**Automation Interview questions:**

**1. What is Selenium? What are its components?**

**Answer: Selenium is an open-source automation testing tool for web applications. It supports multiple programming languages (e.g., Java, Python, C#) and browsers.**

**Components of Selenium:**

* **Selenium IDE: Record and playback tool.**
* **Selenium WebDriver: Automates browser actions using programming.**
* **Selenium Grid: Runs tests on multiple machines and browsers in parallel.**

**2. What is the difference between findElement() and findElements()?**

**Answer:**

* **findElement(): Returns the first web element matching the locator. Throws NoSuchElementException if no element is found.**
* **findElements(): Returns a list of all web elements matching the locator. Returns an empty list if no elements are found.**

**3. What are locators in Selenium?**

**Answer: Locators are used to identify elements on a webpage. Types of locators:**

* **ID**
* **Name**
* **Class Name**
* **XPath**
* **CSS Selector**
* **Tag Name**
* **Link Text**
* **Partial Link Text**

**4. What is the difference between Absolute XPath and Relative XPath?**

**Answer:**

* **Absolute XPath: Specifies the complete path starting from the root node. Example: /html/body/div[1]/div[2].**
* **Relative XPath: Begins from the current node or a specific node. Example: //div[@class='example'].**

**5. What are the advantages of Selenium WebDriver over Selenium IDE?**

**Answer:**

* **Supports multiple programming languages.**
* **Can handle dynamic web elements.**
* **Enables advanced actions like handling alerts, frames, and file uploads.**
* **Allows integration with frameworks like TestNG, Maven, etc.**

**Intermediate Questions**

**6. How do you handle dynamic elements in Selenium?**

**Answer: Dynamic elements have IDs or attributes that change with every page load. To handle them:**

* **Use dynamic XPath or CSS Selectors:**

**java**

**CopyEdit**

**driver.findElement(By.xpath("//button[contains(text(),'Login')]"));**

**driver.findElement(By.xpath("//input[starts-with(@id,'user\_')]"));**

* **Use regular expressions or strategies like contains(), starts-with() in XPath.**

**7. What is TestNG, and why is it used in Selenium?**

**Answer: TestNG (Testing Next Generation) is a testing framework in Java used with Selenium for:**

* **Defining test methods using annotations like @Test, @BeforeMethod, etc.**
* **Running tests in parallel.**
* **Generating detailed reports.**
* **Managing dependencies and grouping tests.**

**8. How do you handle alerts in Selenium?**

**Answer: Alerts are handled using the Alert interface:**

**java**

**CopyEdit**

**Alert alert = driver.switchTo().alert();**

**alert.accept(); // To click OK**

**alert.dismiss(); // To click Cancel**

**String alertText = alert.getText(); // To get the alert message**

**9. How do you handle a dropdown in Selenium?**

**Answer: Use the Select class to handle dropdowns:**

**java**

**CopyEdit**

**Select dropdown = new Select(driver.findElement(By.id("dropdownId")));**

**dropdown.selectByVisibleText("Option1");**

**dropdown.selectByValue("value1");**

**dropdown.selectByIndex(2);**

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

**Answer: POM is a design pattern in Selenium where each web page is represented as a class. The class contains:**

**Each page in the application is required to have its own corresponding java class in the project framework**

**Web UI elements on a page,will be stored as object in the corresponding java classes in the framework.**

* **Web elements as variables.**
* **Methods for performing actions on the web page.**

