Actions

The Actions class in Selenium with Java provides a way to perform complex user interactions, such as mouse and keyboard actions, on a web page. It is part of the org.openqa.selenium.interactions package. The Actions class is often used for performing actions like drag-and-drop, mouse hovering, key press/release, etc

Create an instance of the Actions class

Actions actions = new Actions(driver);

For Mouse Hover :

WebElement elementToHover = driver.findElement(By.id("elementId"));

actions.moveToElement(elementToHover).build().perform();

For Click :

WebElement elementToClick = driver.findElement(By.id("elementId"));

actions.click(elementToClick).perform();

For Double Click

WebElement elementToDoubleClick = driver.findElement(By.id("elementId"));

actions.doubleClick(elementToDoubleClick).perform();

For Right Click :

WebElement elementToRightClick = driver.findElement(By.id("elementId"));

actions.contextClick(elementToRightClick).perform();

For Drag and Drop :

```
WebElement sourceElement = driver.findElement(By.id("sourceElementId"));

WebElement targetElement = driver.findElement(By.id("targetElementId"));

actions.dragAndDrop(sourceElement, targetElement).perform();
```

Day - 3

Handling Pop ups

Handling Alerts/pop-ups in Selenium WebDriver using Java involves using the Alert interface. Here's a basic guide :

First we have to switch to the alert using switchto() :

```
// Switch to the alert

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

// Perform actions on the alert

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

// or

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

// Get text from the alert

String alertText = alert.getText();
```

Handling Prompt Pop ups :

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

// Send data to the prompt

alert.sendKeys("YourInput");
```

```
// Accept or dismiss

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

// or

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

// Get text from the alert

String alertText = alert.getText();
```

What are Frames in Selenium?

The term "frames" in Selenium refers to segmenting an HTML document into several portions, each of which can contain a separate HTML document. These frames are also referred to as iframes (inline frames). Frames are frequently used when a web page has numerous sections or pages that must be displayed simultaneously or interacted with independently.

We must first identify the frame to handle frames in selenium using WebDriver commands. This can be done in three ways index, by name or id, and by web elements. Selenium WebDriver has a few simple steps to handle frames:

-
- Switch the driver's focus to the frame using the switchTo().frame() method.
- Using web driver commands, interact with the elements of the frame and perform the operations.
- Switch back to the web content by the switchTo().defaultContent() method

By Index

Use the switchTo().frame() method in Selenium WebDriver and supply the frame's index as an argument to switch to a frame by index. The index of frames on a page starts from 0, so the first frame is at index 0 the next frame at index one, and so on

driver.switchTo().frame(0);

By Name or ID

You can use the name or ID as a parameter to switch to the frame using the switchTo().frame() method in Selenium WebDriver. It's important to make sure the frame name or ID is unique to the frame because there may be other frames on a page with the same name or ID

```
// Switch to the frame by name

driver.switchTo().frame("frame-name");

// Switch to the frame by ID

driver.switchTo().frame("frame-id");
```

By Web Element

We can also switch to the frame by web elements. Pass the web element as an argument in the switchTo().frame() method to switch to the frame using web elements. Use any locator method to locate the frame element before providing it to the switchTo command.

```
// Find the frame element

WebElement frameElement = driver.findElement(By.id("frame-id"));

// Switch to the frame using the web element
```

```
driver.switchTo().frame(frameElement);

// Switch to the parent frame

driver.switchTo().frame("parent-frame");
```

Handling Nested Frames in Selenium WebDriver

In Selenium WebDriver, nested frames mean a situation when a web page has numerous frames nested inside a frame. It means a frame also contains frames as its element. Moving the driver's focus to the appropriate frame is important to interact with the items inside a nested frame when using Selenium.

