When we get the driver object, the below are the methods that we can perform operation on a driver. In IDE like eclipse, when we enter driver. and click on space bar, it will show all the below methods.All these come under WebElement

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

Other methods:

14. getAttribute()
15. getLocation()
16. click()
17. clear()

1.Method Name :- get()

Syntax: get(url)
Example: driver.get();
Purpose: It will load a new web page in the current browser window. This is done using an http get operation, and the method will block until the load is complete.
Parameters: URL - The URL to load and it should be a fully qualified URL

2.Method Name: getCurrentUrl()

Syntax: getCurrentUrl()
Example: driver.getCurrentUrl();
Purpose: Gets a string representing the current URL that the browser is opened.
Returns: The URL of the page currently loaded in the browser

3.Method Name: getTitle()

Syntax: getTitle()
Example: driver.getTitle();
Purpose: Gets the title of the current web page.
Returns: The title of the current page, with leading and trailing white space stripped, or null if one is not already set

4.Method Name: findElements()

Syntax: findElements(By by)
Example: driver.findElements(By.xpath("//");

Purpose: Find all elements within the current page using the given mechanism.
Parameters: By - The locating mechanism to use
Returns: A list of all WebElements, or an empty list if nothing matches

5.Method Name: findElement()

Syntax: WebElement findElement(By by)
Example: driver.findElements(By.xpath("//");
Purpose: Find the first WebElement using the given method.
Parameters: By - The locating mechanism
Returns: The first matching element on the current page Throws: NoSuchElementException - it will return exception if no matching elements are found

6.Method Name: getPageSource()

Syntax: getPageSource()
Example: driver.getPageSource();
Purpose: Get the source of the currently loaded page. If the page has been modified after loading (for example, by Javascript) there is no guarantee that the returned text is that of the modified page.
Returns: The source of the current page

7.Method Name: close()

Syntax: void close()
Example: driver.close();
Purpose: Close the current window, if there are multiple windows, it will close the current window which is active and quits the browser if it's the last window opened currently.

8.Method Name: quit()

Syntax: void quit()
Example: driver.quit();
Purpose: Quits this driver instance, closing every associated window which is opened.

9.Method Name: getWindowHandles()

Syntax: Set getWindowHandles()
Example: driver.getWindowHandles();
Purpose: Return a set of window handles which can be used to iterate over all the open windows of this Webdriver instance by passing them to switchTo().WebDriver.Options.window()
Returns: A set of window handles which can be used to iterate over all the open windows.

10.Method Name: getWindowHandle()

Syntax: String getWindowHandle()
Example: driver.getWindowHandle();
Parameter: Return an opaque handle to this window that uniquely identifies it within this driver instance. This can be used to switch to this window at a later date switchTo

Returns: A Target Locator which can be used to switch or select a frame or window

WebDriver.TargetLocator switchTo() The next future commands will be performed to a different frame or window.

11.Method Name: navigate()

Syntax: WebDriver.Navigation navigate()
Example: driver.navigate.to("");
Purpose: An abstraction allowing the driver to access the browser's history and to navigate to a given URL.
Returns: A WebDriver.Navigation that allows the selection of what to do next
Click here to know more on Navigation methods

12.Method Name: manage()

Syntax: WebDriver.Options manage()

Purpose: Gets the Option interface

Returns: An option interface

13.switchTo()

Syntax: Webdriver.TargetLocator()switchto()

Purpose: Send future commands to a different frame or window.