**Advantages of POM:**

* **Improves readability and maintainability.**
* **Reusability of code.**

**Advanced Questions**

**11. How do you perform cross-browser testing in Selenium?**

**Answer: Use WebDriver to specify the browser driver for testing:**

**java**

**CopyEdit**

**WebDriver driver;**

**if (browser.equals("chrome")) {**

**System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");**

**driver = new ChromeDriver();**

**} else if (browser.equals("firefox")) {**

**System.setProperty("webdriver.gecko.driver", "path/to/geckodriver");**

**driver = new FirefoxDriver();**

**}**

**You can parameterize the browser name using TestNG or Maven.**

**12. How do you handle a file upload in Selenium?**

**Answer: For file uploads, use sendKeys to specify the file path:**

**java**

**CopyEdit**

**WebElement upload = driver.findElement(By.id("fileUpload"));**

**upload.sendKeys("C:\\path\\to\\file.txt");**

**13. What are Explicit Waits, Implicit Waits, and Fluent Waits?**

**Answer:**

* **Implicit Wait: Sets a global wait time for locating elements.**

**java**

**CopyEdit**

**driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);**

* **Explicit Wait: Waits for a specific condition to be met.**

**java**

**CopyEdit**

**WebDriverWait wait = new WebDriverWait(driver, 10);**

**wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("elementId")));**

* **Fluent Wait: Specifies polling intervals while waiting for a condition.**

**java**

**CopyEdit**

**Wait<WebDriver> fluentWait = new FluentWait<>(driver)**

**.withTimeout(Duration.ofSeconds(10))**

**.pollingEvery(Duration.ofSeconds(2))**

**.ignoring(NoSuchElementException.class);**

**fluentWait.until(ExpectedConditions.elementToBeClickable(By.id("elementId")));**

**14. How do you capture a screenshot in Selenium?**

**Answer: Use TakesScreenshot interface:**

**java**

**CopyEdit**

**File screenshot = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);**

**FileUtils.copyFile(screenshot, new File("path/to/save.png"));**

**15. What is Selenium Grid, and how do you use it?**

**Answer: Selenium Grid is used for running tests on multiple machines and browsers in parallel.**

**Steps to Use Selenium Grid:**

1. **Start the hub: java -jar selenium-server-standalone.jar -role hub**
2. **Start the nodes: java -jar selenium-server-standalone.jar -role node**
3. **Configure tests to point to the hub:**

**java**

**CopyEdit**

**DesiredCapabilities capabilities = new DesiredCapabilities();**

**capabilities.setBrowserName("chrome");**

**WebDriver driver = new RemoteWebDriver(new URL("http://localhost:4444/wd/hub"), capabilities);**

**Scenario-Based Questions**

**16. How do you validate broken links on a webpage?**

**Answer: Use HttpURLConnection in Java to check the HTTP response code:**

**java**

**CopyEdit**

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

**for (WebElement link : links) {**

**String url = link.getAttribute("href");**

**HttpURLConnection conn = (HttpURLConnection) new URL(url).openConnection();**

**conn.setRequestMethod("HEAD");**

**int responseCode = conn.getResponseCode();**

**if (responseCode != 200) {**

**System.out.println("Broken Link: " + url);**

**}**

**}**

**17. How do you scroll a webpage using Selenium?**

**Answer: Use JavaScriptExecutor for scrolling:**

**java**

**CopyEdit**

**JavascriptExecutor js = (JavascriptExecutor) driver;**

**js.executeScript("window.scrollBy(0,500)"); // Scroll down**

**18. How do you debug a failing Selenium test?**

**Answer:**

1. **Check the error message and stack trace.**
2. **Validate the locators used (e.g., XPath, CSS).**
3. **Use debugging tools like breakpoints in the IDE.**
4. **Capture screenshots for failed tests.**
5. **Add explicit waits for synchronization issues.**

