Selenium 300 Q&A

## Q1. What is Selenium?

Selenium is an open-source tool to automate web browsers.

## Q2. What are Selenium components?

IDE, WebDriver, and Grid (RC is old).

## Q3. What is Selenium IDE?

A browser extension that records and replays tests.

## Q4. What is Selenium WebDriver?

It lets code control web browsers directly.

## Q5. What is Selenium Grid?

It runs tests in parallel across browsers and machines.

## Q6. Why use Selenium?

It is free, supports many languages, browsers, and OS.

## Q7. Which languages does Selenium support?

Java, Python, C#, JavaScript, Ruby, Kotlin.

## Q8. Which browsers are supported by Selenium?

Chrome, Firefox, Safari, Edge, Opera.

## Q9. Which OS are supported by Selenium?

Windows, macOS, Linux.

## Q10. Can Selenium automate desktop apps?

No, Selenium only works with web apps.

## Q11. What is a locator in Selenium?

A way to find an element on a webpage.

## Q12. What are types of locators?

ID, Name, Class, Tag, LinkText, CSS, XPath.

## Q13. Which locator is fastest?

ID, if it is unique.

## Q14. Difference between CSS and XPath?

CSS is faster, XPath is more flexible.

## Q15. What is absolute XPath?

Full path from root, e.g., /html/body/div.

## Q16. What is relative XPath?

Shorter path starting with //.

## Q17. How to find element by text in XPath?

Use //tag[text()='value'].

## Q18. What does contains() do in XPath?

Matches partial text or attributes.

## Q19. How to find multiple attributes in XPath?

Use and/or, e.g., //tag[@id='x' and @name='y'].

## Q20. Which is more stable, CSS or XPath?

CSS is usually faster and simpler.

## Q21. What is WebElement in Selenium?

An object representing a page element.

## Q22. What does click() do?

Clicks the element.

## Q23. What does sendKeys() do?

Types text into an input field.

## Q24. What does clear() do?

Clears text from an input field.

## Q25. What does getText() do?

Gets the visible text of an element.

## Q26. What does getAttribute() do?

Gets the value of an attribute of element.

## Q27. What does isDisplayed() do?

Checks if element is visible.

## Q28. What does isEnabled() do?

Checks if element is enabled for use.

## Q29. What does isSelected() do?

Checks if checkbox/radio is selected.

## Q30. What is difference between findElement() and findElements()?

findElement finds one, findElements finds many.

## Q31. What does getSize() return?

Height and width of element.

## Q32. What does getLocation() return?

X and Y position of element.

## Q33. What does getRect() return?

Location and size in one object.

## Q34. How to take screenshot of element?

Use getScreenshotAs().

## Q35. What are waits in Selenium?

They make driver wait for elements.

## Q36. What is implicit wait?

Default wait time for elements.

## Q37. What is explicit wait?

Wait for a certain condition before continuing.

## Q38. What is fluent wait?

Wait with custom polling interval.

## Q39. Why not use Thread.sleep()?

It is fixed and slows tests, better use waits.

## Q40. What is WebDriverWait?

A class for explicit waits.

## Q41. What are ExpectedConditions?

Predefined conditions like visibility, clickable.

## Q42. What is a window handle?

A unique ID for each browser window.

## Q43. How to switch to another window?

Use driver.switchTo().window(handle).

## Q44. How to switch to a frame?

Use driver.switchTo().frame().

## Q45. How to go back to main page from frame?

Use driver.switchTo().defaultContent().

## Q46. How to handle alerts?

Use driver.switchTo().alert().

## Q47. What does alert.accept() do?

Clicks OK on alert.

## Q48. What does alert.dismiss() do?

Clicks Cancel on alert.

## Q49. How to get alert text?

Use alert.getText().

## Q50. How to type into alert prompt?

Use alert.sendKeys().

## Q51. What is Select class used for?

To handle dropdowns.

## Q52. How to select option by index?

selectByIndex().

## Q53. How to select option by value?

selectByValue().

## Q54. How to select option by visible text?

selectByVisibleText().

## Q55. How to deselect option?

Use deselect methods.

## Q56. What is Actions class?

For advanced mouse and keyboard actions.

## Q57. How to do double click?

Use actions.doubleClick().perform().

## Q58. How to do right click?

Use actions.contextClick().perform().

## Q59. How to hover mouse?

Use actions.moveToElement().perform().

## Q60. How to drag and drop?

Use actions.dragAndDrop().perform().

## Q61. What is JavaScriptExecutor?

Runs JavaScript code in browser.

## Q62. How to scroll page?

Use js.executeScript('window.scrollBy(...)').