Returns:A TargetLocator which can be used to select a frame or window

3. Xpath by attribute:

Using relative xpath we can reduce the length of expression but it may match with multiple elements even after using index.In order to identify the element uniquely we can use attribute in the xpath expression using following syntax:

//tag[@AttributeName='AttributeValue']

Example: //input[@placeholder='Username']

IQ 1]Can we use multiple attribute in the xpath expression

Ans: Yes

Example:1) //input[@placeholder='Username' AND@name='username']

//input[@placeholder='Username' OR name='username']

2)//input[@value='Log In']

3)//input[@id='next']

4)//input[@value='Next']

4.Xpath by text():

If attribute is not present (or) attribute is matching with multiple elements in such cases we can use xpath by text,which has following syntax:

//tag[text()='textvalue']

Example: 1)//div[text()='Login']

2)//div[text()='USERS']

NOTE: In the same xpath expression we can specify both attribute and text()

HANDLING NON-BREAKABLE SPACE:

1)developer can give the space in the value using spacebar or using Keyword [non breakable space]

2)when we inspect the element in the browser we cannot make out whether   is used or not

3)If value has   then xpath will not identify such elements

Example:

```
<html>
<body>
   <button type="submit>&nbspOK </button>
</body>
</html>
```

IN FIREBUG:

*<button type="submit"> OK </button> → not identify the element

*//button[text()=' OK '] → not identify the element

To handle non breakable space we should use contains() which has following syntax:

//tag[contains(text(),'textvalue')]

//tag[contains(@AttributeName,'AttributeValue')]

Example:

1) attribute example
   button[contains(text(),'OK')]

Example: //input[contains(@value,'Create Type of Work')]login to actitime>settings>types of work>create types of work> create types of work>

2)text example

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

NOTE:

We use contains() if value has non-breakable space or if value is keep changing.

Example://span[contains(text(),'Inbox')]

XPATH TRAVERSING:

We can derive a xpath expression which can navigate from one element to another element which is called as traversing.It supports 2 types of traversing;

1) Forward Traversing

2) Backward Traversing

EXAMPLE:

```
<html>
    <body>
        <table border="1">
        <tbody>
            <tr>
<td>1</td>
                <td>Unix</td>
                <td>300</td>
            </tr>
            <tr>
                <td>2</td>
                <td>Java</td>
                <td>400</td>
            </tr>
        </tbody>
```

</table>

</body>

</html>

1) FORWARD TRAVERSING:

Navigating from parent element to any of its child element is called as forward traversing.

Example:

Navigating from;

table to Unix;

```
//table/tbody/tr[1]/td[2]
```

Table to java;

```
//table/tbody/tr[2]/td[2]
```

2) BACKWARD TRAVERSING:

Navigating from child element to any of its parent element is called as backward traversing

Example:

Navigating from;

Unix to table;

```
//td[text()='Unix']/../../..
```

Java to table;

```
//td[text()='Java']/../../..
```

5) INDEPENDENT DEPENDENT XPATH:

If the element is completely changing (or) it duplicate with some other elements we can use independent dependent concept of xpath to identify it.

EXAMPLE:

Derive xpath to identify cost of UNIX

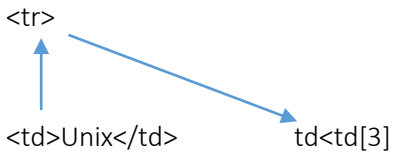
Step#1: Inspect the independent element and Note down its source code.

Step#2: Place the mouse pointer on the source code of independent element and move the mouse pointer in upward direction step by step till it highlights both independent and dependent element. This will be the common parent ,add it to the html tree

Step#3:

Use arrow key to navigate till dependent element, add its path to the html tree.

Step#4: Using the tree derive the xpath expression which navigates from independent element to common parent and then to dependent element



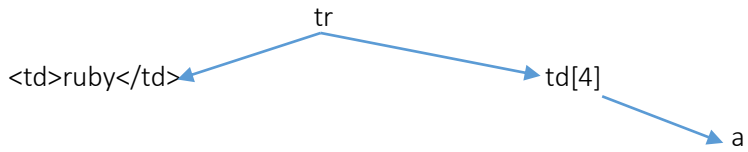
//td[text()='Unix']/../td[3]

NOTE:

While doing forward traversing we should navigate till end i.e, expand all the + sign.