**Would you like me to expand on any specific question or provide a coding example for a particular scenario?**

**4o**

**Top of Form**

**O**

**Bottom of Form**

**ChatGPT can mak**

**Automation framework questions and answers:**

1. **What are the different components of your automation framework?**
2. **Difference between data-driven and keyword -driven framework?which is more preffered framework?**
3. **What version control platform you are using?**
4. **What is the command used to commit code , and push code from local to repository?**
5. **Do you execute test locally or do you use the CI/CD pipeline?Explain the process?**
6. **How do you generate test result?**
7. **How do you take screenshot?**
8. **What are the coding standards you follow for test scripting?(Camel case standards,adding comments,avoiding sleep commands,code review,vesion controlling)**
9. **What is POM?**
10. **How do you pass test data to selenium tests?**
11. **What are the improvements you have made to enhance your test automation framework?**
12. **Without select class can we automate dropdown?**
13. **What type of exception you have seen dring automation?**
14. **What is Xpath?**
15. **When do you use Explicit wait?**

Explicit wait in Selenium is a synchronization mechanism that allows the WebDriver to wait for a specific condition to occur before proceeding with the next step in the code.

There are 2 types of Explicit wait types of Explicit wait

* **Webdriver wait**-Wait for the condition or element to be appear on the web page default polling time will be half sec.ear on the web page default polling time will be half sec.
* **Fluent wait:** Wait for the condition it a advanced method of wait where we can customized the polling time.

1. **How do you handle dynamic web elements?**
2. **How do you handle multiple windows?**
3. **How will you test broken links in webpage?**
4. **How will you test the HTML page source using selenium?**
5. **How will you scroll down to a particular web elements in a webpage?**
6. **What are the challenges you have faced during automation .how did you overcome it?**
7. **Can we run particular test from a class file?if so how?**
8. **Explain the structure of Testng.xml file?**
9. **What are the TestNG annotations used in your frame work?**
10. **What is the purpose of using annotations?**
11. **Have you done parallel execution ?how many browsers you are running test scripts?**
12. **Use of dataprovider annotation?**
13. **What is assertion?**
14. **Can final keyword be used with static keyword?**
15. **Difference between method overloading and overidding?**
16. **Difference between abstract class and interface?**
17. **Use of static keyword?**
18. **Use of finally block?**
19. **What is Hashmap and where have you used Hashmap in your selenium tests?**
20. **Is Arraylist better that array?**
21. **How are objects allocated in memory?**
22. **Why is main method void and static?**
23. **Can we override static methods?**
24. **Can you explain about Xpath using contains?**
25. **What is BDD?**
26. **How you will drag and drop in selenium?**
27. **What is the use of drivers in selenium?**
28. **What is page factory why we use it?**
29. **What are the new fetures in selenium 4?**
30. **When we use robot class in selenium?**
31. **What is fluent wait?**
32. **HTTP methods available in crud operations with example?**
33. **What do you mean by Bearer token?**
34. **What are the different status code and expalinabout 429 status code?**
35. **Why srting immutable in java?**
36. **Difference between function overloading and function overriding?**
37. **Can we override the main method and why?**
38. **What is abstract class?**
39. **What is super keyword?**
40. **Difference between final anf finally?**
41. **What is constructor?**
42. **What is difference between list and as set?**
43. **What art the access modifiers in java?**
44. **What is collection?**
45. **WAP to print occurnece of elements and count of elements from given in a array?**
46. **What challenge did you face while doing automation?**
47. **How to read data from excel?**
48. **What are the differenct tesng annotations**
49. **How does microservices are build**
50. **How Jenkins pipeline is built?**
51. **What is delete bug cycle**
52. **What is SDLC**

**Programming question asked to automation is**

1. **Write a JAVA program to display Anagrams**
   1. **String S1=automates**
   2. **String S2=Tomatesau**
   3. **Iterate both the string and check the characters are same**
2. **Write java program to print characters occurring once in a string?**
   1. **Input -ajayautomates**
   2. **Output-jyuomes**

**3.Write a program for reverse sentence(Space should not change)**

**4.Write down the code for scroll a page?**

**5.A list of number is available in a table how to find the duplicate number?**

**6.Write a program to find duplicate words in string?**

**what does testng frame work contains**

**ChatGPT said:**

ChatGPT

TestNG is a popular testing framework for Java that provides a rich set of functionalities to support testing of applications. Here's what the TestNG framework typically contains:

