**\*Automation Testing or SDET\***

1) UI (User Interface) Automation or Web Based Automation

2) API (Application Programming interface) Automation

3) Database Automation.

1) UI (User Interface) Automation or Web Based Automation

1) Java

2) Selenium WebDriver

3) TestNG [Test Next Generation]

4) Java Script Executor

5) Cucumber

6) Page Object Model (POM) Design Pattern

7) Data Driven Framework

8) Hybrid Framework

9) BDD Framework

10) GitHub using command Mode and UI.

11) Maven Build tool

12) Basic HTML Code

13) Jenkins (CICD)

2) API (Application Programming interface) Automation

1) Java

2) Cucumber

3) TestNG

4) GitHub

5) Maven

6) Jenkins

7) Rest assured

8) Json Path

9) Jackson-databind

10) Hamcrest

11) Json Schema Validator

12) Lombok

13) POJO or Bean

14) Gson

15) Google json simple

16) Rest Assured with Hybrid Framework

17) Rest Assured with BDD Framework

18) Spring boot Framework using MVC design Pattern

3) Database Automation using MySQL

1) Java

2) Java Database connectivity

3) Combine the API and Database Automation.

4) API and Database Automation using BDD framework

Manual Testing

1) Manual Testing

2) JIRA

3) Database testing using SQL Server

4) Basic Unix OS commands

5) API testing using Postman

6) Telecom Project

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Selenium\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**What is Selenium? VVVIMMMPPP**

- Selenium is suite of tool, is used to automate the web Based application.

- Selenium is open source tool, everyone we can use free of cost.

- Selenium is used to achieve the functional and regression testing.

- Selenium is not single tool to achieve the UI/GUI/Web based automation.

- there are different tools available in market to achieve the UI Automation testing.

1) selenium WebDriver

2) UFT/QTP

3) TOSCA

4) Silk Test

5) Test Complete

6) RFT

7) Protector

etc...

- QTP tool is used to achieve the Web Based automation as well as Desktop based Automation.

-but Selenium tool is used to achieve the Web Based Automation only.

- If u want achieve the Desktop based automation in selenium webdriver we use """AutoIT tool. ----> VVVVIMMMPPP

**How to upload files in Selenium?**  **VVVVIMMMPPP**

We use AutoIT library/tool to upload the files in selenium.

What are the different language supported by selenium? **VVVVIMMMPPP**

1) java

2) Python

3) Java Script

4) PHP

5) C#

6) Perl

7) Ruby

8) Groovy Script

What are the different OS Supported by Selenium?

1) Window

2) Mac

3) Linux

What are the different browser is supported by selenium?

1) HTML Unit Driver

2) Chrome Driver

3) Edge Driver

4) Opera Driver

5) Firefox Driver

6) IE driver

7) Safari Driver

**which is fastest driver in selenium? VVVVVVVIMMMPPPPPPPPPPP**

**HTML Unit driver is fastest driver in selenium**

--

Selenium is suite of tool

there are 4 different tools present in selenium

1) Selenium IDE (Integrated Development Environment)

2) Selenium RC (Remote Control) --->out dated tool---->it has same features as like WebDriver.

3) Selenium WebDriver

4) Selenium Grid

1) Selenium IDE (Integrated Development Environment)

- Selenium IDE is used to record test case and play back same.

- Selenium IDE is used to create the test cases, edit test case, execute the test case, debug test cases, enhance test cases, create test suite.

- We can create the test cases in Selenium IDE using """Selenese command""". VVVVVVVIMMMPPPPPPPPPPP

- Selenium IDE test cases default format is .html (Hypertext markup language).

- We can export the test cases from .html format to java/python/php/c#/perl/ruby/groovy script.

- Selenium IDE it only support for ""Firefox browser"".

- Selenium IDE is not supported for programming logic to write the test case/enhance test case.

- Selenium IDE does not support for Data driven testing.

- Selenium IDE is not support for complex test cases.

how to write the test cases in Selenium IDE?

We use """Selenese command""" to write test cases.

2) Selenium RC (Remote Control)

- Selenium RC is similar like Selenium WebDriver but it is outdated tool.

3) Selenium WebDriver **VVVVVIMMMPPPPPP**

- Selenium WebDriver is an programming interface.

- Selenium WebDriver is used to create the test cases, edit test cases, enhance test cases, debug test cases, execute the test cases,.

- Selenium WebDriver is support for different languages

1) Java

2) Python

3) Java Script

4) PHP

5) C#

6) Perl

7) Ruby

8) Groovy Script

- Selenium WebDriver support different OS

1) Window

2) Linux

3) Mac

- selenium Webdriver support different browser

1) HTML unit Driver

2) Chrome Driver

3) Edge Driver

4) Firefox driver

5) Opera Driver

6) Safari Driver

7) IE driver

**- Which is fastest driver in selenium WebDriver?**

- **HTML unit driver** is fastest driver in selenium WebDriver.

- Using Element Locator and Selenium WebDriver methods we create test cases in Selenium WebDriver.

- What are the different Element/Object present in Web Based application?

1) Text box

2) Drop down

3) Radio Button

4) Button/Register button/submit/save button/cancel button/reset button/show/proceed button/dismiss/etc..

5) Link

6) Checkbox

7) Image

8) Mouse Events

7) upload/download

8) Physical Text /label

9) Pop Up

10) Keyboard Actions

11) Drag And Drop

12) Scroll up and down

13) calendar

14) Browsers (back, forward, refresh, maximize , minimize, full screen, browser dimension/close tab/close window)/Navbar

15) open a new tab or new window

16) click and hold

17) click/right click/ double click.

18) Frames

19) static Web Table/ dynamic Web tables

20) Sliders

21) Screenshot

**- What are the different locators in selenium?**  VVVVVVIMMMPPPPPPPP

- it used to locate the Object/Element in Web Page.

**there are 8 types of locators in selenium WebDriver.**

1) Id

2) name

3) className

4) tagName

5) linkText

6) partialLinkText

7) xpath

8) cssSelector

- Locators are common for all the browsers.

- We can use anyone locator to locate the object in web page.

**what is fastest locator in selenium WebDriver? VVVVVVIMMMPPPPPPPP**

- id fastest locator in selenium webdriver

4) Selenium Grid

- Selenium Grid is used to run the test cases.

- we cannot create test cases in selenium grid.

What are the different operations we can perform on browser using selenium WebDriver?

1) Open a chrome browser

2) Open a URL in browser

3) Navigate from One URL to another URL

4) Navigate back to previous page

5) Navigate forward

6) Refresh the browser

7) Capture the browser URL

8) Capture title of browser window

9) minimize the browser

10) maximize the browser

11) full screen the browser

12) open a new tab

13) open a new window

14) close a single tab

15) close all tab/window

16) open a browser in incognito mode.

----

Selenium WebDriver 3.141.59 version. **VVVVVIMMMPPPP**

----

**how to connect to the actual browser?**

- connect to the actual browser we use System.setProperty()method

- if we want to connect to the browser we need same browser driver of same version.

- just imagine we use chrome browser 124.0 version then we need same chrome driver version.

- chrome browser and chrome driver version should be same otherwise we will get Runtime error.

syntax:

System.setProperty("webdriver.chrome.driver","path of chrome driver");

e.g.

System.setProperty("webdriver.chrome.driver","

----

open a browser

//Up casting .

Reference of parent class and object of child class, this allowed to access all the methods from parent class.

Reference of WebDriver interface and Object of ChromeDriver class, this allowed to access all the methods from WebDriver interface.

syntax:

WebDriver driver new ChromeDriver();

it will open a blank browser means without any URL

------

Selenium WebDriver methods VVVIMMMPPPP

**1) get() method**

- get() method is used to open specified URL in browser window.

- return type is void.

- get() method present inside the WebDriver interface.

syntax:

webdriverobjectName.get("url");

e.g.

driver.get("https://www.facebook.com");

step 1: connect to the actual browser

System.setProperty("webdriver.chrome.driver","path of chromedriver");

step 2: up casting concept

WebDriver driver new ChromeDriver();

step 3: open a url in browser

driver.get("https://www.facebook.com");

package Tutorial1;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo2 {

public static void main(String[] args) {

// step 1: connect to the actual browser.

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver-win64 (4)\\chromedriver-win64\\chromedriver.exe");

// Step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open facebook url in browser window

driver.get("https://www.facebook.com");

}

}

**2) getTitle() method**

- getTitle()method is used to capture the title of the browser window

- return type is string

- getTitle() method present inside the WebDriver interface.

syntax:

webdriverobjectname.getTitle();

e.g.

String a driver.getTitle();

step 1: connect to the actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://www.amazon.com");

step 4:capture the browser title

String a driver.getTitle();

System.out.println(a);

package Tutorial1;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo3 {

public static void main(String[] args) {

//step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver", "C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver-win64 (4)\\chromedriver-win64\\chromedriver.exe");

//step 2: up casting

WebDriver driver new ChromeDriver();

//step 3: open a url

driver.get("https://www.amazon.in");

//step 4: capture the title

String abc driver.getTitle();

System.out.println(abc);

}

}

-

**3) getCurrentUrl() method**

- getCurrentUrl() method is used to capture the URL of browser window.

- return type string

- getCurrentUrl() method is present inside the WebDriver interface.

syntax:

webdriverobjectname.getCurrentUrl();

e.g.

Sting abc driver.getCurrentUrl();

step 1: connect to the actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3; open URL

driver.get("https://www.redbus.com");

step 4: capture the current url

String abc driver.getCurrentUrl();

package Tutorial1;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo4 {

public static void main(String[] args) {

//step 1: connect to the actual browser

System.setProperty("webdriver.chrome.driver", "C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver-win64 (4)\\chromedriver-win64\\chromedriver.exe");

//step 2: up casting

WebDriver driver new ChromeDriver();

//step 3: open a url

driver.get("https://www.redbus.com");

//step 4: capture the current url

String abc driver.getCurrentUrl();

System.out.println(abc);

}

}

Ass:

open a https://opensource-demo.orangehrmlive.com/web/index.php/auth/login

capture the title

and verify HRM keyword is present or not If present then print "Title test case is pass" otherwise print "Title test cases is fail".

capture the URL

and verify demo keyword is present or not If present then print "URL test case is pass" otherwise print "URL test cases is fail".

Ass:

open a https://www.facebook.com/

capture the title

reverse the captured title

**Selenium WebDriver Methods**

1) get() method

- it used to open a specified URL in browser window

- return type is void.

- get() method present inside the WebDriver interface.

syntax:

driver.get("https://www.google.com");

--

2) getTitle()method

- it is used to capture the title of the browser window

- return type is String.

-getTitle() method present inside the WebDriver interface.

syntax:

String title driver.getTitle();

--

3) getCurrentUrl() method

- it is used to capture the current URL browser window

- return type is String

- -getCurrentUrl() method present inside the WebDriver interface.

syntax:

String url driver.getCurrentUrl();

--

**4) navigate() method**

- it used to navigate from one url to another url, navigate back to previous page, navigate forward page and refresh the browser.

- return type is Navigation interface.

- navigate() method present inside the WebDriver interface.

syntax;

Navigation nvt driver.navigate();

--

**Navigation interface method**

1) to() method

2) back()method

3) forward() method

4) refresh() method

**1) to() method**

- to() method used to navigate from one URL or another URL.

- also to()method is used to open url in browser window.

- return type is void.

- to() method present inside the Navigation interface.

syntax:

Navigation nvt driver.navigate();

nvt.to("https://www.google.com");

or

driver.navigate().to("https://www.google.com");

**can we open a new URL by using driver.navigate().to() method? VVVIMMMPPPPP**

yes we can open but it will not wait till all component to be loaded in page.

**diff between driver.get() method and driver.navigate().to() method? VVVVIMMMPPPP**

- driver.get() method is used to open a specified url in browser window and it will wait till all component to be loaded in webpage.

- get() method present inside the WebDriver interface.

- driver.navigate().to() method is used to open specified url in browser window as well it is navigate from one url to another url and

it will not wait till all component loaded in webpage.

- to() method is present inside the Navigation interface.

--

Scenario:

open a facebook application

capture the title

navigate to google application

capture the URL

step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a URL

driver.get("https://www.facebook.com");

step 4: capture the title

String abc driver.getTitle();

step 5: Navigate to google URL

Navigation nvt driver.navigate();

nvt.to("https://www.google.com");

or

driver.navigate().to("https://www.google.com");

step 6: capture the URL

String url driver.getCurrentUrl();

package Tutorial2;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

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

// Connect to the actual broser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// up casting

WebDriver driver new ChromeDriver();

// open a url

driver.get("https://www.facebook.com");

// capture the title

String a driver.getTitle();

System.out.println(a);

// Wait 5 seconds

Thread.sleep(5000);

// navigate to google.com

driver.navigate().to("https://www.google.com");

String b driver.getCurrentUrl();

System.out.println(b);

}

}

2) back() method

- it used to navigate single item back in browser history

- return type is void

- back() method present inside the Navigation interface.

syntax:

Navigation nvt driver.navigate();

nvt.back();

or

driver.navigate().back();

Scenario:

open a amazon url

capture the title

navigate to flipkart url

capture the title

navigate back amazon url

capture the Url

step 1: connect to the actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a amazon URL

driver.get("https://www.amazon.in");

step 4: capture the title

String a driver.getTitle();

step 5; navigate to flipkart url

driver.navigate().to("https://www.flipkart.com");

step 6: capture the title

String b driver.getTitle();

step 7: navigate back

driver.navigate().back();

step 8: capture the current url

String c driver.getCurrentUrl();

package Tutorial2;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo2 {

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

// connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// up casting

WebDriver driver new ChromeDriver();

// open a amazon url

driver.get("https://www.amazon.in");

// capture the title

String a driver.getTitle();

System.out.println(a);

// Wait for 5 seconds

Thread.sleep(5000);

// navigate to flipkart url

driver.navigate().to("https://www.flipkart.com");

// capture the title

String b driver.getTitle();

System.out.println(b);

// Wait for 5 seconds

Thread.sleep(5000);

// navigate back

driver.navigate().back();

// capture the Url

String c driver.getCurrentUrl();

System.out.println(c);

}

}

**3) forward() method**

- it navigate single item forward in browser history.

- return type is void

- forward() method present inside the Navigation interface.

syntax:

Navigation nvt driver.navigate();

nvt.forward();

or

driver.navigate().forward();

Scenario:

- open orange hrm url

- capture the title

- navigate to guru99 url

capture the title

- navigate back

- capture url

- navigate forward

- capture url

step 1: connect to the actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a orange hrm url

driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

step 4: capture title

String a driver.getTitle();

step 5: navigate to guru99 url

driver.navigate().to("https://demo.guru99.com/test/newtours/register.php");

step 6: capture title

String b driver.getTitle();

step 7: navigate back

driver.navigate().back();

step 8: capture url

String c driver.getCurrentUrl();

step 9: navigate forward

driver.navigate().forward();

step 10: capture url

String d driver.getCurrentUrl();

package Tutorial2;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebDriver.Navigation;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo3 {

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

// Step 1: Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Step 2: up casting

WebDriver driver new ChromeDriver();

// Step 3: open a orange HRM url

driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

// Step 4: capture title

String a driver.getTitle();

System.out.println(a);

// Wait for 5 seconds

Thread.sleep(5000);

//Step 5: navigate to Guru99 URL

Navigation nvt driver.navigate();

nvt.to("https://demo.guru99.com/test/newtours/register.php");

//or

//driver.navigate().to("https://demo.guru99.com/test/newtours/register.php");

//step 6: capture title

String b driver.getTitle();

System.out.println(b);

//wait for 5 seconds

Thread.sleep(5000);

//step 7:navigate back

driver.navigate().back();

//step 8: capture url

String c driver.getCurrentUrl();

System.out.println(c);

//wait for 5 seconds

Thread.sleep(5000);

//step 9: navigate forward

driver.navigate().forward();

//step 10: capture url

String d driver.getCurrentUrl();

System.out.println(d);

}

}

4) refresh()method

- it used to refresh the web page or reload the webpage

- return type is void

- this method present inside the Navigation interface.

syntax:

Navigation nvt driver.navigate();

nvt.refresh();

or

driver.navigate().refresh();

scenario:

open a flipkart url

capture title and url

refresh the webpage

Step 1: connect to the actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a flipkart url

driver.get("https://www.flipkart.com");

step 4: capture title

String a driver.getTitle();

step 5: capture url

String b driver.getCurrentUrl();

step 6; refresh the browser

driver.navigate().refresh();

package Tutorial2;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo4 {

public static void main(String[] args) {

// Step 1: Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Step 2: up casting

WebDriver driver new ChromeDriver();

//step 3: open a url

driver.get("https://www.flipkart.com");

//step 4: capture title

System.out.println(driver.getTitle());

//step 5: capture url

System.out.println(driver.getCurrentUrl());

//step 6: refresh the web page

driver.navigate().refresh();

}

}

1) switchTo() method

- switchTo() method is used to

i) switch focus to parent frame,

ii) switch focus to child frame,

iii) switch focus to top frame,

iv) switch focus to alter pop up,

v) switch focus to window

vi) switch focus to tab,

vii) open new window

viii) open a new tab

- return type of switchTo() method is TargetLocator interface.

- switchTo() method present inside the WebDriver interface.

syntax:

TargetLocator trg driver.switchTo();

**1) newWindow(WindowType.TAB) ---> \*\*\*\* this method present inside** the Selenium 4 and above version.

- it used to open new Tab.

- return type is WebDriver interface.

- this method is present inside the TargetLocator interface.

syntax:

TargetLocator trg driver.switchTo();

trg.newWindow(WindowType.TAB);

or

driver.switchTo().newWindow(WindowType.TAB);

Scenario:

- open a google url

- open a new tab

- in new tab open facebook url

Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chromedriver");

step 2; up casting

WebDriver driver new ChromeDriver();

step 3: open a google url

driver.get("https://www.google.com");

step 4: open new tab

driver.switchTo().newWindow(WindowType.TAB);

step 5: open fb url in new tab

driver.get("https://www.facebook.com");

package Tutorial2;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WindowType;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo5 {

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

// Step 1: Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://www.facebook.com");

// wait for 7 seconds

Thread.sleep(7000);

// step 4: open a new tab

driver.switchTo().newWindow(WindowType.TAB);

// wait for 7 seconds

Thread.sleep(7000);

// step 5: open a google url

driver.get("https://www.google.com");

}

}

-----

**2) newWindow(WindowType.WINDOW);**

- it used to open a new window

- this method present inside the TargetLocator interface.

- return type is WebDriver interface.

syntax:

TargetLocator trg driver.switchTo();

trg.newWindow(WindowType.WINDOW);

or

driver.switchTo().newWindow(WindowType.WINDOW);

Scenario;

open a google url

open new window

open amazon url in new window

step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open google url

driver.get("https://www.google.com");

step 4: open a new window

driver.switchTo().newWindow(WindowType.WINDOW);

step 5: open amazon url in new window

driver.get("https://www.amazon.in");

package Tutorial2;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WindowType;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo6 {

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

// Step 1: Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://www.google.com");

// wait for 7 second

Thread.sleep(7000);

// step 4: open a new window

driver.switchTo().newWindow(WindowType.WINDOW);

// wait for 7 second

Thread.sleep(7000);

// step 5: open a amazon url

driver.get("https://www.amazon.in");

}

}

**1) close() method**

- it used to close the focused window or tab

- return type is void

- this method is present inside the WebDriver interface.

syntax:

driver.close();

**2) quit() method**

- it is used to close all window/tab.

- return type is void

- this method is present inside the WebDriver interface.

syntax:

driver.quit();

Scenario:

open a google url

open a new tab

open facebook url in new tab

close a tab

package Tutorial2;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WindowType;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo7 {

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

// Step 1: Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://www.google.com");

Thread.sleep(5000);

// step 4:open a new tab

driver.switchTo().newWindow(WindowType.TAB);

Thread.sleep(5000);

// step 5: open a fb

driver.get("https://www.facebook.com");

Thread.sleep(5000);

driver.close();

}

}

package Tutorial2;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WindowType;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo7 {

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

// Step 1: Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://www.google.com");

Thread.sleep(5000);

// step 4:open a new tab

driver.switchTo().newWindow(WindowType.TAB);

Thread.sleep(5000);

// step 5: open a fb

driver.get("https://www.facebook.com");

Thread.sleep(5000);

driver.quit();

}

}

package Tutorial2;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WindowType;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo7 {

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

// Step 1: Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://www.google.com");

Thread.sleep(5000);

// step 4:open a new tab

driver.switchTo().newWindow(WindowType.WINDOW);

Thread.sleep(5000);

// step 5: open a fb

driver.get("https://www.facebook.com");

Thread.sleep(5000);

// Session is running state after using the close() method

driver.close();

}

}

package Tutorial2;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WindowType;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo7 {

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

// Step 1: Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://www.google.com");

Thread.sleep(5000);

// step 4:open a new tab

driver.switchTo().newWindow(WindowType.WINDOW);

Thread.sleep(5000);

// step 5: open a fb

driver.get("https://www.facebook.com");

Thread.sleep(5000);

// Session is close after using the quit() method

driver.quit();

}

}

1) driver.navigate().to("https://www.google.com");

2) driver.navigate().back();

3) driver.navigate().forward();

4) driver.navigate().refresh();

- navigate()method present inside the WebDriver interface

- return type of navigate()method is Navigation interface.

- to(),back(),forward() and refresh() method present inside the Navigation interface

- return type of to(),back(),forward() and refresh() method is void.

5) driver.switchTo().newWindow(WindowType.TAB);

6) driver.switchTo().newWindow(WindowType.WINDOW);

- switchTo() method present inside the WebDriver interface

- return type of switchTo() method is TargetLocator interface.

