**1. What is Selenium?**

Selenium is a tool for automating everything you can do inside a web browser. It automates functional aspects of web-based applications and can run across all browsers and platforms.

[**Selenium WebDriver with Java -Basics to Advanced+Frameworks**](https://www.udemy.com/course/selenium-real-time-examplesinterview-questions/)

Last Updated June 2021

**Bestseller**

* 350 lectures

* All Levels

**4.6** (62,998)

“TOP RATED (BEST SELLER) #1 Master SELENIUM java course” -5 Million students learning worldWide with great collaboration | By Rahul Shetty

[**Explore Course**](https://www.udemy.com/course/selenium-real-time-examplesinterview-questions/)

**2. Which are the components of the Selenium suite**

Selenium is not just a single tool or a utility. Rather, it’s a bundle of multiple test tools, referred to as a suite. Each of the tools is designed to meet specific criteria for testing and testing environments.

The suite package contains the following set of tools:

* **Selenium IDE (Integrated Development Environment)** – Open-source record and playback test automation for the web. This is distributed as a plug-in to Firefox and Chrome.
* **Selenium RC** **(Remote Control)** – Selenium RC is a server that allows a user to build test scripts in a desired programming language. It also enables test scripts to be performed across a wide spectrum of browsers.
* **Selenium WebDriver** – WebDriver is an entirely different tool, with different advantages over Selenium RC. WebDriver interacts directly with the web browser and makes use of its native compatibility to automate.
* **Selenium Grid** – Selenium Grid is used to simultaneously distribute the test execution on multiple platforms and environments.

**3. What are the advantages of Selenium in the automation testing world?**

Selenium is a free open-source platform and supports a wide array of users and support communities. It’s compatible across systems, making it friendly for Windows, Linux, and Mac users. It supports Java, Python, Perl, PHP, C#, and Ruby programming languages.

Since Selenium is compatible with cross-browser testing, it works on almost all common browsers including Chrome, Firefox, Edge, Internet Explorer and Safari. Another benefit is that Selenium Grid allows for parallel execution.

**4. What is WebDriver?**

WebDriver is a simple and concise programming interface. As an object-oriented and lightweight API, it drives the browser through interface-type reference variables.

**5. What is Selenium 4?**

Selenium 4 is the latest version of Selenium WebDriver, which will be fully standardized with [W3C](https://www.w3.org/). Since most browsers such as Chrome, Safari, Firefox, Internet Explorer, and Edge follow W3C standardization, browser drivers will interact with Selenium WebDriver in W3C standard protocol.

**6. What are some noteworthy Selenium 4 features?**

* Selenium Grid – The hubs and nodes are smooth to set up 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 for the Chromium driver for Microsoft Edge browser
* Support of Relative Locators
* In one browser window, running with different browsers at the same time and with different tabs

**7. What are all the relative locators in Selenium 4?**

We have several methods of finding a (relative) element close by. These are:

* oLeftOf() : Element located to the left of specified element.
* toRightOf() : Element located to the right of the specified element.
* above() : Element located above with respect to the specified element.
* below() : Element located below with respect to the specified element.
* near() : Element is a range of at most 50 pixels from the specified element. The pixel value is adjustable

**8. What are the different waits in Selenium WebDriver?**

There are three types of waits in WebDriver:

1. **Implicit Waits**: This is a wait timeout that applies to an instance on a WebDriver. It implies that all actions of this instance are time-out only after waiting for a period of time.
2. **Explicit Waits**: This is an exclusive time-out 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.
3. **Fluent Waits:**This defines the maximum amount of time to wait for a particular condition and frequency to test the condition before throwing an exception to “ElementNotVisibleException.”

**9. What is the difference between WebDriver.findElement vs WebElement.findElement?**

* **WebDriver.findElement():**This finds the element from the entire page using the given selector.
* **WebElement.findElement():** First, this generates the WebElement. Then, the child elements of the given element are searched based on the given selector.

**10. What are all the design patterns in the Selenium framework?**

* Page Object Model (POM) Pattern
* Factory Pattern
* Facade Pattern
* Strategy Pattern

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

* **Page Object Model (POM):**POM is a Selenium design pattern. It’s seen as a repository where we store all the WebElements. POM is popular because it’s easy to manage, offers code reusability, and eliminates duplication of code. 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. It follows the same principle of keeping repository objects or page classes separate from test classes. Here we use @FindBy to find elements and to initialize WebElements using initElements process.

