1. **What is Selenium?** Selenium is an open-source tool used for automating web browsers. It supports multiple programming languages like Java, Python, C#, etc.
2. **What are the different types of locators in Selenium?**
   * ID
   * Name
   * Class Name
   * Tag Name
   * Link Text
   * Partial Link Text
   * CSS Selector
   * XPath
3. **How do you handle alerts in Selenium?**

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

alert.accept(); // To accept the alert

alert.dismiss(); // To dismiss the alert

1. **What is TestNG?** TestNG is a testing framework inspired by JUnit and NUnit, designed to simplify a broad range of testing needs, from unit testing to integration testing [[1]](https://www.toolsqa.com/testng/testng-interview-questions/).
2. **What are the advantages of TestNG?**
   * Parallel execution of test methods
   * Defining dependencies between test methods
   * Assigning priorities to test methods
   * Grouping test methods into test groups
   * Parameterizing test cases using @Parameters annotation
   * Data-driven testing using @DataProvider annotation
   * Detailed HTML reports[[2]](https://www.softwaretestingmaterial.com/testng-interview-questions/)
3. **What are the annotations available in TestNG?**
   * @BeforeTest
   * @AfterTest
   * @BeforeClass
   * @AfterClass
   * @BeforeMethod
   * @AfterMethod
   * @BeforeSuite
   * @AfterSuite
   * @BeforeGroups
   * @AfterGroups
   * @Test
4. **How do you run a TestNG script?** You can run a TestNG script by right-clicking on the TestNG class and selecting Run As -> TestNG Test [[1]](https://www.toolsqa.com/testng/testng-interview-questions/).
5. **How to create and run testng.xml?** In TestNG, you need to create a testng.xml file to handle multiple test classes. This file allows you to configure test runs, set test dependencies, include or exclude tests, and set priorities [[2]](https://www.softwaretestingmaterial.com/testng-interview-questions/).
6. **How to pass parameters through testng.xml file to a test case?**

<suite name="Suite">

<test name="Test">

<parameter name="paramName" value="paramValue"/>

<classes>

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

</classes>

</test>

</suite>

1. **How do you handle dropdowns in Selenium?**

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

dropdown.selectByVisibleText("OptionText");

dropdown.selectByIndex(1);

dropdown.selectByValue("optionValue");

1. **How do you perform drag and drop actions in Selenium?**

Actions actions = new Actions(driver);

WebElement source = driver.findElement(By.id("sourceID"));

WebElement target = driver.findElement(By.id("targetID"));

actions.dragAndDrop(source, target).perform();

1. **How do you wait for an element to be visible in Selenium?**

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

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

1. **How do you handle cookies in Selenium?**

// Adding a cookie

Cookie cookie = new Cookie("cookieName", "cookieValue");

driver.manage().addCookie(cookie);

// Getting a cookie

Cookie retrievedCookie = driver.manage().getCookieNamed("cookieName");

// Deleting a cookie

driver.manage().deleteCookieNamed("cookieName");

1. **How do you run tests in parallel using TestNG?**

<suite name="Suite" parallel="tests" thread-count="2">

<test name="Test1">

<classes>

<class name="com.example.TestClass1"/>

</classes>

</test>

<test name="Test2">

<classes>

<class name="com.example.TestClass2"/>

</classes>

</test>

</suite>

1. **How do you use @Listeners annotation in TestNG?**

@Listeners(com.example.MyListener.class)

public class TestClass {

@Test

public void testMethod() {

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

}

}

1. **How do you skip a test method in TestNG?**

@Test(enabled = false)

public void testMethod() {

System.out.println("This test method will be skipped");

}

1. **How do you retry failed tests in TestNG?**

public class RetryAnalyzer implements IRetryAnalyzer {

private int count = 0;

private static final int maxTry = 3;

@Override

public boolean retry(ITestResult result) {

if (count < maxTry) {

count++;

return true;

}

return false;

}

}

@Test(retryAnalyzer = RetryAnalyzer.class)

public void testMethod() {

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

}

1. **How do you handle JavaScript alerts in Selenium?**

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

alert.accept(); // To accept the alert

alert.dismiss(); // To dismiss the alert

alert.sendKeys("Text"); // To send text to the alert

1. **What is the Page Object Model (POM) in Selenium?** Page Object Model is a design pattern in Selenium that creates an object repository for web elements. It helps in reducing code duplication and improves test maintenance.
2. **How do you handle file uploads in Selenium?**

WebElement uploadElement = driver.findElement(By.id("uploadID"));

uploadElement.sendKeys("path/to/file");

1. **How do you handle AJAX calls in Selenium?**

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

wait.until(ExpectedConditions.elementToBeClickable(By.id("elementID")));

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

Dynamic elements can be handled using various strategies such as waiting for the element to appear, using XPath or CSS selectors that are less likely to change, or using JavaScript executors.

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

WebElement dynamicElement = wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//div[@class='dynamicElement']")));

1. **How do you handle browser pop-ups in Selenium?**

Browser pop-ups can be handled using the Alert interface.

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

alert.accept(); // To accept the alert

alert.dismiss(); // To dismiss the alert

1. **How do you perform right-click actions in Selenium?**

Actions actions = new Actions(driver);

WebElement element = driver.findElement(By.id("elementID"));

actions.contextClick(element).perform();

1. **How do you handle SSL certificates in Selenium?**

SSL certificates can be handled by configuring the browser capabilities.

DesiredCapabilities capabilities = new DesiredCapabilities();

capabilities.setCapability(CapabilityType.ACCEPT\_SSL\_CERTS, true);

WebDriver driver = new ChromeDriver(capabilities);

1. **How do you use @Parameters annotation in TestNG?**

@Parameters({"paramName"})

@Test

public void testMethod(String paramName) {

System.out.println("Parameter value is: " + paramName);

}

1. **How do you use @DataProvider annotation in TestNG?**

@DataProvider(name = "dataProvider")

public Object[][] dataProviderMethod() {

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

}

@Test(dataProvider = "dataProvider")

public void testMethod(String data) {

System.out.println("Data is: " + data);

}

1. **How do you use @Factory annotation in TestNG?**

@Factory

public Object[] factoryMethod() {

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

}

public class TestClass {

private String data;

public TestClass(String data) {

this.data = data;

}

@Test

public void testMethod() {

System.out.println("Data is: " + data);

}

}

1. **How do you use @Listeners annotation in TestNG?**

@Listeners(com.example.MyListener.class)

public class TestClass {

@Test

public void testMethod() {

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

}

}

1. **How do you handle AJAX calls in Selenium?**

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

wait.until(ExpectedConditions.elementToBeClickable(By.id("elementID")));

1. **How do you handle file downloads in Selenium?** File downloads can be handled by configuring the browser settings.

HashMap<String, Object> chromePrefs = new HashMap<>();

chromePrefs.put("profile.default\_content\_settings.popups", 0);

chromePrefs.put("download.default\_directory", "path/to/download");

ChromeOptions options = new ChromeOptions();

options.setExperimentalOption("prefs", chromePrefs);

WebDriver driver = new ChromeDriver(options);

1. **How do you handle browser navigation in Selenium?**

driver.navigate().to("http://example.com");

driver.navigate().back();

driver.navigate().forward();

driver.navigate().refresh();

1. **How do you handle JavaScript execution in Selenium?**

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("alert('Hello World');");