## Q63. How to get title of page?

Use driver.getTitle().

## Q64. How to get current URL?

Use driver.getCurrentUrl().

## Q65. How to get page source?

Use driver.getPageSource().

## Q66. How to navigate back?

Use driver.navigate().back().

## Q67. How to navigate forward?

Use driver.navigate().forward().

## Q68. How to refresh page?

Use driver.navigate().refresh().

## Q69. What is POM in Selenium?

Page Object Model, each page has a class.

## Q70. What is PageFactory?

Helper to create POM objects easily.

## Q71. What is @FindBy annotation?

Used to locate elements in PageFactory.

## Q72. What is a base class?

Parent class with setup and teardown.

## Q73. Why use config.properties file?

To store reusable settings.

## Q74. What is data-driven testing?

Testing with external data like Excel.

## Q75. What is constants class?

Stores common fixed values.

## Q76. What is TestNG?

A testing framework for Selenium.

## Q77. Difference between TestNG and JUnit?

TestNG has more features, JUnit is simpler.

## Q78. What is testng.xml?

File that controls test execution.

## Q79. What are groups in TestNG?

They allow running certain test sets.

## Q80. What is DataProvider in TestNG?

It provides multiple sets of input data.

## Q81. What is retry analyzer?

It reruns failed tests automatically.

## Q82. How to run tests in parallel?

Set parallel in testng.xml.

## Q83. What is Selenium Grid hub?

The central server in Grid.

## Q84. What is Selenium Grid node?

Machines connected to hub for testing.

## Q85. What is RemoteWebDriver?

Runs Selenium tests on remote machine.

## Q86. What is Docker-Selenium?

Run Selenium Grid inside Docker containers.

## Q87. What is BrowserStack?

A cloud platform for Selenium testing.

## Q88. What is Sauce Labs?

Another cloud Selenium platform.

## Q89. Why use cloud grids?

They provide real browsers without setup.

## Q90. What is Extent Report?

Library for HTML test reports.

## Q91. What is Allure Report?

Another reporting tool with charts.

## Q92. What is Log4j?

Logging framework for Java tests.

## Q93. Difference between reports and logs?

Reports summarize tests, logs show details.

## Q94. How to attach screenshot in report?

Take screenshot and add to report file.

## Q95. What are listeners in TestNG?

Hooks for test events like pass/fail.

## Q96. What is NoSuchElementException?

Thrown when element not found.

## Q97. What is StaleElementException?

When element is outdated in DOM.

## Q98. What is TimeoutException?

When wait time is exceeded.

## Q99. How to run browser headless?

Set ChromeOptions/FirefoxOptions headless.

## Q100. Difference between driver.quit() and driver.close()?

quit closes all windows, close closes one.

## Q101. What is a test framework?

A setup that manages test scripts, reports, and execution.

## Q102. Why use frameworks with Selenium?

They make tests reusable, structured, and easy to maintain.

## Q103. What is hybrid framework?

A mix of data-driven and keyword-driven approaches.

## Q104. What is keyword-driven framework?

Uses keywords for test steps stored in external files.

## Q105. What is data-driven framework?

Runs tests with different inputs from files like Excel/CSV.

## Q106. What is modular framework?

Divides app into modules and tests separately.

## Q107. What is Page Object Model (POM)?

Each page is a class with locators and methods.

## Q108. Why use POM?

It makes tests easier to read and maintain.

## Q109. What is PageFactory?

Helper to initialize elements in POM.

## Q110. What is @FindBy?

Annotation to locate elements in PageFactory.

## Q111. What is a base class?

Parent class with setup and teardown code.

## Q112. What are utility classes?

Helpers for common functions like waits or screenshots.

## Q113. What is config.properties file?

Stores environment and browser settings.

## Q114. What is constants class?

Keeps fixed values like URLs, timeouts.

## Q115. What is parameterization?

Running tests with multiple data sets.

## Q116. How to read data from Excel in Selenium?

Use Apache POI library.

## Q117. How to read data from CSV in Selenium?

Use OpenCSV or FileReader.

## Q118. How to read data from JSON in Selenium?

Use JSON libraries like Jackson or Gson.

## Q119. How to connect DB with Selenium?

Use JDBC in Java for queries.

## Q120. Why store test data outside code?

So tests are flexible without code change.

## Q121. What is TestNG?

A framework that supports grouping, parallel tests, and reports.

## Q122. What is JUnit?

Another Java testing framework, simpler than TestNG.

## Q123. Difference between TestNG and JUnit?

TestNG has more features like groups, parallel runs.

