

# ***SELENIUM FULL MATERIAL***

## **CONTENTS**

- 1. INTRODUCTION OF AUTOMATION TESTING**
- 2. BROWSER LAUNCHING**
- 3. WEB DRIVER METHODS**
- 4. NAVIGATION COMMANDS**
- 5. LOCATORS**
- 6. WEBELEMENT METHODS**
- 7. RADIO BUTTON**
- 8. HANDLING MULTIPLE WEBELEMENTS AND AUTO SUGGESTION**
- 9. CHECK BOX**
- 10. DROP DOWN**
- 11. WEB TABLE**
- 12. JAVASCRIPT EXECUTOR**
- 13. SCROLL UP AND SCROLL DOWN**
- 14. SCREENSHOT**
- 15. MOUSE HOVER ACTION**
- 16. DRAG & DROP AND RIGHT CLICK**
- 17. WINDOW HANDLING**
- 18. FRAME HANDLING**
- 19. ALERTS**
- 20. WAIT**
- 21. FILE UPLOADING USING ROBOT CLASS AND SENDKEYS**
- 22. BROKEN LINKS**

## **INTRODUCTION OF AUTOMATION TESTING**

### **Important Java concepts required for selenium:**

- Conditions(if, if-else, switch)
- Loops(for, while, do-while, for-each)
- OOps(Inheritance, Polymorphism, Encapsulation, Abstraction)
- Method overloading, overriding
- Constructors
- String
- Type casting, Upcasting
- Code optimization
- Collection(List and Set)

### **Automation:**

- Performing any task by using a tool or machine is called as automation.

### **Advantages:**

1. Save the time.
2. Faster
3. Requires less manual effort
4. Restless.
5. Accuracy will be more
6. Multi-task
7. Requires less human resources

### **Dis Advantages:**

1. Initial investment will be more.
2. It requires constant maintenance
3. It requires additional skill sets.

### **Automation testing:**

- Testing an application by using any automation tools is called as automation testing.

### **Automation Tool:**

- It's a software or an application which is used to automate any applications.
  - Ex: Selenium, QTP, Appium, AutoIT etc,

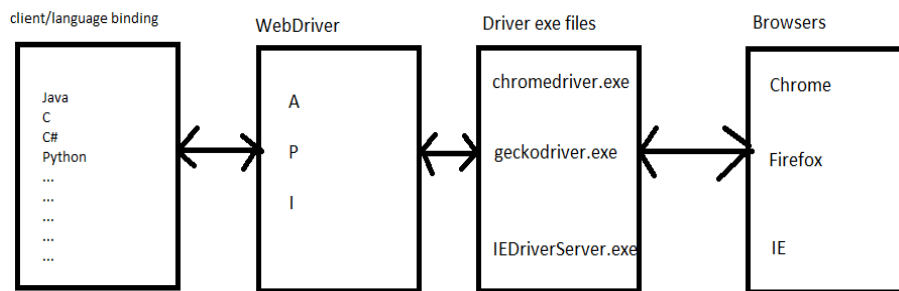
## Selenium:

- It's a free and open source automation tool which is used to automation any web based applications.

## Advantages of selenium:

- It is freely available automation tool. To make use of selenium for commercial purpose we don't have to buy any license. It is available in below website.
  - <https://www.seleniumhq.org/download/>
- Anyone can view source code of selenium which is available in below website.
  - <https://github.com/SeleniumHQ/selenium>
- Using selenium we can automate any web based applications such as gmail, facebook, flipkart etc...
- It supports for 14 programming languages.
- It supports for multiple platforms such as Windows, Mac, Linux.
- It supports all most all the browsers such as chrome, firefox, ie, safari, opera.

## High Level Architecture of selenium:



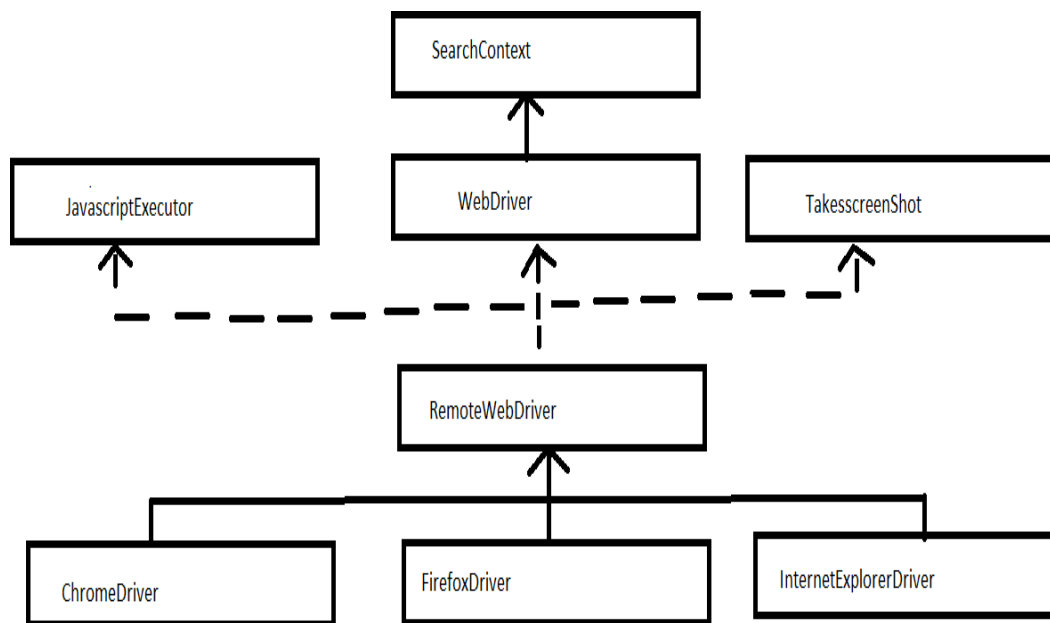
- Selenium supports for multiple languages such as java, c, python etc,
- Each of these languages contains generic libraries(Re-useable codes) which is called as client/language binding.
- These client binding will communicates with as API called WebDriver
- Here client binding will communicate the actions which has to be performed on the browser such as open the browser, enter url/text, click, close etc.

- To perform action on the browser WebDriver will communicate with driver executable files such as chromedriver.exe, geckodriver.exe, IEDriverServer.exe etc

**Note:**

1. To perform action on chrome browser, webdriver will communicate with chromedriver.exe
2. To perform action on firefox browser, webdriver will communicate with geckodriver.exe
3. To perform action on ie browser, webdriver will communicate with IEDriverServer.exe

**Java selenium Architecture:**



**Interface:**

1. SearchContext
2. JavaScriptExecutor

### 3. WebDriver

#### **Classes:**

##### 1. RemoteWebDriver

##### 2. ChromeDriver

##### 3. InternetExplorerDriver

- SearchContext is the super most interface which contains 2 methods,
  1. findElement()
  2. findElements()
- SearchContext interface is inherited by another interface called WebDriver which contains 13 methods including findElement() and findElements()
- There are other 2 interfaces such as
  1. JavascriptExecutor:- which is used to execute javascript statements
  2. TakesScreenshot:- which is used to take screenshots of webpage
- All these interfaces are implemented in a class called RemoteWebDriver (Super most class in selenium)
- All the methods of RemoteWebDriver class are overridden in respective browser classes such as, ChromeDriver, FirefoxDriver, InternetExplorerDriver.

