# Selenium Exceptions and Their Handling

## NoSuchElementException

Cause: The element cannot be found using the given locator.

Handling: Verify the locator (XPath, CSS, ID, etc.). Use explicit waits for dynamic elements.

Example:

try {  
 driver.findElement(By.id("nonexistentElement"));  
} catch (NoSuchElementException e) {  
 System.out.println("Element not found!");  
}

## TimeoutException

Cause: A wait exceeds the specified timeout while waiting for a condition.

Handling: Use explicit waits (e.g., WebDriverWait).

Example:

WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));  
try {  
 wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("elementId")));  
} catch (TimeoutException e) {  
 System.out.println("Element not visible within timeout!");  
}

## ElementNotInteractableException

Cause: The element is present in the DOM but not interactable (e.g., hidden).

Handling: Use waits to ensure visibility. Scroll the element into view.

Example:

try {  
 driver.findElement(By.id("hiddenButton")).click();  
} catch (ElementNotInteractableException e) {  
 System.out.println("Element not interactable!");  
}

## StaleElementReferenceException

Cause: The referenced element is no longer attached to the DOM.

Handling: Re-locate the element before interacting.

Example:

try {  
 element.click();  
} catch (StaleElementReferenceException e) {  
 element = driver.findElement(By.id("someElement"));  
 element.click();  
}

## WebDriverException

Cause: A general WebDriver issue (e.g., driver misconfiguration).

Handling: Restart the WebDriver session or debug the error.

Example:

try {  
 driver.get("https://example.com");  
} catch (WebDriverException e) {  
 System.out.println("WebDriver issue: " + e.getMessage());  
}

## NoSuchWindowException

Cause: Switching to a window that doesn't exist.

Handling: Use driver.getWindowHandles() to confirm available windows.

Example:

try {  
 driver.switchTo().window("nonexistentWindow");  
} catch (NoSuchWindowException e) {  
 System.out.println("Window does not exist!");  
}

## NoSuchFrameException

Cause: Switching to a frame that doesn't exist.

Handling: Verify the frame is present before switching.

Example:

try {  
 driver.switchTo().frame("nonexistentFrame");  
} catch (NoSuchFrameException e) {  
 System.out.println("Frame does not exist!");  
}

## SessionNotCreatedException

Cause: Browser or WebDriver version incompatibility.

Handling: Update browser and WebDriver to compatible versions.

Example:

try {  
 WebDriver driver = new ChromeDriver();  
} catch (SessionNotCreatedException e) {  
 System.out.println("Session not created: " + e.getMessage());  
}

## InvalidSelectorException

Cause: Malformed or incorrect locator (XPath or CSS).

Handling: Verify and correct the selector syntax.

Example:

try {  
 driver.findElement(By.xpath("//\*[@id='invalid']"));  
} catch (InvalidSelectorException e) {  
 System.out.println("Invalid selector!");  
}

## MoveTargetOutOfBoundsException

Cause: Performing an action on an element outside the viewport.

Handling: Scroll the element into view.

Example:

try {  
 actions.moveToElement(element).perform();  
} catch (MoveTargetOutOfBoundsException e) {  
 System.out.println("Element out of bounds!");  
}

## UnhandledAlertException

Cause: An unexpected alert is present.

Handling: Switch to the alert and accept/dismiss it.

Example:

try {  
 driver.findElement(By.id("triggerAlert")).click();  
} catch (UnhandledAlertException e) {  
 driver.switchTo().alert().accept();  
}

## ElementClickInterceptedException

Cause: Another element blocks the target element.

Handling: Use JavaScriptExecutor for a direct click.

Example:

try {  
 element.click();  
} catch (ElementClickInterceptedException e) {  
 ((JavascriptExecutor) driver).executeScript("arguments[0].click();", element);  
}

# Best Practices for Handling Exceptions

Use explicit waits to avoid timing-related issues.

Log exceptions for debugging.

Verify locators to ensure accuracy.

Always confirm elements are interactable before interacting.