**1. Annotations**

TestNG annotations help control how tests are executed. They are used to define setup, execution, and teardown steps. Common annotations include:

* @Test: Marks a method as a test case.
* @BeforeSuite and @AfterSuite: Execute methods before or after the entire test suite.
* @BeforeTest and @AfterTest: Execute methods before or after a specific <test> tag in the XML configuration.
* @BeforeClass and @AfterClass: Execute methods before or after a test class.
* @BeforeMethod and @AfterMethod: Execute methods before or after each test method.
* @DataProvider: Provides data to test methods for parameterized testing.
* @Factory: Used to execute test cases multiple times with different sets of data.

**2. TestNG XML Configuration**

* TestNG uses an XML file (testng.xml) for test configuration.
* It allows grouping of tests, parallel execution, setting dependencies, and controlling test execution order.
* Example:

xml

CopyEdit

<suite name="TestSuite">

<test name="Test1">

<classes>

<class name="com.example.TestClass" />

</classes>

</test>

</suite>

**3. Parallel and Sequential Execution**

TestNG supports the execution of test cases in parallel or sequentially, making it efficient for large test suites:

* Parallel execution can be done at method, class, or suite level.
* Controlled using the <suite> tag in the XML file.

**4. Data-Driven Testing**

* TestNG supports parameterized testing using @DataProvider.
* Example:

java

CopyEdit

@DataProvider(name = "testData")

public Object[][] createData() {

return new Object[][] { { "data1" }, { "data2" } };

}

@Test(dataProvider = "testData")

public void testMethod(String data) {

System.out.println("Test data: " + data);

}

**5. Assertions**

* TestNG provides assertion methods to verify expected outcomes in tests.
* Examples:
  + Assert.assertEquals(actual, expected);
  + Assert.assertTrue(condition);
  + Assert.assertFalse(condition);
  + Assert.fail("Forcefully fail the test");

**6. Test Dependencies**

* TestNG allows defining dependencies between tests using the dependsOnMethods and dependsOnGroups attributes in the @Test annotation.
* Example:

java

CopyEdit

@Test

public void loginTest() {

System.out.println("Login successful");

}

@Test(dependsOnMethods = "loginTest")

public void dashboardTest() {

System.out.println("Access dashboard");

}

**7. Test Groups**

* Tests can be grouped together using the groups attribute in the @Test annotation.
* Example:

java

CopyEdit

@Test(groups = "smoke")

public void test1() {

System.out.println("Smoke test");

}

@Test(groups = "regression")

public void test2() {

System.out.println("Regression test");

}

**8. Reporting**

* TestNG automatically generates HTML and XML reports that include the results of the test execution.
* The reports show the number of tests passed, failed, or skipped.

**9. Exception Testing**

* TestNG provides a way to verify if a test case throws a specific exception using the expectedExceptions attribute.
* Example:

java

CopyEdit

@Test(expectedExceptions = ArithmeticException.class)

public void exceptionTest() {

int result = 1 / 0;

}

**10. Test Prioritization**

* TestNG allows setting the order of test execution using the priority attribute in the @Test annotation.
* Example:

java

CopyEdit

@Test(priority = 1)

public void testA() {

System.out.println("Test A");

}

@Test(priority = 2)

public void testB() {

System.out.println("Test B");

}

**11. Skipping Tests**

* You can skip tests programmatically by using throw new SkipException("Reason").
* Example:

java

CopyEdit

@Test

public void skipTest() {

throw new SkipException("Skipping this test case");

}

**12. Parameterized Testing**

* TestNG allows passing parameters to tests via the XML file using the @Parameters annotation.
* Example: XML:

xml

CopyEdit

<parameter name="browser" value="Chrome" />

Test:

java

CopyEdit

@Test

@Parameters("browser")

public void testParameter(String browser) {

System.out.println("Testing on browser: " + browser);

}

**13. Retry Logic**

* TestNG supports retrying failed tests using the IRetryAnalyzer interface.