#### **WebDriver methods:**

- |    |                    |
|----|--------------------|
| 1  | get()              |
| 2  | getTitle()         |
| 3  | getCurrentUrl()    |
| 4  | getPageSource()    |
| 5  | findElement()      |
| 6  | findElements()     |
| 7  | getWindowHandle()  |
| 8  | getWindowHandles() |
| 9  | switchTo()         |
| 10 | manage()           |
| 11 | navigate()         |
| 12 | close()            |
| 13 | quit()             |

#### **JavascriptExecutor methods:**

- |   |                      |
|---|----------------------|
| 1 | executeScript()      |
| 2 | executeAsyncScript() |

### **TakesscreenShot methods:**

1. getScreenShotAs()

### **Note:**

1. ChromeDriver class is used to work with chrome browser.
2. FirefoxDriver class is used to work with firefox browser.
3. InternetExplorerDriver class is used to work with ie browser.

### **Tools Required:**

- jdk[1.8 & above]
- Eclipse[Mars, Neon, Oxygen]
- Selenium jar file
- Driver executable files
- Latest version of browsers
- Application Under Testing

### **Steps to install selenium:**

- Download selenium jar file and driver exe files from following website.  
URL→ <http://www.seleniumhq.org/download>
- Extract all the driver exe files
- In eclipse, create a java project with the name Automation.
- Under the Java project create 2 folders with the name drivers and jars  
To create folder, Right click on project→new→create folder
- Store all extracted driver exe files under drivers folder
- Store selenium jar file under jars folder
- Associate selenium jar file with java project  
To associate the jar file, right click on jar file→Build path→Add to build path

### **Note:**

- Before launching the browser we have to set path of the driver exe file.
- We can set path of the driver exe file by using setProperty() of System class.
- setProperty() is a static method which takes 2 args of type String. They are,
  - key
  - value
- key for geckcodriver is, webdriver.gecko.driver
- value is the path of the driver exe file. We can specify the value in any of the following ways,
  - C:\\Users\\Venkat\\Desktop\\capgemini\\Automation\\drivers\\geckodriver.exe
  - C:/Users/Venkat/Desktop/capgeminis/Automation/drivers/geckodriver.exe

- ./drivers/geckodriver.exe(---> path of the current java project)
- If we do not set path of the driver exe file, then it will throw IllegalStateException.

## **BROWSER LAUNCHING**

1. Open the browser

Firefox browser:

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

        // to configure driver
        System.setProperty("webdriver.gecko.driver",

"C:/Users/siva/workspace/Selenium/driver/geckodriver.exe");
        // create the firefox driver

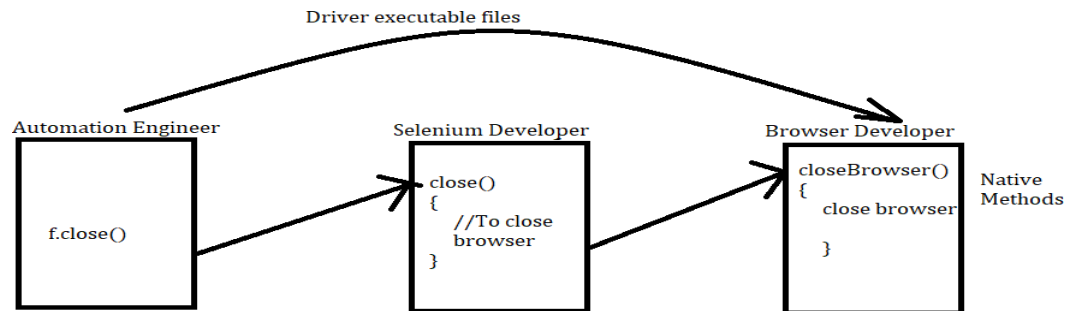
        WebDriver driver = new FirefoxDriver();
        // url mention
        driver.get("https://www.facebook.com/");
        driver.close();
    }
}
```

**Q1: How does selenium performs the action?**

Ans: By calling native methods of the browsers.

**Q2: How do you call native methods of the browsers?**

Ans: By using driver executable files.



### Close();

- It is used to close the application.
- It will close the current browser.

### Quit();

- Destroy the object.
- It will close all browser windows opened by selenium webdriver

### Chrome Launching:

chrome driver-v60=chrome driver-v2.30

```

public class GmailAccount {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver",
            "C:/Users/siva/workspace/Selenium/driver/chromedriver.exe");
    }
}
  
```

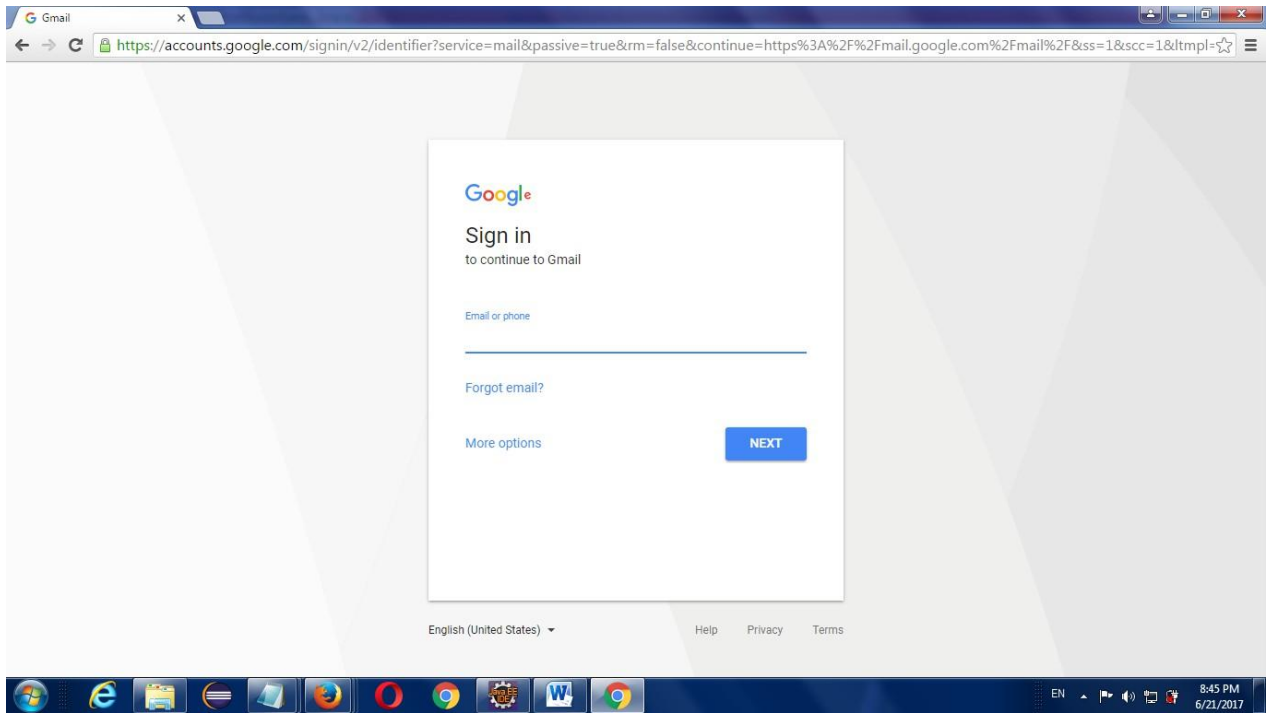
```

WebDriver driver = new ChromeDriver();
driver.get("https://www.gmail.com/");
driver.manage().window().maximize();
  
```