IQ2)Derive the xpath to match with download link of ruby present in download page of selenium

Ans:



//td[text()='Ruby']/../td[4]/a

Note :

The above xpath identifies the link which is present in 4th column only.To identify the download link even if column is keep changing we can use below xpath;

//td[text()='Ruby']/..//a[text()='Download']

Example in actiTime: //a[text()='Vidya']/../..//a[text()='set by default']

Assignment:

1) **Derive a xpath to identify the price of Mi 4i(Blue 16GB) present in flipkart**
   //span[text()=' Rs. 11,998']
2) **Derive a xpath to identify add to compare checkbox of Redmi2 Prime(Grey 16 GB) in flipkart**
   //input[@id='MOBE9T7GTHERTDAC']

3) Derive a xpath to identify phone number of Mumbai present in isrtc.com

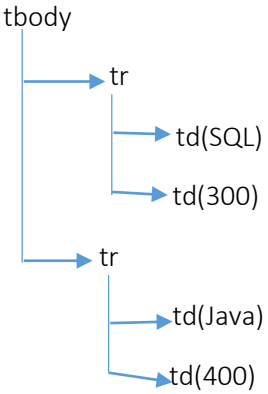
---

4) Identify help icon(?) present in actitime application;

(//div[@class='popup menu arrow'])[3]

Note:

Sometimes we may not be able to identify the element even after using all types of xpath which is previously discussed.In such case we use GROUP INDEX(GI)



*//td[1]->SQL

*(//td)[1]->SQL

*(//td)[3]->Java

*(//td[1])[2]->java

IQ3)What is the difference between //a, //a[1], (//a)[1]

1.all the links

2.all the first links

3.only first link

IQ4)Derive the xpath which matches with last checkbox.

Answer: (//input[@type='checkbox'])[last()]

IQ5)Write a xpath to select first and last checkbox

(//input[@type='checkbox'])[1]|(//input[@type='checkbox'])[last()]

|||ly

*(//input[@type='checkbox']

*(//input[@type='checkbox'])[1]

*(//input[@type='checkbox'])[6]

IMPORTANT LOCATORS:

1) Id

2) name

3) linkText

4) xpath

Note:

In very few situations xpath written using a browser(Firefox)may not work in some other browser,in such cases we can use CssSelector.

CONVERTING XPATH TO CSSSELECTOR:

XPATH	CSSSELECTOR
1) //button[@type='submit']	button[type='submit']
2) //input[@id='UN']	input#UN
3) //input[@class='c1']	input.c1
4) //a	a
5) //tr/td	tr>td
6) //table//a	table a
7) //td/..	Backward traversing is not supported in selenium
8) //td[text()='Java']	text() is not supported in selenium

IQ6) Can we use independent dependent concept in selenium

Ans; No,because backward traversing is not supported in selenium

Note: 1) get() will enter the Url and wait till the page is completely loaded.Waiting time of get() is infinite.

2)findElement() will search for specified element in the current page,if it is present it will returns the address of the element.If it is not there it will trow NoSuchElementFoundException IMMEDIATELY.

SYNCHRONIZATION:

Process of matching Selenium speed with application is called as synchronization .In order to synchronize the script we can use sleep() of Thread class as shown below;

```
Try
{
Thread.sleep(30000);
}
catch(InterruptedException e)
{
}
```

IMPLICITLY WAIT:

If we use sleep(),It will drastically increase maintenance consumes lot of time and space.Instead of this we can use implicitly wait statement of selenium.

Example: driver.manage().timeouts.implicitlyWait(30,TimeUnit.SECONDS);

In the above example 30SECONDS is used by all the findElement method and each findElement() waits upto 30 SECONDS.

After every half second it will search for element in page till the time out or till the element is located which ever comes earlier,This is called POLLING PERIOD.This is specified in a class called FluentWait.