**14. Custom Listeners**

* TestNG provides listeners to perform actions during test execution. Common listeners include:
  + ITestListener
  + ISuiteListener
  + IAnnotationTransformer
* Example:

java

CopyEdit

public class TestListener implements ITestListener {

@Override

public void onTestStart(ITestResult result) {

System.out.println("Test Started: " + result.getName());

}

}

**15. Integration with Build Tools**

* TestNG integrates seamlessly with tools like **Maven** and **Gradle** for running tests as part of CI/CD pipelines.
* Example: Adding TestNG to Maven:

xml

CopyEdit

<dependency>

<groupId>org.testng</groupId>

<artifactId>testng</artifactId>

<version>7.7.0</version>

<scope>test</scope>

</dependency>

**Summary**

TestNG is a powerful testing framework that offers robust features such as annotations, parallel execution, data-driven testing, reporting, dependency management, and integration with tools. It simplifies writing and managing tests for Java-based applications.

**2.Letxpath key features of different locater element identification?**

**Key Differences**

| **XPath Type** | **What It Targets** |
| --- | --- |
| **Class-Based XPath** | Locates elements using their class attribute. |
| **Text-Based XPath** | Locates elements using their visible text. |
| **Following-Sibling XPath** | Locates sibling elements that appear after a specific element. |
| **Parent Attribute XPath** | Locates the parent element based on its attributes. |
| **Parent Class XPath** | Locates the parent element specifically by its class attribute. |

Certainly, here are some additional **Infosys interview questions for Automation Testing with Selenium and Java**, reflecting recent trends and topics from 2024:

**Core Java Questions**

**1. Explain the difference between final, finally, and finalize in Java.**

**Answer:**

* **final**: A keyword used to declare constants, prevent method overriding, and inheritance.
* **finally**: A block in exception handling that executes regardless of whether an exception is handled or not.
* **finalize**: A method called by the garbage collector before an object is destroyed.

*Source:* [*Infosys Selenium Automation Test Engineer Interview Questions*](https://www.glassdoor.com/Interview/Infosys-Selenium-Automation-Test-Engineer-Interview-Questions-EI_IE7927.0%2C7_KO8%2C41.htm)

**2. How do you achieve multiple inheritance in Java?**

**Answer:**

Java does not support multiple inheritance with classes to avoid complexity and simplify the design. However, it can be achieved using interfaces:

java

CopyEdit

interface InterfaceA {

void methodA();

}

interface InterfaceB {

void methodB();

}

class MyClass implements InterfaceA, InterfaceB {

public void methodA() {

// Implementation

}

public void methodB() {

// Implementation

}

}

*Source:* [*Infosys Selenium Automation Tester Interview Questions*](https://www.ambitionbox.com/interviews/infosys-interview-questions/selenium-automation-tester)

**3. Write a Java program to check if a number is a prime number.**

**Answer:**

java

CopyEdit

public class PrimeCheck {

public static boolean isPrime(int num) {

if (num <= 1) return false;

for (int i = 2; i <= Math.sqrt(num); i++) {

if (num % i == 0) return false;

}

return true;

}

public static void main(String[] args) {

int number = 29;

if (isPrime(number)) {

System.out.println(number + " is a prime number.");

} else {

System.out.println(number + " is not a prime number.");

}

}

}

*Source:* [*Infosys Selenium Automation Test Engineer Interview Questions*](https://www.glassdoor.com/Interview/Infosys-Selenium-Automation-Test-Engineer-Interview-Questions-EI_IE7927.0%2C7_KO8%2C41.htm)

**Selenium and Automation Testing Questions**

**4. How do you handle dynamic web elements in Selenium?**

**Answer:**

Dynamic web elements can be handled using various strategies:

