

Agenda

Switch To Commands:

Handle Frames

Handle Windows

Types of Alerts

1) Simple Alert

Simple alert displays some information or warning on the screen.

2) Confirmation Alert

This confirmation alert asks permission to do some type of operation.

3) Prompt Alert

Prompt Alert asks some input from the user and selenium webdriver can enter the text using `sendKeys("input....")`.

Handle IFrames / Frames

In some application for better visibility developer use frame concept in web pages.

What is iFrame? An iFrame (Inline Frame) is an HTML document embedded inside the current HTML document on a website.

iFrame HTML element is used to insert content from another source, such as an advertisement, into a Web page.

A Web designer can change an iFrame's content without making them reload the complete website.

A website can have multiple frames on a single page. And a frame can also have inner frames (Frame in side a Frame)

We can handle frames/iframes present in the webpage using **driver.switchTo()** command in selenium webdriver.

Frame/iFrame is nothing but **another webelement in html page, which displays another part of webpage.**

How to Identify Frame ?

There are many ways to identify the Frame. We will see the easiest way to identify the frame. Let's see the step by step procedure-

Right click on the element. Check if “This Frame” option is available. If This frame option is available, it means that the element is inside a frame.

View page source of the web page and check if any tag is available for ‘iframe’.

It will show the iFrame tag for the frame inside which the webelement is present.

To work with the frames you need to switch to frame first.

In Selenium to work with Frames, we have different ways to handle frame depending on the need.

On Chrome : View frame source option get enable when you mouse hover on element

The screenshot shows a web browser window with the address bar displaying `https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_frame_cols`. The page content consists of three frames: Frame A (purple), Frame B (orange), and Frame C (yellow). Frame A contains the text: "Note: The frameset, frame, and noframes elements supported HTML5." A context menu is open over Frame A, listing various actions. The "View frame source" option is highlighted. The code editor on the left shows the following HTML code:

```
<!DOCTYPE html>
<html>

<frameset cols="25%,*,25%">
  <frame src="frame_a.htm">
  <frame src="frame_b.htm">
  <frame src="frame_c.htm">
</frameset>

</html>
```

The context menu options are:

- Back (Alt+Left Arrow)
- Forward (Alt+Right Arrow)
- Reload (Ctrl+R)
- Save as... (Ctrl+S)
- Print... (Ctrl+P)
- Cast...
- Translate to English
- IDM Integration Module
- View page source (Ctrl+U)
- View frame source (highlighted)
- Reload frame
- Inspect (Ctrl+Shift+I)

The Windows taskbar at the bottom shows the search bar with "Type here to search" and the system clock displaying 22:30 on 21-10-2018.

On Firefox : This frame option get enable

The screenshot shows a Firefox browser window with the address bar displaying `https://www.w3schools.com/tags/tryit.asp?filename=tryhtml_frame_cols`. The page content includes a Sony Bravia banner and a frameset with three frames: Frame A (purple), Frame B (orange), and Frame C (yellow). A context menu is open over Frame A, showing options like 'This Frame', 'View Page Source', and 'Inspect Element'. The code editor on the left shows the HTML frameset code.

```
<!DOCTYPE html>
<html>

<frameset cols="25%,*,25%">
  <frame src="frame_a.htm">
  <frame src="frame_b.htm">
  <frame src="frame_c.htm">
</frameset>

</html>
```

Result Size: 662 x 470

Frame A

Note: The frameset, frame, and noframes elements are not supported in HTML5.

Frame B

Frame C

Context Menu Options:

- Save Page As...
- Save Page to Pocket
- Send Page to Device
- View Background Image
- Select All
- This Frame
- View Page Source
- View Page Info
- Inspect Element
- Show Only This Frame
- Open Frame in New Tab
- Open Frame in New Window
- Reload Frame
- Bookmark This Frame
- Save Frame As...
- Print Frame...
- View Frame Source
- View Frame Info

```
1 <html>
2 <head><title>Welcome to Selenium iframes Tutorial</title>
3 </head>
4 <body>
5 <div>
6   <iframe id="frame1">
7     <iframe id="frame2">
8       <body>
9         <input type="text" id="input2">UserName</input>
10      </body >
11    </iframe>
12
13    <body>
14      <input type="text" id="input1">Password</input>
15    </body >
16  </iframe>
17
18  <body>
19    <button name="btnG">OK</button>
20  </body >
21 </div>
22 </body>
23 </html>
```

Switch to Frames by Name or ID

Select a frame by its name or ID. Frames located by matching name attributes are always given precedence over those matched by ID.