- newWindow()method present inside the TargetLocator interface.

- return type of newWindow()method is WebDriver interface.

7) driver.close();

8) driver.quit();

- close() and quit() method is present inside the WebDriver.

- return type is close() and quit() method is void.

Assignment:

// open a edge/chrome browser

System.setProperty("webdriver.chrome.driver","D:\Softech Class 2024\Softech Class - 27 April 2024 Automation Testing\chromedriver-win64\chromedriver-win64\chromedriver.exe");

WebDriver drivernew ChromeDriver();

//open a url https://copyright.gov.in/UserRegistration/frmNewUser.aspx

driver.get("https://copyright.gov.in/UserRegistration/frmNewUser.aspx");

//capture the title

String titledriver.getTitle();

System.out.println(title);

//verify the "Registration" keyword is present or not if present the print title test cases is pass else fail

if(title.contains("Registration"))

{

System.out.println("Title test case is pass");

}

else

{

System.out.println("Title test case is fail");

}

//open new tab

driver.switchTo().newWindow(Window.TYPE,TAB)

//open a url https://demoqa.com/buttons

driver.get("https://demoqa.com/buttons");

//capture the url

String urldriver.getCurrentUrl();

System.out.println(url);

//navigate to https://ui.cogmento.com/ url

driver.navigate().to("https://ui.cogmento.com/ url

");

//capture the title

driver.getTitle();

//navigate to url https://www.irctc.co.in/nget/train-search

driver.navigate().to("https://www.irctc.co.in/nget/train-search

");

//capture url

driver.getCurrentUrl();

//navigate back

driver.navigate().back();

//capture url

driver.getCurrentUrl();

//navigate forward

driver.navigate().forward();

//capture title

driver.getTitle();

//close all window

driver.quit();

**Selenium WebDriver methods**

1) get() method

2) getTitle() method

3) getCurrentUrl() method

4) close() method

5) quit() method

6) switchTo() method

7) navigate() method

**Navigation interface**

1) to() method

2) back()method

3) forward() method

4) refresh() method

----------

**TargetLocator interface**

**1) newWindow(WindowType.TAB)**

**2) newWindow(WindowType.WINDOW)**

**1) maximize() method**

- it is used to maximize the window

syntax:

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

manage() method present inside the WebDriver interface.

return type of manage() method is Options interface.

window() method present inside the Options interface.

return type of window()method is Window interface.

maximize() method present the Window interface.

return type of maximize()method is void.

----------

**2) minimize() method -------> this method present inside the selenium 4 and above** **version.**

-it is used to minimize the window

syntax:

driver.manage().window().minimize();

manage() method present inside the WebDriver interface.

return type of manage() method is Options interface.

window() method present inside the Options interface.

return type of window()method is Window interface.

minimize() method present the Window interface.

return type of minimize()method is void.

----------

**3) fullScreen() method**

-it is used to full screen the window

syntax:

- driver.manage().window().fullScreen();

manage() method present inside the WebDriver interface.

return type of manage() method is Options interface.

window() method present inside the Options interface.

return type of window()method is Window interface.

fullScreen() method present the Window interface.

return type of fullScreen()method is void.

Scenario:

open application

maximize the window

Step 1; connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 2: open a url

driver.get("https://www.google.com");

step 3: maximize the window

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

package Tutorial3;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

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

// Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// up casting

WebDriver driver new ChromeDriver();

// open a google url

driver.get("https://www.google.com");

Thread.sleep(5000);

//maximize

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

Thread.sleep(7000);

//fullScreen

driver.manage().window().fullscreen();

Thread.sleep(7000);

//minimize

driver.manage().window().minimize();

Thread.sleep(7000);

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

Thread.sleep(7000);

driver.quit();

}

}

**How to change the browser dimension? VVVVIMMMPPPP**

if we want to change the browser dimension then we use Dimension class.

we have to create object of Dimension class by passing width and height of the browser

syntax;

Dimension d new Dimension(400,500);

then we use setSize()method by passing Dimension class object name.

syntax;

driver.manage().window().setSize(d);

manage() method present inside the WebDriver interface

return type is Options interface.

window() method present inside the Options interface

return type is Window interface

setSize() method present inside the Window interface

return type is void.

Scenario;

Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driever","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open google url

driver.get("https://www.google.com");

step 4: change the browser dimension

Dimension d new Dimension(400,500);

driver.manage().window().setSize(d);

package Tutorial3;

import org.openqa.selenium.Dimension;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo2 {

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

// Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// up casting

WebDriver driver new ChromeDriver();

// open a google url

driver.get("https://www.google.com");

Thread.sleep(5000);

// change browser dimension

Dimension d new Dimension(2000, 500);

driver.manage().window().setSize(d);

}

}

**How to open browser in incognito mode? VVVVVIMMMPPPPPPPPPPP**

first we have to create object of ChromeOptions class.

we use addArguments() method from ChromeOptions class by passing --incognito keyword

then we have pass ChromeOptions class object name inside the ChromeDriver constructor.

syntax:

ChromeOptions opt new ChromeOptions();

opt.addArguments("--incognito");

WebDriver driver new ChromeDriver(opt);

step 1:connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: Create Object ChromeOptions class

ChromeOptions opt new ChromeOptions();

step 3: use addArguments() method

opt.addArguments("--incognito");

step 4: up casting concept by passing ChromeOptions class instance name / object name

WebDriver driver new ChromeDriver(opt);

step 5: open a facebook url

driver.get("https://www.facebook.com");

package Tutorial3;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

public class Demo3 {

public static void main(String[] args) {

// Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Step 2: create Object of ChromeOptions class

ChromeOptions opt new ChromeOptions();

// step 3: use addArguments() method and by passing --incognito keyword

opt.addArguments("--incognito");

// step 4: up casting and by passing ChromeOptions class object name in

// ChromeDriver constructor

WebDriver driver new ChromeDriver(opt);

driver.get("https://www.facebook.com");

}

}

**5) findElement() method**

- it is used to find the element/object in webpage by using given locator.

- return type of findElement() method is WebElement interface.

- findElement() method present inside the WebDriver interface as well as WebElement interface.

syntax;

WebElement wb driver.findElement(By.id("id value"));

WebElement wb driver.findElement(By.name("name value"));

WebElement wb driver.findElement(By.className("className value"));

WebElement wb driver.findElement(By.linkText("linkText value"));

WebElement wb driver.findElement(By.partialLinkText("partialLinkText value"));

WebElement wb driver.findElement(By.xpath("xpath value"));

WebElement wb driver.findElement(By.cssSelector("cssSelector value"));

WebElement wb driver.findElement(By.tagName("tagName value"));

-

**WebElement Interface**

1) sendKeys() method

- it used to send text to the text box

- return type is void

- this method present inside the WebElement interface.

Scenario:

open a facebook login page url

find the username text box using findElement() method and name locator

enter the text in text box

step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://www.facebook.com");

step 4: find the username text box using findElement() method and using name locator

WebElement wb driver.findElement(By.name("email"));

step 5: enter the text using sendKeys() method

wb.sendKeys("rahul@gmail.com");

package Tutorial3;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo4 {

public static void main(String[] args) {

// Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// up casting

WebDriver driver new ChromeDriver();

// open facebook url

driver.get("https://www.facebook.com");

//find the username text box using findElement() method and using name locator

WebElement wb driver.findElement(By.name("email"));

//enter the text using sendKeys()method

wb.sendKeys("dhanshri@gmail.com");

}

}

1) Open a https://demo.guru99.com/test/newtours/register.php

2) find the first name text box by using findElement() method and using name locator

3) enter the first name

4) find the last name text box by using findElement() method and using name locator

5) enter the last name

6) find the phone text box by using findElement() method and using name locator

7) enter the phone

8) find the email text box by using findElement() method and using name locator

9) enter email id

10) find the address text box by using findElement() method and using name locator

11) enter the address

12) find the city text box by using findElement() method and using name locator

13) enter the city

14) find the state text box by using findElement() method and using name locator

15) enter the state

16) find the pincode text box by using findElement() method and using name locator

17) enter the pincode

18) find the username text box by using findElement() method and using name locator

19) enter the username

20) find the password text box by using findElement() method and using name locator

21) enter the password

22) find the submit text box by using findElement() method and using name locator

WebElement wb1 driver.findElement(By.name("submit"));

23) click on submit

wb1.click();

package Tutorial3;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo5 {

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

// Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// up casting

WebDriver driver new ChromeDriver();

driver.get("https://demo.guru99.com/test/newtours/register.php");

WebElement wb1 driver.findElement(By.name("firstName"));

wb1.sendKeys("Alfiya");

WebElement wb2 driver.findElement(By.name("lastName"));

wb2.sendKeys("Shaikh");

Thread.sleep(5000);

WebElement wb3 driver.findElement(By.name("submit"));

wb3.click();

}

}

1) Open facebook url https://www.facebook.com/ -----> don't share code in group--->

2) find the username text box using findElement() method and by using name locator

3) enter the username

4) find the password text box using findElement() method and by using name locator

5) enter the password

6) find the login button using findElement() method and by using name locator

7) click on login button

-

**What are the different locator in selenium?** VVVVIMMMPPPP

there are 8 types of locator in selenium

1) id()

2) name()

3) className()

4) tagName()

5) linkText()

6) partialLinkText()

7) cssSelector()

8) xpath()

**which is fasted locator in selenium? VVVVIMMMPPPP**

**-id is faster loaded in selenium**

**what is use of locator?**

- locator are used to locate the element/object in webpage uniquely.

- locators are the html properties of web elements/objects.

**why we need different locator?**

- developer may not provide all locators for all the web elements or objects.

- sometimes some locators are duplicates.

- so we have choose anyone unique locators to recognize the object/element in webpage.

----------

what are the different operation we can perform on text box?

find the text box using findElement() method and using locators.

-verify text box is displayed or not. isDisplayed() boolean

-verify Text box is enabled or not. isEnabled() boolean

- Enter text in Text box sendKeys() void

- capture the entered Text/value getAttribute("value") String

- clear the text box clear() void

----------

what are the different operation we can perform on Radio button?

find the Radio button using findElement() method and using any locator?

-verify Radio button is displayed or not isDisplayed() boolean

-verify radio button is enabled or not? isEnabled() boolean

-verify Radio button is selected or not? isSelected() boolean

- click on radio button click() void

-verify Radio button is selected or not? isSelected() boolean

what are the different operation we can perform on checkbox?

find the checkbox using findElement() method and using given locator?

-verify checkbox is displayed or not isDisplayed() boolean

-verify checkbox is enabled or not isEnabled() boolean

-verify checkbox is selected or not isSelected() boolean

-click on checkbox click() void

-verify checkbox is selected or not isSelected() boolean

**WebElement Interface method**

1) sendKeys() method

- it is used to enter the text in text box

- return type of sendKeys() method is void,

- this method present inside the WebElement interface.

syntax:

WebElement wb driver.findElement(By.name("name value"));

wb.sendKeys("Dipali");

**2) isDisplayed() method**

- it used to check object /element is displayed or not in webpage.

or

- it used to check textbox/radio button/checkbox/link etc.. is displayed or not in webpage.

- return type is boolean

- this method is present inside the WebElement interface.

syntax:

WebElement wb driver.findElement(By.name("name value"));

boolean abc wb.isDisplayed();

System.out.println(abc);

----------

**3) isEnabled() method**

- it used to check element/object is enabled or not in webpage

- return type is boolean

- this method is present inside the WebElement interface.

syntax:

WebElement wb driver.findElement(By.name("name values"));

boolean abc wb.isEnabled();

System.out.println(abc);

**4) getAttribute() method**

- getAttribute()method is used to capture the entered text in text box or it used to capture the inner HTML attribute values by passing the key name.

- return type is String.

- this method is present inside the WebElement interface.

syntax:

WebElement wb driver.findElement(By.name("name value"));

String abc wb.getAttribute("value"); //entered text in text box

System.out.println(abc);

**5) clear() method**

- it is used to clear the values from text box

- return type is void

- this method is present inside the WebElement interface.

syntax:

WebElement wb driver.findElement(By.name("name value"));

wb.clear();

Scenario:

- open a facebook register page application get() method void

- find the first name text box findElement() method WebElement

- check first name text box displayed status isDisplayed() boolean

- check first name text box enabled status isEnabled() boolean

- enter the values in text box sendKeys() void

- capture the entered value getAttribute("value") String

- clear the values clear() void

Step 1: connect to the actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: Open a facebook register page application

driver.get("https://www.facebook.com/reg");

step 4: find the first name text box

WebElement wb driver.findElement(By.name("firstname"));

step 5: check first name text box displayed status

boolean a wb.isDisplayed();

System.out.println(a);

step 6: check first name text box enabled status

boolean b wb.isEnabled();

System.out.println(b);

step 7: enter the values in text box

wb.sendKeys("Poonam");

step 8: capture the entered value

String c wb.getAttribute("value");

System.out.println(c);

step 9: clear the values

wb.clear();

package Tutorial4;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

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

// step 1: connect the actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open fb register page url

driver.get("https://www.facebook.com/reg");

// step 4: find the first name text box

WebElement wb driver.findElement(By.name("firstname"));

// step 5: verify the first name text box displayed status

boolean a wb.isDisplayed();

System.out.println(a);// true

// step 6: verify the first name text box enabled status

boolean b wb.isEnabled();

System.out.println(b);

// step 7: enter the text in first name text box

wb.sendKeys("Supriya");

// step 8: captured the entered values from first name txt box

String c wb.getAttribute("value");

System.out.println(c);

// Wait for 7 seconds

Thread.sleep(7000);

// step 9: clear the text box

wb.clear();

// Wait for 7 seconds

Thread.sleep(7000);

// step 10: close the session

driver.quit();

}

}

Scenario for radio button:

- open a https://copyright.gov.in/UserRegistration/frmNewUser.aspx get() method void

- find the General User Radio button findElement() method WebElement

- check the General User Radio button displayed status isDisplayed() boolean

- check the General User Radio button enabled status isEnabled() boolean

- check the General User Radio button selected status isSelected() boolean

- click on General User Radio button click() void

- check the General User Radio button selected status isSelected() boolean

- find the Society User radio button findElement() WebElement

- click on Society User radio button click() void

- check the General User Radio button selected status isSelected boolean

step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open https://copyright.gov.in/UserRegistration/frmNewUser.aspx

driver.get("https://copyright.gov.in/UserRegistration/frmNewUser.aspx");

step4 : find the General User Radio button

WebElement wb driver.findElement(By.id("ctl00\_ContentPlaceHolder1\_rdoGeneral"));

step 5: check the General User Radio button displayed status

boolean a wb.isDisplayed();

System.out.println(a);

step 6: check the General User Radio button enabled status

boolean b wb.isEnabled();

System.out.println(b);

step 7: check the General User Radio button selected status

boolean c wb.isSelected();

System.out.println(c);

step 8: click on General User Radio button

wb.click();

step 9: check the General User Radio button selected status

boolean d wb.isSelected();

System.out.println(d);

step 10: find the Society User radio button

WebElement wb1 driver.findElement(By.id("ctl00\_ContentPlaceHolder1\_rdoSociety"));

step 11: click on Society User radio button

wb1.click();

step 12: check the General User Radio button selected status

boolean e wb.isSelected();

System.out.println(e);

package Tutorial4;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo2 {

public static void main(String[] args) {

// step 1: connect the actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://copyright.gov.in/UserRegistration/frmNewUser.aspx");

// step 4: find the general radio button

WebElement wb driver.findElement(By.id("ctl00\_ContentPlaceHolder1\_rdoGeneral"));

// step 5: check general radio button displayed status

boolean a wb.isDisplayed();

System.out.println(a);// true

// step 6: check general radio button enabled status

boolean b wb.isEnabled();

System.out.println(b);// true

// step 7: check general radio button selected status

boolean c wb.isSelected();

System.out.println(c);// true--- default general radio button is selected

// step 8: find the society radio button

WebElement wb1 driver.findElement(By.id("ctl00\_ContentPlaceHolder1\_rdoSociety"));

// step 9: click on society radio button

wb1.click();

// step 10: check the general radio button selected status.

boolean d wb.isSelected();

System.out.println(d);// false

System.out.println("-");

/// step 11: society radio button displayed,enabled and selected status

System.out.println(wb1.isDisplayed());// true

System.out.println(wb1.isEnabled());// true

System.out.println(wb1.isSelected());// true

}

}

Scenario for checkbox:

open a url https://demo.automationtesting.in/Register.html get() method void

- find the Cricket checkbox findElement() WebElement

- Cricket checkbox displayed status isDisplayed() boolean

- Cricket checkbox enabled status isEnabled() boolean

- Cricket checkbox selected status isSelected() boolean

- click Cricket checkbox click() void

- Cricket checkbox selected status isSelected() boolean

- find the Movies checkbox findElement() WebElement

- click on movies checkbox click void

- Cricket checkbox selected status isSelected boolean

- Movies checkbox displayed status isDisplayed boolean

- Movies checkbox enabled status isEnabled boolean

- Movies checkbox selected status isSelected boolean

step 1: connect to the actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

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

step 4: find the Cricket checkbox

WebElement wb driver.findElement(By.id("checkbox1"));

step 5: Cricket checkbox displayed status

boolean a wb.isDisplayed();

step 6: Cricket checkbox enabled status

boolean b wb.isEnabled();

step 7: Cricket checkbox selected status

boolean c wb.isSelected();

step 8: click Cricket checkbox

wb.click();

step 9: find the Movies checkbox

WebElement wb1 driver.findElement(By.id("checkbox2"));

step 10: Cricket checkbox selected status

boolean d wb.isSelected();

step 11: Movies checkbox displayed status

boolean e wb1.isDisplayed();

step 12: Movies checkbox enabled status

boolean f wb1.isEnabled();

step 13: Movies checkbox selected status

boolean g wb1.isSelected();

package Tutorial4;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo3 {

public static void main(String[] args) {

// step 1: connect the actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

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

// step 4: find the cricket checkbox

WebElement wb driver.findElement(By.id("checkbox1"));

/// step 5: check the cricket check box displayed status

boolean a wb.isDisplayed();

System.out.println(a);//true

/// step 6: check the cricket check box enabled status

boolean b wb.isEnabled();

System.out.println(b);//true

//step 7: check the cricket checkbox selected status

boolean c wb.isSelected();

System.out.println(c);//false

///step 8:click on cricket checkbox

wb.click();

//step 9:check the cricket checkbox selected status

boolean d wb.isSelected();

System.out.println(d);//true

//step 10: find the movies checkbox

WebElement wb1 driver.findElement(By.id("checkbox2"));

//step 11: click on movies checkbox

wb1.click();

//step 12: check cricket checkbox selected status

boolean f wb.isSelected();

System.out.println(f);//true

//step 13:movies checkbox displayes

boolean g wb1.isDisplayed();

System.out.println(g);

//step 14: movies checkbox enabled status

boolean h wb1.isEnabled();

System.out.println(h);

//step 15: movies check box selected status

boolean i wb1.isSelected();

System.out.println(i);

}

}

-

https://prafulp1030.github.io/AllHtmlElement.html

https://demo.automationtesting.in/Register.html

https://www.saucedemo.com/inventory.html

open a url https://www.saucedemo.com/inventory.html

find username

- check displayed status

- check enabled status

- enter value

- capture the entered value

find password

- check displayed status

- check enabled status

- enter value

- capture the entered value

find the login button

- check displayed status

- check enabled status

- click on button

- capture the home page title and url

- find the Sauce Labs Backpack add to cart

- click on it

- find the Sauce Labs Bike Light add to cart

- click on it

------

https://copyright.gov.in/UserRegistration/frmNewUser.aspx

https://accounts.digilocker.gov.in/signup/smart\_v2/1bad09c305565fabf8122991da6bd537--en

1. https://demo.guru99.com/test/newtours/register.php

2. https://demoqa.com/automation-practice-form

3. https://www.facebook.com/reg

4. https://www.saucedemo.com/

5. https://opensource-demo.orangehrmlive.com/web/index.php/auth/login

6. https://ui.cogmento.com/

7. https://www.salesforce.com/in/form/signup/freetrial-sales/?dtopnav2-btn-ft

8. https://www.india.gov.in/user/register

9. https://portal2.passportindia.gov.in/AppOnlineProject/user/RegistrationBaseAction?request\_localeen

10. https://oag.iitk.ac.in/Oa\_Rec\_Pg/NewUser.do

11. https://www.nseindia.com/portfolioTracker/content/mynse/registration.htm

12. https://apps2.fcc.gov/fccUserReg/pages/createAccount.htm ------> Pratiksha [26th dec Evening]

13. https://www.indiapost.gov.in/VAS/pages/userregistration.aspx

14. https://www.goibibo.com/trains/create-irctc-account/

15. https://www.automationanywhere.com/new-user-registration

16. https://register.usp.org/register/public/NewUser

17. https://www.microfocus.com/selfreg/jsp/createAccount.jsp

18. https://cds.climate.copernicus.eu/user/register > [ Om-- 26th Dec 26th Dec [Morning]

19. https://www.name.com/account/create

20. https://dynamicforms.ngwebsolutions.com/Account/CreateAccount

21. https://indiankanoon.org/members/signup/

22. https://opcfoundation.org/register -----> [ Monika-- 26th Dec [Morning] \*\*\*\*\*\*\*\*\* later

23. https://visionease.com/support/create-online-account/

24. https://visionease.com/support/create-online-account/ ---------->

25. https://edistrict.kerala.gov.in/registerPortalUser.do

26. https://edistrict.kerala.gov.in/registerPortalUser.do

27. https://accounts.digilocker.gov.in/signup/smart\_v2/1bad09c305565fabf8122991da6bd537--en

28. https://www.vsp.com/create-account -> without select tag drop down

29. https://copyright.gov.in/UserRegistration/frmNewUser.aspx

alert Popup URL:

https://demoqa.com/alerts

https://mail.rediff.com/cgi-bin/login.cgi

https://rahulshettyacademy.com/AutomationPractice/

Actions class link

https://demoqa.com/buttons

https://demo.guru99.com/test/simple\_context\_menu.html

https://swisnl.github.io/jQuery-contextMenu/demo.html

https://www.amazon.com/

http://www.dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html

https://jqueryui.com/droppable/

https://prafpawar11.github.io/

https://rahulshettyacademy.com/AutomationPractice/

https://testkru.com/Interactions/KeyboardActions

https://demoqa.com/buttons

https://demo.guru99.com/test/simple\_context\_menu.html

https://swisnl.github.io/jQuery-contextMenu/demo.html

https://www.amazon.com/

http://www.dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html

https://jqueryui.com/droppable/

https://prafpawar11.github.io/

https://rahulshettyacademy.com/AutomationPractice/

https://selenium08.blogspot.com/2019/11/double-click.html

https://www.qafeast.com/demo

https://testkru.com/Elements/Buttons

https://kybarg.github.io/bootstrap-dropdown-hover/

Frames:

https://chercher.tech/practice/frames-example-selenium-webdriver

https://demo.automationtesting.in/Frames.html

https://www.hyrtutorials.com/p/frames-practice.html

https://praf002.github.io/

https://prafpawar11.github.io/

**What are the different types of locators in selenium WebDriver?**

**1)id()**

**2) name()**

**3) className()**

**4) tagName()**

**5) linkText()**

**6) partialLinkText()**

**7) xpath()**

**8) cssSelector()**

What are the different operation we can perform on link?