* **Dynamic XPath**: Using functions like contains(), starts-with(), or ends-with() to create flexible XPath expressions.

java

CopyEdit

WebElement element = driver.findElement(By.xpath("//button[contains(text(),'Submit')]"));

* **Explicit Waits**: Waiting for elements to be present or visible before interacting.

java

CopyEdit

WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));

WebElement element = wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("dynamicElement")));

*Source:* [*Infosys Test Automation Engineer Interview Questions*](https://www.glassdoor.com/Interview/Infosys-Test-Automation-Engineer-Interview-Questions-EI_IE7927.0%2C7_KO8%2C32.htm)

**5. What is the Page Factory in Selenium, and how does it differ from the Page Object Model (POM)?**

**Answer:**

* **Page Object Model (POM)**: A design pattern that creates an object repository for web elements. Each web page is represented as a class with methods that interact with the page.
* **Page Factory**: An extension of POM that uses annotations to initialize web elements. It provides the @FindBy annotation and the initElements method for initializing elements.

java

CopyEdit

public class LoginPage {

@FindBy(id = "username")

WebElement usernameField;

@FindBy(id = "password")

WebElement passwordField;

@FindBy(id = "loginButton")

WebElement loginButton;

public LoginPage(WebDriver driver) {

PageFactory.initElements(driver, this);

}

public void login(String username, String password) {

usernameField.sendKeys(username);

passwordField.sendKeys(password);

loginButton.click();

}

}

*Source:* [*Top 90+ Selenium Interview Questions and Answers for 2025*](https://www.simplilearn.com/tutorials/selenium-tutorial/selenium-interview-questions-and-answers)

**6. How do you handle SSL certificate errors in Selenium WebDriver?**

**Answer:**

SSL certificate errors can be handled by configuring the browser to accept insecure certificates:

* **For Chrome:**

java

CopyEdit

ChromeOptions options = new ChromeOptions();

options.setAcceptInsecureCerts(true);

WebDriver driver = new ChromeDriver(options);

* **For Firefox:**

java

CopyEdit

FirefoxOptions options = new FirefoxOptions();

options.setAcceptInsecureCerts(true);

WebDriver driver = new FirefoxDriver(options);

*Source:* [*Top 90+ Selenium Interview Questions and Answers for 2025*](https://www.simplilearn.com/tutorials/selenium-tutorial/selenium-interview-questions-and-answers)

**7. Explain the concept of TestNG Listeners and their use in Selenium testing.**

**Answer:**

TestNG Listeners are interfaces that allow you to modify the default behavior of TestNG. They can be used for:

* Logging
* Reporting
* Taking actions on test start, success, failure, etc.

Common listeners include:

* **IAnnotationTransformer**: Alters annotations at runtime.
* **ITestListener**: Listens to test events like start, success, failure.
* **ISuiteListener**: Listens to events before and after the suite runs.

Implementation example:

java

CopyEdit

public class CustomListener implements ITestListener {

@Override

public void onTestFailure(ITestResult result) {

// Code to take a screenshot on test failure

}

// Implement other methods as needed

}

*Source:* [*Top 90+ Selenium Interview Questions and Answers for 2025*](https://www.simplilearn.com/tutorials/selenium-tutorial/selenium-interview-questions-and-answers)

**8. How do you manage test data in your automation framework?**

**Answer:**

Test data management strategies include:

* **External Files**: Using Excel, CSV, or JSON files to store test data.
* **Databases**: Storing test data in databases and retrieving it during test execution.
* **Property Files**: Using .properties files for configuration and test data.
* **Data Providers**: Utilizing TestNG's @DataProvider to supply test data to test methods.

Example using TestNG @DataProvider:

java

CopyEdit

@DataProvider(name = "loginData")

public Object[][] getData() {

::contentReference[oaicite:0]{index=0}

4o

Top of Form

O

Bottom of Form

ChatGPT can make