```

}}
  
```

Output of the above program:



### **IE Launching:**

IE-v11.0= internet driver server-v3.4

### **Internet Explorer browser:**

Before launching IE browser we have to do the following settings.

- open the ie browser
- goto tools and click on internet options
- goto security tab
- Select enable protected mode checkbox in all the zone
  - Internet
  - Local intranet
  - trusted sites
  - Restricted sites
- Set the zoom level to 100%

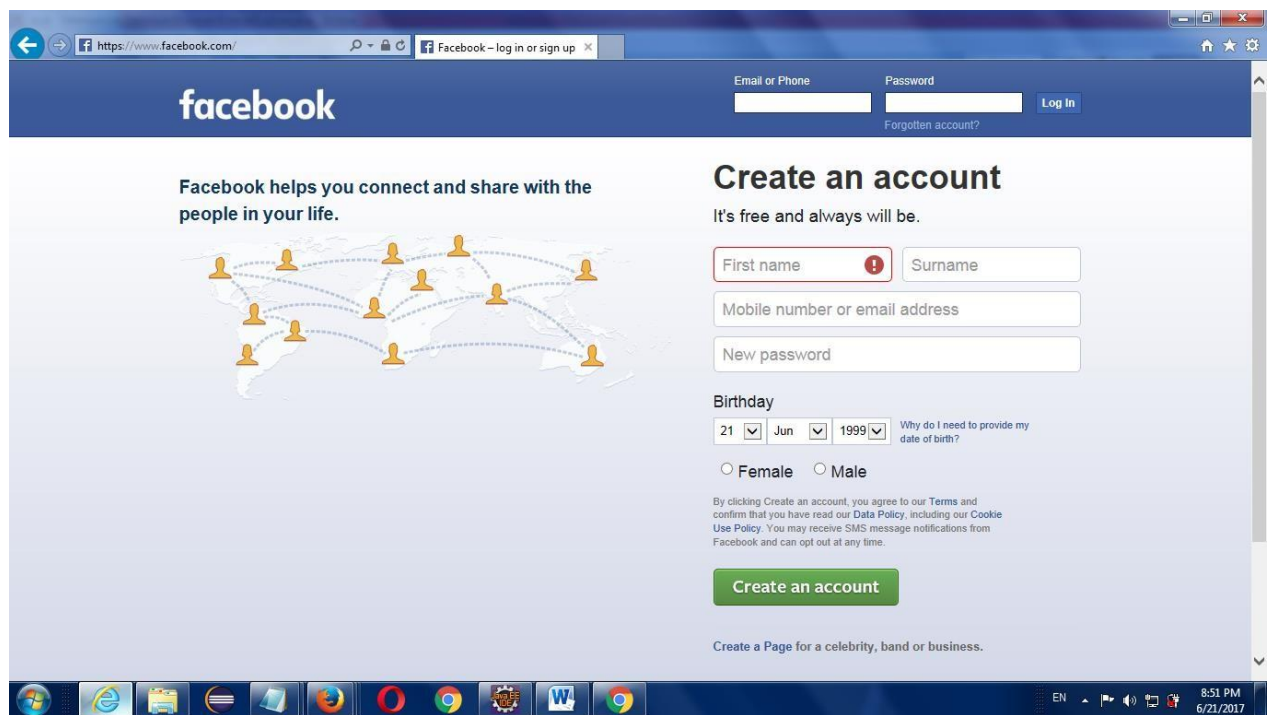
### **IE Launching:**

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

```
System.setProperty("webdriver.ie.driver",
"C:/Users/siva/workspace/Selenium/driver/IEDriverServer.exe");
```

```
WebDriver driver = new InternetExplorerDriver();
driver.get("https://www.facebook.com/");
driver.manage().window().maximize();
}
}
```

Output of the program



### **Opera Launching:**

Opera-v46=opera driver-v2.27

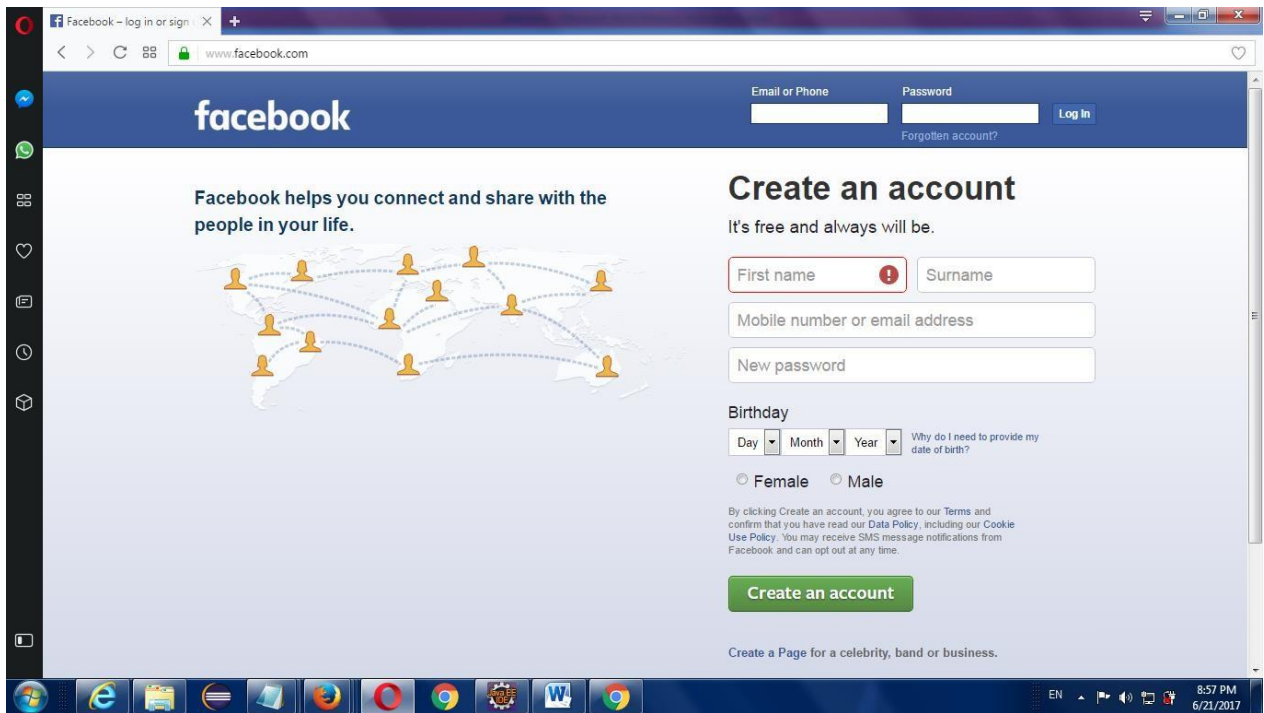
```
public class Opera {

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

```
System.setProperty("webdriver.opera.driver","C:/Users/siva/workspace/Selenium/
driver/operadriver.exe" );
```

```
WebDriver driver = new OperaDriver();
driver.get("https://www.facebook.com/");
driver.manage().window().maximize();
}
}
```

### Output:



### Write a Script to open and close the browser based on user input

```
public class Demo
{
    public static void main(String[] args) throws InterruptedException
    {
        Scanner sc = new Scanner(System.in);
```

```

System.out.println("Enter browser Name:");
String browser = sc.nextLine();

WebDriver driver = null;

if(browser.equals("Firefox"))
{
    System.setProperty("webdriver.gecko.driver",
"./drivers/geckodriver.exe");
    driver = new FirefoxDriver();
}
else
if(browser.equals("Chrome"))
{
    System.setProperty("webdriver.chrome.driver",
"./drivers/chromedriver.exe");
    driver = new ChromeDriver();
}
else
{
    System.out.println("Invalid browser");
}
Thread.sleep(2000);
driver.close();
}
}

```

Note: The above script is an example for Run Time Polymorphism.

