***Action Class Selenium***

The Actions class in Selenium is a feature-rich class that enables you to perform complex interactions on a web page, such as mouse and keyboard actions, multi-step interactions, and more. It's especially useful for simulating user interactions that involve multiple steps or require precise control over mouse movements, clicks, and keyboard input.

Key features and functionalities of the Actions class in Selenium:

1. **Importing the Actions Class:** To use the Actions class, you need to import it from the **org.openqa.selenium.interactions** package.
2. **Creating an Actions Object:** To perform actions on a web page, you create an instance of the Actions class by providing the WebDriver instance.

javaCopy code

WebDriver driver = new ChromeDriver();

Actions actions = new Actions(driver);

1. **Chaining Actions:** The Actions class supports method chaining, allowing you to build a sequence of actions to be performed in a specific order.

javaCopy code

actions.moveToElement(element).click().sendKeys("Hello").perform();

1. **Common Actions:** Some common actions that can be performed using the Actions class include:
   * **click(WebElement element)**: Clicks an element.
   * **doubleClick(WebElement element)**: Performs a double-click on an element.
   * **contextClick(WebElement element)**: Right-clicks on an element.
   * **moveToElement(WebElement element)**: Moves the mouse to the center of an element.
   * **sendKeys(CharSequence keysToSend)**: Sends keyboard input to the active element.
   * **dragAndDrop(source, target)**: Drags an element from source to target.

**Combining Actions:** You can combine multiple actions to form a sequence, allowing you to perform more complex interactions.

javaCopy code

actions.moveToElement(element).click().sendKeys("Hello").build().perform();

1. **Performing Actions:** The **perform()** method is used to execute the sequence of actions that have been defined. It triggers the actions to be performed on the web page.

javaCopy code

actions.moveToElement(element).click().perform();

1. **Release Method:** After performing an action like dragging an element, you might need to release the mouse button using the **release()** method.

javaCopy code

actions.dragAndDropBy(source, xOffset, yOffset).release().perform();

The Actions class is particularly useful for automating scenarios that involve mouse hover actions, context menus, drag-and-drop interactions, and keyboard input. It provides a way to simulate user behavior more accurately, helping you create comprehensive and realistic test cases for your web applications.