## Q124. What are TestNG annotations?

@Test, @BeforeMethod, @AfterMethod, @BeforeClass, @AfterClass.

## Q125. What is testng.xml?

A config file to define test suites and order.

## Q126. What is a test suite?

A collection of test cases.

## Q127. What is grouping in TestNG?

Running selected groups of tests.

## Q128. What is priority in TestNG?

Controls order of test execution.

## Q129. What is dependency in TestNG?

One test depends on another's result.

## Q130. What is DataProvider in TestNG?

Feeds multiple input sets to tests.

## Q131. What is a listener in TestNG?

Monitors events like test start/fail.

## Q132. What is retry analyzer in TestNG?

Reruns failed tests automatically.

## Q133. How to run tests in parallel in TestNG?

Set parallel attribute in testng.xml.

## Q134. How to generate reports in TestNG?

TestNG creates default HTML reports.

## Q135. What is JUnit annotation?

@Test, @Before, @After, @BeforeClass, @AfterClass.

## Q136. What is @BeforeClass in JUnit?

Runs once before all tests in class.

## Q137. What is @AfterClass in JUnit?

Runs once after all tests in class.

## Q138. What is @Before in JUnit?

Runs before each test.

## Q139. What is @After in JUnit?

Runs after each test.

## Q140. Which is better: TestNG or JUnit?

TestNG for advanced, JUnit for simple projects.

## Q141. What is Selenium Grid?

It runs tests in parallel on many browsers/machines.

## Q142. What is hub in Selenium Grid?

The central controller that manages tests.

## Q143. What is node in Selenium Grid?

Machine where tests actually run.

## Q144. What is RemoteWebDriver?

Lets you run tests on a remote machine.

## Q145. What is desired capabilities?

Key-value pairs to set browser, OS, version.

## Q146. What are Grid setups?

Standalone, Hub-Node, and Distributed.

## Q147. What is Docker-Selenium?

Running Selenium Grid inside Docker containers.

## Q148. Why use Docker-Selenium?

Easy setup, scalable, portable.

## Q149. What is BrowserStack?

Cloud Selenium testing platform.

## Q150. What is Sauce Labs?

Another cloud testing platform for browsers.

## Q151. What is advantage of cloud grids?

Real devices, no local setup, faster runs.

## Q152. What is Extent Report?

A tool to create HTML reports with details.

## Q153. What is Allure Report?

Another reporting library for nice reports.

## Q154. What is the difference between reports and logs?

Reports show summary, logs show step details.

## Q155. How to attach screenshot in report?

Capture screenshot and add file path.

## Q156. What is Log4j?

Logging framework for Java Selenium projects.

## Q157. What is SLF4J?

Another Java logging framework.

## Q158. Why use logging in Selenium?

To trace and debug test execution.

## Q159. What are listeners in Selenium?

Hooks to capture test events.

## Q160. What is a test failure screenshot?

Screenshot captured when test fails.

## Q161. What is NoSuchElementException?

Element is not found on the page.

## Q162. What is TimeoutException?

Waited too long, condition not met.

## Q163. What is ElementNotVisibleException?

Element is hidden, cannot be used.

## Q164. What is StaleElementException?

Element reference is outdated in DOM.

## Q165. What is WebDriverException?

Generic exception when driver fails.

## Q166. What is IllegalStateException?

Occurs when driver path is wrong.

## Q167. What is SessionNotFoundException?

Session ended before command finished.

## Q168. What is ElementClickInterceptedException?

Another element blocked the click.

## Q169. What is FileNotFoundException?

External file not found for test data.

## Q170. What is NullPointerException?

Null object reference used in test.

## Q171. What is headless browser testing?

Run browser without showing UI.

## Q172. Why use headless mode?

Faster and useful for CI servers.

## Q173. How to enable Chrome headless?

Use ChromeOptions.addArguments('headless').

## Q174. How to enable Firefox headless?

Use FirefoxOptions.setHeadless(true).

## Q175. What are options classes?

ChromeOptions, FirefoxOptions to set browser prefs.

## Q176. How to maximize window in Selenium?

Use driver.manage().window().maximize().

## Q177. How to delete cookies in Selenium?

Use driver.manage().deleteAllCookies().

## Q178. How to add cookies in Selenium?

Use driver.manage().addCookie().

## Q179. How to get all cookies?

Use driver.manage().getCookies().

## Q180. How to handle SSL certificates?

Use DesiredCapabilities or Options to accept.

## Q181. What is WebDriverManager?

Tool to auto-manage driver binaries.

## Q182. What is Maven?

Build tool to manage dependencies.

## Q183. What is pom.xml?

Config file in Maven projects.