- To run same script on multiple browsers, we are converting sub class object into interface type (upcasting).

```

WebDriver driver = new ChromeDriver();

```

```

WebDriver driver = new FirefoxDriver();

```

## **WEB DRIVER METHODS:**

### **Methods of WebDriver Interface:**

1	get()	To enter the url
2	getTitle()	To get the title of current web page
3	getCurrentUrl()	To get the url of current web page
4	getPageSource()	To get the page source of current web page
5	findElement()	To get single webElements
6	findElements()	To get multiple webElements
7	getWindowHandle()	To get the id of parent window
8	getWindowHandles()	To get the id of All windows
9	switchTo()	Used to switch one window to other window
10	manage()	1. Window 2. Cookies
11	navigate()	1. Enter the URL 2. Navigate to previous page 3. Navigate to next page 4. Refresh current web page
12	close()	To close the current/parent browser
13	quit()	To close all the browsers opened by selenium

```
public class Demo
{
    public static void main(String[] args) throws InterruptedException
    {
        //open the browser
        System.setProperty("webdriver.chrome.driver",
        "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();

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

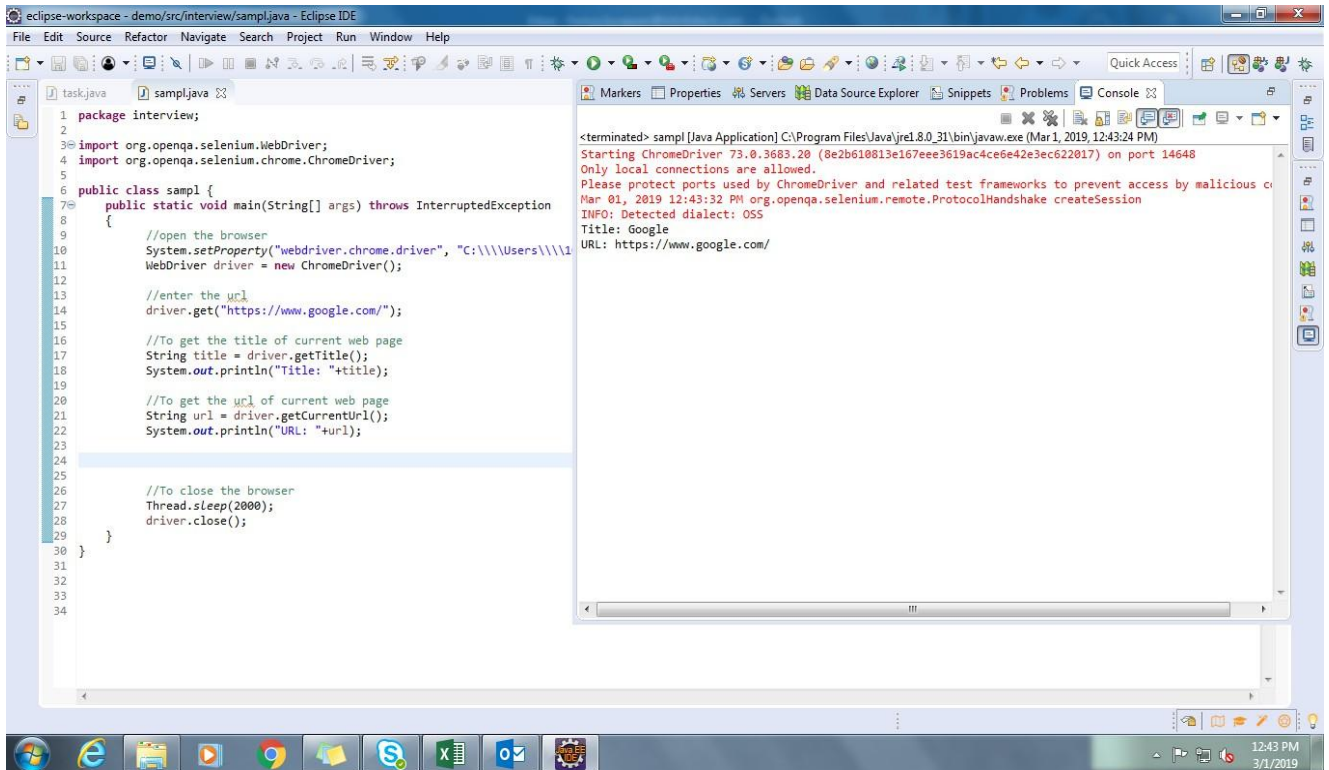
        //To get the title of current web page
        String title = driver.getTitle();
        System.out.println("Title: "+title);

        //To get the url of current web page
        String url = driver.getCurrentUrl();
        System.out.println("URL: "+url);
    }
}
```

```

        //To close the browser
        Thread.sleep(2000);
        driver.close();
    }
}

```



**Write a script for the following:**

- **Open the browser**
- **Delete all cookies**
- **Set size of the window**
- **Set position of the window**
- **Maximize the window**

```

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

```

```

        //To open the browser
        System.setProperty("webdriver.chrome.driver",
"./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        Thread.sleep(2000);

        //To delete cookies
        driver.manage().deleteAllCookies();

        //To set the size of the window
        Dimension d = new Dimension(500, 500);
        driver.manage().window().setSize(d);
        Thread.sleep(2000);

        //To set the position of the window
        Point p = new Point(250, 250);
        driver.manage().window().setPosition(p);
        Thread.sleep(2000);

        //To maximize the window
        driver.manage().window().maximize();
    }
}

```

### **NAVIGATION COMMANDS**

1. Navigate().to()
2. Refresh()
3. Back()
4. Forward()

```

public class Demo
{
    public static void main(String[] args) throws InterruptedException
    {
        //open the browser
        System.setProperty("webdriver.chrome.driver",
"./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();

        //To maximize the window
        driver.manage().window().maximize();

        //To delete the cookies
    }
}

```

```

        driver.manage().deleteAllCookies();

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

        //To enter the url
        driver.navigate().to("https://www.facebook.com/");
        Thread.sleep(1000);

        //To navigate to previous page
        driver.navigate().back();
        Thread.sleep(1000);

        //To navigate to next page
        driver.navigate().forward();
        Thread.sleep(1000);

        //Refresh current web page
        driver.navigate().refresh();
    }
}

public class Demo
{
    public static void main(String[] args) throws InterruptedException
    {
        //open the browser
        System.setProperty("webdriver.chrome.driver",
"./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.naukri.com/");

        Thread.sleep(2000);

        //To close all the browsers
        driver.quit();
    }
}

```

### **Difference Between get() and navigate():**

get	navigate
It will just enter the URL	1. It will enter the URL 2. It will navigate to previous page 3. It will navigate to next page 4. It will refresh the current web page
After entering the URL it will not allow any statements to execute until the page loads completely	After entering the URL it will not wait until the page loads completely

### **WEBELEMENTS METHODS:**

#### **WebElement:**

- Anything which is present on the webpage is called as webelement.  
Ex: text box, link, image, listbox, checkbox etc,
- These Web elements are developed by using HTML.
- HTML stands for HyperText Markup Language.
- The components of HTML are,
  - i) Tags
  - ii) Attributes
  - iii) Text

We can develop the web pages by using the notepad.

