RahulShettyAcademy.com   
 Advance Selenium + Core Java Interview Ques

1. **What are the Challenges with Selenium Automation?**

No Support for Non-Web Automation

Ex: if a PDF gets generated we can’t scan it. Any non web related items cannot be automated with Selenium.

Window or Desktop based applications cannot be automated with Selenium. Need to rely on external utilities to perform such actions.

Timeout or Sync Issues

Ex: Selenium will wait only for mentioned amount of time. Cypress can understand browser behavior. It waits until the page/element is loaded

Test Execution Slowness in Internet Explorer

Limited Reporting

Ex: By default Selenium will not provide any reporting. Need to rely on external utilities like Extent reports /HTML reports or TestNG/JUnit Reporter Log

1. **What are new Selenium 4 features?**

* WebDriver is developed completely by W3C Standardization
* The Selenium IDE support for Chrome is available now

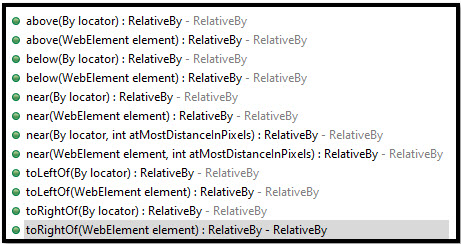
In earlier versions Selenium IDE support is there only for Firefox

* Selenium Grid - The hubs and nodes are smooth to setup and handle now. Once a Selenium server is started, the grid will act both as a hub and node.
* Taking screenshots at the Element level, Section level and Page Level is possible now.
* Support of Relative Locators

<https://blog.testproject.io/2020/07/13/selenium-4-relative-locators/#:~:text=The%20purpose%20of%20Relative%20Locators,parameter%20or%20WebElement%20element%20parameter>.

The purpose of Relative Locators is to find a specific element related to the position of another element. There’s a total of 5 overloaded methods with an option of 2 parameters. We can use the By locator parameter or WebElement element parameter.The following is a list of all 5 Relative Locator methods in alphabetical order:

1. **above()** – finds an element or elements located **above** a fixed element.
2. **below()**– finds an element or elements located **below** a fixed element.
3. **near()** – finds an element or elements located **near** a fixed element.
4. **toLeftOf()** – finds an element or elements located **to the left of** a fixed element.
5. **toRightOf()** – finds an element or elements located **to the right of** a fixed element.



Relative Locators join the existing group of Selenium’s 8 locators: id, name, className, tagName, cssSelector, linkText, partialLinkText, and xpath.

1. **What is difference between WebDriver.findElement vs WebElement.findElement ?**

**driver.findElement():**

It is for finding the element from the entire page using the given selector.

**WebElement.findElement():**

First, it generates the WebElement, and then only the child elements of the given element are searched based on the given selector.

1. **What is the difference between Page Object Model and Page Factory?**

**Page Object Model (POM):**

POM is a Selenium design pattern which helps in separating page objects and scripts; we can see it as a repository where we store all the WebElements. This has become very popular in industry these days, because it is very easy to manage, reusability of code and eliminates duplication of code.

The key benefit is if UI changes in the future, then we can update WebElements to Page Classes in POM or Object Repository accordingly

**Page Factory:**

Page Factory in Selenium WebDriver is an integrated concept or API. Here we follow again the same principle of keeping repository objects or page classes separate from test classes.

Selenium WebDriver provides a class called PageFactory which helps to assist Page Object of Design patterns. Here we use an annotation ‘@FindBy’ to find web elements. There is a method called an ‘initElements’ method, which is used for initializing the elements of the web.

<https://www.geeksforgeeks.org/difference-between-pom-and-pagefactory/>

1. **What are all the locators that support selenium?**

A locator should uniquely identify or locate the element on the web page

**ID**: An ID is unique to a web element

**Name**: Same as ID although it is not unique

**CSS** **Selector**: Works on element tags and attributes

**XPath**: Searches elements in the DOM (based on hierarchy in an HTML document), Reliable but slow

**Class** **name**: Uses the class name attribute

**TagName**: Uses HTML tags to locate web elements

**LinkText**: Uses anchor text to locate web elements

**Partial** **Link Text**: Uses partial anchor link text to find web elements

1. **How To Overcome StaleElementReferenceException in Selenium**

The reference to an element is now "stale"; the element will no longer appear on the page's DOM. In simple words, when you started interacting with it, the element you located using the findElement method disappeared.

Adding exception handling to your action and, if the exception is stale, try to locate the element after a simple wait for 500 milliseconds and repeat these actions until the action or max iterations have been successful.