## Q184. Difference between src/test/java and src/main/java?

src/test/java has tests, src/main/java has code.

## Q185. How to read properties file?

Use Properties class in Java.

## Q186. Why use constants in Selenium?

To avoid magic numbers and repeated values.

## Q187. How to set timeouts in Selenium?

Use implicitlyWait, pageLoadTimeout, setScriptTimeout.

## Q188. What is pageLoadTimeout?

Max time to wait for page to load.

## Q189. What is setScriptTimeout?

Max time for async scripts to run.

## Q190. How to run specific test from suite?

Use testng.xml to select test.

## Q191. How to skip test in TestNG?

Use throw new SkipException() or @Ignore.

## Q192. How to disable test in JUnit?

Use @Ignore annotation.

## Q193. What is dependsOnMethods in TestNG?

Runs only if other test passes.

## Q194. What is alwaysRun in TestNG?

Runs test even if dependency fails.

## Q195. How to create parameter in TestNG?

Define in testng.xml and use @Parameters.

## Q196. What is parallel attribute in testng.xml?

Used for parallel test execution.

## Q197. What is threadPoolSize in TestNG?

Runs same test in parallel threads.

## Q198. What is invocationCount in TestNG?

Runs same test multiple times.

## Q199. What is timeout in TestNG?

Fails test if it runs longer than set time.

## Q200. What is soft assert in TestNG?

Does not stop test when fail, collects all fails.

## Q201. What is WebDriverManager?

A tool that auto-downloads and manages browser drivers.

## Q202. Why use WebDriverManager?

It removes need to set driver path manually.

## Q203. What is Maven in Selenium?

A build tool to manage Selenium dependencies.

## Q204. What is pom.xml?

Maven config file listing dependencies and plugins.

## Q205. What is Gradle?

Another build tool for Java projects.

## Q206. Difference between Maven and Gradle?

Maven uses XML, Gradle uses Groovy/Kotlin scripts.

## Q207. What is src/test/java folder?

It contains test classes.

## Q208. What is src/main/java folder?

It contains application code.

## Q209. Why keep test data separate?

To change inputs without changing code.

## Q210. What types of test data sources are used?

Excel, CSV, JSON, Properties files.

## Q211. How to read Excel in Selenium?

Using Apache POI library.

## Q212. How to read CSV in Selenium?

Using OpenCSV or FileReader.

## Q213. How to read JSON in Selenium?

Using Jackson or Gson libraries.

## Q214. How to read Properties file?

Using Properties class in Java.

## Q215. How to set implicit wait?

driver.manage().timeouts().implicitlyWait().

## Q216. How to set explicit wait?

Use WebDriverWait with ExpectedConditions.

## Q217. How to set page load timeout?

driver.manage().timeouts().pageLoadTimeout().

## Q218. How to set script timeout?

driver.manage().timeouts().setScriptTimeout().

## Q219. Difference between implicit and explicit wait?

Implicit applies globally, explicit applies to condition.

## Q220. Which wait is better?

Explicit wait is more reliable.

## Q221. How to navigate back in Selenium?

driver.navigate().back().

## Q222. How to navigate forward in Selenium?

driver.navigate().forward().

## Q223. How to refresh page in Selenium?

driver.navigate().refresh().

## Q224. How to open URL in Selenium?

driver.get("url") or driver.navigate().to("url").

## Q225. How to get page title?

driver.getTitle().

## Q226. How to get current URL?

driver.getCurrentUrl().

## Q227. How to get page source?

driver.getPageSource().

## Q228. What are cookies in Selenium?

Small data stored in browser, can be added or removed.

## Q229. How to add cookie in Selenium?

driver.manage().addCookie().

## Q230. How to get all cookies?

driver.manage().getCookies().

## Q231. How to delete all cookies?

driver.manage().deleteAllCookies().

## Q232. What is Actions class?

Handles mouse and keyboard advanced actions.

## Q233. How to hover mouse in Selenium?

actions.moveToElement().perform().

## Q234. How to double click in Selenium?

actions.doubleClick().perform().

## Q235. How to right click in Selenium?

actions.contextClick().perform().

## Q236. How to drag and drop?

actions.dragAndDrop().perform().

## Q237. What is Keys class?

Contains special keys like TAB, ENTER.

## Q238. What is Robot class?

Helps with keyboard/mouse outside WebDriver control.

## Q239. How to upload file in Selenium?

Use sendKeys() with file path or Robot class.

## Q240. How to download file in Selenium?

Set browser preferences with Options class.

## Q241. How to check element is displayed?

element.isDisplayed().

## Q242. How to check element is enabled?

element.isEnabled().