Steps to create web pages.

1. Open the notepad
2. Write the html code
3. Save the file with .html
4. Open the file with any web browsers

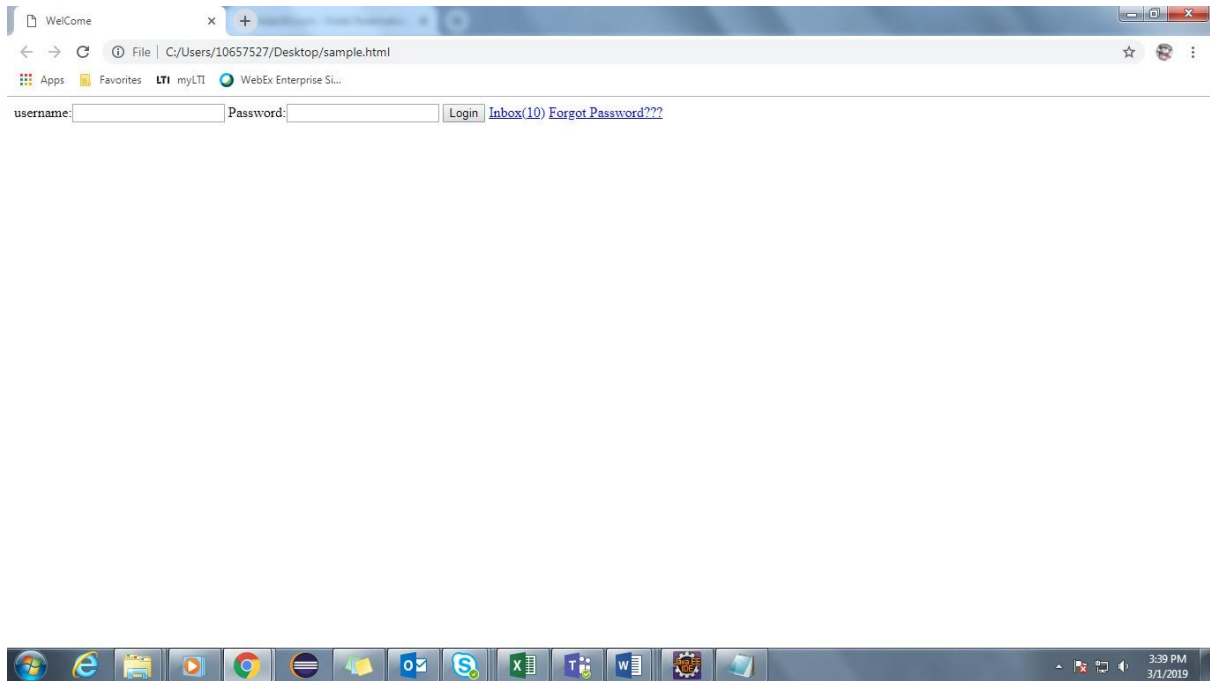
Ex:

```
<html>
```

```
    <head>
```

```
<title>WelCome</title>
</head>
<body>
  username:<input type="text">
  Password:<input type="password">
  <input type="button" value="Login">
  <a href="http://www.gmail.com">Inbox(10)</a>
  <a href="http://www.google.com" id="fp" name="forgot" class="pass">Forgot
Password???
```

WelCome		Header
username:	<input type="text"/>	password: <input type="password"/>
Login: <input type="button" value="Login"/>		
<a href="#">Inbox(10)</a> <a href="#">Forgot Password???</a>		Body



Before performing action on any elements, we have to perform following steps.

1. Inspect the element
2. Identify/locate the element
3. Find the element
4. Perform the action

### **Inspect the element:**

- Fetching the source code of an element is called as inspect the element.
- To inspect the elements, Right click on element → click on Inspect element, which will give source code of that element
- In some applications, for security purpose right click option will be disabled. In such cases
  - Press F12 which will open developers tool
  - Select inspect button (mouse icon available on top left corner)
  - click on the element

### Methods of WebElement Interface:

1	sendKeys()	1. To enter the value in textbox 2. To Handle some keyboard action
2	clear()	To clear the textbox value
3	Click()	To click the particular WebElement(Button)
4	getCssValue()	To get the color/size/font of the particular webelement
5	findElement()	To get single webElements
6	findElements()	To get multiple webElements
7	getText()	To get the text of the particular webelement
8	getAttribute()	To get the text of the particular attribute(id, name, value...etc)
9	getTagName()	To get the tagname of particular webelement
10	getLocation()	To get the X axis and Y axis location of particular webelement
11	getSize()	To get the size of particular webelement(textbox, text...etc)
12	isDisplayed()	To check whether the particular webelement is displaying or not( logo, textbox, text...etc)
13	isEnabled()	To check whether the textbox is enabled to enter the text or not
14	isSelected()	To check whether the radiobutton/dropdown is selected or not
15	submit()	To click on an element only if the type of the element is submit Ex: <input type="submit" id="s" value="Submit">

### LOCATORS:

- Static methods which are used to identify the elements which are present the webpage.
- All these locators are present in a class called **By** which is an Abstract class.
- There are 8 types of locators and all the locators takes argument of type string. They are,
  1. Id(String)
  2. name(String)
  3. className(String)
  4. tagName(String)
  5. linkText(String)
  6. partialLinkText(String)
  7. cssSelector(String)
  8. xpath(String)

**Note:**

- Id, name, className are available as attributes of an element.
- In order to Handle the single element we use findElement().
- Return type of findElement() is WebElement.
- In findElement(), if the specified locator is not matching with any element it will throw NoSuchElementException
- In findElement(), if the specified locator is matching with multiple element it will return the address of 1<sup>st</sup> matching element
- If the specified element is link, then we can identify that element by using linkText
- If the specified element is link and if it partially dynamic, then we can identify that element by using partialLinkText

**cssSelector**

- If we can not identify the elements by using any of the above locators, then we can identify that element by using cssSelector.
- Syntax:
  - tagName[attributeName='attributeValue']  
ex: input[type='password']
- In order to verify cssSelector expression in firefox browser, click on TX→select queryselectorAll option→ specify the expression in expression field and click on enter

**Note:**

- In cssSelector,
  - Id can be represented by using #,  
Ex:- input#email
  - Class can be represented by using .  
Ex:- tagName.classValue

Ex,

**Adactin hotel login using id/name:**

**public class** Login {

```
    public static void main(String[] args) throws Throwable {  
  
        System.setProperty("webdriver.chrome.driver",  
                            "C:\\Users\\10657527\\Downloads\\chromedriver_win32  
(1)\\chromedriver.exe");
```

```

WebDriver driver = new ChromeDriver();

driver.get("https://adactin.com/HotelApp/index.php");

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

WebElement x = driver.findElement(By.id("username"));

x.sendKeys("vengatram");

WebElement x1 = driver.findElement(By.name("password"));

x1.sendKeys("vengat@123445");

WebElement x2 = driver.findElement(By.id("login"));

x2.click();

    }
}

```

### LinkText:

<a href="ForgotPassword.php">Forgot Password?</a>

Text: >Forgot Password?<

linkText: Text which is available in **a** tag called linktext.

```

public class Login {

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

        System.setProperty("webdriver.chrome.driver",
                           "C:\\Users\\10657527\\Downloads\\chromedriver_win32
(1)\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();

        driver.get("https://adactin.com/HotelApp/index.php");

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