```
// Switch to the child frame

driver.switchTo().frame("child-frame");

// Interact with elements inside the child frame

WebElement childElement = driver.findElement(By.id("child-element"));

childElement.click();

// Switch back to the parent frame

driver.switchTo().parentFrame();

// Interact with elements inside the parent frame

WebElement parentElement = driver.findElement(By.id("parent-element"));

parentElement.click();
```

```
// Switch back to the default content

driver.switchTo().defaultContent();
```

Working with windows and tabs

Get window handle

WebDriver does not make the distinction between windows and tabs. If your site opens a new tab or window, Selenium will let you work with it using a window handle. Each window has a unique identifier which remains persistent in a single session. You can get the window handle of the current window by using:

```
driver.getWindowHandle();  - > window handle of parent
```

How to get the window handle of Child tab?

We use a method called driver.getWindowHandles(); -> window handle of all(Parent + child tabs)

Syntax for Switching to a window :

```
Driver.switchTo().window(WIndowHandle_ID);
```

getwindowHandle()	**getWindowHandles**()
Gets the window handle of Parent Window	Gets the window handle of All the Windows present(Parent + Child)
Return type of getwindowHandle() is String	Return type of getwindowHandles() is Set<String>

This will give us the window handles of all the windows currently present.

Get the handle of the parent window using the command:

String parentWindowHandle = driver.getWindowHandle();

Print the window handle of the parent window.

Find the element on the web page using an ID which is an element locator.

Open multiple child windows.

Iterate through child windows.

Get the handles of all the windows that are currently open using the command:

Set<String> allWindowHandles = driver.getWindowHandles();

which returns the set of handles.

Use the SwitchTo command to switch to the desired window and also pass the URL of the web page

```
public static void main(String[] args) throws Exception {

System.setProperty("webdriver.chrome.driver","Path to the driver");

WebDriver driver = new ChromeDriver();

driver.manage().window().maximize();

// Load the website

driver.get("http://www.naukri.com/");

// It will return the parent window name as a String

String parent=driver.getWindowHandle();

Set<String>s=driver.getWindowHandles();

// Now iterate using Iterator

Iterator<String> I1= s.iterator();
```

```
while(I1.hasNext())

{

String child_window=I1.next();

if(!parent.equals(child_window))

{

driver.switchTo().window(child_window);

System.out.println(driver.switchTo().window(child_window).getTitle());

driver.close();

}

}

//switch to the parent window

driver.switchTo().window(parent);

}
```

Programs :

```
package Day3;

import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class alertstest {

	public static void main(String[] args) throws InterruptedException {
```

```
		WebDriver driver = new ChromeDriver();

		driver.get("https://www.hyrtutorials.com/p/alertsdemo.html#google_vignette");

		driver.manage().window().maximize();

		driver.findElement(By.id("alertBox")).click();

		Thread.sleep(2000);

		// After clicking above button , ALert will Pop up.

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

		al.accept();


	}

}
```

```
package Day3;

import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class ConfirmationALert {

	public static void main(String[] args) throws InterruptedException {
		// TODO Auto-generated method stub
		WebDriver driver = new ChromeDriver();

		driver.get("https://demo.automationtesting.in/Alerts.html");

		driver.manage().window().maximize();

		driver.findElement(By.xpath("//a[.='Alert with Textbox ']")).click();


		driver.findElement(By.xpath("//button[@onclick='promptbox()']")).click();


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

		Thread.sleep(1000);
```

```
		alert.sendKeys("okkk");

		Thread.sleep(1000);

		alert.accept();// dismiss() is used to click on Cancel
	}

}

package Day3;

import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;

public class DoubleClickTest {

	public static void main(String[] args) throws InterruptedException {
		// TODO Auto-generated method stub
		//



		WebDriver driver = new ChromeDriver();


		driver.get("https://demo.guru99.com/test/simple_context_menu.html");


		driver.manage().window().maximize();


		WebElement ele = driver.findElement(By.xpath("//*[.='Double-Click Me To
See Alert']"));

		// Step 1 - Create object of Actions class

		Actions act = new Actions(driver);


		act.doubleClick(ele).build().perform();

		Thread.sleep(2000);

		// We are getting an ALert
```

```
                // Step 1 - TO handle alert - use switchto()

                 Alert al = driver.switchTo().alert(); // this command will switch to ALert

                 // Step 2 - use accept()
                 al.accept();









        }

}
```

```
package Day3;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;

public class Draganddrop {

        public static void main(String[] args) throws InterruptedException {


                WebDriver driver = new ChromeDriver();

                driver.get("https://v1.training-support.net/selenium/drag-drop");