-find the link in Web page using findElement() method and using given locators

- verify link is displayed or not

- verify link is enabled or not

- capture the link text

- click on link

What are the different operation we can perform on button?

- find the button in web page using findElement() method and using given locators

- verify button is displayed or not

- verify button is enabled or not

- capture the button text

- click on button

**What are the different operation we can perform on drop down? VVVVVIMMPPPPPP**

or

How to handle drop down in selenium?

or

how to verify specific value is present or not in dropdown

- find the drop down using findElement() and using given locators

- verify drop down is displayed or not

- verify the drop down is enabled or not

- select the values from drop down

- capture all drop down values

- check specific value present or not in drop down

**1) getText() method VVVVVVVVIMMMPPPPP**

- it used to capture physical text in webpage

- return type is String

- this method present inside the WebElement interface.

syntax:

WebElement wb driver.findElement(By.id("id value"));

String a wb.getText();

System.out.println(a);

----

What are the different operation we can perform on link?

- find the link in web page using findElement() method and using given locator findElement() WebElement

-verify the link is displayed or not isDisplayed() boolean

- verify the link is enabled or not? isEnabled() boolean

- capture the link Text getText() String

- click on link click() void

----------

Scenario:

- open a fb register page url

-find the link in web page using findElement() method and using given locator

-verify the link is displayed or not

- verify the link is enabled or not?

- capture the link Text

- click on link

Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://www.facebook.com/reg");

step 4: find the link using findElement() method and given locators

WebElement wb driver.findElement(By.linkText("Already have an account?"))

or

WebElement wb driver.findElement(By.partialLinkText("account?"));

step 5: verify the link in displayed or not

boolean a wb.isDisplayed();

System.out.println(a);

step 6: verify the link in enabled or not

boolean b wb.isEnabled();

System.out.println(b);

step 7: capture the link text

String c wb.getText();

System.out.println(c);

step 8: click on link

wb.click();

-

package Tutorial5;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

public static void main(String[] args) {

// connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a fb register page url

driver.get("https://www.facebook.com/reg");

// step 4: find the link in web page using findElement() and given locators

WebElement wb driver.findElement(By.linkText("Already have an account?"));

// step 5: verify the link is displayed or not

boolean a wb.isDisplayed();

System.out.println(a);

// step 6: verify the link is enabled or not?

boolean b wb.isEnabled();

System.out.println(b);

// step 7: capture the link text or physical text

String c wb.getText();

System.out.println(c);

}

}

package Tutorial5;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

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

// connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a fb register page url

driver.get("https://www.facebook.com/reg");

// step 4: find the link in web page using findElement() and given locators

WebElement wb driver.findElement(By.partialLinkText("account?"));

// step 5: verify the link is displayed or not

boolean a wb.isDisplayed();

System.out.println(a);

// step 6: verify the link is enabled or not?

boolean b wb.isEnabled();

System.out.println(b);

// step 7: capture the link text or physical text

String c wb.getText();

System.out.println(c);

Thread.sleep(5000);

// step 8: click on link

wb.click();

}

}

**diff between getText() and getAttribute() method**  VVVVVVVVVVVVIMMMPPPPPPP

- getText() method is used to capture the physical text in webpage

- getAttribute()method is used to capture the inner HTML attribute in web page.

- getText() method does not accept the any arguments

- getAttribute() method accept the String type of arguments

- return type of getText() method is String

- return type of getAttribute() method is String

- both method present inside the WebElement interface.

**What are the different operation we can perform on button?**

- find the button in web page using findElement() and using given locators findElement() WebElement

- verify the button is displayed or not? isDisplayed() boolean

- verify the button is enabled or not isEnabled() boolean

- capture the button physical text

Note:

I) if button tagName start with button then we capture the values using getText() method getText() String

2) if button tagName start with input then we can capture the button values using getAttribute() method getAttribute() String

- click on button click() void

Scenario for button

- open a facebook url

- find the button using findElement() method and using given locators

- verify the button is displayed or not?

- - verify the button is enabled or not

- capture the button physical text

- click on button

Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

Step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a facebook url

driver.get("https://www.facebook.com/reg");

step 4: - find the button using findElement() method and using given locators

WebElement wb driver.findElement(By.name("websubmit"));

step 5: - verify the button is displayed or not?

boolean a wb.isDisplayed();

step 6: - - verify the button is enabled or not

boolean b wb.isEnabled();

step 7: - capture the button physical text

String c wb.getText();

step 8: - click on button

wb.click();

package Tutorial5;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo2 {

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

// connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a fb register url

driver.get("https://www.facebook.com/reg");

// step 4:find the button in webpage

WebElement wb driver.findElement(By.name("websubmit"));

// step 5: verify the displayed status

boolean a wb.isDisplayed();

System.out.println(a);

// step 6: verify the button enabled status

boolean b wb.isEnabled();

System.out.println(b);

// step 7: capture the button physical text

String c wb.getText();

System.out.println(c);

// wait for 5 seconds

Thread.sleep(5000);

// step 8: click on button

wb.click();

}

}

**How to handle drop down in selenium VVVVVVIMMMPPPPPP**

**or**

**how to verify the specific value is present or not in drop down**

**or**

**capture the all the drop down values and print as per the ascending order?**

**or**

**what different operation we can perform on drop down?**

- find the drop down in web page using findElement() and using given locator

WebElement wb driver.findElement(By.id("month"));

note:

- if drop down tag name starts with select tag in html code, then we handle drop down using Select class from selenium webdriver.

- we have to create Object of Select class by passing drop down WebElement instance name or object name

Select sel new Select(wb);

- we can select the drop down values by using 3 ways

**I) by using selectByVisibleText() method**

**2) by using selectByValue() method**

**3) by using selectByIndex() method**

- capture all the drop down values

Scenario 1:

Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2:up casting

WebDriver driver new ChromeDriver();

step 3:open fb register url

driver.get("https://www.facebook.com/reg");

step 4: find the month drop down

WebElement wb driver.findElement(By.id("month"));

step 5: create object of Select class by passing the WebElement instance name or Object name

Select sel new Select(wb);

step 6: select the Nov month using selectByVisibleText() method

sel.selectByVisibleText("Nov");

package Tutorial5;

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.support.ui.Select;

public class Demo3 {

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

// connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a fb register url

driver.get("https://www.facebook.com/reg");

// step 4: find the months drop down

WebElement wb driver.findElement(By.id("month"));

// step 5: create Object of Select class by passing WebElement instance name or

// object name

Select sel new Select(wb);

Thread.sleep(5000);

// step 6: select the Nov month using selectByVisibleText() method

sel.selectByVisibleText("Nov");

}

}

Scenario 1:

Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2:up casting

WebDriver driver new ChromeDriver();

step 3:open fb register url

driver.get("https://www.facebook.com/reg");

step 4: find the month drop down

WebElement wb driver.findElement(By.id("month"));

step 5: create object of Select class by passing the WebElement instance name or Object name

Select sel new Select(wb);

step 6: select the Aug month using selectByValue() method

sel.selectByValue("8");

package Tutorial5;

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.support.ui.Select;

public class Demo4 {

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

// connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a fb register url

driver.get("https://www.facebook.com/reg");

// step 4: find the months drop down

WebElement wb driver.findElement(By.id("month"));

// step 5:create object of Select class by passing WebElement instance name or

// object name

Thread.sleep(5000);

Select sel new Select(wb);

//step 6: select the Aug month using selectByValue() method

sel.selectByValue("8");

}

}

Scenario 1:

Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2:up casting

WebDriver driver new ChromeDriver();

step 3:open fb register url

driver.get("https://www.facebook.com/reg");

step 4: find the month drop down

WebElement wb driver.findElement(By.id("month"));

step 5: create object of Select class by passing the WebElement instance name or Object name

Select sel new Select(wb);

step 6: select the Mar month using selectByIndex() method

sel.selectByIndex(2);

package Tutorial5;

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.support.ui.Select;

public class Demo5 {

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

// connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a fb register url

driver.get("https://www.facebook.com/reg");

// step 4: find the months drop down

WebElement wb driver.findElement(By.id("month"));

// step 5:create Object of Select class by passing WebElement instance name or

// object name

Select sel new Select(wb);

Thread.sleep(5000);

// step 6:select drop down values using selectByIndex() method

sel.selectByIndex(10);

}

}

**check how many values present in dropdown**

Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2:up casting

WebDriver driver new ChromeDriver();

step 3:open fb register url

driver.get("https://www.facebook.com/reg");

step 4: find the month drop down

WebElement wb driver.findElement(By.id("month"));

step 5: create object of Select class by passing the WebElement instance name or Object name

Select sel new Select(wb);

step 6: capture all drop down values using getOptions() method

List<WebElement> ls sel.getOptions();

Step 7: check how many values present in dropdown

int abc ls.size();

System.out.println(abc);

package Tutorial5;

import java.util.List;

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.support.ui.Select;

public class Demo6 {

public static void main(String[] args) {

// connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a fb register url

driver.get("https://www.facebook.com/reg");

// step 4: find the year drop down

WebElement wb driver.findElement(By.id("year"));

//step 5: create Object of Select class by passing WebElement instance or Object name

Select sel new Select(wb);

//step 6: capture the all drop down values using getOptions() method

List<WebElement> ls sel.getOptions();

//step 7: check how many values present in dropdown using size() method

int abc ls.size();

System.out.println(abc);

}

}

print all down values dropdown

Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2:up casting

WebDriver driver new ChromeDriver();

step 3:open fb register url

driver.get("https://www.facebook.com/reg");

step 4: find the month drop down

WebElement wb driver.findElement(By.id("month"));

step 5: create object of Select class by passing the WebElement instance name or Object name

Select sel new Select(wb);

step 6: capture all drop down values using getOptions() method

List<WebElement> ls sel.getOptions();

Step 7: print all dropdown value

for(WebElement abc: ls)

{

String a abc.getText();

System.out.println(a);

}

package Tutorial5;

import java.util.List;

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.support.ui.Select;

public class Demo7 {

public static void main(String[] args) {

// connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a fb register url

driver.get("https://www.facebook.com/reg");

// step 4: find the day drop down

WebElement wb driver.findElement(By.id("month"));

// step 5: create object of Select class by passing WebElement instance name

Select sel new Select(wb);

// step 6: capture all the drop down values using getOptions method

List<WebElement> ls sel.getOptions();

// step 7:print all drop down values

for (WebElement abc : ls) {

String a abc.getText();

System.out.println(a);

}

}

}

https://copyright.gov.in/UserRegistration/frmNewUser.aspx

https://www.facebook.com/reg

https://demo.guru99.com/insurance/v1/register.php

**How to handle drop down in selenium? VVVVVIMMPPPPP**

Select class Method

1) selectByVisibleText() method

2) selectByIndex() method

3) selectByValue() method

4) getOptions() method

5) getFirstSelectedOption() method

**1) selectByVisibleText() method**

- selectByVisibleText() method used to select the values from drop down.

- return type is void.

- this method present inside the Select class.

- this method it accept the String type arguments.

syntax:

WebElement wb driver.findElement(By.id("id value"));

Select sel new Select(wb);

sel.selectByVisibleText("drop down value");

-------

**2) selectByIndex() method**

-selectByIndex() method is used to select the values from drop down

- return type is void

- this method is present inside the Select class.

- this method accept the int type of arguments.

syntax:

WebElement wb driver.findElement(By.id("id values"));

Select sel new Select(wb);

sel.selectByIndex(index position);

-------

**3) selectByValue() method**

-selectByValue() method is used to select the values from drop down

- return type is void

- this method is present inside the Select class.

- this method it accept the String type of arguments.

syntax:

WebElement wb driver.findElement(By.id("id values"));

Select sel new Select(wb);

sel.selectByValue("drop down values from html code");

**4) getOptions() method**

- it is used to capture the all the drop down values.

- return type is List<WebElement>

- this method present inside the Select class.

syntax:

WebElement wb driver.findElement(By.id("id values"));

Select sel new Select(wb);

List<WebElement> ls sel.getOptions();

-------

Scenario:

- Open a url https://demo.guru99.com/test/newtours/register.php

- find the country drop down

- capture all drop down values.

- print in console

Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3:open a url

driver.get("https://demo.guru99.com/test/newtours/register.php");

step 4: find the country drop down

WebElement wb driver.findElement(By.name("country"));

step 5: create object of Select class by passing WebElement instance name or object name

Select sel new Select(wb);

step 6: capture all drop down values

List<WebElement> ls sel.getOptions();

step 7: iterate the loop and print all values in console.

for(WebElement abc: ls)

{

String a abc.getText();

System.out.println(a);

}

package Tutorial6;

import java.util.List;

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.support.ui.Select;

public class Demo1 {

public static void main(String[] args) {

// Step 1: connect browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://demo.guru99.com/test/newtours/register.php");

// step 4: find the drop down web element

WebElement wb driver.findElement(By.name("country"));

// step 5: create object of Select class by passing WebElement instance name or

// object name

Select sel new Select(wb);

// step 6: capture all drop down values

List<WebElement> ls sel.getOptions();

// step 7: iterate all drop down values and print in console

for (WebElement abc : ls) {

String a abc.getText();

System.out.println(a);

}

// step 8: count how many values is present in drop down

int b ls.size();

System.out.println(b);

}

}

https://demo.guru99.com/test/newtours/register.php

verify the ""INDIA""" value is present or not in country drop down VVVVVIMMMPPPP

step 1: connect browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2; up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://demo.guru99.com/test/newtours/register.php");

step 4: find the drop down

WebElement wb driver.findElement(By.name("country"));

step 5: create object of Select class by passing WebElement instance.

Select sel new Select(wb);

step 6: capture all drop down values

List<WebElement> ls sel.getOptions();

step 7: iterate all drop down values using for loop

for(WebElement abc: ls)

{

String a abc.getText();

//step 7a) use if condition to verify the INDIA values.

if(a.equals("INDIA"))

{

System.out.println("Test cases is pass");

}

}

package Tutorial6;

import java.util.List;

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.support.ui.Select;

public class Demo2 {

public static void main(String[] args) {

// Step 1: connect browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://demo.guru99.com/test/newtours/register.php");

// step 4: find the drop down web element

WebElement wb driver.findElement(By.name("country"));

// step 5: create object of Select class by passing WebElement instance

Select sel new Select(wb);

// step 6: capture the all drop down values

List<WebElement> ls sel.getOptions();

// step 7: iterate all drop down value using for loop

for (WebElement abc : ls) {

String a abc.getText();

// step 7a: verify the INDIA value is present or not

if (a.equals("INDIA")) {

System.out.println("test cases is pass " + a);

}

}

}

}

**5) getFirstSelectedOption() method**

- it is used to capture the selected values from drop down

- return type is WebElement interface.

- this method present inside the Select class.

syntax

WebElement wb driver.findElement(By.name("name value"));

Select sel new Select(wb);

sel.selectByVisibleText("text");

or

sel.selectByIndex(index);

or

sel.selectByValue("value");

WebElement abc sel.getFirstSelectedOption();

String a abc.getText();

System.out.println(a);

Scenario:

open a fb register url

select the Oct month value.

capture the selected values.

Step 1: connect to actual browser

step 2: up casting

step 3: open a url

step 4: find the drop down using findElement() method and using given locator.

WebElement wb driver.findElement(By.id("month"));

step 5: create object of Select class by passing WebElement instance name

Select sel new Select(wb);

step 6: select the drop down

sel.selectByVisibleText("Oct");

step 7: capture the selected values.

WebElement abc sel.getFirstSelectedOption();

String a abc.getText();

System.out.println(a);

package Tutorial6;

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.support.ui.Select;

public class Demo3 {

public static void main(String[] args) {

// Step 1: connect browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://www.facebook.com/reg");

// step4: find the drop down

WebElement wb driver.findElement(By.id("month"));

// step 5: create Object of Select class by passing WebElement instance

Select sel new Select(wb);

// step 6: select the drop down values

// sel.selectByVisibleText("Dec");

//sel.selectByIndex(7);

sel.selectByValue("9");

// step 7: capture the selected values

WebElement abc sel.getFirstSelectedOption();

String a abc.getText();

System.out.println(a);

}

}

**How to handle alert Pop in selenium? VVVVVVIMMMPPPPPPPPPP**

- we can not inspect the alert pop up object / elements.

- if u want to handle alert pop in selenium then first we have switch our focus on alert pop using switchTo().alert() method.

- alert() method is present inside the TargetLocator interface.

- return type of alert() method is Alert interface.

- Alert interface it define the 4 method which is used to handle the alert pop up.

**1) accept()**

- if u want to click on ok button then we use accept method

- return type is void.

- this method present inside the Alert interface.

syntax:

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

alt.accept();

**2) dismiss() method**

- if u want to click on cancel button then we have to use dismiss() method.

- return type is void.

- this method present inside the Alert interface.

syntax:

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

alt.dismiss();

**3) getText() method**

- if u want to capture the alert pop text then we use getText() method

- return type is String.

- this method present inside the Alert interface.

syntax:

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

String a alt.getText();

**4) sendKeys() method**

- if u want to enter the text in alert text box then we use sendKeys() method

- return type is void.

- this method is present inside the Alert interface.

syntax:

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

alt.sendKeys("enter text");

Scenario:

-open a https://demoqa.com/alerts

- find click me button findElement() method WebElement

- click on click me button click() void

- switch focus to alert window driver.switchTo().alert() Alert interface

- capture the alert pop up text getText() String

- click on ok button accept() void.

step 1: connect to actual browser

step 2: up casting

step 3: open a url

driver.get("https://demoqa.com/alerts");

step 4: find the click on button

WebElement wb driver.findElement(By.id("alertButton"));

step 5: click on click me button

wb.click();

step 6: switch focus to alert pop up

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

step 7: capture the alert pop physical text

String a alt.getText();

System.out.println(a);

step 8: click on ok button

alt.accept();

package Tutorial6;

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;

public class Demo4 {

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

// Step 1: connect browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://demoqa.com/alerts");

Thread.sleep(5000);

// step 4: find the click me button

WebElement wb driver.findElement(By.id("alertButton"));

// step 5: click on click me button

wb.click();

// step 6: switch focus to alert pop up window

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

/// step 7: capture the alert pop up physical text

String a alt.getText();

System.out.println(a);

Thread.sleep(5000);

// step 8: click on ok button

alt.accept();

}

}

Scenario:

-open a https://demoqa.com/alerts

- find 4th click me button WebElement findElement()

- click on click me button void click()

- switch focus to alert window Alert interface driver.switchTo().alert()

- enter the text in alert pop up text void sendKeys

- click on cancel void dismiss()

Step 1: connect to actual browser

step 2: up casting

step 3: open a url

driver.get("https://demoqa.com/alerts");

step 4: find the 4th click on button

WebElement wb driver.findElement(By.id("promtButton"));

step 5: click on 4th click on button

wb.click();

step 6: switch focus to alert pop up window

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

step 7: enter the text in alert pop up window

alt.sendKeys("selenium");

step 8: click on cancel button

alt.dismiss();

package Tutorial6;

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;

public class Demo5 {

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

// Step 1: connect browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://demoqa.com/alerts");

// step 4: find the 4th click on button

WebElement wb driver.findElement(By.id("promtButton"));

// step 5: click on 4th click on button

wb.click();

Thread.sleep(5000);

// step 6: switch focus to alert pop window

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

// step 7: enter the text in alert pop up text box

alt.sendKeys("selenium testing");

Thread.sleep(5000);

// step 8; click on cancel button

alt.dismiss();

}

}

Scenario:

open https://rahulshettyacademy.com/AutomationPractice/

enter name in text box

click on confirm button

- capture alert pop text

- and verify the name is present or not in captured text.

