

Advanced Browser Actions

Command	Use
<code>driver.navigate().to("URL");</code>	Navigate to a URL
<code>driver.navigate().refresh();</code>	Refresh the page
<code>driver.navigate().back();</code>	Navigate back in browser history
<code>driver.navigate().forward();</code>	Navigate forward in browser history
<code>driver.manage().deleteAllCookies();</code>	Delete all browser cookies

Handling JavaScript Alerts

Command	Use
<code>Alert alert = driver.switchTo().alert();</code>	Switch to an alert
<code>alert.accept();</code>	Click OK in alert
<code>alert.dismiss();</code>	Click Cancel in alert
<code>alert.getText();</code>	Get text from alert
<code>alert.sendKeys("text");</code>	Enter text in alert prompt

Handling File Uploads & Downloads

Command	Use
<code>element.sendKeys("C:\\path\\file.txt");</code>	Upload a file using sendKeys
Use AutoIt/Sikuli	Handle file upload popups (Windows)

Handling Mouse & Keyboard Actions (Actions Class)

Command	Use
<code>Actions actions = new Actions(driver);</code>	Create an Actions object
<code>actions.moveToElement(element).perform();</code>	Hover over an element
<code>actions.clickAndHold(element).perform();</code>	Click and hold an element
<code>actions.doubleClick(element).perform();</code>	Double-click an element
<code>actions.contextClick(element).perform();</code>	Right-click on an element
<code>actions.dragAndDrop(source, target).perform();</code>	Drag and drop an element
<code>actions.sendKeys(Keys.ENTER).perform();</code>	Simulate pressing ENTER

Handling Keyboard Events (Robot Class)

Command	Use
<code>Robot robot = new Robot();</code>	Create a Robot object
<code>robot.keyPress(KeyEvent.VK_ENTER);</code>	Press ENTER key
<code>robot.keyRelease(KeyEvent.VK_ENTER);</code>	Release ENTER key
<code>robot.mouseMove(x, y);</code>	Move mouse to coordinates

Handling Screenshots

Command	Use
<code>TakesScreenshot scrShot = (TakesScreenshot) driver;</code>	Convert WebDriver to Screenshot type
<code>File src = scrShot.getScreenshotAs(OutputType.FILE);</code>	Capture screenshot
<code>FileUtils.copyFile(src, new File("path"));</code>	Save screenshot to a file

Handling Web Table Data

Command	Use
<pre>List<WebElement> rows = driver.findElements(By.xpath("//table//tr"));</pre>	Get all rows from a table
<pre>List<WebElement> cols = driver.findElements(By.xpath("//table//th"));</pre>	Get all column headers from a table
<pre>WebElement cellData = driver.findElement(By.xpath("//table//tr[2]//td[3]"));</pre>	Get specific cell data

Verifying Element Status

Command	Use
<pre>element.isDisplayed();</pre>	Check if element is visible
<pre>element.isEnabled();</pre>	Check if element is enabled
<pre>element.isSelected();</pre>	Check if checkbox/radio is selected

Handling Dynamic Elements (Fluent Wait)

Command	Use
<pre>FluentWait<WebDriver> wait = new FluentWait<>(driver);</pre>	Create Fluent Wait object
<pre>wait.withTimeout(Duration.ofSeconds(30));</pre>	Set max wait time
<pre>wait.pollingEvery(Duration.ofSeconds(5));</pre>	Set polling frequency
<pre>wait.ignoring(NoSuchElementException.class);</pre>	Ignore specific exceptions
<pre>wait.until(ExpectedConditions.elementToBeClickable(By.id("button")));</pre>	Wait until element is clickable

Handling Dynamic Web Elements (JavaScript Executor)

Command	Use
<code>JavascriptExecutor js = (JavascriptExecutor) driver;</code>	Create JavaScript Executor object
<code>js.executeScript("window.scrollTo(0,500)");</code>	Scroll down the page
<code>js.executeScript("arguments[0].click();", element);</code>	Click using JavaScript
<code>js.executeScript("document.getElementById('id').value='text'");</code>	Set value using JavaScript

Running Tests in Headless Mode

Command	Use
<code>ChromeOptions options = new ChromeOptions();</code>	Create ChromeOptions object
<code>options.addArguments("--headless");</code>	Enable headless mode
<code>WebDriver driver = new ChromeDriver(options);</code>	Run browser in headless mode

Handling Browser Permissions & Notifications

Command	Use
<code>ChromeOptions options = new ChromeOptions();</code>	Create ChromeOptions object
<code>options.addArguments("--disable-notifications");</code>	Disable browser notifications
<code>WebDriver driver = new ChromeDriver(options);</code>	Run browser with disabled notifications

Handling AJAX Elements

Command

Use

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

Create explicit wait object

```
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("ajaxElement")));
```

Wait for AJAX element to be visible

Handling Browser Console Logs

Command

Use

```
LogEntries logs = driver.manage().logs().get(LogType.BROWSER);
```

Get browser console logs

```
for (LogEntry entry : logs) { System.out.println(entry.getMessage()); }
```

Print console logs