```

```

        WebElement x2 = driver.findElement(By.linkText("Forgot Password?"));

        x2.click();

    }
}

```

### PartialLinkText:

```

package com.lnt.test;

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

public class Login {

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

        System.setProperty("webdriver.chrome.driver",
                           "C:\\\\Users\\10657527\\Downloads\\chromedriver_win32
(1)\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();

        driver.get("https://adactin.com/HotelApp/index.php");

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

        WebElement x2 = driver.findElement(By.partialLinkText("Forgot"));

        x2.click();

    }
}

```

### ***Xpath:***

- Path of an element present in the webpage.
  - Absolute
  - Relative

### **Sample webpage:**

<div>

A:<input type="text" value="A">

B:<input type="text" value="B">

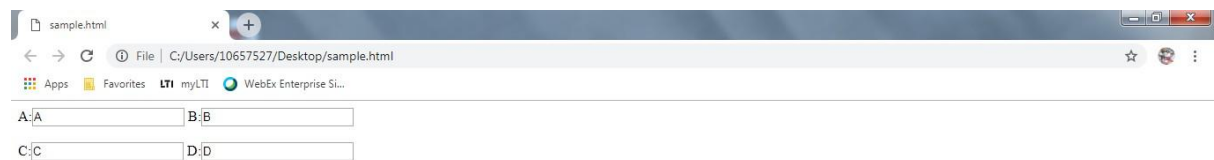
</div><br/>

<div>

C:<input type="text" value="C">

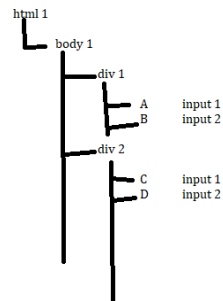
D:<input type="text" value="D">

</div>



In selenium, we represent the webpage in the form of HTML Tree Structure.

**Ex:**



**Absolute:**

- Complete path of an element from root of the webpage(html)
- Represented by using /--> immediate child

**Ex:**

/html/body/div[1]/input[2]

**Relative xpath:**

- Path of any element which is present on web page.
- It is represented by using // which means any child/element

Syntax:

1. //tagName→ all the matching elements
2. //tagName[1]→ all the 1<sup>st</sup> matching elements
3. //tagName[last()]→ all the last matching elements
4. //\*→ all the elements
5. //\*[@attribute='value']
6. //tagName[@attribute='value']

**Ex:**

```
//div[1]/input[2]
//div[1]/input
//div[2]/input
//input[1]
//div[1]/input[2]| //div[2]/input
//input
```

### Xpath by attribute:

- To identify the specified elements, if we use index it may not work properly when we use the index values because whenever the position of an element changes its index value will also changes.
- To overcome the above problem in place of index we can include attributes which is called as xpath by attributes.
- It is applicable for both Absolute and Relative xpath.
- **Syntax:**
  - tagName[@attributeName='attributeValue']
- **Example:**
  - Absolute → /html/body/div/input[@value='B']
  - Relative → //input[@value='B']
- In an xpath we can pass multiple attributes by using or operator.
- **Example:**
  - //input[@value='B' or @value='C']

### Assignment:

- **Derive the xpath expression for the elements which are present in FaceBook login or sign up page.**
  - Email or Phone: //input[@type='email']
  - Day list box: //select[@aria-label='Day']
  - Male: //input[@value='2']

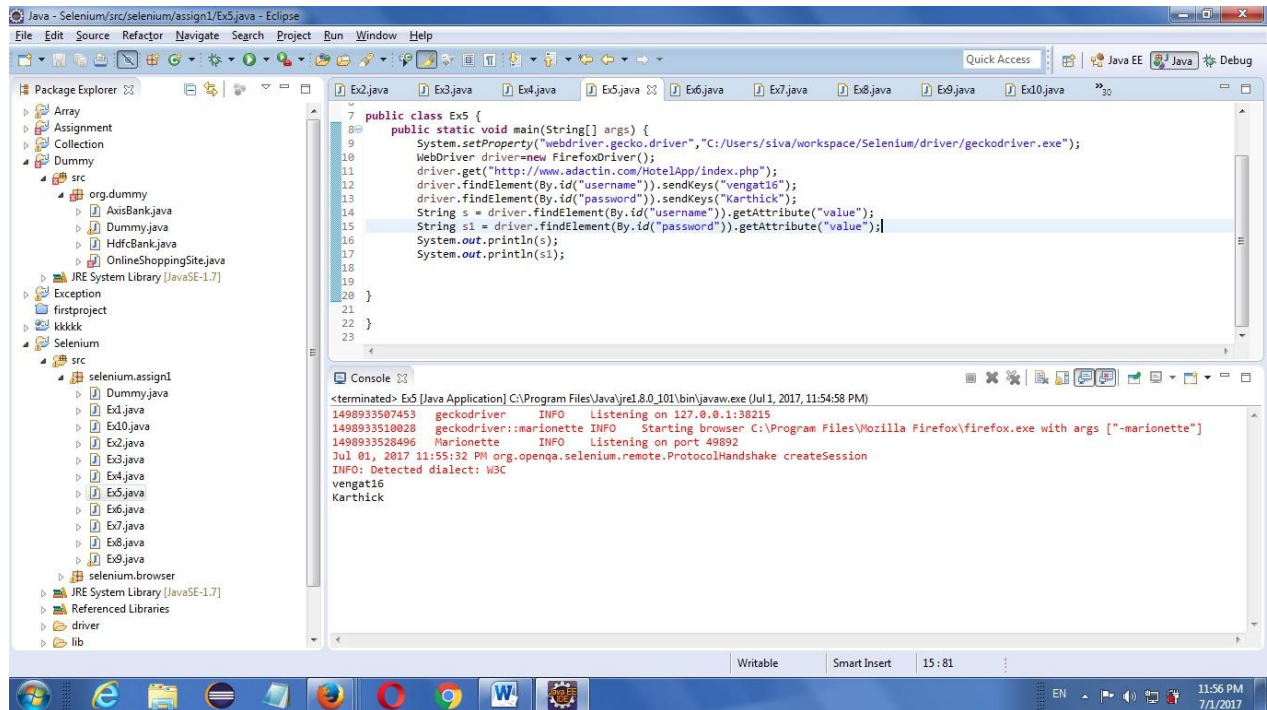
### getAttribute() and getText():

It is a method, used to print the value whatever you gave in the text box

### *Example program:*

```
public class Ex5 {  
    public static void main(String[] args) {  
  
        System.setProperty("webdriver.gecko.driver", "C:/Users/siva/workspace/Selenium/  
driver/geckodriver.exe");  
        WebDriver driver=new FirefoxDriver();  
        driver.get("http://www.adactin.com/HotelApp/index.php");  
        driver.findElement(By.id("username")).sendKeys("vengat16");  
        driver.findElement(By.id("password")).sendKeys("Karthick");  
        String s = driver.findElement(By.id("username")).getAttribute("value");  
        String s1 = driver.findElement(By.id("password")).getAttribute("value");  
        System.out.println(s);  
        System.out.println(s1);  
    }  
}
```

Output:



### NoSuchElementException:

- It is throws when particular id/xpath/class or whatever is not available in DOM structure