- click on ok button

https://demo.automationtesting.in/Alerts.html

**1) get() method**

- used to open a specified URL in browser window

- return type is void

- this method present inside the WebDriver interface,

**diff between Relative xpath and absolute xpath? VVVVVVIMMMPPPPPPPPPPPPPPPP**

**at least 2 or 3 questions on real time website to find the xpath using Relative xpath? VVVVVVIMMMPPPPPPPPPPPPPPPP**

Xpath **VVVVVVIMMMPPPPPPPPPPPPPPPP**

**there are 2 types of xpath**

**1) absolute xpath**

**2) Relative xpath**

-

**2) Relative xpath**

- in relative xpath we never mention the long xpath

- relative xpath is starts with double forward slash (//).

- Relative xpath is also known as custom xpath or own xpath.

- we can start searching the element from middle of the DOM(Document Object Model) HTML structure.

Single attribute Relative xpath

syntax:

//tagname[@attribute'value']

e.g.

//input[@name'firstname']

WebElement wb driver.findElement(By.xpath("//input[@name'firstname']"));

package Tutorial7;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

public static void main(String[] args) {

WebDriver driver new ChromeDriver();

driver.get("https://www.facebook.com/reg");

WebElement wb driver.findElement(By.xpath("//input[@name'firstname']"));

wb.sendKeys("selenium testing");

}

}

-

Facebook year dropdown xpath

1) //select[@aria-label'Year']

2) //select[@name'birthday\_year']

3) //select[@id'year']

4) //select[@title'Year']

WebElement wb driver.findElement(By.xpath("//select[@id'year']"));

package Tutorial7;

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.support.ui.Select;

public class Demo1 {

public static void main(String[] args) {

WebDriver driver new ChromeDriver();

driver.get("https://www.facebook.com/reg");

WebElement wb driver.findElement(By.xpath("//select[@id'year']"));

Select sel new Select(wb);

sel.selectByVisibleText("2001");

}

}

-

2) relative xpath using text() method

syntax;

//tagname[text()'text value']

e.g.

//div[text()'Login']

https://ui.cogmento.com/

login button

1) //div[@class'ui fluid large blue submit button']

2) //div[text()'Login']

WebElement wb driver.findElement(By.xpath("//div[text()'Login']"));

package Tutorial7;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo2 {

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

WebDriver driver new ChromeDriver();

driver.get("https://ui.cogmento.com/");

Thread.sleep(5000);

WebElement wb driver.findElement(By.xpath("//div[text()'Login']"));

wb.click();

}

}

forgot button link

1) //a[@href'https://register.cogmento.com/password/reset/request/?langen-GB']

2) //a[text()'Forgot your password?']

3) //a[contains(text(),'password')]

3) relative xpath using contains() and text()method

syntax:

//tagname[contains(text(),'text value')]

e.g.

//a[contains(text(),'password')]

package Tutorial7;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo2 {

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

WebDriver driver new ChromeDriver();

driver.get("https://ui.cogmento.com/");

Thread.sleep(5000);

WebElement wb driver.findElement(By.xpath("//a[contains(text(),'password')]"));

wb.click();

}

}

https://demo.guru99.com/test/newtours/register\_sucess.php

1) //a[@href'login.php']

2) //a[text()' sign-in ']

3)//a[contains(text(),'sign-in')]

WebElement wb driver.findElement(By.xpath("//a[contains(text(),'sign-in')]"));

package Tutorial7;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo3 {

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

WebDriver driver new ChromeDriver();

driver.get("https://demo.guru99.com/test/newtours/register\_sucess.php");

Thread.sleep(5000);

WebElement wb driver.findElement(By.xpath("//a[contains(text(),'sign-in')]"));

wb.click();

}

}

Relative xpath with contains() method with attribute

//tagname[contains(@attribute,'value')]

e.g.

//input[contains(@name,'email\_\_')]

Username text box in facebook register page

1) //input[contains(@name,'email\_\_')]

2) //input[contains(@aria-label,'Mobile')]

3) //input[contains(@id,'u\_0\_g')]

package Tutorial7;

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.support.ui.Select;

public class Demo1 {

public static void main(String[] args) {

WebDriver driver new ChromeDriver();

driver.get("https://www.facebook.com/reg");

WebElement wb driver.findElement(By.xpath("//input[contains(@id,'u\_0\_g')]"));

wb.sendKeys("Selenium@test.com");

}

}

Relative xpath with 2 attribute

1) //tagname[@attribute'value'][@attribute'value']

2) //tagname[text()'value'][@attribute'value']

3) //tagname[@attribute'value'][text()'value']

4) //tagname[contains(@attribute,'value')][@attribute'value']

5) //tagname[contains(text(),'value')][@attribute'value']

Password text box in fb Register page

1) //input[@id'password\_step\_input'][@name'reg\_passwd\_\_']

2) //input[@id'password\_step\_input'][@type'password']

3) //input[contains(@id,'password')][@name'reg\_passwd\_\_']

4) //input[contains(@id,'password')][contains(@name,'reg')]

forgot password link in fb register page

1) //a[@aria-label'Already have an account?'][text()'Already have an account?']

2) //a[contains(@aria-label,'Already')][text()'Already have an account?']

3) //a[contains(@aria-label,'Already')][contains(text(),'account')]

or operator: if any one condition is true then result is true

and operator if both condition is true then result is true

Relative xpath with 2 attribute with OR operator

1) //tagname[@attribute'value' or @attribute'value']

2) //tagname[@attribute'value' or text()'text value']

3) //tagname[text()'value' or @attribute'value']

4) //tagname[contains(@attribute,'value') or @attribute'value']

5) //tagname[contains(text(),'text value') or @attribute'value']

Relative xpath with 2 attribute with AND operator

1) //tagname[@attribute'value' and @attribute'value']

2) //tagname[@attribute'value' and text()'text value']

3) //tagname[text()'value' and @attribute'value']

4) //tagname[contains(@attribute,'value') and @attribute'value']

5) //tagname[contains(text(),'text value') and @attribute'value']

facebook register page last name text box

1)//input[@name'lastname' and @type'text']

2) //input[@name'lastname' or contains(@id,'u\_0\_d')]

3) //input[@name'lastname' and contains(@id,'u\_0\_d')]

4) //input[@name'lastname' and @class'inputtext \_58mg \_5dba \_2ph-']

5) //input[@name'lastname' and @type'text']

6) //input[@aria-label'Surname' and contains(@name,'lastname')]

7) //input[@aria-label'Surname' or contains(@name,'lastname')]

WebElement wb driver.findElement(By.xpath("//input[@aria-label'Surname' and contains(@name,'lastname')]"));

package Tutorial7;

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.support.ui.Select;

public class Demo1 {

public static void main(String[] args) {

WebDriver driver new ChromeDriver();

driver.get("https://www.facebook.com/reg");

WebElement wb driver.findElement(By.xpath("//input[@aria-label'Surname' and contains(@name,'lastname')]"));

wb.sendKeys("Selenium@test.com");

}

}

Relative xpath with starts-with() method

syntax:

//tagname[starts-with(@attribute,'value')]

//tagname[starts-with(text(),'value')]

1) //a[starts-with(text(),'Already')]

1) //div[starts-with(@class,'mbs')]

//div[@class'FPdoLc lJ9FBc']/descendant::input[@name'btnK']

//div[@class'FPdoLc lJ9FBc']/child::center/input[@name'btnK']

//div[@class'FPdoLc lJ9FBc']//child::input[@name'btnK']

//div[@class'M8H8pb']/following::input[@name'btnK']

//div[@class'M8H8pb']/following-sibling::div/center/input[1]

//div[@class'M8H8pb']/following-sibling::div/center/input[@class'gNO89b']

//div[@class'o3j99 qarstb']/preceding::div[@class'FPdoLc lJ9FBc']/descendant::input[@name'btnK']

**Relative Xpath axes VVVVIMMMPPP**

1) following keyword

2) following-sibling keyword

3) preceding keyword

4) preceding-sibling keyword

5) parent keyword

6) child keyword

7) ancestor keyword

8) descendant keyword

Solve using only xpath

https://www.saucedemo.com/

enter valid credentials

capture title and url

click on add to cart

click on cart button

click on checkout button

enter firstname last name and zip code

click on continue button

click on finish button

click on back home

click on menu icon

click on logout

Solve using only xpath

open a https://opensource-demo.orangehrmlive.com/web/index.php/auth/login

enter valid credentials

capture title and url

click on PIM link

click on add employee button

enter firstname, last name and click on save button

capture the employee id using getText() method

select gender click on save button

click on employee list

enter the employee id

click on search button

select the checkbox

click on delete selected button

click on yes delete

click on profile

click on logout

**Relative xpath Axes in Selenium VVVVIMMMPPPPP**

1) following keyword

2) preceding keyword

3) parent keyword

4) child keyword

5) ancestor keyword

6) descendant keyword

7) following-sibling keyword

8) preceding-sibling keyword

**1) following keyword**

- it is sued to find all element after the current tags.

syntax:

//tagname[@attribute'value']/following::tagname[@attribute'value']

//tagname[text()'value']/following::tagname[@attribute'value']

//tagname[contains(text(),'value')]/following::tagname[@attribute'value']

//tagname[contains(text(),'value')]/following::tagname[index position]

//tagname[starts-with(text(),'value')]/following::tagname[index position]

Password text box in Facebook register page

1) //input[@name'reg\_email\_\_']/following::input[2]

2) //input[@name'reg\_email\_\_']/following::input[@id'password\_step\_input']

3) //input[@name'reg\_email\_\_']/following::input[@name'reg\_passwd\_\_']

4) //input[@name'reg\_email\_\_']/following::input[@type'password']

Year drop down in fb register page

1) //input[@name'reg\_email\_\_']/following::select[@name'birthday\_year']

2) //input[@name'reg\_email\_\_']/following::select[@id'year']

3) //input[@name'reg\_email\_\_']/following::select[@aria-label'Year']

4) //input[@name'reg\_email\_\_']/following::select[3]

sign button in facebook register page

1) //input[@name'reg\_email\_\_']/following::button[1]

2) //input[@name'reg\_email\_\_']/following::button[@name'websubmit']

3) //input[@name'reg\_email\_\_']/following::button[starts-with(@id,'u\_0\_s')]

**2) preceding keyword**

It will find all previous element before the current tags

syntax:

1) //tagname[@attribute'value']/preceding::tagname[@attribute'value']

2) //tagname[text()'value']/preceding::tagname[@attribute'value']

3) //tagname[contains(@attribute,'value')]/preceding::tagname[@attribute'value']

4) //tagname[contains(text(),'value')]/preceding::tagname[@attribute'value']

5) //tagname[starts-with(@attribute,'value')]/preceding::tagname[@attribute'value']

6) //tagname[starts-with(text(),'value')]/preceding::tagname[@attribute'value']

first name text box in facebook register page

1) //input[@name'reg\_email\_\_']/preceding::input[2]

2) //input[@name'reg\_email\_\_']/preceding::input[@name'firstname']

3) //input[@name'reg\_email\_\_']/preceding::input[starts-with(@id,'u\_0\_b')]

4) //input[@name'reg\_email\_\_']/preceding::input[@aria-label'First name']

fb logo in register page

1) //input[@name'reg\_email\_\_']/preceding::img

2) //input[@name'reg\_email\_\_']/preceding::img[@class'fb\_logo \_agiv img']

3) //input[@name'reg\_email\_\_']/preceding::img[@alt'Facebook']

package Tutorial8;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

public static void main(String[] args) {

WebDriver driver new ChromeDriver();

driver.get("https://www.facebook.com/reg");

WebElement wb driver

.findElement(By.xpath("//input[@name'reg\_email\_\_']/following::input[@name'reg\_passwd\_\_']"));

wb.sendKeys("Selenium testing");

}

}

package Tutorial8;

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.support.ui.Select;

public class Demo1 {

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

WebDriver driver new ChromeDriver();

driver.get("https://www.facebook.com/reg");

Thread.sleep(5000);

WebElement wb driver

.findElement(By.xpath("//input[@name'reg\_email\_\_']/following::select[@aria-label'Year']"));

Select sel new Select(wb);

sel.selectByVisibleText("2010");

}

}

package Tutorial8;

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.support.ui.Select;

public class Demo1 {

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

WebDriver driver new ChromeDriver();

driver.get("https://www.facebook.com/reg");

Thread.sleep(5000);

WebElement wb driver

.findElement(By.xpath("//input[@name'reg\_email\_\_']/following::button[starts-with(@id,'u\_0\_s')]"));

wb.click();

}

}

package Tutorial8;

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.support.ui.Select;

public class Demo1 {

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

WebDriver driver new ChromeDriver();

driver.get("https://www.facebook.com/reg");

Thread.sleep(5000);

WebElement wb driver.findElement(By.xpath("//input[@name'reg\_email\_\_']/preceding::input[2]"));

wb.sendKeys("Selenium testing");

}

}

----

package Tutorial8;

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.support.ui.Select;

public class Demo1 {

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

WebDriver driver new ChromeDriver();

driver.get("https://www.facebook.com/reg");

Thread.sleep(5000);

WebElement wb driver.findElement(By.xpath("//input[@name'reg\_email\_\_']/preceding::img"));

boolean a wb.isDisplayed();

System.out.println(a);

}

}

--

**3) parent keyword**

It is used to find the immediately parent tag of current tag

syntax:

//tagname[@attribute'value']/parent::tagName

//tagname[text()'value']/parent::tagName

//tagname[contains(@attribute,'value')]/parent::tagName

//tagname[starts-with(@id,'value')]/parent::tagName

1) //input[@id'password\_step\_input']/parent::div

--

**4) child keyword:**

it is used to find the immediately child tag of current tag

syntax:

syntax:

//tagname[@attribute'value']/child::tagName

//tagname[text()'value']/child::tagName

//tagname[contains(@attribute,'value')]/child::tagName

//tagname[starts-with(@id,'value')]/child::tagName

syntax:

//div[@id'reg\_form\_box']/child::div

--

5) ancestor keyword

- it is used to find the parent tag, grand parent tag, grand of grand parent tag of current tags,

syntax:

//tagname[@attribute'value']/ancestor::tagName

//tagname[text()'value']/ancestor::tagName

//tagname[contains(@attribute,'value')]/ancestor::tagName

//tagname[starts-with(@id,'value')]/ancestor::tagName

//input[@id'password\_step\_input']/ancestor::div

--

**6) descendant keyword:**

- it used to find the child tag, grand child, grand of grand child tag of current tags.

syntax:

//tagname[@attribute'value']/descendant::tagName

//tagname[text()'value']/descendant::tagName

//tagname[contains(@attribute,'value')]/descendant::tagName

//tagname[starts-with(@id,'value')]/descendant::tagName

//div[@id'reg\_box']/descendant::div

//div[@cel\_widget\_id'MAIN-SEARCH\_RESULTS-5']/descendant::span[@class'a-price-whole']

//div[@data-asin'B0CHX6N27Y']/descendant::span[contains(@aria-label,'FREE delivery ')]/span[@class'a-color-base a-text-bold']

//input[@id'twotabsearchtextbox']/following::span[@class's-heavy' and text()' cover']

//span[@class's-heavy' and text()' cover']

//textarea[@id'APjFqb']/following::div[@id'gLuoSb']/div[@class'wM6W7d']/span

//div[@class'FPdoLc lJ9FBc']/descendant::input[@name'btnK']

//div[@class'FPdoLc lJ9FBc']/descendant::input[@name'btnK']

//label[@class'oxd-label' and contains(text(),'License Number')]/parent::div/following-sibling::div/input

**How to handle frames in selenium? VVVVVVVVVVVVVIMPPPPPPPPPPPPPPPPPP**

**or**

**how to handle multiple frames in selenium? VVVVVVVVVVVVVIMPPPPPPPPPPPPPPPPPP**

**or**

**can we switch from child frame to child frame? VVVVVVVVVVVVVIMPPPPPPPPPPPPPPPPPP**

**or**

**diff between parentFrame() method and defaultContent() method? VVVVVVVVVVVVVIMPPPPPPPPPPPPPPPPPP**

**or**

**how many ways we can switch to child frame? VVVVVVVVVVVVVIMPPPPPPPPPPPPPPPPPP**

**Handle frames in Selenium VVVVVVVVVVVVVIMPPPPPPPPPPPPPPPPPP**

- HTML frame are used to divide the browser window into the multiple sections where each section can hold separate HTML web page.

- frame are sections of web page displayed on top window, whenever we access the pages then we by default we focus on top window or frame.

- there are 4 different ways we can switch to child frame

1) by using frame id

2) by using frame name

3) by using frame index

4) by using frame web element

- if u want to switch to main frame or window to child frame then we use switchTo().frame() method by passing frame id or frame name or frame index

position, or frame web element instance name.

- switchTo() method present inside the WebDriver interface and return type is TargetLocator interface.

- frame() method present inside the TargetLocator interface and """return type is WebDriver interface""".

**1) by using frame id**

syntax:

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

**2) by using frame name**

syntax:

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

**3) by using frame index position**

syntax:

driver.switchTo().frame(int frame index);

**4) by using frame web element**

syntax:

WebElement wb driver.findElement(By.id("id values"));

driver.switchTo().frame(frame web Element);

**How to switch focus from child frame to main frame or main window?**

- we use switchTo().defaultContent() method.

- switchTo() method present inside the WebDriver interface and return type is TargetLocator interface.

defaultContent() method is present inside the TargetLocator interface and return type is WebDriver interface.

**diff between parentFrame() method and defaultContent() method**

- defaultContent() method is used to switch our focus to main frame or top frame.

- parentFrame() method is used to switch focus to immediate parent frame.

- both method is present inside the TargetLocator interface and return type is WebDriver interface.

-

Scenario 1:

open a https://prafpawar11.github.io/mainFrame.html

enter the text in name text box

select the TestNG and POM checkbox

Step 1: connect to the browser code

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2:up casting

WebDriver driver new ChromeDriver();

step 3 open a url

driver.get("https://prafpawar11.github.io/mainFrame.html");

step 4: find the name text box and enter the values

WebElement wb driver.findElement(By.id("name"));

wb.sendKeys("Ajit");

step 5:switch focus to child frame or checkbox frame

driver.switchTo().frame("mainframe");

step 6: find the TestNG and POM checkbox and click on it.

WebElement wb1 driver.findElement(By.id("TestNG"));

wb1.click();

WebElement wb2 driver.findElement(By.id("POM"));

wb2.click();

package Tutorial9;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

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

// step 1: connect to the actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://prafpawar11.github.io/mainFrame.html");

Thread.sleep(5000);

// step 4: find the name text box and enter the values

WebElement wb driver.findElement(By.id("name"));

wb.sendKeys("ajit");

// step 5: switch focus to checkbox frame or child frame

// driver.switchTo().frame("mainframe");

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

WebElement framewb driver.findElement(By.xpath("//iframe[@id'mainframe']"));

driver.switchTo().frame(framewb);

// step 6: find the TestNG and POM check box and click on it

WebElement wb1 driver.findElement(By.xpath("//input[@id'TestNG']"));

wb1.click();

WebElement wb2 driver.findElement(By.xpath("//input[@id'POM']"));

wb2.click();

}

}

Scenario 2:

open a https://prafpawar11.github.io/twoFrame.html

enter the full name

select the Jenkins checkbox

select the BDD framework values from drop down

step 1: connect actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://prafpawar11.github.io/twoFrame.html");

step 4; find the full name text box and enter the values

WebElement wb driver.findElement(By.id("name"));

wb.sendKeys("abc xyz");

step 5: switch focus to checkbox frame or child frame1

driver.switchTo().frame("chk");

step 6: find the Jenkins checkbox and click on it

WebElement wb1 driver.findElement(By.id("Jenkins"));

wb1.click();

step 7: switch to the top frame or main frame or main window

driver.switchTo().defaultContent();

step 8: switch to drop down frame or child frame 2

driver.switchTo().frame(1);

step 9: find the drop down and select the BDD framework values

WebElement wb2 driver.findElement(By.id("course"));

Select sel new Select(wb2);

sel.selectByVisibleText("BDD Framework");

package Tutorial9;

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.support.ui.Select;

public class Demo2 {

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

// step 1: connect to the actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://prafpawar11.github.io/twoFrame.html");

Thread.sleep(5000);

// step 4:find the full name text box and enter the values

WebElement wb1 driver.findElement(By.id("name"));

wb1.sendKeys("abc xyz");

// step 5: switch focus to child frame1 or checkbox frame

driver.switchTo().frame("chk");

Thread.sleep(5000);

// step 6: find the Jenkins checkbox and click on it

WebElement wb2 driver.findElement(By.id("Jenkins"));

wb2.click();

// step 7: switch focus to main frame or top frame or top window

driver.switchTo().defaultContent();

// step 8: switch to child frame 2 or drop down frame

driver.switchTo().frame(1);

Thread.sleep(5000);

// step 9: find the drop down and select the BDD Framework value

WebElement wb3 driver.findElement(By.id("course"));

Select sel new Select(wb3);

sel.selectByVisibleText("BDD Framework");

}

}

Scenario 3:

open a url https://prafpawar11.github.io/frame.html

find the name text box and enter the name

find address text box and enter the address

find the cucumber check box and select

capture the entered address

and enter new address

capture the enter name and clear the name

unselect the cucumber checkbox

step 1: connect browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://prafpawar11.github.io/frame.html");

step 4: find the name text box and enter the name

WebElement wb driver.findElement(By.id("name"));

wb.sendKeys("abc xyz");

step 5: switch focus to address frame

driver.switchTo().frame("mainframe");

step 6: find address text box and enter the address

WebElement wb1 driver.findElement(By.id("add"));

wb1.sendKeys("Pune");

step7:switch focus to checkbox frame

driver.switchTo().frame("child1");

step 8: find the cucumber check box and select

WebElement wb3 driver.findElement(By.id("Cucumber"));

wb3.click();

step 9:switch focus to address frame

driver.switchTo().defaultContent();

driver.switchTo().frame("mainframe");

or

driver.switchTo().parentFrame();

step 10: capture the entered address and enter new address

String a wb1.getAttribute("value");

syso(a);

wb1.clear();

wb1.sendKeys("Mumbai");

step 11: switch focus to top frame

driver.switchTo().defaultContent();

step 12: capture the enter name and clear the name

String b wb.getAttribute("value");

syso(b);

wb.clear();

step 13: switch focus to address frame

driver.switchTo().frame(0);

step 14: switch focus to checkbox frame

driver.switchTo().frame(0);

step 15: unselect the cucumber checkbox

wb3.click();

package Tutorial9;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo3 {

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

// step 1: connect to the actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://prafpawar11.github.io/frame.html");

