



**GroTechMinds**  
Making Life Better Everyday

# 50 Selenium Interview Questions For Freshers



## 1. What is Selenium?

Selenium is an open-source automation testing tool for web applications across different browsers and platforms.

## 2. What are the different components of Selenium?

Selenium has four main components:

- Selenium IDE
- Selenium RC (deprecated)
- Selenium WebDriver
- Selenium Grid

## 3. What are the advantages of Selenium?

- Open-source and free
- Supports multiple browsers and platforms
- Supports various programming languages (Java, Python, C#, etc.)
- Strong community support
- Integration with other tools like TestNG, JUnit, Maven, etc.

## 4. What are the limitations of Selenium?

- Cannot test desktop applications
- No built-in reporting
- Limited support for handling CAPTCHA and images
- Requires external tools for handling mobile testing

## 5. What is Selenium WebDriver?

Selenium WebDriver is a web automation framework that allows you to execute cross-browser tests. It directly communicates with the browser without needing a separate server.



## **6. What is the difference between Selenium IDE and WebDriver?**

- Selenium IDE is a record-and-playback tool that's simpler but limited.
- WebDriver allows advanced scripting and supports multiple browsers, programming languages, and frameworks.

## **7. What browsers does Selenium support?**

Selenium supports major browsers like Chrome, Firefox, Safari, Edge, and Internet Explorer.

## **8. What is the difference between `findElement()` and `findElements()`?**

- `findElement()`: Returns a single `WebElement`.
- `findElements()`: Returns a list of `WebElements` or an empty list if no elements are found.

## **9. Can Selenium be used for mobile testing?**

Selenium itself cannot test mobile apps directly, but tools like Appium extend Selenium WebDriver functionality to mobile platforms.

## **10. What is Selenium Grid?**

Selenium Grid allows parallel test execution on multiple machines and browsers to speed up the testing process.





## **11. What is the difference between implicit and explicit waits?**

- Implicit Wait: Applies to all elements and sets a default wait time if an element is not found immediately.
- Explicit Wait: Applies to specific conditions or elements.

## **12. What is Fluent Wait in Selenium?**

Fluent Wait is a type of explicit wait where you can set the polling frequency and conditions for waiting, allowing customization beyond default waits.

## **13. What is a headless browser?**

A headless browser performs the same functionality as a standard browser but without the graphical user interface (GUI). Common examples include Chrome headless and PhantomJS.

## **14. Can Selenium handle windows-based popups?**

Selenium cannot directly handle windows-based popups because it's a web-based automation tool. Third-party tools like AutoIt or Robot class are needed.

## **15. How can you handle JavaScript pop-ups in Selenium?**

JavaScript alerts, prompts, and confirmation boxes can be handled using Selenium's `switchTo().alert()` method.

## **16. What is the difference between a window handle and a window title?**

- Window Handle: A unique identifier assigned to each window opened by the browser.
- Window Title: The name displayed in the title bar of the browser window.



## **17. What is the Page Object Model (POM)?**

POM is a design pattern that creates an object repository for web elements, allowing tests to be more maintainable and readable.

## **18. What is Page Factory in Selenium?**

Page Factory is an implementation of the POM, providing annotations like `@FindBy` to initialize web elements easily.

## **19. How do you take a screenshot in Selenium?**

Selenium WebDriver provides the `TakesScreenshot` interface for capturing screenshots.

## **20. What are some common exceptions in Selenium?**

- `NoSuchElementException`
- `TimeoutException`
- `ElementNotVisibleException`
- `WebDriverException`
- `StaleElementReferenceException`

## **21. What is the purpose of `driver.quit()` and `driver.close()`?**

- `driver.quit()`: Closes all browser windows opened by the WebDriver.
- `driver.close()`: Closes the current browser window.

## **22. How can you switch between different browser windows or tabs?**

Use the `getWindowHandles()` method to get all the window handles and then switch using `switchTo().window(handle)`.





### **23. What is the difference between isDisplayed(), isEnabled(), and isSelected()?**

- isDisplayed(): Checks if the element is visible on the page.
- isEnabled(): Checks if the element is enabled for interaction.
- isSelected(): Checks if a checkbox or radio button is selected.

### **24. How do you handle frames in Selenium?**

Use the switchTo().frame() method to handle frames and move between them.