**12. What are all the locators that support Selenium?**

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

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

**XPath**: Searches elements in the DOM. 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 link text to find web elements

**13. What are the differences between XPath and CSS Selector?**

* XPath is slower than CSS Selector
* XPath supports text, while CSS Selector does not
* XPath supports index
* XPath can move in both forward and backward directions whereas CSS Selector can only move forward

**14. 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)

**15. How can one handle alerts in Selenium WebDriver?**

WebDriver provides an API to handle alert dialogs.  Alerts cannot able to inspect if there is no Alert in the screen, you will get ‘NoAlertPresentException’

The Alert interface contains a number of APIs to execute different actions, including:

* 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

**16. What are the exceptions in Selenium WebDriver?**

* WebDriverException
* NoSuchElementException
* NoSuchFrameException
* NoAlertPresentException
* NoSuchWindowException
* ElementNotVisibleException
* ElementNotInteractableException
* SessionNotCreatedExceptio
* TimeoutException
* InvalidSelectorException
* IllegalStateException
* StaleElementReferenceException

**17. How would you overcome StaleElementReferenceException in Selenium?**

Because the reference to an element is now stale the element will no longer appear on the page’s DOM. In other words, the element you located using the findElement method disappeared.

To resolve this, add exception handling to your action. If the exception is stale, locate the element after a short wait of 500 milliseconds and repeat these actions until the action or max iterations have been successful.

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

A framework is a charter 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
* Behavioral Driven Framework

**19. How should one handle Windows-based alerts or pop-ups in Selenium?**

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

Use the Java-based Robot class utility to simulate keyboard and mouse actions. The KeyPress and KeyRelease methods simulate the user pressing and releasing a specific key on the keyboard.

**20. What are TestNG annotations frequently used with Selenium?**

TestNG annotations prioritize the calling of a test method over others. Here are the ones to use with Selenium:

* @BeforeSuite – to run before all tests
* @AfterSuite – to run only once after all tests
* @BeforeClass – to run only once before the first test method
* @AfterClass – to run only once after all the test methods of the current class finish execution
* @BeforeTest – to run before any test method inside the “Test” tag
* @AfterTest – to run after any test method inside the “Test” tag

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

Listeners is an interface that modifies the behavior of the system allowing customization of reports and logs.

There are two primary types of Listeners, WebDriver Listeners, and TestNG Listeners.

**22. What is a soft assertion and a hard assertion in Selenium?**

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

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

**23. What are static blocks or static initializers in Java?**

Static blocks or static initializers are used to initialize static fields in java. We declare static blocks when we want to initialize static fields in our class. Static blocks are executed exactly once when the class is loaded. Static blocks are executed even before the constructors are executed.

**24. How should you use the keywords This and Super in Java?**

**This**: This keyword can only be used for the constructor. It initializes class level variables in the constructor using local variables. We can use more than one **This** Keyword within the Constructor at a time.

**Super**: Using Super keyword to call Parent class constructor. We can only have one Super keyword declared within the constructor at a time. You should always write Super keyword in child class constructor

**25. Why is the main() method is public, static, and void in Java?**

* **public**: This is an access specifier which can be used outside the class. When main method is declared public, it means it can be used outside class.
* **static**: To call a method we require object, though sometimes it may be required to call a method without the help of object. Then we declare that method as static. JVM calls the main() method without creating object by declaring keyword static.
* **void**: The void return type is used when a method doesn’t return any value. main() method doesn’t return any value, so main() is declared as void.

**26. What are the differences between Method Overloading and Method Overriding in Java?**

|  |  |
| --- | --- |
| **Method Overloading** | **Method Overriding** |
| Method overloading occurs within the same class | Method overriding occurs between two classes superclass and subclass |
| Since it involves only one class, inheritance is not involved | Since method overriding occurs between superclass and subclass, inheritance is involved |
| In overloading, return type need not be the same | In overriding, return type must be same |
| Parameters must be different for overloading | Parameters must be same for overriding |
| Static polymorphism is achieved using method overloading | Dynamic polymorphism is achieved using method overriding |
| In overloading, one method can’t hide the another | In overriding, subclass method hides the superclass method |

**27. What is an exception in Java?**

In Java, exception is an object. Exceptions are created when an abnormal situation is raised in the program. Exceptions can be created by JVM or by our application code. All exception classes are defined in java.lang. In other words, we can see this as a runtime error.

**28. What are the differences between interface and abstract class?**

**Interface:**

1) Interface contains only abstract methods

2) Access Specifiers for methods in interface must be public

3) Defined variables must be public, static, final