StaleElementReferenceException can be avoided by using some workarounds like refreshing the webpage before accessing the web element, using the try-catch block, using explicit wait(wait for the expected condition or wait for the refresh of DOM), or using POM.

In [***POM design pattern***](https://www.toolsqa.com/selenium-webdriver/page-object-model/) or while using the principles of [***Page Factory***](https://www.toolsqa.com/selenium-webdriver/page-factory-in-selenium/), we locate an element using the @FindBy annotation. It helps in updating the reference of the web element each time before any action is performed on it. This becomes very useful in eliminating the occurrence of StaleElementReferenceException.

<https://toolsqa.com/selenium-webdriver/what-is-stale-element#:~:text=StaleElementReferenceException%20can%20be%20avoided%20by,DOM)%2C%20or%20using%20POM>.

1. **Different Between XPath and CSS Selector?**

* Xpath is slower than CSS, whereas CSS Selector is faster than XPath.
* XPath supports text though CSS Selector does not allow Text.
* XPath can move in both forward and backward directions whereas CSS Selector can move only in forward direction.

1. **How to access the CSS selector using the nth element?**

Here is a syntax for using the CSS selector to access the nth attribute: <type>:nth-child(n)

Ex: tr:nth-child(2)

1. **How to handle alerts in Selenium WebDriver?**

WebDriver provides an API to handle alert dialogs. You will get ‘NoAlertPresentException’ when there is no alert and still selenium tries to find it.

The Alert interface contains a number of APIs to execute different actions.

The following list:

Alert alert = driver.switchto().alert();

alert.accept(); This is equivalent to the OK button action on the dialog

alert.dismiss(); This is equivalent to clicking on the CANCEL action button

alert.sendKeys("String"); This will allow to type in some text into the alert

alert.getText(); This will return the text that appears on the dialog

1. **What are the different exceptions you faced in Selenium WebDriver?**

* WebDriverException

When the Browser is not invoked properly with WebDriver object

* NoSuchElementException

If there is No Element present

* NoSuchFrameException

If there is No frame available

* NoAlertPresentException

If there is No Alert present

* NoSuchWindowException

If there is No window available

* ElementNotVisibleException

If the Element is present but not visible

* ElementNotInteractableException

For example if a link is hidden by a dropdown menu or any advertisement

* SessionNotCreatedException

When browser itself is not properly invoked or When there is issue with browser drivers

* TimeoutException

When timeout is provided with Implicit or Explicit waits and if element is not identified within that given time

* InvalidSelectorException

When invalid element locator is provided i.e., wrong xpath or css is provided

* IllegalStateException

It’s a JAVA error that may occur during compilation issues

* StaleElementReferenceException

1. **What is a framework? What are the different types of frameworks available?**

A framework is a set of rules and best practices for the systemic resolution of a problem.

There are different kinds of automation frameworks:

* Data-Driven Testing Framework
* Keyword Driven Testing Framework
* Hybrid Testing Framework
* Behavioural Driven Framework

1. **How to run Tests in Headless Mode with Chrome?**

ChromeOptions chromeOptions = new ChromeOptions();

// chromeOptions.addArguments("--no-sandbox"); //optional

chromeOptions.addArguments("--headless");

driver = new ChromeDriver(chromeOptions)

If we can run test cases in headless mode(browser will not open) performance of execution can be improved. To change Chrome configuration required Arguments can be added.

1. **How to handle windows-based alerts/popups in selenium?**

driver.switchto().alert(); 🡪 this is Java Alert

Selenium only supports web applications and does not provide a way to automate Windows-based applications. However, the following approaches can help.

Need to depend on external tools(like Robot class or AutoIT) to handle windows based pop-ups

* Use the Robot class (Java-based) utility to simulate the keyboard and mouse actions. That is how you can handle the window based pop-ups.
* AutoIt Integration with Selenium helps to automate Window Based Popups

1. **What are Listeners in Selenium?**

Listeners is an interface that modifies the behavior of the system under test. Listeners allow customization of reports and logs.

When Test execution is done Pass or Fail status is to be sent to reports. That code is written in Listeners to display status in the reports.

For example if a test fails Listeners will capture the failure and execute required methods like onTestFailure() or if a test passes onTestSuccess()

Listeners mainly comprise of two types, namely

* WebDriver listeners
* TestNG listeners

### What are the differences between StringBuffer and StringBuilder?

StringBuffer is *synchronized*, i.e., thread safe. It means two or more threads(tests) can't call the methods of StringBuffer simultaneously.

StringBuilder is *non-synchronized*,i.e., not thread safe. It means two or more threads(tests) can call the methods of StringBuilder simultaneously.