Thread.sleep(5000);

// step 4:find the full name text box and enter the values

WebElement wb driver.findElement(By.id("name"));

wb.sendKeys("abc xyz");

// step 5: switch focus address frame

driver.switchTo().frame("mainframe");

// step 6: find the address text box and enter the values

WebElement wb1 driver.findElement(By.id("add"));

wb1.sendKeys("Pune");

Thread.sleep(5000);

// step 7: switch focus to checkbox frame

driver.switchTo().frame("child1");

// step 8: find the Cucumber checkbox and click on it

WebElement wb2 driver.findElement(By.id("Cucumber"));

wb2.click();

// step 9: switch focus to address frame

// driver.switchTo().defaultContent();

// driver.switchTo().frame("mainframe");

// or

driver.switchTo().parentFrame();

Thread.sleep(5000);

// step 10: capture the entered addres and enter new address

String a wb1.getAttribute("value");

System.out.println(a);

wb1.clear();

wb1.sendKeys("Mumbai");

// step 11: switch focus to top frame

driver.switchTo().defaultContent();

Thread.sleep(5000);

// step 12:capture the entered text and clear it

String b wb.getAttribute("value");

System.out.println(b);

wb.clear();

Thread.sleep(5000);

// step 13: switch focus to address frame

driver.switchTo().frame(0);

// step 14: switch focus to checkbox frame

driver.switchTo().frame(0);

// step 15: unselect the checkbox

wb2.click();

}

}

Ass:

open a https://praf002.github.io/

find the full name text box and enter the value

find the GitHub checkbox and click on it

find and select the Database Automation values from drop down

find and Enter the mobile numbers

Find and Enter the address

unselect the GitHub checkbox

click on selenium click

capture the entered full name and enter new name

select the Java Script Executor value

capture entered address and enter new address

Ass:

open a url https://chercher.tech/practice/frames-example-selenium-webdriver

enter the topic name

select the avatar

select the checkbox

select the cat

capture the entered text and enter new text

unselect the checkbox

Ass:

open a url https://demo.automationtesting.in/Frames.html

- click on single iframe

enter text in text box

- click on iframe with iframe button

- enter the text in text box

- click on single iframe

-- capture the entered text and enter new text

- click on iframe with iframe button

- capture the entered text and enter new text

**How to handle table in selenium? VVVVVVIMMMMMMPPPPPPPPPPPPP**

**or**

**How to handle static table in selenium? VVVVVVIMMMMMMPPPPPPPPPPPPP**

**or**

**How to handle dynamic table in selenium? VVVVVVIMMMMMMPPPPPPPPPPPPP**

**or**

**how to handle calendar in selenium? VVVVVVIMMMMMMPPPPPPPPPPPPP**

----------

HTML code structure:

- table it starts with table tagname

- table heading tagname starts with th or thead tag (table columns).

- table row tagname starts with tr tag.

- table data tagname starts with td tag.

----------

**Operation on static table**

- count how many rows in table

- capture all the rows value

- count how many columns/table heading in table

- capture all columns names/values

- capture all table values

- check specific value is present or not?

**- count how many rows in table**

List<WebElement> ls driver.findElements(By.tagName("tr"));

int rows ls.size();

- capture all the rows value

List<WebElement> ls driver.findElements(By.tagName("tr"));

for(WebElement abc: ls)

{

String a abc.getText();

System.out.println(a);

}

Scenario 1:

open a https://prafpawar11.github.io/emptable.html

count rows and print all rows values in console

step 1: connect browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://prafpawar11.github.io/emptable.html");

step 4: find all rows using findElements() method

List<WebElement> ls driver.findElements(By.tagName("tr"));

step 5: count rows using size() method

int a ls.size();

System.out.println(a);

step 6: capture all rows values using for loop and getText() method

for(WebElement abc : ls)

{

String b abc.getText();

System.out.println(b);

}

package Tutorial10;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

public static void main(String[] args) {

// Connect browser

// up casting

WebDriver driver new ChromeDriver();

// open a url

driver.get("https://prafpawar11.github.io/emptable.html");

// find all rows using findElements() method

List<WebElement> ls driver.findElements(By.tagName("tr"));

// count how many rows present

int a ls.size();

System.out.println(a);

// capture all rows values in console

for (WebElement abc : ls) {

String b abc.getText();

System.out.println(b);

}

}

}

Scenario 2:

open https://prafpawar11.github.io/emptable.html

find and count total columns present in table and capture all values

Step 1: connect browser

System.setProperty("webdriver.chrome.driver","path of chrome");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://prafpawar11.github.io/emptable.html");

step 4: find all columns using findElements() method

List<WebElement> ls driver.findElements(By.tagName("th"));

step 5: count how many columns using size() method

int a ls.size();

System.out.println(a);

step 6: capture all columns values

for(WebElement abc : ls)

{

String b abc.getText();

System.out.println(b);

}

package Tutorial10;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo2 {

public static void main(String[] args) {

// Connect browser

// up casting

WebDriver driver new ChromeDriver();

// open a url

driver.get("https://prafpawar11.github.io/emptable.html");

// find all columns in table

List<WebElement> ls driver.findElements(By.tagName("th"));

// count columns using size() method

int a ls.size();

System.out.println(a);

// capture all colums names in console

for (WebElement abc : ls) {

String b abc.getText();

System.out.println(b);

}

}

}

package Tutorial10;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo2 {

public static void main(String[] args) {

// Connect browser

// up casting

WebDriver driver new ChromeDriver();

// open a url

driver.get("https://prafpawar11.github.io/emptable.html");

// find all columns in table

List<WebElement> ls driver.findElements(By.tagName("th"));

// count columns using size() method

int a ls.size();

System.out.println(a);

// capture all colums names in console

for (WebElement abc : ls) {

String b abc.getText();

// // Verify the Salary column is present or not

if (b.equals("Salary")) {

System.out.println("test cases is pass");

}

}

}

}

Scenario 3:

- count how many values present inside the table

- capture all table values

- check specific value is present or not?

Step 1: connect browser

System.setProperty("webdriver.chrome.driver","path of chrome");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://prafpawar11.github.io/emptable.html");

step 4: find all tables values using findElements() method

List<WebElement> ls driver.findElements(By.tagName("td"));

step 5: count how many values present inside the table

int a ls.size();

System.out.println(a);

step 6: capture all table values

for(WebElement abc : ls)

{

String b abc.getText();

System.out.println(b);

}

step 7) check specific value is present or not?

for(WebElement abc : ls)

{

String b abc.getText();

if(b.equals("Radhika"))

{

System.out.println("test cases is pass");

}

}

package Tutorial10;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo3 {

public static void main(String[] args) {

// Connect browser

// up casting

WebDriver driver new ChromeDriver();

// open a url

driver.get("https://prafpawar11.github.io/emptable.html");

// find all table values using findElements() method

List<WebElement> ls driver.findElements(By.tagName("td"));

// count total records present in table

int a ls.size();

System.out.println(a);

// capture all table values

for (WebElement abc : ls) {

String b abc.getText();

System.out.println(b);

}

// verify specific value is present or not in table

for (WebElement abc : ls) {

String b abc.getText();

if (b.contains("Radhika")) {

System.out.println("Test Cases is pass");

}

}

}

}

https://prafpawar11.github.io/statictable.html

**How to handle dynamic table in selenium? VVVVVVIMMMMMMPPPPPPPPPPPPP**

**or**

**how to handle calendar in selenium? VVVVVVIMMMMMMPPPPPPPPPPPPP**

Scenario 4:

open a url https://www.irctc.co.in/nget/train-search

click on date object

run the infinite loop

find and capture the displayed month and year

compare capture values with expected month and year values

if values is matches then select the expected date and break the loop

if it both values not matches then click on next button

Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://www.irctc.co.in/nget/train-search");

step 4: find the date object and click on it.

WebElement wb driver.findElement(By.xpath("//span[@class'ng-tns-c58-10 ui-calendar']/input"));

wb.click();

step 5: run the infinite while loop

while(true)

{

//step 5a) find and capture the displayed month and year values

WebElement monthwb driver.findElement(By.xpath("//span[@class'ui-datepicker-month ng-tns-c58-10 ng-star-inserted']"))

WebElement yearwb driver.findElement(By.xpath("//span[@class'ui-datepicker-year ng-tns-c58-10 ng-star-inserted']");

String a monthwb.getText();

String b yearwb.getText();

String monthYear a.concat(b);

//step 5b) compare with expected month and year

if(c.equals("July2024"))

{

//select the date

WebElement datewb driver.findElement(By.xpath("//a[text()'10']"));

datewb.click();

//brake the while loop;

break;

}

else

{

//step c) click on next button

WebElement nextwb driver.findElement(By.xpath("//span[@class'ui-datepicker-next-icon pi pi-chevron-right ng-tns-c58-10']"));

nextwb.click();

}

}

package Tutorial10;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo4 {

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

// step 1: connect browser

// step 2: up casting

WebDriver driver new ChromeDriver();

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

// step 3: open a url

driver.get("https://www.irctc.co.in/nget/train-search");

Thread.sleep(2000);

// step 4: find the date object and click on it

WebElement wb driver.findElement(By.xpath("//span[@class'ng-tns-c58-10 ui-calendar']/input"));

wb.click();

// step 5: run infinite while loop

while (true) {

Thread.sleep(5000);

// capture the displayed month and year values

WebElement monthwb driver

.findElement(By.xpath("//span[@class'ui-datepicker-month ng-tns-c58-10 ng-star-inserted']"));

WebElement yearwb driver

.findElement(By.xpath("//span[@class'ui-datepicker-year ng-tns-c58-10 ng-star-inserted']"));

String monthText monthwb.getText();

String yearText yearwb.getText();

String monthYearText monthText.concat(yearText);

// compare captured values with expected value

if (monthYearText.equalsIgnoreCase("September2024")) {

// select the date

WebElement datewb driver.findElement(By.xpath("//a[text()'7']"));

datewb.click();

// break the while loop

break;

} else {

// click on next button

WebElement nextwb driver.findElement(

By.xpath("//span[@class 'ui-datepicker-next-icon pi pi-chevron-right ng-tns-c58-10']"));

nextwb.click();

}

}

}

}

Scenario 5:

open a https://www.goibibo.com/

find and click on date

run infinite loop

- capture current displayed month and year value

- compare captured values with expected month and year value

- if capture value and expected value is equal then click on date and break the loop

- capture value and expected value is not equal click on next button

Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://www.goibibo.com/");

step 4: find and click on date

WebElement wb driver.findElement(By.xpath("//span[text()'Departure']/following::span[@class'sc-12foipm-8 eXKWBG fswDownArrow']"));

wb.click();

step 5: run the infinite loop

while(true)

{

//step 5a) capture the displayed month and year values.

WebElement monthYearWb driver.findElement(By.xpath("(//div[@class'DayPicker-Caption'])[1]/div"));

String monthYearText monthYearWb.getText();

//step b)compare captured values with expected month and year value

if(monthYearText.equals("February 2025")

{

//select the date

WebElement dateWb driver.findElement(By.xpath("(//p[text()'5'])[1]"));

dateWb.click();

//break the while loop

break;

}

else

{

//step 5c) click on next button

WebElement nextWb driver.findElement(By.xpath("//span[@class'DayPicker-NavButton DayPicker-NavButton--next']"));

nextWb.click();

}

}

package Tutorial10;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo5 {

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

// step 1: connect browser

// step 2: up casting

WebDriver driver new ChromeDriver();

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

// step 3: open a url

driver.get("https://www.goibibo.com/");

Thread.sleep(5000);

// step 4: find and click on date object

WebElement wb driver.findElement(

By.xpath("//span[text()'Departure']/following::span[@class'sc-12foipm-8 eXKWBG fswDownArrow']"));

wb.click();

Thread.sleep(5000);

// step 5: run infinite loop

while (true) {

// capture the displayed month and year value

WebElement monthYearWb driver.findElement(By.xpath("(//div[@role'heading'])[1]/div"));

String monthYearText monthYearWb.getText();

Thread.sleep(5000);

// compare the captured values with expected value

if (monthYearText.equalsIgnoreCase("April 2025")) {

// if captured values with expected value is equal then select date and break

// while loop.

WebElement dateWb driver.findElement(By.xpath("(//p[text()'22'])[1]"));

dateWb.click();

break;

} else {

// if captured values with expected value is not equal then click on next button

WebElement nextwb driver.findElement(By.xpath("//span[@aria-label'Next Month']"));

nextwb.click();

}

}

}

}

Ass:

open a url https://www.makemytrip.com/

find and enter from city as Pune

and suggestion list click on first Object

find and enter to city as Delhi

and suggestion list click on first Object

find and select Departure 20 March 2025

find and click on Travellers & Class

select 5 ADULTS

and select the CHILDREN

and select the Business class

click on apply

click on search button

Ass:

open a https://www.redbus.in/

find from text box and enter value

find to text box and enter the values

select 10 Dec 2024 date

**Actions class Events**

- Actions class is used to perform mouse over and keyboard events

- there are 2 main category in Actions class

1) Mouse Events

2) Keyboard Events

--

1) Mouse Events

- By using mouse we can perform different activities as like

i) click on button/click click(WebElement wb);

ii) double click doubleClick(WebElement wb)

iii) scroll up and scroll down ------> by using java script Executor

iv) right click contextClick(WebElement wb) VVVVVVIMMPP

v) drag and drop dragAndDrop(WebElement src, WebElement trg)

vi) mouse over on element moveToElement(WebElement wb) VVVVVVIMMPP

vii) click and hold element without drop clickAndDrop(WebElement wb) VVVVVVIMMPP

--

**How many ways we can click on element? VVVVVVIMMMMMPPPPPPP**

there are 5 ways we can click on element

1) by using WebElement interface click() method

2) by using WebElement interface submit() method

3) by using Actions class click() method

4) by using Actions class ENTER keyword

5) by using JavascriptExecutor interface click()method

--

**diff between build() method and perform() method VVVVVVIMMMMMPPPPPPP**

- build() method is used to combine the multiple actions in single statement

- perform() method is used to execute combined each and every actions.

- build() method is present inside the Actions class.

- perform() method is present inside the Action interface.

- return type of build() method is Action interface.

- return type of perform() method is void.

--

**i) click() method from Actions class**

- it is used to click on element or object.

- return type of click() method is Actions class.

- click() method is present inside the Actions class.

- if u want to use Actions class click() method then

- first we find the Element or object in webpage then we create Object of Actions class by passing WebDriver instance or object name.

- then we click() method by passing WebDriver instance and then build() method and perform() method.

- click() method is used to click on element

- build() method is used to combine the multiple actions in single statements and this build method is present inside the Actions class and return type of build() method is Action interface.

- perform() method is used to execute each and every combined actions and perform() method present inside the Action interface and

return type is void.

syntax:

WebElement wb driver.findElement(By.id("id value"));

Actions act new Actions(driver);

act.click(wb).build().perform();

or

act.click(wb).perform();

Scenario 1:

open a https://www.facebook.com/reg

click on forgot link using Actions class.

Step 1: connect to browser code

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://www.facebook.com/reg");

step 4: find the element

WebElement wb driver.findElement(By.xpath("//a[text()'Already have an account?']"));

step 5: create object of Actions class by passing """WebDriver instance."""

Actions act new Actions(driver);

step 6: use click() method by passing WebDriver instance from Actions class then build() and perform() method

act.click(wb).build().perform();

package Tutorial11;

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 Demo1 {

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

// Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://www.facebook.com/reg");

Thread.sleep(5000);

// step 4:find the forgot link

WebElement wb driver.findElement(By.xpath("//a[text()'Already have an account?']"));

// step 5: create object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

// step 6: use click() by passing WebElement instance from Actions class then

// use build() method and perform() method

act.click(wb).build().perform();

}

}

**2) doubleClick(WebElement wb) method**

- doubleClick() method is used to double on click on element.

- doubleClick() method is present inside the Actions class

- return type of doubleClick() method is Actions class.

- if u want to use doubleClick()method then first we find the web element then we create object of Actions class by passing WebDriver instance.

- then use doubleClick() method by passing WebElement instance and then build() and perform() method

- doubleClick() method use to doubleClick() on element

- build() method is used to combine the multiple actions in single statement, and this build method is present inside the Actions class and return

type is Action interface.

- perform() method is used to execute the combined each and every actions, this perform() method is present inside the Action interface and

return type is void.

syntax:

WebElement wb driver.findElement(By.id("id value"));

Actions act new Actions(driver);

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

Scenario:

https://demoqa.com/buttons

find the double click button and double click on it.

Step 1: connect to browser code

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://demoqa.com/buttons");

step 4: find the double click button

WebElement wb driver.findElement(By.id("doubleClickBtn"));

step 5: create object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

step 6: use doubleClick() method by passing WebElement instance from Actions class and then build() and perform() method

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

package Tutorial11;

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 Demo2 {

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

// Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://demoqa.com/buttons");

Thread.sleep(5000);

// step 4: find the double click button/ element or object

WebElement wb driver.findElement(By.id("doubleClickBtn"));

// step 5: create object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

// step 6: use doubleClick() method by passing WebElement instance from Actions

// class and then use build() and perform() method

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

// find the message text

WebElement wb1 driver.findElement(By.id("doubleClickMessage"));

String a wb1.getText();

System.out.println(a);

}

}

**3) contextClick(WebElement wb) method VVVVVVVVIMMMMPPPP**

- contextClick() method is used to right click on element

- contextClick() method is present inside the Actions class and return type is Actions class.

- if u want to use contextClick() method then first we have to find Object or element in webpage, then

we create Object of Actions class by passing WebDriver instance.

then we use contextClick() method by passing WebElement instance from Actions class then we use build() and perform() method.

**-contextClick() method is used to right click on element**

- build() method is used combine the multiple actions in single statement and this build() method is present inside Actions class and

return type is Action interface.

- perform() method is used to execute the combined each and every actions and perform()method is present inside the Action interface and

return type is void.

syntax:

WebElement wb driver.findElement(By.id("id value"));

Actions act new Actions(driver);

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

Scenario 3:

open https://demoqa.com/buttons

find the right click button and right click on it.

Step 1: connect to browser code

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://demoqa.com/buttons");

step 4: find the right click button

WebElement wb driver.findElement(By.id("rightClickBtn"));

step 5: create object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

step 6: use contextClick() method by passing WebElement instance from Actions class and then use build().perform() method

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

package Tutorial11;

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 Demo3 {

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

// Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://demoqa.com/buttons");

Thread.sleep(5000);

// step 4: find the right click button/ element or object

WebElement wb driver.findElement(By.id("rightClickBtn"));

// step 5: create object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

// step 6: use contextClick() method by passing WebElement instance from Actions

// class and then build() and perform() method

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

// step 7: find the right click message and capture it

String a driver.findElement(By.id("rightClickMessage")).getText();

System.out.println(a);

}

}

Ass:

open a https://swisnl.github.io/jQuery-contextMenu/demo.html

find and right click on right click me button

find and click on copy button

switch focus to alert pop up, capture the text and click on ok button

**4) moveToElement(WebElement wb)**  VVVVVVVVVVIMMMMMPPPPPPP

- moveToElement() method is used to mouse over on element without click.

- moveToElement() method present inside the Actions class and return type is Actions class.

- if u want to use moveToElement() method then first we have find the element and then we have to create object of Actions class by passing

WebDriver instance, then we use moveToElement() method by passing WebElement instance from Actions class and then we use build() and perform() method.

- moveToElement() method is used to mouse over on element and

build() method is used to combine the multiple actions in single statement and build() method is present inside the Actions class and

return type is Action interface.