### **25. What is an iframe, and how do you handle it?**

An iframe is an inline frame used to embed another document within the current HTML document. Handle iframes using switchTo().frame().

### **26. What are some Selenium alternatives for browser automation?**

Alternatives include Cypress, Puppeteer, TestCafe, and Playwright.

### **27. Can Selenium be integrated with test management tools?**

Yes, Selenium can be integrated with tools like TestNG, JUnit, Jenkins, and others for reporting, test case management, and CI/CD.

### **28. What is the use of Actions class in Selenium?**

The Actions class allows you to simulate keyboard and mouse events such as drag-and-drop, hover, and right-click



## **29. What is XPath?**

XPath is a query language used to select nodes from an XML or HTML document.

## **30. What is the difference between absolute and relative XPath?**

- Absolute XPath: Starts from the root node and uses a full path (/html/body/div).
- Relative XPath: Starts from the middle of the DOM structure (//div[@id='example']).

## **31. What are the different types of waits in Selenium?**

- Implicit Wait
- Explicit Wait
- Fluent Wait

## **32. How can you maximize a browser window in Selenium?**

Use `driver.manage().window().maximize()` to maximize the browser window.

## **33. How do you run tests in multiple browsers using Selenium?**

Use WebDriver's interface for different browsers (ChromeDriver, FirefoxDriver, etc.), or use tools like TestNG with browser parameters for cross-browser testing.

## **34. What are WebElements in Selenium?**

WebElements represent elements on a web page (e.g., buttons, input fields) that Selenium can interact with.





### **35. What is the use of get() and navigate().to()?**

- get(): Navigates to a specified URL and waits for the page to load completely.
- navigate().to(): Also navigates to a URL but doesn't wait for the full page load.

### **36. How do you perform a double-click action in Selenium?**

Use the Actions class and call doubleClick() on the WebElement.

### **37. What is the role of TestNG in Selenium?**

TestNG is a testing framework integrated with Selenium for managing test cases, running parallel tests, generating reports, and handling assertions.

### **38. How can you upload files in Selenium?**

You can simulate file upload by using the sendKeys() method to input the file path into an upload field.

### **39. What are some common strategies to locate elements in Selenium?**

- ID
- Name
- Class Name
- Tag Name
- Link Text / Partial Link Text
- CSS Selector
- XPath





#### **40. What is the difference between `getText()` and `getAttribute()`?**

- `getText()`: Retrieves the visible inner text of a `WebElement`.
- `getAttribute()`: Retrieves the value of a specific attribute of the `WebElement`.

#### **41. Can Selenium handle HTTPS certification issues?**

Yes, Selenium can handle these issues using browser-specific settings or `DesiredCapabilities`.

#### **42. What is the role of `JavascriptExecutor` in Selenium?**

`JavascriptExecutor` allows executing JavaScript code within the browser, which can be useful for handling dynamic elements or scrolling.

#### **43. What is the importance of `DesiredCapabilities` in Selenium?**

`DesiredCapabilities` is used to set browser properties like browser name, version, platform, etc., before starting the `WebDriver` session.

#### **44. What is the difference between `WebDriver` and `RemoteWebDriver`?**

- `WebDriver`: Local browser automation.
- `RemoteWebDriver`: Automates a browser on a remote server (used in Selenium Grid).

#### **45. What is a proxy in Selenium, and how do you use it?**

A proxy server can be used to route traffic through a middle layer for testing purposes. You can configure proxies using `DesiredCapabilities`.



#### **46. What is a WebDriverWait in Selenium?**

WebDriverWait is an explicit wait applied to wait for a specific condition or element before throwing an exception.

#### **47. Can you automate Captcha using Selenium?**

No, Captcha is designed to prevent automation, so you cannot automate it using Selenium.

#### **48. What is the difference between `navigate().refresh()` and `driver.get(driver.getCurrentUrl())`?**

- `navigate().refresh()`: Refreshes the browser window.
- `driver.get(driver.getCurrentUrl())`: Loads the current URL again, which also results in a page refresh.

#### **49. What are Selenium's advantages over QTP/UFT?**

- Open-source (free)
- Supports multiple programming languages
- Cross-platform
- Larger community support

#### **50. Can Selenium automate desktop applications?**

No, Selenium can only automate web applications. Desktop automation requires tools like AutoIt or Robot Framework.

This list will help you navigate key concepts around Selenium without touching on any coding-related aspects.