If your Tests run in Parallel mode use StringBuffer if they run in sequential mode use StringBuilder.

These are the 2 classes to create strings in Java. If tests are running in sequential mode we can use either of these classes. When parallel tests are are to be executed in the framework then it is better to use StringBuffer as it is Thread safe.

1. **What are the advantages of selenium in automation testing world?**

* It is an open source platform free to use. This method does not need to be allotted budget
* It works on all operating systems such as Windows, Linux and Mac, as it is compatible across systems
* It works on almost all common browsers including Chrome, Firefox, Edge, Internet Explorer and Safari, since it is compatible with cross-browser testing
* It supports Java, Python, Perl, PHP, C #, Ruby programming languages
* Selenium Grid concepts allow parallel execution
* Continuous integration With Jenkins and Hudson we will achieve daily execution
* It provide a wide base of users and support communities because this is an open platform

1. **What is soft and hard assertion in selenium?**

**Soft Assertion**: Soft Assert will not throw an exception when an assert fails, and after the assertion execution will continue with the next step.

Your test will not fail. Test execution will resume when assertion fails but at the end it will report where the assertions failed

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

1. **What is the purpose of static methods and variables?**

When the methods or variables are defined as static they are shared by all the objects of the class. The static is a part of the class and not of the object. The static variables are stored in the class area, and we do not need to create the object to access such variables. Therefore, static is used in the case, where we need to define variables or methods which are common to all the objects of the class.

Static variable will get memory only once and retain its value. Where as Instance variable i.e., normal class variable will get memory each time when we create object of the class.

Static variables and methods belong to class and not objects.

Static methods can be called directly with class name without creating any object of that class.

Static methods will only accept static variables. Static method cannot have a non static varible.

<https://www.javatpoint.com/static-keyword-in-java>

1. **Usage of This and Super Keywords in Java?**

**This**:

* This keyword is used to initialize class level variables in the constructor using local variables
* We use this keyword to distinguish local variables(formal arguments) and instance variables. [this keyword refers to current class local variable]
* This keyword can only be used for constructor
* We can use more than one **This** Keyword within the Constructor at a time.

**Super**:

* We can use super keyword to access the data members or fields of parent class. It is used if parent class and child class have same fields to avoid conflict.
* The super keyword can also be used to invoke parent class method. It is used if parent class and child class have same methods to avoid conflict.
* super is used to invoke parent class constructor

## <https://www.javatpoint.com/this-keyword> <https://www.javatpoint.com/super-keyword>

1. **Difference Between Array and Array List?**

Array is a fixed length data structure whereas ArrayList is a variable length Collection class. We cannot change length of array once created in Java but ArrayList can be dynamically changed.

int[] integerList = new int[10];

ArrayList<Integer> integerList= new ArrayList<Integer>();  
integerList.add(1);

integerList.add(2);

1. **Difference between abstract class and interface?**

A method without any implementation or body is called abstract method.

**Interface:**

1) Interface contains only abstract methods

2) Access Specifiers for methods in interface must be public

3) Variables defined must be public, static, final 🡪 variables are constant

4) To implement an interface we use implements keyword

5) Interface supports multiple inheritance.

**Abstract Class:**

1. Abstract class can contain abstract methods, concrete methods or both
2. Abstract class can have final, non-final, static and non-static variables.

3) Abstract class doesn't support multiple inheritance.

<https://www.javatpoint.com/difference-between-abstract-class-and-interface>

# **Difference between HashMap and Hashtable** HashMap is **non synchronized**. It is not-thread safe and can't be shared between many threads without proper synchronization code.

# Hashtable is **synchronized**. It is thread-safe and can be shared with many threads.

If your Tests run in Parallel mode use Hashtable if they run in sequential mode use Hashmap.

# HashMap **allows one null key and multiple null values**.

Hashtable **doesn't allow any null key or value**.

1. **Difference between final, finally and finalize**

<https://www.javatpoint.com/difference-between-final-finally-and-finalize>

We use finally block in our Framework to close the browser or delete the cookies. As we need to close the browser no matter if the test is passed or failed

1. **When should I 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, testing under different environments and saving execution time remarkably.

1. **How many objects will be created in the following code?**
2. String s1="Welcome";
3. String s2="Welcome";
4. String s3="Welcome";

Only one object will be created using the above code because in Java strings are immutable.  
Each time you create a string literal, the JVM checks the "string constant pool" first. If the string already exists in the pool, a the reference to pooled instance is returned. If the string doesn't exist in the pool, a new string instance is created and placed in the pool. String objects are stored in a special memory area known as the **string constant pool**