- perform() method is used to execute the combine each and every actions and perform() method is present inside the Action interface and return type is void,

syntax;

WebElement wb driver.findElement(By.id("id value"));

Actions act new Actions(driver);

act.moveToElement(wb).build().perform();

Scenario 4:

open a https://www.amazon.com/

find and mouse on Account and Lists

click on Orders link

Step 1: connect to browser code

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://www.amazon.com/");

step 4: find the Account and List link

WebElement wb driver.findElement(By.id("nav-link-accountList"));

step 5: create object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

step 6: use moveToElement() method by passing WebElement instance from Actions class and then build() and perform() method

act.moveToElement(wb).build().perform();

step 7: find and click on Orders link

WebElement wb1 driver.findElement(By.xpath("//span[text()'Orders']"));

wb1.click();

package Tutorial11;

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 Demo4 {

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

// Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

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

// step 3: open a url

driver.get("https://www.amazon.in");

Thread.sleep(5000);

// step 4: find the Account and Lists link

WebElement wb driver.findElement(By.id("nav-link-accountList"));

// step 5: create Object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

// step 6: use moveToElement() method by passing WebElement instance from

// Actions class and build and perform method

act.moveToElement(wb).build().perform();

Thread.sleep(5000);

// step 7: find and click on Order links

WebElement wb1 driver.findElement(By.xpath("//span[text()'Your Orders']"));

wb1.click();

}

}

package Tutorial11;

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 Demo5 {

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

// Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

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

// step 3: open a url

driver.get("https://www.amazon.in");

Thread.sleep(5000);

// step 4: find the Account and Lists link

WebElement wb1 driver.findElement(By.id("nav-link-accountList"));

// step 5: find Orders link

WebElement wb2 driver.findElement(By.xpath("//span[text()'Your Orders']"));

// step 6: create Object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

// step 7: use moveToElement() by passing WebElement from Actions class and use

// click() method by passing WebElement instance from Actions class and then

// build() and perform() method

act.moveToElement(wb1).click(wb2).build().perform();

}

}

Ass:

Open a https://www.flipkart.com/

mouse over on Electronics

mouse over on Laptop and Desktop

and click on Laptop

Mouse on Login button

click on SignUp link

Ass:

open a https://www.tanishq.co.in/#home

Mouse over on GOLD

Click on Rings

click on sort drop down and select high to low price

click on first item

click on I am interested

enter name, email , mobile number, and

select city

- click on city drop down

- find all list

- capture all city name

then use if condition, and if value is present then click on it break the loop

Ass:

open a https://www.firstcry.com/

- mouse over on ALL CATEGORIES

- Mouse over on toys

- click on Coloring, Sequencing & Engraving Art

- select sort by top rated

- select 1t product and click on add to cart

- click on cart button

handle window/tab/window pop up

wait in selenium

take screenshot

upload files - using AutoIT dependency

xpath

-

Actions class Events

--

Mouse Events in Actions class

--

1) click(WebElement wb)

2) doubleClick(WebElement wb)

3) contextClick(WebElement wb)

4) moveToElement(WebElement wb)

5) clickAndHold(WebElement wb)

6) release(WebElement wb)

7) dragAndDrop(WebElement src, WebElement trg)

8) scroll up and scroll down ----> we perform using JavascriptExecutor interface.

-------

**5) clickAndHold(WebElement wb)** VVVVVVIMMMPPPPPPPP

- if you want to hold the specific element without dropping then we use clickAndHold() method.

- clickAndHold() method is present inside the Actions class and return type is Actions class.

- if u want to use clickAndHold()method, first we find the web element using findElement() method and then

we create Object of Actions class by passing WebDriver instance.

- then we use clickAndHold() method by passing WebElement instance from Actions class and then build() and perform() method.

- clickAndHold() method is used to hold the element and build()method is used to combine the multiple actions in single statement and build() method present inside the Actions class and return type is Action interface. and perform() method used to execute each and every combined actions.

syntax:

WebElement wb driver.findElement(By.id(""));

Actions act new Actions(driver);

act.clickAndHold(wb).build().perform();

Scenario 1:

open a http://www.dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html

find the Washington element and hold the Washington element

Step 1: Connect to actual browser

System.setProperty("webdriver.chrome.driver", "chrome path");

Step 2: Up casting

WebDriver driver new ChromeDriver();

Step 3: open a URL

driver.get("http://www.dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html");

Step 4: find the WebElement using findElement() method and using given locator

WebElement wb driver.findElement(By.id("box3"));

Step 5: create Object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

step 6: use clickAndHold() to hold the element from Actions class and then build() and perform() method

act.clickAndHold(wb).build().perform();

package Tutorial12;

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 Demo1 {

public static void main(String[] args) {

// Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Up casting

WebDriver driver new ChromeDriver();

// open a URL

driver.get("http://www.dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html");

// find the WebElement using findElement() method and using given locator

WebElement wb driver.findElement(By.id("box3"));

// create Object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

// use clickAndHold() method by passing WebElement instance and build() and

// perform() methodF

act.clickAndHold(wb).build().perform();

}

}

------

**6) release(WebElement wb)**  VVVVVIMMMPPPP

- release() method is used to release the hold elements.

- release() method present inside the Actions class and return type is Actions class.

- if u want to release() method then first we find the element or object using findElement() method and then we create object of

Actions class by passing WebDriver instance and

then we use release() method by passing WebElement instance from Actions class and then build() method and perform() method.

release() method is used to release the hold element and build() method used to combine the multiple Actions in single statement

and this build() method is present inside the Actions class and return type is Action interface and perform() method is used to

execute each and every combined actions and perform() method is present inside the Action interface and return type is void.

syntax:

WebElement wb driver.findElement(By.id(""));

Actions act new Actions(driver);

act.clickAndHold(wb).build().perform();

WebElement wb1 driver.findElement(By.id(""));

act.release(wb1).build().perform();

Scenario 1:

open a http://www.dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html

find the Washington element and hold the Washington element

and release Washington elements in Spain box.

Step 1: Connect to browser code

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("http://www.dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html");

step 4: find the Washington elements or box

WebElement wb1 driver.findElement(By.id("box3"));

step 5: create Object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

step 6: hold the Washington elements by using clickAndHold() method and then build() and perform() method

act.clickAndHold(wb1).build().perform();

step 7: find the Spain elements or box

WebElement wb2 driver.findElement(By.id("box107"));

step 8: release the Washington elements on Spain box.

act.release(wb2).build().perform();

---

act.clickAndHold(wb1).release(wb2).build().perform();

---

package Tutorial12;

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 Demo2 {

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

// Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Up casting

WebDriver driver new ChromeDriver();

// open a URL

driver.get("http://www.dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html");

// find the WebElement using findElement() method and using given locator

WebElement wb1 driver.findElement(By.id("box3"));

// create object of Actions class by pasing WebDriver instance

Actions act new Actions(driver);

// step 6: hold the Washington elements by using clickAndHold() method and then

// build() and perform() method

act.clickAndHold(wb1).build().perform();

Thread.sleep(10000);

// find the Spain elements or box

WebElement wb2 driver.findElement(By.id("box103"));

// release the Washington elements on Spain box.

act.release(wb2).build().perform();

}

}

------

package Tutorial12;

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 Demo2 {

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

// Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Up casting

WebDriver driver new ChromeDriver();

// open a URL

driver.get("http://www.dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html");

// find the WebElement using findElement() method and using given locator

WebElement wb1 driver.findElement(By.id("box3"));

// create object of Actions class by pasing WebDriver instance

Actions act new Actions(driver);

// find the Spain elements or box

WebElement wb2 driver.findElement(By.id("box103"));

// hold the Washington elements by using clickAndHold() method and then

// release the Washington elements on US box.

act.clickAndHold(wb1).release(wb2).build().perform();

}

}

**2) dragAndDrop(WebElement src, WebElement trg)**

- dragAndDrop() method used to drag the element from source location and drop the target location,

- dragAndDrop() method present inside the Actions and return type is Actions class.

- if u want to use dragAndDrop()method then we find the source element and target element using findElement() method and

then we create Object of Actions class by passing WebDriver instance and then we dropAndDrop() method by passing

source web element location and target element location from Actions class and then build() and perform() method

dropAndDrop() is used to drag the element from source location and drop the target location,

build() method is used to combine the multiple actions and this method present inside the Actions class and return type is Action interface. and perform() method is used to execute each and every combined actions.

syntax:

WebElement src driver.findElement(By.id("id value"));

WebElement trg driver.findElement(By.id("id value"));

Actions act new Actions(driver);

act.dragAndDrop(src,trg). build().perform();

Scenario :

open a http://www.dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html

find the Washington element

find the Spain Element

and drag the Washington element and drop Washington elements in Spain box.

Step 1: connect browser

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("http://www.dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html");

step 4; find the Washington element

WebElement src driver.findElement(By.id("box3"));

step 5: find Spain Element

WebElement trg driver.findElement(By.id("box107"));

step 6: create Object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

step 7: use dragAndDrop() method by passing source and target location then build() and perform() method

act.dragAndDrop(src,trg).build().perform();

package Tutorial12;

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 Demo3 {

public static void main(String[] args) {

// Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Up casting

WebDriver driver new ChromeDriver();

// open a URL

driver.get("http://www.dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html");

// step 4; find the Washington element

WebElement src driver.findElement(By.id("box3"));

// step 5: find US Element

WebElement trg driver.findElement(By.id("box103"));

// step 6: create Object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

// step 7: use dragAndDrop() method by passing source and target location then

// build() and perform() method

act.dragAndDrop(src, trg).build().perform();

}

}

**2) Keyboard events in Actions class**.

there are 3 main method present inside the Actions class keyboard events

**1) keyDown(Keys keys)**  it used to press any key

this present inside the Actions and return type Actions

**2) keyUp(Keys keys**) it is used to release the pressed key

this present inside the Actions and return type Actions.

**3) sendKeys(WebElement wb, String string)**

it is used to send text to the elements or object.

how many ways we can send text to text box? **VVVVVIMMMMPPPPPP**

there are 3 ways we can send text to text box

1) by using WebElement sendKeys() method

2) by using Actions class sendKeys() method

3) by using JavascriptExecutor interface value attribute

**Enter the value in Text box using Actions class VVVVVVIMMMPPPPP**

Scenario:

open a fb register page application

find the first name text box

enter values

Step 1: connect browser

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://www.facebook.com/reg");

step 4: find the first name text box

WebElement wb driver.findElement(By.name("firstname"));

step 5: create object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

step 6: use sendKeys() method by passing WebElement instance and text from Actions class and then build() and perform() method

act.sendKeys(wb,"selenium").build().perform();

package Tutorial12;

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 Demo4 {

public static void main(String[] args) {

// Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Up casting

WebDriver driver new ChromeDriver();

// open a URL

driver.get("https://www.facebook.com/reg");

// find the first name text box

WebElement wb driver.findElement(By.name("firstname"));

// create object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

// use sendKeys() method by passing WebElement instance and text from Actions

// class and then build and perform() method

act.sendKeys(wb, "poonam").build().perform();

}

}

**Enter Upper case Letter in text box VVVVVVVIMMMMPPPP**

Scenario

open fb register page and enter upper case value in first name text box

Actions act new Actions(driver);

act.keyDown(Keys.SHIFT).sendKeys(wb,"anjali").build().perform();

Step 1: connect browser

step 2: up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://www.facebook.com/reg");

step 4: find the first name text box

WebElement wb driver.findElement(By.name("firstname"));

step 5: create object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

step 6: use keyDown()method to press the key by passing Keys.SHIFT key and then use sendKeys() method by passing WebElement instance and text from Actions class then build() and perform() method

act.keyDown(Keys.SHIFT).sendKeys(wb,"anjali").build().perform();

package Tutorial12;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.interactions.Actions;

public class Demo5 {

public static void main(String[] args) {

// Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Up casting

WebDriver driver new ChromeDriver();

// open a URL

driver.get("https://www.facebook.com/reg");

// find the first name text box

WebElement wb driver.findElement(By.name("firstname"));

// create object of Actions class by passing WebDriver instance

Actions act new Actions(driver);

// use keyDown() to press key by passing Keys.SHIFT keyword and then use

// sendKeys()method by passing

// WebElement instance and text from Actions class and then build() and

// perform() method

act.keyDown(Keys.SHIFT).sendKeys(wb, "anjali").build().perform();

}

}

Scenario:

open fb register page application

find the first name text box and

enter value in first name ,last name text box, username text box and password text box

package Tutorial12;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.interactions.Actions;

public class Demo6 {

public static void main(String[] args) {

// Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Up casting

WebDriver driver new ChromeDriver();

// open a URL

driver.get("https://www.facebook.com/reg");

// find the first name text box

WebElement wb driver.findElement(By.name("firstname"));

Actions act new Actions(driver);

act.sendKeys(wb,"Anjali").build().perform();

//it will focus on last name

act.keyDown(Keys.TAB).build().perform();

//enter value in last name text box

act.sendKeys("Patil").build().perform();

//focus on username text box by pressing tab

act.keyDown(Keys.TAB).build().perform();

//enter username

act.sendKeys("anjali@gmail.com").build().perform();

//press tab

act.keyDown(Keys.TAB).build().perform();

//enter the confirm username

act.sendKeys("anjali@gmail.com").build().perform();

//press tab

act.keyDown(Keys.TAB).build().perform();

//enter password

act.sendKeys("Anjali@123").build().perform();

}

}

package Tutorial12;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.interactions.Actions;

public class Demo7 {

public static void main(String[] args) {

// Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Up casting

WebDriver driver new ChromeDriver();

// open a URL

driver.get("https://www.facebook.com/reg");

// find the first name text box

WebElement wb driver.findElement(By.name("firstname"));

Actions act new Actions(driver);

act.sendKeys(wb,"Anjali")

.keyDown(Keys.TAB).sendKeys("Patil")

.keyDown(Keys.TAB).sendKeys("anjali@gmail.com")

.keyDown(Keys.TAB).sendKeys("anjali@gmail.com")

.keyDown(Keys.TAB).sendKeys("Anjali@123")

.build().perform();

}

}

Scenario:

open https://www.saucedemo.com/

find the username text box

and enter value in username text box, password text box and click on login button

package Tutorial12;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.interactions.Actions;

public class Demo8 {

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

// Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Up casting

WebDriver driver new ChromeDriver();

// open a URL

driver.get("https://www.saucedemo.com/");

WebElement wb driver.findElement(By.id("user-name"));

Actions act new Actions(driver);

act.sendKeys(wb,"standard\_user").build().perform();

act.keyDown(Keys.TAB).build().perform();

act.sendKeys("secret\_sauce").build().perform();

act.keyDown(Keys.TAB).build().perform();

//click on login button

act.sendKeys(Keys.ENTER).build().perform();

}

}

package Tutorial12;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.interactions.Actions;

public class Demo8 {

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

// Connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// Up casting

WebDriver driver new ChromeDriver();

// open a URL

driver.get("https://www.saucedemo.com/");

WebElement wb driver.findElement(By.id("user-name"));

Actions act new Actions(driver);

act.sendKeys(wb,"standard\_user")

.keyDown(Keys.TAB).sendKeys("secret\_sauce")

.keyDown(Keys.TAB).sendKeys(Keys.ENTER)

.build().perform();

}

}

----

Ass:

open a fb register page https://www.facebook.com/reg

enter the value in first name text, copy the value from first name text box and paste in last name box

----------

ass:

open a fb register page https://www.facebook.com/reg

enter value in first name box and delete the last 2 character.

--

Ass:

open https://www.amazon.com/

navigate end of page

wait for 5 seconds

navigate home page

Ass:

open a fb register page https://www.facebook.com/reg

enter the value in

first name text,

last name

username

confirm username

password

select date

select month

select year

select gender

----------

open https://demo.guru99.com/test/newtours/register.php

find the first name text box only

enter value

First Name:

Last Name:

Phone:

Email:

Address:

City:

State/Province:

Postal Code:

select Country INDIA

User Name:

Password:

Confirm Password:

click on submit button

click on sign up link

enter username

enter password

click on submit button

**What is synchronization?**

**or**

**diff between implicitlyWait and explicit wait?**

**or**

**diff between explicit wait and fluent wait**

**or**

**which type of wait we use to handle the exception in selenium webdriver?**

-------

**What is synchronization?**

- it is process of matching the speed of application under test and test tool in order to get proper execution flow then we use synchronization.

why we need synchronization?

- During the test execution selenium tool give one by one instruction with same speed, but application under test takes less time for some elements and takes more time for some elements, in order to keep then in synchronize then we use synchronization concept.

**there are 2 main types of synchronization**

**1) unconditional synchronization**

**2) conditional synchronization**

-------

**1) unconditional synchronization**

- in unconditional synchronization we specify the time out values in milliseconds, we will make selenium tool to wait certain amount of time then processed to next step.

syntax:

Thread.sleep(10000);

- once we use Thread dot sleep() method by passing time in milliseconds as 10 seconds then it will wait for 10 seconds and once the

10 seconds is completed then it will start executing the remaining lines of code.

-Thread dot sleep() method we define 10 seconds as wait then compulsory it will wait for 10 seconds , once it over then it start executing the remaining lines of code that is reason we call unconditional synchronization is an static wait.

-------

**2) conditional synchronization**

- conditional synchronization it apply only for """"findElement() and findElements() method.""""

- conditional synchronization does not apply for the all commands in application or method in selenium webdriver.

- there are 3 types of conditional synchronization

i) implicitly wait

ii) explicit wait

iii) fluent wait.

-------

**i) implicitly wait**  **VVVVVIMMMPPPPPPPPPPPPPPPPPP**

- implicitly wait is called as global wait.

- implicitly wait is also called as dynamic wait.

- once we define the implicitly wait then it is applicable for all elements in web page.

- implicitly wait is applicable only for findElement() and findElements() method.

Scenario 1:

- if we define implicitly wait in program as 30 seconds, and if element found within the 2 seconds then it will ignore the remaining all seconds.

Scenario 2:

- if we define the implicitly wait in program as 30 seconds and if element is not found within the mentioned time then it throws NoSuchElementException.

- default time for implicit wait is zero.

- default searching time for elements in 500 milliseconds.

- define the implicitly wait syntax

syntax:

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30));

- manage() method present inside the WebDriver interface and return type is Options interface.

- timeouts() method present inside the Options interface and return type is Timeouts interface.

- implicitlyWait() method is present inside the Timeouts interface and return type is Timeouts interface.

package Tutorial14;

import java.time.Duration;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

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

WebDriver driver new ChromeDriver();

driver.get("https://demo.guru99.com/test/newtours/register.php");

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30)));

WebElement wb1 driver.findElement(By.name("firstName"));//30

wb1.sendKeys("Amruta");

WebElement wb2 driver.findElement(By.name("lastName13434444"));//30

wb2.sendKeys("Patil");

WebElement wb3 driver.findElement(By.name("phone"));//30

wb3.sendKeys("90909090");

WebElement wb4 driver.findElement(By.name("userName"));//30

wb4.sendKeys("amruta@gmail.com");

Thread.sleep(5000);

driver.quit();

}

}

**2) Explicit wait VVVVVIMMMPPPPPPPPPPPPPPPPPP**

- Explicit wait is applicable for """"single element in web page."""

- Explicit wait is also called as local wait.

- Explicit wait is also called as dynamic wait.

- Explicit wait is applicable only for findElement() and findElements() method.

Scenario 1:

- if we define the Explicit wait as 30 seconds and if element is found within the 3 seconds then it will ignore remaining all seconds.

Scenario 2:

- if we define the explicit wait as 30 seconds and if element is not found within the mention time then it will throws the TimeoutsException.

- how to define the explicit wait.

Step 1: we have to create Object of WebDriverWait class by passing WebDriver instance and time in seconds.

WebDriverWait wait new WebDriverWait(Duration.ofSeconds(30));

step 2: we mention ExpectedConditions by using until() method

WebElement wb wait.until(ExpectedConditions.visibilityOfElementLocated(By.name("name value")));

return type of until() method is WebElement interface.

step 3: use WebElement instance to perform operation.

wb.sendKeys("Patil");

----------

WebDriverWait wait new WebDriverWait(driver, Duration.ofSeconds(30));

WebElement wb wait.until(ExpectedConditions.visibilityOfElementLocated(By.name("name value")));

wb.sendKeys("selenium");

----------

package Tutorial14;

import java.time.Duration;

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.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

public class Demo2 {

public static void main(String[] args) {

WebDriver driver new ChromeDriver();

driver.get("https://demo.guru99.com/test/newtours/register.php");

WebElement wb1 driver.findElement(By.name("firstName"));

wb1.sendKeys("Amruta");

// Explicit wait

// create Object of WebDriverWait by passing WebDriver instance and time in

// seconds.

WebDriverWait wait new WebDriverWait(driver, Duration.ofMinutes(2));

// use until() method from WebDriverWait class by passing ExpectedConditions as

// visibilityOfElementLocated

WebElement wb2 wait.until(ExpectedConditions.visibilityOfElementLocated(By.name("lastName")));

// perform the operation

wb2.sendKeys("Patil");

WebElement wb3 driver.findElement(By.name("phone"));

wb3.sendKeys("90909090");

WebElement wb4 driver.findElement(By.name("userName"));

wb4.sendKeys("amruta@gmail.com");

}

}

can we use implicitlyWait and explicit wait in same in class.

yes

we can apply same implicitlyWait and explicit wait for same web elements then which one wait it will apply for elements.

- yes, but it will apply maximize time duration for elements as per the wait.

package Tutorial14;

import java.time.Duration;

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.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

public class Demo2 {

public static void main(String[] args) {

WebDriver driver new ChromeDriver();

driver.get("https://demo.guru99.com/test/newtours/register.php");

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30));

WebElement wb1 driver.findElement(By.name("firstName"));

wb1.sendKeys("Amruta");

// Explicit wait

// create Object of WebDriverWait by passing WebDriver instance and time in

// seconds.

WebDriverWait wait new WebDriverWait(driver, Duration.ofSeconds(45));

// use until() method from WebDriverWait class by passing ExpectedConditions as

// visibilityOfElementLocated

WebElement wb2 wait.until(ExpectedConditions.visibilityOfElementLocated(By.name("lastName")));

// perform the operation

wb2.sendKeys("Patil");

WebElement wb3 driver.findElement(By.name("phone"));

wb3.sendKeys("90909090");

WebElement wb4 driver.findElement(By.name("userName"));

wb4.sendKeys("amruta@gmail.com");

}

}

diff between implicitlyWait and explicit wait **VVVVIMPPPPPPPPPP**

- implicitlyWait is called global wait

- explicit wait is called as local wait.

- implicitlyWait it applicable for all element in web page

- Explicit wait is applicable for single element in web page.

- in implicitlyWait if element is not found then it throws NoSuchElementException.

- in Explicit wait if element it not found then it throws TimeoutsException.

- both wait are called as dynamic wait because if element is found within few seconds then it ignore remaining times,

- both wait it applicable only of findElement() and findElements() method.

- in both wait we can mentions time in sec, ms, min, hrs, and days.