### isDisplayed():

It is a method, used to check the particular id/value is available or not

### Example program:

```
public class Ex6 {
    public static void main(String[] args) {

        System.setProperty("webdriver.gecko.driver", "C:/Users/siva/workspace/Selenium/
driver/geckodriver.exe");
        WebDriver driver=new FirefoxDriver();
        driver.get("https://www.facebook.com/");
```

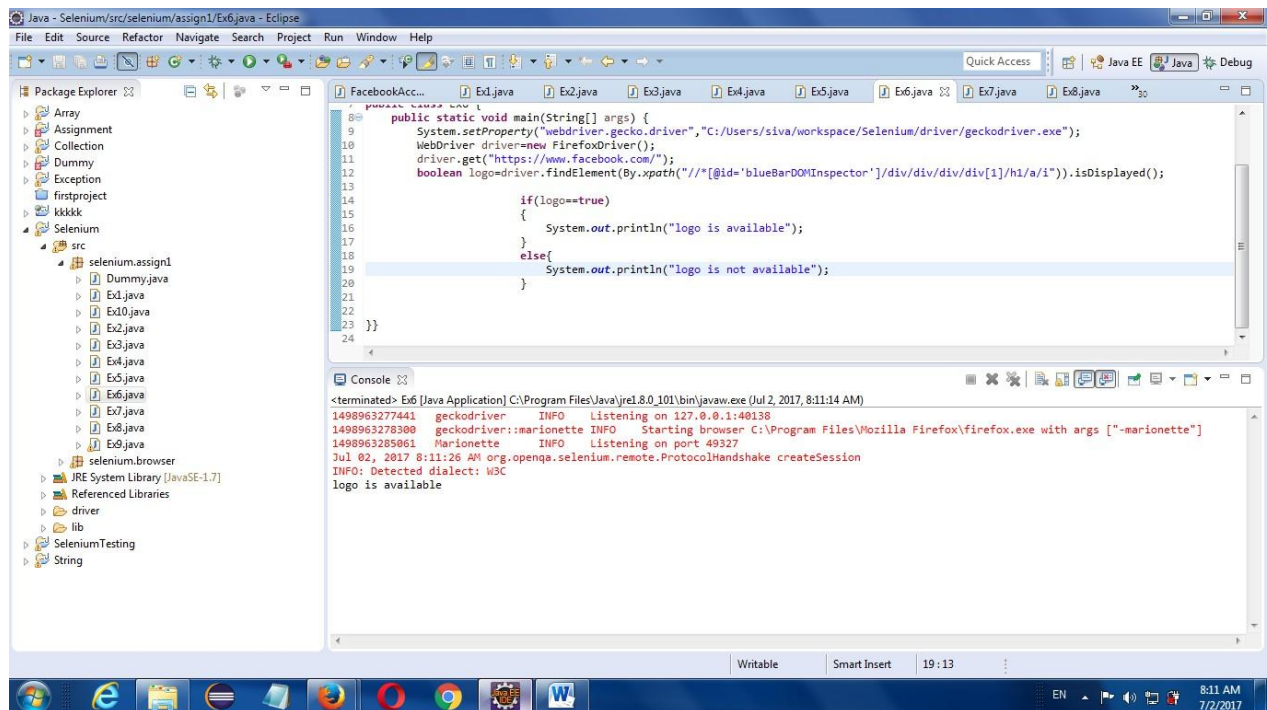
**boolean**

```
logo=driver.findElement(By.xpath("//*[@id='blueBarDOMInspector']/div/div/div/div[1]/h1/a/i")).isDisplayed();
```

```
if(logo==true)
{
    System.out.println("logo is available");
}
else{
    System.out.println("logo is not available");
}
```

```
}}
```

Output:



**isEnabled:**

- It is a method, is used to check particular text box is enable to print or not

**Example program:**

```
public class Dummy {
```

```

public static void main(String[] args) {
    System.setProperty("webdriver.gecko.driver",

"C:/Users/siva/workspace/Selenium/driver/geckodriver.exe");
    WebDriver driver = new FirefoxDriver();
    driver.get("https://www.facebook.com/");
    boolean logo = driver.findElement(By.xpath("//*[@id='email']"))
        .isEnabled();

    if (logo == true) {
        System.out.println("Text box is enable to print");
    } else {
        System.out.println("not enable");
    }

}
}

```

Output:

```

<terminated> Dummy (1) [Java Application] C:\Program Files\Java\jre1.8.0_101\bin\javaw.exe (Jul 2, 2017, 8:20:59 AM)
14908963862152  geckodriver: INFO Listening on 127.0.0.1:45252
14908963862887  geckodriver::marionette INFO Starting browser C:\Program Files\Mozilla Firefox\firefox.exe with args [\"-marionette\"]
14908963871744  Marionette INFO Listening on port 49469
Jul 02, 2017 8:21:13 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: W3C
Text box is enable to print

```

### isSelected():

- It is a method, used to check the particular radio button is selected or not

### *Example program:*

```
public class Dummy {
    public static void main(String[] args) {
        System.setProperty("webdriver.gecko.driver",
            "C:/Users/siva/workspace/Selenium/driver/geckodriver.exe");
        WebDriver driver = new FirefoxDriver();
        driver.get("https://www.facebook.com/");
        driver.findElement(By.xpath("//*[@id='u_0_g']"))
            .click();
        boolean logo = driver.findElement(By.xpath("//*[@id='u_0_g']"))
            .isSelected();
        if (logo == true) {
            System.out.println("button is selected");
        } else {
            System.out.println("not selected");
        }
    }
}
```

### Xpath by text():

- If the specified element does not contain any attributes and if it contains text then we can identify that element by using xpath by text()
- It is applicable for both absolute and relative xpath
- Syntax:
  - tagName[text()='textValue']
- Example:
  - //td[text()='Java']
- text() can be represented by using dot(.)
- Example:
  - //td[.='Java']
- Attribute values and the text values are case and space sensitive.
- Example:
  - //div[text()='Login ']

### **Xpath by contains():**

- It is used to handle the partial dynamic elements
- It is applicable for both absolute and relative xpath.
- **Syntax 1:** if text value is partially dynamic,
  - **tagName[contains(text(),'textValue')]**
- **Example:**
  - **//noabr[contains(text(),'actiTIME')]**
- **Syntax 2:** if attribute value is partially dynamic,
  - **tagName[contains(@attributeName,'attributeValue')]**
- **Example:**
  - **//img[contains(@src,'/img/default/login/timer.png?hash')]**

### **Handling special characters:**

- While developing the application developers will be using some special characters like &amp;, &nbsp; etc.
- If the element contains any special characters then we can identify that element by using xpath by contains().
- If any value contains & symbol then it is the special character.
- **Example:** Derive the xpath to identify Forgotten Password? Link present on facebook login or sign up page.
  - **//a[contains(@href,'https://www.facebook.com/recover/initiate?lwv')]**

### **Traversing:**

- Navigating from one element to another element using xpath is called traversing.
- To navigate from one element to another element xpath uses axis.
- The different types of axis are,
  1. child
  2. parent
  3. descendant
  4. ancestor
  5. following-sibling
  6. preceding-sibling
- Syntax:

**/axis::tagName**

**Sample web page:**

```
<select>
  <option value="j">Jan</option>
  <option value="f">Feb</option>
  <option value="m">Mar</option>
  <option value="a">Apr</option>
  <option value="m">May</option>
</select>
```

**Child:**

- Navigate from one element to it's immediate child.  
Ex: `//select//child::option[1]`

**Parent:**

- Navigate from one element to it's immediate parent.  
Ex: `//select[@option='j']/parent::select`

**Descendant:**

- Navigate from one element to any of it's child present on the webpage.  
Ex: `/html/descendant::option[1]`

**Ancestor:**

- Navigate from one element to any of it's parent present on the webpage.  
Ex: `//select[@option='j']/ancestor::html`

**Following-sibling:**

- The elements which are present below the specified element, under same parent are called as following-sibling.