## Q243. How to check element is selected?

element.isSelected().

## Q244. How to get element size?

element.getSize().

## Q245. How to get element location?

element.getLocation().

## Q246. How to get element rectangle?

element.getRect().

## Q247. How to check broken links?

Get href and check HTTP status with HttpURLConnection.

## Q248. How to handle hidden elements?

Use JavaScriptExecutor to click or send text.

## Q249. How to handle dynamic elements?

Use contains() or starts-with() in XPath.

## Q250. How to handle tooltips?

Use getAttribute("title") or Actions hover.

## Q251. How to handle modal dialogs?

Switch using driver.switchTo().alert() or frame().

## Q252. How to handle date pickers?

Use sendKeys() or JavaScriptExecutor.

## Q253. How to handle sliders?

Use Actions dragAndDropBy().

## Q254. How to handle infinite scroll?

Loop with JS scroll until no new data.

## Q255. How to handle iframes?

Use driver.switchTo().frame().

## Q256. How to return from iframe?

Use driver.switchTo().defaultContent().

## Q257. How to emulate mobile view?

Use ChromeOptions setExperimentalOption('mobileEmulation').

## Q258. How to test network speed?

Use Chrome DevTools Protocol in Selenium 4.

## Q259. How to capture console logs?

Use driver.manage().logs().get('browser').

## Q260. How to record video of tests?

Use third-party tools or cloud providers.

## Q261. How to test broken images?

Check img src with HTTP status.

## Q262. How to check page load performance?

Use Navigation Timing API with JSExecutor.

## Q263. How to take full page screenshot?

Use AShot library or Selenium 4 features.

## Q264. How to highlight element?

Use JSExecutor to change CSS style temporarily.

## Q265. How to simulate slow network?

Use Chrome DevTools in Selenium 4.

## Q266. How to handle multiple tabs?

Use driver.getWindowHandles() and switchTo().

## Q267. How to handle multiple windows?

Same as tabs with handles.

## Q268. How to close current window?

driver.close().

## Q269. How to quit browser?

driver.quit().

## Q270. What is headless mode?

Browser runs without opening UI.

## Q271. Why use headless mode?

It is faster and good for CI/CD.

## Q272. How to set Chrome headless?

chromeOptions.addArguments('headless').

## Q273. How to set Firefox headless?

firefoxOptions.setHeadless(true).

## Q274. What is Jenkins integration with Selenium?

Run tests automatically in Jenkins jobs.

## Q275. How to publish reports in Jenkins?

Use HTML publisher plugin.

## Q276. How to run Selenium in Docker?

Use Docker-Selenium images with hub and nodes.

## Q277. What are cloud Selenium providers?

BrowserStack, Sauce Labs, LambdaTest.

## Q278. Why use cloud providers?

Real devices, many browsers, no local setup.

## Q279. How to run tests in parallel?

Use TestNG or JUnit parallel execution.

## Q280. How to handle flaky tests?

Use explicit waits, retries, and stable locators.

## Q281. How to debug Selenium tests?

Add logs, screenshots, and run step by step.

## Q282. How to capture video in CI?

Use tools like Selenium Grid extras or cloud.

## Q283. How to rerun failed tests?

Use TestNG retry analyzer or rerun plugins.

## Q284. What is CI/CD pipeline?

Automated flow for build, test, deploy.

## Q285. How is Selenium used in CI/CD?

Tests run automatically after builds.

## Q286. What is the challenge in CI/CD?

Flaky tests and environment mismatch.

## Q287. How to improve stability?

Use waits, page objects, and retries.

## Q288. What is the future of Selenium?

More support for CDP and parallel execution.

## Q289. What is Selenium 4?

Latest version with new features like relative locators.

## Q290. What are relative locators?

Find elements near others using above(), below().

## Q291. What is Chrome DevTools Protocol in Selenium?

Allows network and console access.

## Q292. What are new window/tab methods?

driver.switchTo().newWindow().

## Q293. What is Selenium Manager?

Auto-manages driver binaries in Selenium 4.

## Q294. What is BiDi API?

Bi-directional API for better debugging.

## Q295. What is Shadow DOM testing?

Use JSExecutor to access shadow root elements.

## Q296. What are challenges in Selenium testing?

Flaky tests, waits, dynamic elements.

## Q297. How to handle CAPTCHA?

Use manual input or external service.

## Q298. How to handle OTP testing?

Use test numbers or API mocks.

## Q299. What are Selenium best practices?

Use POM, waits, logs, retries, reports.

## Q300. What are Selenium alternatives?

Cypress, Playwright, Puppeteer.