- in implicitlyWait we never mention Expected Conditions.

- in Explicit wait compulsory we have to mention ExpectedConditions.

- to define the implicitlyWait we use implicitlyWait() method from Timeouts interface.

- to define the Explicit wait we have to create object of WebDriverWait by passing WebDriver instance and time in seconds then we mention ExpectedConditions using until() method.

--------

implicit wait

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30));

Explicit wait

WebDriverWait wait new WebDriverWait(driver,Duration.ofSeconds(45));

WebElement wb wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("id value"));

--------

**3) FluentWait**

- FluentWait is applicable for single element in web page.

- FluentWait is also called as local wait.,

- FluentWait is also called as dynamic wait.

- FluentWait is used to change the element searching time.

- FluentWait is used to handle the Exception

scenario 1:

-if define FluentWait as 30 seconds and if element found within the 2 seconds then it ignore remaining all seconds.

Scenario 2:

-if define FluentWait as 30 seconds and if element is not found within the mentioned seconds then it will throws TimeoutsException.

- how to define the FluentWait.

step 1:

we have to create Object of FluentWait class by passing WebDriver instance and by using WebDriver generic and by calling withTimeout() and passing duration in withTimeout() method and then pollingEvery() method and passing duration in seconds, and then by calling ignoring() method and by passing Exception name in ignoring method.

FluentWait<WebDriver> wait new FluentWait<WebDriver(driver)

.withTimeout(Duration.ofSeconds(30))

.pollingEvery(Duration.ofSeconds(1))

.ignoring(Exception.class);

- withTimeout() method is used to mention timeout in FluentWait

- pollingEvery()method used to change the element searching time.

ignoring() method is used to ignore the exceptions.

step 2: use until() method by passing new Function of generic as Webdriver and WebElement and by using apply() method by passing WebDriver parameters in apply method and by finding the elements in apply() method.

WebElement wb wait.until(new Function<WebDriver, WebElement>() {

public WebElement apply(WebDriver driver)

{

return driver.findElement(By.id("id value"));

}

}

);

step 3: perform operation on web elements

wb.sendKeys("Patil");

package Tutorial14;

import java.time.Duration;

import java.util.function.Function;

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.support.ui.FluentWait;

public class Demo3 {

public static void main(String[] args) {

WebDriver driver new ChromeDriver();

driver.get("https://demo.guru99.com/test/newtours/register.php");

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30));

WebElement wb1 driver.findElement(By.name("firstName"));

wb1.sendKeys("Amruta");

// we have to create Object of FluentWait class by passing WebDriver instance

// and using WebDriver generic, and by calling withTimeout() method and by

// passing time duration in withTimeout() method

// and by calling pollingEvery() method ad by passing duration in seconds in

// pollingEvery() method

// and calling ignoring() method and passing Exception name.

FluentWait<WebDriver> wait new FluentWait<WebDriver>(driver)

.withTimeout(Duration.ofSeconds(30))

.pollingEvery(Duration.ofSeconds(5))

.ignoring(Exception.class);

// use until() method by passing the new Function() and generic as WebDriver and

// WebElement and then use apply() method by passing WebDriver arguments and

// inside the apply method find the web elements

WebElement wb2 wait.until(new Function<WebDriver, WebElement>() {

public WebElement apply(WebDriver driver) {

return driver.findElement(By.name("lastName9887777"));

}

}

);

wb2.sendKeys("Patil");

}

}

implicit Wait

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30);

Explicit Wait

WebDriverWait wait new WebDriverWait(driver,Duration.ofSeconds(45));

WebElement wb wait.until(ExpectedConditions.visibilityOfElementLocated(By.name("name value")));

Fluent Wait

FluentWait<WebDriver> wait new FluentWait<WebDriver>(driver)

.withTimeout(Duration.ofSeconds(45))

.pollingEvery(Duration.ofSeconds(1))

.ignoring(Exception.class);

wait.until(new Function<WebDriver, WebElement>()

{

public WebElement apply(WebDriver driver)

{

return driver.findElement(By.id("id value"));

}

}

);

which wait we use to handle the Exception VVVVVVVVVVVVVVIMMPP

we use FluentWait to handle the Exception.

**How to take screenshot in selenium VVVVIMMMPPPPPPPPPPP**

-

- TakesScreenshot is an interface present inside the Selenium webdriver.

- if u want to capture the screenshot then we have use getScreenshotAs() method from TakesScreenshot interface.

- then we use copyFile() method from FileUtils class to store the screenshot from source location to destination location.

Step 1: convert the WebDriver Object into the TakesScreenshot here we perform down casting concept.

TakesScreenshot ts (TakesScreenshot)driver;

step 2: capture the screenshot using getScreenshotAs() by passing OutputType.FILE

File src ts.getScreenshotAs(OutputType.FILE);

- getScreenshotAs() method is used to capture the Screenshot

- getScreenshotAs() method inside the TakesScreenshot interface.

- return type of getScreenshotAs() method is File class.

step 3: if u want to store the Screenshot in destination location then we have to create Object of File class by passing File destination location.

File dest new File("destination location");

step 4: if u want to copy the screenshot from source location to destination location then we use copyFile() method by passing source and destination location.

FileUtils.copyFile(src, dest);

-

Scenario 1:

open a guru 99 application

enter the first name and last name

and takes the screenshot

Step 1: Connect to the actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2; up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://demo.guru99.com/test/newtours/register.php");

step 4: find the first name text box and enter the values

driver.findElement(By.name("firstName")).sendKeys("dipali");

step 5: find the last name text box and enter the values

driver.findElement(By.name("lastName")).sendKeys("Joshi");

step 6":convert the WebDriver object into TakesScreenshot

TakesScreenshot ts (TakesScreenshot)driver;

step 7: use getScreenshotAs() method by passing OutputType.FILE to capture the actual screenshot

File src ts.getScreenshotAs(OutputType.FILE);

step 8:if want to store the screenshot in destination location then we have to create object of File class by passing file location.

File dest new File("destination location");

step 9: copy the screenshot from source location to destination location using copyFile() method by passing source and destination location

FileUtils.copyFile(src, dest);

----

Note:

if u want to use copyFile() method from FileUtils class then we have to add """"common io dependency in pom.xml file""".

<!-- https://mvnrepository.com/artifact/commons-io/commons-io -->

<dependency>

<groupId>commons-io</groupId>

<artifactId>commons-io</artifactId>

<version>2.11.0</version>

</dependency>

----

package Tutorial15;

import java.io.File;

import java.io.IOException;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.By;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

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

// step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a URL

driver.get("https://demo.guru99.com/test/newtours/register.php");

// step 4: find the first name text box and enter the values

driver.findElement(By.name("firstName")).sendKeys("Aboli");

// step 5: find the last name text box and enter the values

driver.findElement(By.name("lastName")).sendKeys("Shinde");

// step 6: convert the WebDriver object into the TakesScreenshot

TakesScreenshot ts (TakesScreenshot) driver;

// step 7: use getScreenshotAs() method by passing OutputType.FILE to capture

// the actual screenshot in file format

File src ts.getScreenshotAs(OutputType.FILE);

// step 8: store the screenshot in destination location then we have to create

// Object of File class by passing file destination location

File dest new File("C:\\Users\\praf0\\OneDrive\\Desktop\\"

+ "Automation Testing 27th April 2024 Batch\\Screenshot\\xyz.png");

// step 9: copy the screenshot from source location to destination location by

// using copyFile() and by passing

// source and destination location

FileUtils.copyFile(src, dest);

}

}

package Tutorial15;

import java.io.File;

import java.io.IOException;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.By;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo2 {

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

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

WebDriver driver new ChromeDriver();

driver.get("https://demo.guru99.com/test/newtours/register.php");

driver.findElement(By.name("firstName")).sendKeys("Anuja");

// convert the WebDriver Object into the TakesScreenshot

TakesScreenshot ts (TakesScreenshot) driver;

// capture the actual screenshot using getScreenshotAs() method and by passing

// OutputType.FILE to capture the screenshot in file format.

File src ts.getScreenshotAs(OutputType.FILE);

// store the destination location then we have to create object of File class by

// passing destination location

File dest new File(System.getProperty("user.dir")+"\\PassScreenshot\\b1.png");

// copy the screenshot from source location and store in destination location.

FileUtils.copyFile(src, dest);

}

}

**Print current date and time in console.**

step 1: create Object of SimpleDateFormat class by passing date and time format.

SimpleDateFormat simple new SimpleDateFormat("ddMMyyyy\_HHmmss");

step 2: use format() method by passing new Date() object from SimpleDateFormat class.

String abc simple.format(new Date());

Step 3: print in abc variable in console

System.out.println(abc);

package Tutorial15;

import java.text.SimpleDateFormat;

import java.util.Date;

public class Demo3 {

public static void main(String[] args) {

// Step 1: create object of SimpleDateFormat class by passing date and time

// format

SimpleDateFormat simple new SimpleDateFormat("ddMMyyyy\_HHmmss");

// step 2: use format() method by passing new Date() object from

// SimpleDateFormat class.

String abc simple.format(new Date());

System.out.println(abc);

}

}

Scenario 1;

open a facebook

enter the first name

capture screenshot and provide the screenshot name as facebook14052024\_195788.png

step 1: connect actual browser

System.setProperty("webdriver.chrome.driver","path of chrome driver");

step 2: up casting

WebDriver driver new ChromeDriver();

step 3; open a url

driver.get("https://www.facebook.com/reg");

step 4: find the first name text box and enter the value

driver.findElement(By.name("firstname")).sendKeys("Pooja")

step 5: convert the WebDriver object into the TakesScreenshot

TakesScreenshot ts (TakesScreenshot)driver;

step 6: use getScreenshotAs() method by passing OutputType.FILE

File src ts.getScreenshotAs(OutputType.FILE);

step 7;create object of SimpleDateFormat class by passing date and time format

SimpleDateFormat simple new SimpleDateFormat("ddMMyyyy\_HHmmss");

step 8: use format() method by passing Date class object

String abc simple.format(new Date());

step 9: create object of File class by passing file Destination location

capture screenshot and provide the screenshot name as facebook14052024\_195788.png

File dest new File(System.getProperty("user.dir")+"\\PassScreenshot\\facebook"+abc+".png");

step 10: copy the screenshot from source location to destination location

FileUtils.copyFile(src, dest);

package Tutorial15;

import java.io.File;

import java.io.IOException;

import java.text.SimpleDateFormat;

import java.util.Date;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.By;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo4 {

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

// Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// up casting

WebDriver driver new ChromeDriver();

// open a url

driver.get("https://www.facebook.com/reg");

// find the first name text box

driver.findElement(By.name("firstname")).sendKeys("Rahul");

// convert the WebDriver object into the TakesScreenshot

TakesScreenshot ts (TakesScreenshot) driver;

// use getScreenshotAs() method by passing OutputType.FILE

File src ts.getScreenshotAs(OutputType.FILE);

// create object of SimpleDateFormat class by passing date and time format

SimpleDateFormat simple new SimpleDateFormat("ddMMyyyy\_HHmmss");

// use format() method by passing Date class object

String abc simple.format(new Date());

// store the screenshot in destination location by creating object of File

// class and passing File destination location

File dest new File(System.getProperty("user.dir") + "\\PassScreenshot\\facebook" + abc + ".png");

// copy the screenshot from source location and store in destination location

// using copyFile() method

FileUtils.copyFile(src, dest);

}

}

**How to capture the Full Screenshot in selenium WebDriver VVVVVIMMMPPPPP**

- AShot library is used to take the full page screenshot of visible part of the web page and then it scroll to capture the rest of the page

screenshot.

Step 1: create object of AShot class

AShot a new AShot();

step 2: use shootingStrategy() method by passing ShootingStrategies.viewPortPasting() method and inside the viewPortPasting method pass the time in milliseconds.

a.shootingStrategy(ShootingStrategies.viewPortPasting(1000));

step 3: use takeScreenshot() method by passing WebDriver instance.

Screenshot sc a.takeScreenshot(driver);

Step 4: capture the actual screenshot using getImage() method

BufferedImage src sc.getImage();

step 4:if u want to store the screenshot in destination location then we create object of File class by passing file destination location

File dest new File("file destination location");

step 5:capture the screenshot from source location and store in destination location using write() method and by passing source location, image type and destination location from ImageIO class.

ImageIO.write(src, "PNG", dest);

-

Scenario 1:

open amazon and take full page screenshot

Step 1: connect to actual browser

step 2: up casting

step 3: open a url

driver.get("https://www.amazon.com");

step 4: create object of AShot class

AShot a new AShot();

Step 5: use shootingStrategy() method to capture the screenshot of visible part of the page and use viewPortPasting() from ShootingStrategies class to capture screenshot of remaining web page by scrolling the page. and by mentioning the scrolling to wait to load the web page.

a.shootingStrategy(ShootingStrategies.viewPortPasting(2000));

step 6: use takeScreenshot() method by passing WebDriver instance.

Screenshot sc a.takeScreenshot();

step 7: capture the actual screenshot in source location then we use getImage() method

BufferedImage src sc.getImage();

step 8: store the screenshot in destination location by creating object of File class

File dest new File("file destination);

step 9: capture the capture the screenshot from source location and store in destination location using write() method and by passing 3 arguments

as source location of screenshot and image type as PNG and destination location of screenshot

ImageIO.write(src,"PNG",dest);

---------

note:

<!-- https://mvnrepository.com/artifact/ru.yandex.qatools.ashot/ashot -->

<dependency>

<groupId>ru.yandex.qatools.ashot</groupId>

<artifactId>ashot</artifactId>

<version>1.5.4</version>

</dependency>

---------

package Tutorial15;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import javax.imageio.ImageIO;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import ru.yandex.qatools.ashot.AShot;

import ru.yandex.qatools.ashot.Screenshot;

import ru.yandex.qatools.ashot.shooting.ShootingStrategies;

public class Demo5 {

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

// Step 1: connect to actual browser

// System.setProperty("webdriver.chrome.driver",

// "C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://github.com/rest-assured/rest-assured/wiki/Usage");

// step 4: create object of AShot class

AShot a new AShot();

// step 5: use shootingStrategy() method to capture the visible part of the web

// page screenshot and use viewPortPasting() method from ShootingStrategies

// class to capture

// screenshot of remaining web page by scrolling web page and by mentioning

// scrolling to wait

a.shootingStrategy(ShootingStrategies.viewportPasting(1000));

// step 6: use takeScreenshot by passing WebDriver instance

Screenshot sc a.takeScreenshot(driver);

// step 7: get the captured screenshot using getImage() method

BufferedImage scr sc.getImage();

// step 8: store screenshot in destination location by creating object of File

// class

File f new File(System.getProperty("user.dir") + "\\PassScreenshot\\p3.png");

// step 8: copy the screenshot from source to destination location using write()

// method and

// by passing source location and image type and destination location.

ImageIO.write(scr, "PNG", f);

}

}

package Tutorial15;

import java.io.File;

import java.io.IOException;

import javax.imageio.ImageIO;

import org.apache.commons.io.FileUtils;

import org.openqa.selenium.OutputType;

import org.openqa.selenium.TakesScreenshot;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import ru.yandex.qatools.ashot.AShot;

import ru.yandex.qatools.ashot.Screenshot;

import ru.yandex.qatools.ashot.shooting.ShootingStrategies;

public class Demo6 {

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

WebDriver driver new ChromeDriver();

// takes screenshot

File src ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(src, new File("destination location"));

// Screenshot using AShot

Screenshot sc new AShot().shootingStrategy(ShootingStrategies.viewportPasting(2000)).takeScreenshot(driver);

ImageIO.write(sc.getImage(), "PNG", new File("destination location"));

}

}

**How to handle Window POP in selenium? VVVVVVIMMMMPPPPPPP**

**or**

**How to handle multiple tabs in selenium? VVVVVVIMMMMPPPPPPP**

**or**

**how to handle multiple window in selenium? VVVVVVIMMMMPPPPPPP**

---

- If u want to handle the ""window pop"" or ""multiple tabs in browser""" or ""multiple window"" in selenium then we use

getWindowHandle() and getWindowHandles() method.

- if u want to switch focus to the window pop /browser tab /browser window then we need the window id number.

- getWindowHandle() method is used to capture the current window id numbers.

- getWindowHandle() method is present inside the WebDriver interface and return type is String.

- getWindowHandle() method it return the window id number which is combination of character and integer numbers.

syntax;

String a driver.getWindowHandle();

- getWindowHandles() method is used to capture all window id numbers which is opened by selenium script.

- getWindowHandles() method is present inside the WebDriver interface and return type is """Set<String>""".

syntax;

Set<String> b driver.getWindowHandles();

-if u want to switch focus to the window then we use switchTo().window() method by passing the window id number.

-switchTo() present inside the WebDriver interface and return type is TargetLocator interface.

- window() method is present inside the TargetLocator interface and return type is WebDriver interface.

syntax;

driver.switchTo().window("window id number");

---

Scenario 1:

- Open a url https://prafpawar11.github.io/multiplewindows.html

- capture current window id number

- Enter the values in first name text box

- click on orange HRM link

- capture all window id Number

- capture the orange HRM id number

- switch focus to orange HRM window

- enter value in username text box

- switch focus to main window

- enter the text in last name text box.

Step 1: connect browser

step 2: up casting

step 3: open a url

step 4: capture current window id number

String parentWindowId driver.getWindowHandle();

step 5: find and enter text in first name text box

driver.findElement(By.name("")).sendKeys("");

step 6: click on orange HRM link

driver.findElement(By.linkText("Visit Orange HRM")).click();

step 7: capture all window id numbers

Set<String> allWindowId driver.getWindowHandles();

step 8: capture the Orange HRM application Id number

String orangeHrmWindowIdnull;

//it will iterate both window id number

for(String abc: allWindowId)

{

//it check condition as if parentWindowId not equal to other id number then run the if block of code.

if(!abc.equals(parentWindowId))

{

orangeHrmWindowIdabc;

}

}

step 9: switch focus to orange HRM window

driver.switchTo().window(orangeHrmWindowId);

step 10: find and enter the values in username text box

driver.findElement(By.name("")).sendKeys("");

step 11: switch focus to main window

driver.switchTo().window(parentWindowId);

step 12: find and enter the values in last name text box

driver.findElement(By.name("")).sendKeys("");

package Tutorial16;

import java.time.Duration;

import java.util.Set;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

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

// Step 1: connect to actual browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30));

// step 3: open a url

driver.get("https://prafpawar11.github.io/multiplewindows.html");

// step 4: capture the current or parent window id number

String parentWindowId driver.getWindowHandle();

System.out.println(parentWindowId);

System.out.println("---");

// step 5: find and enter values in first name text box

driver.findElement(By.name("fname")).sendKeys("Amruta");

Thread.sleep(5000);

// step 6: find and click on orange HRM link

driver.findElement(By.linkText("Visit Orange HRM")).click();

// step 7: capture all window id numbers

Set<String> allWindowId driver.getWindowHandles();

// step 8: capture the child window id or orange HRM application id Number

// step 8a) define the string null variable

String orangeHrmWindowID null;

// step 8a) : iterate the all window id numbers

for (String abc : allWindowId) {

if (!abc.equals(parentWindowId)) {

// reassign value for orangeHrmWindowID

orangeHrmWindowID abc;

}

}

// step 9: switch focus to Orange HRM application

driver.switchTo().window(orangeHrmWindowID);

// step 10: find and enter the value in username text box

driver.findElement(By.name("username")).sendKeys("Admin");

Thread.sleep(5000);

// step 11: switch to parent window

driver.switchTo().window(parentWindowId);

// step 12: find and enter text in last name text box

driver.findElement(By.name("lname")).sendKeys("Patil");

}

}

Scenario 2:

Open a https://prafpawar11.github.io/multiplewindows.html

capture current or parent window id number

find and enter the firstname

click on facebook link

capture all window id number

capture the facebook window id number

switch focus to facebook window

enter first name last name , username and password

switch focus to parent window

find and enter last name

click on orange HRM link

capture all window id number

capture the orange hrm window id number

switch focus to orange hrm window

find and enter username, password and click on login button

switch to parent window

find and enter the address

Step 1: connect to actual browser

step 2; up casting

step 3: open a url

step 4: capture the current or parent window id numbers

String parentWindowId driver.getWindowHandle();

step 5: find and enter value in first name text box

driver.findElement(By.name("")).sendKeys("");

step 6: click on facebook link

driver.findElement(By.name("")).click();

step 7: capture all window id number

Set<String> allWindowID driver.getWindowHandles();

step 8: capture the Facebook window id number

String facebookWindowIDnull;

for(String abc: allWindowID)

{

if(!abc.equals(parentWindowId))

{

facebookWindowIDabc;

}

}

step 9: switch focus to facebook window

driver.switchTo().window(facebookWindowID);

step 10: find and enter value in firstname, last name, username text box

driver.findElement(By.name("")).sendKeys("");

driver.findElement(By.name("")).sendKeys("");

driver.findElement(By.name("")).sendKeys("");

step 11: switch focus to parent window

driver.switchTo().window(parentWindowId);

step 12: find and enter value in last name text box

driver.findElement(By.name("")).sendKeys("");

step 13: find and click on orange HRM link

driver.findElement(By.linkText("")).click();

step 14: capture all window id number

Set<String> allWindowIDNumber driver.getWindowHandles();

step 15: capture the orange HRM window id

String orangeHRMWindowIDnull;

for(String abc: allWindowIDNumber )

{

if(!(abc.equals(parentWindowId) || abc.equals(facebookWindowID))

{

orangeHRMWindowIDabc;