Syntax - `driver.switchTo().frame(String name/id);`

```
driver.switchTo().frame("iframe1");  
driver.quit();
```

Throws: `NoSuchFrameException` - If the frame is not found

Switch to Frame by WebElement

we can switch to an iFrame by simply passing the iFrame WebElement to the `driver.switchTo().frame()` command.

First find the iFrame element using any of the locator strategies and then passing it to `switchTo` command.

```
WebElement iframeElement = driver.findElement(By.id("IF1"));
```

```
//now use the switch command
```

```
driver.switchTo().frame(iframeElement);  
driver.quit();
```

Throws: `NoSuchFrameException` - If the given element is neither an iframe nor a frame element.

Switch to Frames by Index

Index of an iFrame is the position at which it occurs in the HTML page.

Select a frame by its (zero-based) index. That is, if a page has multiple frames (more than 1), the first frame would be at index "0", the second at index "1" and so on.

Once the frame is selected or navigated , all subsequent calls on the WebDriver interface are made to that frame. i.e the driver focus will be now on the frame.

What ever operations we try to perform on pages will not work and throws element not found as we navigated / switched to Frame.

Syntax - `driver.switchTo().frame(int arg0);`

```
driver.switchTo().frame(0);  
driver.quit();
```

Throws: NoSuchElementException - If the frame is not found.

Switch to Frames by Index

Sometimes when there are multiple Frames (Frame inside a frame), we need to first switch to the parent frame and then we need to switch to the child frame.

In the below HTML, you can see we have frame2 inside frame1 and web elements inside the frame2.

//Switch to child frame

```
driver.switchTo().frame("frame1").switchTo().frame("frame2");  
driver.findElement(By.id("input2")).sendKeys("Username");  
driver.findElement(By.id("input1")).sendKeys("Password");
```

```
driver.findElement(By.cssSelector("button[name='btnG']")).click();
```

//switch to parent frame

```
driver.switchTo().frame("frame1");
```

Switching back to Main page from Frame

After working with the frames, main important is to come back to the web page.

if we don't switch back to the default page, driver will throw an exception.

Once you are done with all the task in a particular iFrame we need to switch back to the main page using the `switchTo().defaultContent()`.

```
WebElement iframeElement = driver.findElement(By.id("IF1"));
```

```
//now use the switch command
```

```
driver.switchTo().frame(0);
```

```
//Do all the required tasks in the frame 0
```

```
//Switch back to the main window
```

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

```
driver.quit();
```

Test Case

TC 1: ToolsQA Frame verification

1. Open chrome browser
2. Go to “<https://www.toolsqa.com/iframe-practice-page/>”
3. Click on “ Guest Blogs” link inside first frame
4. Verify element “Interactions” is present in second frame
5. Navigate to Home Page
6. Quit browser

Multiple Windows in Selenium Webdriver

There will be times when we will required to perform some testing, where the testing operations open a new browser/tab, test case may required you to perform some tasks on the newly opened browser window/tab and return back to original window to perform the remaining tasks.

Even if the window/tab is currently on focus but still it is not an active window, so to perform some tasks we need to switch to new browser window/tab in selenium webdriver.

Situations when we are likely to deal with multiple windows:

- Filling forms may require selecting the date from a separately opened window
- Clicking on some link/button can open another window
- Handling Advertisement windows

Switch Window Commands

Selenium WebDriver assigns an alphanumeric id to each window as soon as the WebDriver object is instantiated.

This unique alphanumeric id is called window handle. Selenium uses this unique id to switch control among several windows.

In simple terms, each unique window has a unique ID, so that Selenium can differentiate when it is switching controls from one window to the other.

GetWindowHandle Command

Purpose: To get the window handle of the current window.

//Return a string of alphanumeric window handle

```
String handle= driver.getWindowHandle();
```

Switch Window Commands

GetWindowHandles Command

Purpose: To get the window handle of all the current windows.

//Return a set of window handle

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

SwitchTo Window Command

Purpose: WebDriver supports moving between named windows using the “switchTo” method.

driver.switchTo().window("windowName");

Test Case

TC 1: ToolsQA Windows verification

1. Open chrome browser
2. Go to "https://the-internet.herokuapp.com/"
3. Click on "Multiple Windows" link
4. Click on "Elemental Selenium"
5. Verify title of second window is "Elemental Selenium: Receive a Free, Weekly Tip on Using Selenium like a Pro"
6. Close second window
7. Click on "Click Here" link
8. Verify title if newly open window is "New Window"
9. Close second window
10. Verify title of second window is "The Internet"
11. Quit browser

NOTE: Selenium execution could be too fast, please use Thread.sleep