                WebElement ball = driver.findElement(By.id("draggable"));


                WebElement targetarea = driver.findElement(By.id("droppable"));

                WebElement targetarea2 = driver.findElement(By.id("dropzone2"));

                //Create object of Actions class

                Actions act = new Actions(driver);

                act.dragAndDrop(ball, targetarea).build().perform();
```

```
		Thread.sleep(1000);

		act.dragAndDrop(ball, targetarea2).build().perform();



	}

}

package Day3;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class NestedFrames {

	public static void main(String[] args) throws InterruptedException {
		WebDriver driver = new ChromeDriver();

		driver.get("https://v1.training-support.net/selenium/nested-iframes");

		driver.manage().window().maximize();

		Thread.sleep(1000);

		WebElement outerframe =
driver.findElement(By.xpath("//iframe[@src='/selenium/nested-iframe1']"));

		driver.switchTo().frame(outerframe);

		WebElement innerframe1 =
driver.findElement(By.xpath("//iframe[@src='/selenium/frame1']"));

		driver.switchTo().frame(innerframe1);

		driver.findElement(By.id("actionButton")).click();

		driver.switchTo().defaultContent();

		driver.findElement(By.xpath("//button[@onclick='goBack()']")).click();



	}
```

```
}

package Day3;

import org.openqa.selenium.Alert;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class PromptAlert {

	public static void main(String[] args) throws InterruptedException {


		WebDriver driver = new ChromeDriver();

		driver.get("https://www.hyrtutorials.com/p/alertsdemo.html#google_vignette");

		driver.manage().window().maximize();

		driver.findElement(By.id("promptBox")).click();

		Thread.sleep(2000);

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

		Thread.sleep(1000);

		alert.sendKeys("Hi there");

		Thread.sleep(1000);

		alert.accept();

	}

}

package Day3;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;

public class RightclickTest {
```

```
	public static void main(String[] args) {

		WebDriver driver = new ChromeDriver();

		driver.get("https://demo.guru99.com/test/simple_context_menu.html");

		driver.manage().window().maximize();

		WebElement rightclick_ele = driver.findElement(By.xpath("//*[.='right click me']"));

		// Step 1 - Create an object of Actions class

		Actions act = new Actions(driver);

		act.contextClick(rightclick_ele).build().perform();



	}

}
```

```
package Day3;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class SingleFrameTest {

	public static void main(String[] args) throws InterruptedException {

		WebDriver driver = new ChromeDriver();

		driver.get("https://v1.training-support.net/selenium/iframes");

		driver.manage().window().maximize();
```

```
		Thread.sleep(1000);

		WebElement frame1 =
driver.findElement(By.xpath("//iframe[@src='/selenium/frame1']"));

		//driver.switchTo().frame(0); // Switching to frame using index.

		driver.switchTo().frame(frame1); // Switching to frame using Webelement.

		driver.findElement(By.id("actionButton")).click();

		// Switch back to Main window

		driver.switchTo().defaultContent();

		Thread.sleep(2000);

		driver.findElement(By.xpath("//button[@onclick='goBack()']")).click();

		//Thread.sleep(1000);

		//driver.close();


	}

}
```

```
package Day3;
import java.util.Set;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
public class SwitchWindowTest {
	public static void main(String[] args) {
		WebDriver driver = new ChromeDriver();

		driver.get("https://rahulshettyacademy.com/AutomationPractice/");

		driver.findElement(By.id("opentab")).click();

		// Step 1 - Find window handles of all windows

		String parentid = driver.getWindowHandle();

		Set<String> allwindows = driver.getWindowHandles();
```

```
                String Expected_title = "QAClick Academy - A Testing Academy to Learn, Earn and
Shine";


                // Create a for each loop iterate over these allwindows


                for (String single_id : allwindows) {

                        System.out.println(single_id);
                        driver.switchTo().window(single_id); // Trial and error
                        // First we are switching and then we are checking
                        // whether the switched tab is our expected tab

                        if(driver.getTitle().equals(Expected_title)) {

                                System.out.println("Switched to correct tab");
                                break;
                        }// end of my if block


                }// end of my for loop


                driver.findElement(By.linkText("Access all our Courses")).click();




        }
}
```