}

}

step 16: switch focus to orange HRM window

driver.switchTo().window(orangeHRMWindowID);

step 17:find and enter the value username and password text box and click on login button

driver.findElement(By.name("")).sendKeys("");

driver.findElement(By.name("")).sendKeys("");

driver.findElement(By.name("")).click();

step 18: switch focus to the main window

driver.switchTo().window(parentWindowId);

step 19: find and enter value in address text box

driver.findElement(By.name("")).sendKeys("");

package Tutorial16;

import java.util.Set;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo2 {

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

// Step 1: connect browser

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\praf0\\OneDrive\\Desktop\\Automation Testing 27th April 2024 Batch\\chromedriver.exe");

// step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://prafpawar11.github.io/multiplewindows.html");

// step 4:capture the current or parent window id

String parentWindowId driver.getWindowHandle();

// step 5: find and enter value in first name

driver.findElement(By.name("fname")).sendKeys("Amar");

Thread.sleep(5000);

// step 6: click on facebook link

driver.findElement(By.linkText("Visit Facebook")).click();

// step 7: capture all window id number

Set<String> allWindowId driver.getWindowHandles();

// step 8: capture the facebook window id number

String facebookWindowID null;

for (String abc : allWindowId) {

if (!abc.equals(parentWindowId)) {

facebookWindowID abc;

}

}

// step 9: switch focus to facebook window

driver.switchTo().window(facebookWindowID);

// step 10: find and enter value in first name lastname and username

driver.findElement(By.name("firstname")).sendKeys("Dipali");

driver.findElement(By.name("lastname")).sendKeys("patil");

driver.findElement(By.name("reg\_email\_\_")).sendKeys("dipali@gmail.com");

Thread.sleep(5000);

// step 11: switch focus to main window

driver.switchTo().window(parentWindowId);

// step 12: find and enter value in last name text box

driver.findElement(By.name("lname")).sendKeys("patil");

Thread.sleep(5000);

// step 13: click on orange HRM link

driver.findElement(By.partialLinkText("Orange HRM")).click();

Thread.sleep(5000);

// step 14: capture all window id numbers

Set<String> allwindowIdNumbers driver.getWindowHandles();

// step 15: capture orange hrm window id

String orangeHrmWindowID null;

for (String abc : allwindowIdNumbers) {

if (!(abc.equals(parentWindowId) || abc.equals(facebookWindowID))) {

orangeHrmWindowID abc;

}

}

// step 16: switch focus to orange hrm

driver.switchTo().window(orangeHrmWindowID);

// step 17: find and enter value in username, password and click on login button

driver.findElement(By.name("username")).sendKeys("Admin");

driver.findElement(By.name("password")).sendKeys("admin123");

Thread.sleep(5000);

driver.findElement(By.xpath("//button[text()' Login ']")).click();

Thread.sleep(5000);

// step 18: switch focus to main window

driver.switchTo().window(parentWindowId);

// step 19:find and enter value in address

driver.findElement(By.name("address")).sendKeys("Pune");

}

}

Ass 1:

open https://prafpawar11.github.io/multiplewindows.html

enter value in first name text box

take screenshot

click on sauce demo link

enter valid credential

take screenshot

add 3 products in add to cart

click on add to container

take screenshot

click on checkout

enter first name, last name and pincode and click on continue

click on finish

take screenshot

switch focus to main window

enter lastname

take screenshot

click on cogmento CRM Link link

enter valid credentials prafulp1010@gmail.com Pr@ful0812

take screenshot

click on contact links

click on create button

enter firstname, last name , select status as on Hold, click on do not call toggle switch

take screenshot and click on save button

delete the created user

click on logout

switch focus to main window

take screenshot

-

Ass 2:

open a https://www.flipkart.com/

mouse over Electronics

mouse over Gaming

take screenshot

click on Gaming mouse

click on 1st one item

capture the price

take screenshot

go to main window or search window

click on 2nd item

capture the price

take screenshot

compare the captured price

-

**How to Handle multiple windows? VVVVVIMMPPPPPPPPPP**

**or**

**How to handle Multiple tabs? VVVVVIMMPPPPPPPPPP**

**or**

**How to handle window pop up? VVVVVIMMPPPPPPPPPP**

**--------**

**Handle multiple windows or tabs or window pop up by using ArrayList class.**

Scenario 1:

- open a url https://prafpawar11.github.io/multiplewindows.html

- capture the parent window id number

- find and enter the value in first name text box

- click on sauce demo link

- capture all window id number

- convert the Set<String> into ArrayList<String> because we capture the window id number as per the index position.

- capture sauce demo window id number

- switch focus to sauce demo application

- enter valid credentials

- switch focus to main window

- find and enter values in last name text

Step 1: connect to actual browser

step 2; up casting

WebDriver driver new ChromeDriver();

step 3: open a url

driver.get("https://prafpawar11.github.io/multiplewindows.html");

step 4: find and enter value in first name text box

driver.findElement(By.name("fname")).sendKeys("abc");

step 5: capture the parent/current window id number

String parentWindowId driver.getWindowHandle();

step 6: find and click on sauce demo link

driver.findElement(By.linkText("SauceDemo Link")).click();

step 7: capture all window id numbers

Set<String> allWindow driver.getWindowHandles();

step 8: convert the Set<String> into ArrayList<String> because we can capture the window id number as per the index position.

we have to create object of ArrayList<String> class by passing Set<String> object name or instance name

ArrayList<String> arr new ArrayList<String>(allWindow);

step 9: capture window id of sauce demo application

String sauceDemoWindowID arr.get(1);

step 10: switch focus to sauce demo application

driver.switchTo().window(sauceDemoWindowID);

step 11: find and enter valid credentials

driver.findElement(By.id("user-name")).sendKeys("standard\_user");

driver.findElement(By.id("password")).sendKeys("secret\_sauce");

driver.findElement(By.id("login-button")).click();

step 12: switch focus to main window

String parentId arr.get(0);

driver.switchTo().window(parentId);

step 13: find and enter value in last name text box

driver.findElement(By.name("lname")).sendKeys("abc");

package Tutorial17;

import java.util.ArrayList;

import java.util.Set;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

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

// Step 1: Connect to browser

// Step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://prafpawar11.github.io/multiplewindows.html");

// step 4: find and enter value in first name text box

driver.findElement(By.name("fname")).sendKeys("Akash");

Thread.sleep(5000);

// step 5: find and click on sauce demo link

driver.findElement(By.linkText("SauceDemo Link")).click();

// step 6; capture all window id numbers

Set<String> allwindow driver.getWindowHandles();

// step 7: convert the Set<String> into ArrayList<String> because by using

// ArrayList object we capture window id Number as per the index position

ArrayList<String> arr new ArrayList<String>(allwindow);

// step 8: capture sauce demo window id

String sauceDemoWindowID arr.get(1);

// step 9: switch focus to sauce demo application

driver.switchTo().window(sauceDemoWindowID);

Thread.sleep(5000);

// step 10: find and enter valid credentails

driver.findElement(By.id("user-name")).sendKeys("standard\_user");

driver.findElement(By.id("password")).sendKeys("secret\_sauce");

driver.findElement(By.id("login-button")).click();

Thread.sleep(5000);

// step 11: capture the parent window id

String parentWindowID arr.get(0);

// step 12: switch focus to parent window

driver.switchTo().window(parentWindowID);

// step 13: find and enter value in last name text box

driver.findElement(By.name("lname")).sendKeys("secret\_sauce");

}

}

Scenario 2:

open a https://prafpawar11.github.io/multiplewindows.html

capture title

click on facebook link

capture all window id

convert Set<String> into ArrayList<String> because by using ArrayList object we can capture window id as per the index position.

capture fb window id

switch focus to facebook window

capture the title

capture the main window id

switch focus to main window

capture url

click on cogmento CRM link

capture all window id

convert Set<String> into ArrayList<String> because by using ArrayList object we can capture window id as per the index position.

capture the cogmento CRM window id number

switch focus to cogmento CRM window

capture title

switch focus to fb window

capture url

switch focus to main window

capture url

switch focus to cogmento CRM window

capture url

Step 1: connect to browser

step 2: up casting

step 3: open a url

step 4: capture title

String a driver.getTitle();

step 5: find and click on facebook link

driver.findElement(By.linkText("Visit Facebook")).click();

step 6: capture all window id number

Set<String> allWindow driver.getWindowHandles();

step 7: convert the Set<String> into ArrayList<String> because by using ArrayList object we can capture the values as index position.

ArrayList<String> arr new ArrayList<String>(allWindow);

step 8: capture the facebook window id number

String facebookWindowID arr.get(1);

step 9: switch focus to facebook window

driver.switchTo().window(facebookWindowID);

step 10: capture the title

String b driver.getTitle();

step 11: capture the main window id number

String parentWindowId arr.get(0);

step 12: switch focus to parent window

driver.switchTo().window(parentWindowId);

step 13: capture url

String c driver.getCurrentUrl();

step 14: find and click on cogmento CRM

driver.findElement(By.linkText("cogmento CRM Link")).click();

step 15: capture all window id number

allWindow driver.getWindowHandles();

step 16: convert the Set<String> into ArrayList<String> because by using ArrayList object we can capture window id as per the index position

arr new ArrayList<String>(allWindow);

step 17: capture cogmento crm window id number

String cogmentoCRMWindowID arr.get(2);

step 18; switch focus to cogmento window

driver.switchTo().window(cogmentoCRMWindowID);

step 19:capture title

String d driver.getTitle();

step 20: switch focus to fb window

driver.switchTo().window(facebookWindowID);

step 21:capture url

String e driver.getCurrentUrl();

step 22: switch focus to main window

driver.switchTo().window(parentWindowId);

step 23:capture url

String e driver.getCurrentUrl();

step 24: switch focus to cogmento window

driver.switchTo().window(cogmentoCRMWindowID);

step 25: capture url

String f driver.getCurrentUrl();

package Tutorial17;

import java.util.ArrayList;

import java.util.Set;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.firefox.FirefoxDriver.SystemProperty;

public class Demo2 {

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

// Step 1: Connect to browser

// Step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://prafpawar11.github.io/multiplewindows.html");

// step 4: capture title

System.out.println(driver.getTitle());

Thread.sleep(5000);

// step 5: find and click on sauce demo link

driver.findElement(By.partialLinkText("Facebook")).click();

// step 6: capture all window id

Set<String> allWindow driver.getWindowHandles();

// step 7: convert the Set<String> into ArrayList<String>

ArrayList<String> arr new ArrayList<String>(allWindow);

// step 8: switch focus to fb window

driver.switchTo().window(arr.get(1));

// step 9: capture title

System.out.println(driver.getTitle());

Thread.sleep(5000);

// step 10: switch focus to main window

driver.switchTo().window(arr.get(0));

// step 11: capture url

System.out.println(driver.getCurrentUrl());

Thread.sleep(5000);

// step 12: find and click on cogmento crm link

driver.findElement(By.partialLinkText("CRM")).click();

// step 13: capture all window id

allWindow driver.getWindowHandles();

// step 14: convert Set<String> into ArrayList<String> because by using

// ArrayList object we can capture the values as per the inde posittion

arr new ArrayList<String>(allWindow);

Thread.sleep(5000);

// step 15: switch focus to cogmento window

driver.switchTo().window(arr.get(2));

// step 16: capture url

System.out.println(driver.getCurrentUrl());

Thread.sleep(5000);

// step 17: switch to focus to fb window

driver.switchTo().window(arr.get(1));

// step 18: capture fb url

System.out.println(driver.getCurrentUrl());

// step 19: switch focus to main window

driver.switchTo().window(arr.get(0));

// step 20: capture fb url

System.out.println(driver.getCurrentUrl());

}

}

Scenario:

open a https://prafpawar11.github.io/multiplewindows.html

find and enter value in first name text box in main window

click on fb link

enter fname, lastname , username ,dob

switch focus to main window

find and enter lastname

click on sauce demo

enter valid credentials

take screenshot

switch focus to fb window

take screenshot----------> take full screenshot

switch focus to main window

find and enter address

take screenshot

**Diff between absolute xpath and relative xpath VVVVVIMMMPPPPP**

**absolute xpath**

- absolute xpath is complete path from root node to desired node/element

- absolute xpath it starts from single forward slash.

- in absolute xpath failure chances are more because if there is any changes made in middle of the element then it changes the path or xpath.

- absolute xpath is faster than relative xpath.

- in absolute xpath we can not use HTML tag name attribute.

- in absolute xpath we can not navigate to backward direction,

-in absolute xpath we find the xpath in forward direction only

e.g. /html/body/div/div[1]/span[2]/input[4] after adding new tags it changes the absolute xpath

/html/body/div/div[1]/span[2]/input[3] previous absolute xpath

**Relative xpath**

- In relative xpath, we can simply start referencing the element middle of the html code or DOM structure.

- Relative xpath it starts with double forward slash

- in Relative xpath we don't mention long xpath.

- failure changes are less in Relative xpath because we search element middle of the points.

- Relative xpath is slower than Absolute xpath.

- in Relative xpath we can use any HTML tag name attribute.

- in Relative xpath we can navigate forward as well as backward direction.

- in Relative xpath we can use multiple keywords or methods to locate the element.

- examples we can use contains() method, text() method, starts-with() method, or keyword, and keyword, following keyword, preceding keyword, following-sibling keyword, preceding-sibling keyword, parent keyword, child keyword, ancestor keyword and descendant keyword.

--

Ass 1:

why we don't use absolute xpath?

-

Ass 2:

why we use Relative xpath

--------

//input[@id"state"]/following::span

Result

span class b3

span class b4

span class b5

span class b6

//input[@id"state"]/following::div

Result

div class a2,

div class a3,

div class a4

//input[@id"state"]/following::input

Result

Country input tag

degree input tag

branch input tag

university input tag

project input tag

team input tag

domain input tag

agree input tag

submit input tag

cancel input tag

//input[@id"state"]/following::a

Result

google a tag link

facebook a tag link

------

following-sibling keyword

it is used to find the sibling of next tag in same parent tag

//input[@id'state']/following-sibling::input

Result

country input tag

------

preceding-sibling keyword

it is used to find the sibling of previous tag in same parent tag

//input[@id'state']/preceding-sibling::input

Result

city input tag

//input[@id'state']/preceding-sibling::span

Result

Nothing tag found

------

preceding keyword

it is used to find the previous all tags

//input[@id'state']/preceding::input

Result

first name input tag

middle name input tag

last name input tag

city input tag

//input[@id'state']/preceding::div

Result

Nothing tag found

//input[@id'state']/preceding::span

Result

span b1 class

------

parent keyword

it used to find the immediate parent tag

//input[@id'state']/parent::span

------

ancestor keyword

it used to find the immediate parent tag, parent of parent tag and grand parent of parent tag etc.

//input[@id'state']/ancestor::div

Result

div class a2

div class main

//input[@id'state']/ancestor::span

Result

span class b2

//input[@id'state']/ancestor::\*

Result

html

body

div main class tag

div a1 tag

span b2 tag

------

child keyword:

it is used to find the immediate child

//div[@class'a2']/child::span

Result

span b4 class

span b5 class

------

descendant keyword

it is used to find the child, grand child, grand of grand child etc

//div[@class'a2']/descendant::span

Result

span b4 class

span b5 class

//div[@class'a2']/descendant::input

Result

project input tag

team input tag

domain input tag

------

<!DOCTYPE html>

<html lang"en">

<head>

<meta charset"UTF-8">

<meta name"viewport" content"widthdevice-width, initial-scale1.0">

<title>Document</title>

<style>

.main

{

padding-left: 10px;

padding-top: 10px;

padding-right: 20px;

margin-left: 300px;

margin-right: 70px;

border: 2px;

}

</style>

</head>

<body>

<div class"main">

<div class"a1">

<span class"b1">

First Name<input type"text" id"fname" /> <br> <br>

Middle Name<input type"text" id"mname" /><br> <br>

Last Name<input type"text" id"lname" /> <br> <br>

</span>

<span class"b2">

Area <input type"text" id"area"/> <br>

<br>

City <input type"text" id"city" /> <br> <br>

<label for"state">State</label>

<input type"text" id"state" name"state"></input> <br> <br>

<label for"country">Country</label>

<input type"text" id"country" name"country" /><br> <br>

</span>

<span class"b3">

Degree <input type"text" id"degree" /><br> <br>

Branch <input type"text" id"branch" /><br> <br>

University <input type"text" id"university" /><br> <br>

</span>

</div>

<div class"a2">

<span class"b4">

project name <input type"text" id"pname" /><br> <br>

Team Name <input type"text" id"tname" /><br> <br>

</span>

<span class"b5">

Domain name <input type"text" id"dname" /><br> <br>

</span>

</div>

<div class"a3">

Agree <input type"checkbox" id"agree" /><br> <br>

<input type"button" id"btn"> Submit </input>

<input type"button" id"cancel"> Cancel </input><br> <br>

</div>

<div class"a4">

<span class"b6">

<a href"https://www.google.com"> Google Link </a><br> <br>

<a href"https://www.facebook.com"> Facebook Link </a><br> <br>

</span>

</div>

</div>

</body>

</html>

**When we get StaleElementReferenceException?**  VVVVVVIMMMPPPPPPPPPPPPP

- if we are trying to locate the element and inside the HTML DOM structure and element is not present or removed from HTML DOM structure then we will get StaleElementReferenceException.

- just assume scenario.

we have find the all drop down values and then we have iterated all values and once we find the expected values then we click on it,

and if we are trying to find the next drop down values then we get stale element exception because all drop down values is not is visible state.

-------

**How to handle drop down which is started """""without select tag""" in HTML DOM structure? VVVVVVVIMMMPPPP**

-------

- first we have to find the dropdown element using findElement() method and given locator

- then we have click on drop down element.

- then next we have to capture all dropdown values using findElements() method and then next we iterate all dropdown values using loop.

- and inside the loop we mention if condition and if expected values is present then click on it and break the loop.

syntax:

//click on drop down

WebElement wb1 driver.findElement(By.id("id value"));

wb1.click();

//find all drop down values

List<WebElement> ls driver.findElements(By.xpath("xpath value"));

//iterate all drop down values

for(WebElement abc: ls)

{

//capture drop down text one by one

String xyz abc.getText();

//if expected value is present then run the condition

if(xyz.equals("Expected value"))

{

//condition is true then click on Expected drop down element

abc.click();

break;

}

}

-------

Scenario

Open a https://www.vsp.com/create-account

find the language drop down and click on it

capture all dropdown values

select the Lao language.

Step 1: connect to the actual browser

step 2; up casting

step 3: open a url

driver.get("https://www.vsp.com/create-account");

step 4: find the language drop down and click on it

WebElement wb driver.findElement(By.xpath("//div[@class'ng-input']"));

wb.click();

step 5: find all drop down values and capture all drop down values

List<WebElement> ls driver.findElements(By.xpath("//span[@class'ng-option-label ng-star-inserted']"));

step 6: iterate all drop down and if expected value is present click on it and break the loop.

//iterate all drop down

for(WebElement abc: ls)

{

String xyz abc.getText();

//if expected value is present

if(xyz.equals("Farsi"))

{

// click on it

abc.click();

//break the loop.

break;

}

}

package Tutorial18;

import java.time.Duration;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

public class Demo1 {

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

// Step 1; connect to the actual browser

// Step 2: up casting

WebDriver driver new ChromeDriver();

// step 3: open a url

driver.get("https://www.vsp.com/create-account");

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30));

Thread.sleep(5000);

// step 4: find the dropdown element and click on it

WebElement wb driver.findElement(By.xpath("//div[@class'ng-input']"));

wb.click();

// find all drop down values

List<WebElement> ls driver.findElements(By.xpath("//span[@class'ng-option-label ng-star-inserted']"));

// iterate all drop down values

for (WebElement abc : ls) {

Thread.sleep(2000);

/// capture the drop down text

String xyz abc.getText();

System.out.println(xyz);

if (xyz.equals("Korean")) {

abc.click();

break;

}

}

}

}

package Tutorial18;

import java.time.Duration;

import java.util.List;

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.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

public class Demo2 {

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

WebDriver driver new ChromeDriver();

driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");

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

driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(30));

Thread.sleep(5000);

WebElement username driver.findElement(By.name("username"));

if (username.isDisplayed() && username.isEnabled()) {

username.sendKeys("Admin");

}

// Password Text box

WebDriverWait wait new WebDriverWait(driver, Duration.ofSeconds(45));

WebElement password wait.until(ExpectedConditions.visibilityOfElementLocated(By.name("password")));

password.sendKeys("admin123");

Thread.sleep(5000);

// click on login button

WebElement loginButton wait

.until(ExpectedConditions.elementToBeClickable(By.xpath("//button[text()' Login ']")));

loginButton.click();

Thread.sleep(5000);

WebElement pimLink driver.findElement(By.xpath("//span[text()'PIM']"));

pimLink.click();

Thread.sleep(5000);

// click on drop down element

WebElement jobTitleWb driver.findElement(By.xpath(

"//label[text()'Job Title']/parent::div/following-sibling::div/descendant::div[@class'oxd-select-text--after']"));

jobTitleWb.click();

// capture all drop down values

List<WebElement> ls driver.findElements(By.xpath("//div[@class'oxd-select-option']/child::span"));

// iterate all drop down values

for (WebElement abc : ls) {

Thread.sleep(2000);

// capture drop down element physical text

String xyz abc.getText();

System.out.println(xyz);

if(xyz.equals("IT Manager"))

{

abc.click();

break;

}

}

Thread.sleep(5000);

driver.quit();

}

}

<!-- https://mvnrepository.com/artifact/ru.yandex.qatools.ashot/ashot -->

<dependency>

<groupId>ru.yandex.qatools.ashot</groupId>

<artifactId>ashot</artifactId>

<version>1.5.2</version>

</dependency>