputType.FILE);

## **Advanced Level Selenium Interview Questions**

### 22. Is there a way to type in a textbox without using sendKeys()?

Yes! Text can be entered into a textbox using JavaScriptExecutor

JavascriptExecutor jse = (JavascriptExecutor) driver;

jse.executeScript("document.getElementById(‘email').value=“abc.efg@xyz.com”);

### 23. How to select a value from a dropdown in Selenium WebDriver?

Select class in WebDriver is used for selecting and deselecting options in a dropdown.

The objects of Select type can be initialized by passing the dropdown webElement as a parameter to its constructor.

WebElement testDrop = driver.findElement(By.id("testingDropdown"));

Select dropdown = new Select(testDrop);

WebDriver offers three ways to select from a dropdown:

selectByIndex: Selection based on index starting from 0

   dropdown.selectByIndex(5);

selectByValue: Selection based on value

   dropdown.selectByValue(“Books”);

selectByVisibleText: Selection of option that displays text matching the given argument

   dropdown.selectByVisibleText(“The Alchemist”);

### 24. What does the switchTo() command do?

switchTo() command is used to switch between windows, frames or pop-ups within the application. Every window instantiated by the WebDriver is given a unique alphanumeric value called “Window Handle”.

Get the window handle of the window you wish to switch to

   String  handle= driver.getWindowHandle();

Switch to the desired window

   driver.switchTo().window(handle);

Alternatively

  for(String handle= driver.getWindowHandles())

   { driver.switchTo().window(handle); }

### 25. How to upload a file in Selenium WebDriver?

You can achieve this by using sendkeys() or Robot class method. Locate the text box and set the file path using sendkeys() and click on submit button

Locate the browse button

   WebElement browse =driver.findElement(By.id("uploadfile"));

Pass the path of the file to be uploaded using sendKeys method

    browse.sendKeys("D:\\SeleniumInterview\\UploadFile.txt");

### 26. How to set browser window size in Selenium?

The window size can be maximized, set or resized

To maximize the window

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

To set the window size

  Dimension d = new Dimension(400,600);

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

Alternatively,

The window size can be reset using JavaScriptExecutor

 ((JavascriptExecutor)driver).executeScript("window.resizeTo(1024, 768)");

### 27. When do we use findElement() and findElements()?

findElement() is used to access any single element on the web page. It returns the object of the first matching element of the specified locator.

General syntax:

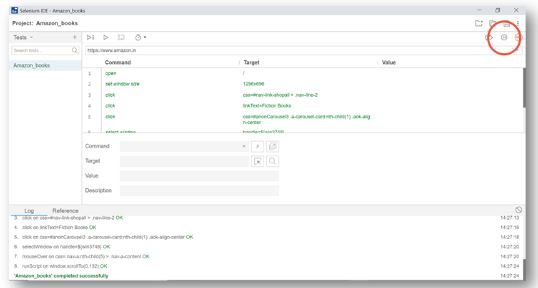
   WebElement element = driver.findElement(By.id(example));

findElements() is used to find all the elements in the current web page matching the specified locator value. All the matching elements would be fetched and stored in the list of Web elements.

General syntax:

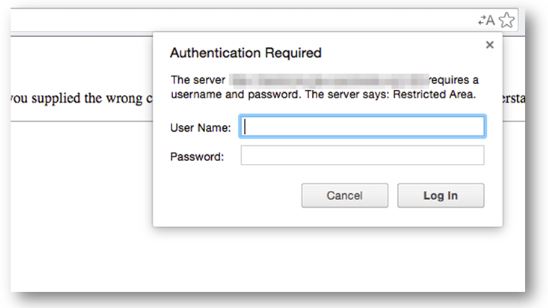
   List <WebElement> elementList = driver.findElements(By.id(example));

### 28. What is a pause on an exception in Selenium IDE?



The user can use this feature to handle exceptions by clicking the pause icon on the top right corner of the IDE. When the script finds an exception it pauses at that particular statement and enters a debug mode. The entire test case does not fail and hence the user can rectify the error immediately.

### 29. How to login to any site if it is showing an Authentication Pop-Up for Username and Password?



To handle authentication pop-ups, verify its appearance and then handle them using an explicit wait command.

Use the explicit wait command

  WebDriverWait wait = new WebDriverWait(driver, 10);

Alert class is used to verify the alert

   Alert alert = wait.until(ExpectedConditions.alertIsPresent());

Once verified, provide the credentials

  alert.authenticateUsing(new UserAndPassword(<username>, <password>));

### 30. What is the difference between single and double slash in Xpath?

Single slash is used to create Xpath with an absolute path i.e. the XPath would be created to start selection from the start node.

/html/body/div[2]/div[1]/div[1]/a

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

//div[class="qa-logo"]/a

### 31. How do you find broken links in Selenium WebDriver?

When we use driver.get() method to navigate to a URL, it will respond with a status of 200-OK

200 – OK denotes that the link is working and it has been obtained. If any other status is obtained, then it is an indication that the link is broken.

Some of the HTTP status codes are :

* 200 – valid Link
* 404 – Link Not Found
* 400 – Bad Request
* 401 – Unauthorized
* 500 – Internal error

As a starter, obtain the links from the web application, and then individually get their status.

Navigate to the interested webpage for e.g. www.amazon.com

Collect all the links from the webpage. All the links are associated with the Tag ‘a‘

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

Create a list of type WebElement to store all the Link elements in it.

  for(int i=0; i<links.size(); i++) {

           WebElement element = links.get(i);

                      String url=element.getAttribute("href");

                       verifyLink(url);  }

Now Create a Connection using URL object( i.e ., link)

  URL link = new URL(urlLink);

Connect using Connect Method

   HttpURLConnection httpConn =(HttpURLConnection)link.openConnection();

Use getResponseCode () to get response code

    if(httpConn.getResponseCode()!== 200)

Through exception, if any error occurred

    System.out.println(“Broken Link”);

With that we have come to the end of the article Seleinium interview questions.