Ex:

```
//select[@option='m']/following-sibling::option → A,M
//select[@option='m']/following-sibling::option[1] → A
//select[@option='m']/following-sibling::option[2] → M
```

**Preceding-sibling:**

- The elements which are present above the specified element, under same parent are called as preceding-sibling.

Ex:

```
//select[@option='m']/preceding-sibling::option→ J,F
//select[@option='m']/preceding-sibling::option[1]→ F
//select[@option='m']/preceding-sibling::option[2]→ J
```

### **Radio button:**

- We can able to select only one at a time

### **Example program:**

```
public class Dummy {
    public static void main(String[] args) {
        System.setProperty("webdriver.gecko.driver",
            "C:/Users/siva/workspace/Selenium/driver/geckodriver.exe");
        WebDriver driver = new FirefoxDriver();
        driver.get("https://www.facebook.com/");
        driver.findElement(By.xpath("//*[@id='u_0_g']"))
            .click();
    }
}
```

### **Xpath:**

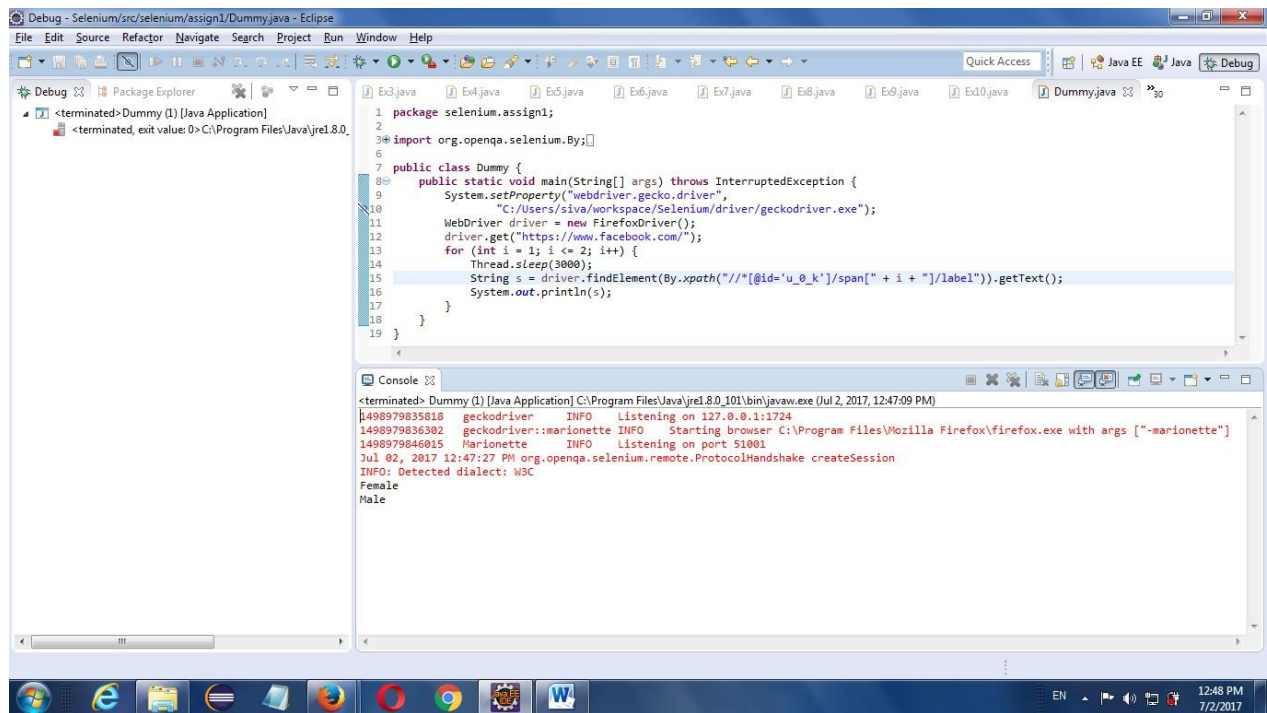
- Structure or combination of absolute path and relative path

### **To print both radio button:**

### **Example program:**

```
public class Dummy {
    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.gecko.driver",
            "C:/Users/siva/workspace/Selenium/driver/geckodriver.exe");
        WebDriver driver = new FirefoxDriver();
        driver.get("https://www.facebook.com/");
        for (int i = 1; i <= 2; i++) {
            Thread.sleep(3000);
            String s = driver.findElement(
                By.xpath("//*[@id='u_0_k']/span[" + i + "]/label"))
                .getText();
            System.out.println(s);
        }
    }
}
```

Output:



KeyBoard Actions using Sendkeys:

```
public class sampl {  
    public static void main(String[] args) throws InterruptedException  
    {  
        //open the browser  
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\\\10655967\\\\eclipse-  
workspace\\\\demo\\\\driver\\\\chromedriver.exe");  
        WebDriver driver = new ChromeDriver();  
  
        driver.get("https://adactin.com/HotelApp/index.php");  
  
        driver.manage().window().maximize();  
  
        WebElement x = driver.findElement(By.id("username"));  
  
        x.sendKeys("vengatram");  
  
        WebElement x1 = driver.findElement(By.name("password"));  
  
        x.sendKeys(Keys.CONTROL,"ac");  
        x1.sendKeys(Keys.CONTROL,"v");  
    }  
}
```

### getLocation() and getSize():

#### Example:

```
public class Demo
{

    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://demo.actitime.com/login.do");

        WebElement un = driver.findElement(By.id("username"));

        //To get the size of an element; height & width
        Dimension s = un.getSize();
        int h = s.getHeight();
        int w = s.getWidth();

        System.out.println("Height: "+h);
        System.out.println("Width: "+w);

        //To get location of an element; x-axis & y-axis
        Point l = un.getLocation();
        int x = l.getX();
        int y = l.getY();

        System.out.println("x-axis: "+x);
        System.out.println("y-axis: "+y);

        Thread.sleep(1000);
        driver.close();
    }
}
```

### getCssValue():

It is used to get the css property (font, color, size) of a web element.

```
public class Login {  
  
    public static void main(String[] args) throws Throwable {  
  
        System.setProperty("webdriver.chrome.driver",  
                            "C:\\Users\\10657527\\Downloads\\chromedriver_win32  
(1)\\chromedriver.exe");  
        WebDriver driver = new ChromeDriver();  
  
        driver.get("https://adactin.com/HotelApp/index.php");  
  
        driver.manage().window().maximize();  
  
        WebElement x =  
driver.findElement(By.xpath("//td[@class='build_title']"));  
  
        String x1 = x.getCssValue("font-size");  
  
        System.out.println(x1);  
  
        String x2 = x.getCssValue("color");  
  
        System.out.println(x2);  
  
        String x3 = x.getCssValue("font-weight");  
  
        System.out.println(x3);  
  
        String x4 = x.getCssValue("font-family");  
  
        System.out.println(x4);  
        String x5 = x.getCssValue("background");  
  
        System.out.println(x5);  
  
    }  
}
```