The implicitelyWait() takes 2 arguments i.e,duration(long type) and Timeunit() such as DAYS,HOURS,SECONDS,MINUTES,MILLISECONDS,MICROSECONDS and NANOSECONDS

If element is not located even after the time out then FindElement() will throw NoSuchElementException

---

EXPLICIT WAIT:

*If the method is other than findElement then to synchronize it we can use explicit wait.

*webDriverWait itself is called as Explicitwait because we specify waiting condition Explicitly

These conditions are available in ExpectedConditions class,these are also called as Predicates.

*If specified condition is not specified even after the specified duration then explicitWait will throw TimeoutException(Selenium unchecked exception)

Code:

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class Demo010216 {

	public static void main(String[] args)
	{
		//script to login to actitime
		//open the browser
		WebDriver driver=new FirefoxDriver();
		//enter the url
		driver.get("http://localhost/login.do");
		//enter the username
		driver.findElement(By.id("username")).sendKeys("admin");
		//enter the password
		driver.findElement(By.name("pwd")).sendKeys("manager");
		//click on login button
		driver.findElement(By.xpath("//div[text()='Login ']")).click();
		//wait till logout link is visible within 30Sec
		WebDriverWait wait=new WebDriverWait(driver,30);
		wait.until(ExpectedConditions.visibilityOfAllElementsLocatedBy(By.id("logoutLink")));
		//get the title of home page and print it
		String title=driver.getTitle();
		System.out.println(title);


	}

}
```

IQ7) What are the differences between ImplicitWait and ExplicitWait

ImplicitWait	ExplicitWait
1) We do not specify any condition	We should specify the condition
2) We can only handle findElement() and findElements()	We can handle any method
3) After the timeout we get NoSuchElementException	After the timeout we get TimeOutException
4) Time unit can be DAYS,HOURS,SECONDS etc	It can be only SECONDS

IQ8) Write a script to login and logout from the application without specifying any type of waiting duration(period)

Answer:
```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class Demo010216 {

	public static void main(String[] args)
	{
		//script to login to actitime
		//open the browser
		WebDriver driver=new FirefoxDriver();
		//enter the url
		driver.get("http://localhost/login.do");
		//enter the username
		driver.findElement(By.id("username")).sendKeys("admin");
		//enter the password
		driver.findElement(By.name("pwd")).sendKeys("manager");
		//click on login button
		driver.findElement(By.xpath("//div[text()='Login ']")).click();
		while(true)
		{
			try
			{
				driver.findElement(By.id("logoutLink")).click();
				break;
			}
			catch(Exception e)
			{
			}
			}
		}

	}
```

IQ9) How do you click on a button without using click()

Answer: By pressing enter key

```
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Demo01216 {

	public static void main(String[] args)
	{
		WebDriver driver = new FirefoxDriver();
		driver.get("https://demo.vtiger.com/");
		driver.findElement(By.xpath("//button[text()='Sign in']")).sendKeys(Keys.ENTER);
		driver.findElement(By.xpath("//button[text()='Sign in']")).submit();
	}}
```

*We can also use submit() if button type=submit

```
	<button type="submit">sign in</button>
```

*We can also use java script to click on a button

IQ10) How do you change the value present in the text box

Ans: Using clear() and sendkeys()

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

Un.clear();

Un.sendkeys("bhanu");

IQ11) How do you remove the value present in the text box without using clear()

Ans: un.sendkeys(Keys.CONTROL+"a"+Keys.DELETE);

IQ12)Write a script to copy paste the value present in one text box into another text box

Ans: un.sendkeys(Keys.CONTROL+"ac"+Keys. CONTROL+"v");

IQ13)Write a script to print value present in the textbox.

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

String v=un.getAttribute("value");

System.out.println(v)

Limitation1:

In Selenium we cannot store the password in encrypted format.

IQ14)How do you retrieve tooltiptext of an element

Ans: using getAttribute("title")

```
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Demo01216 {

	public static void main(String[] args)
	{
		WebDriver driver = new FirefoxDriver();
		driver.get("https://demo.vtiger.com/");
		WebElement chkBox=driver.findElement(By.name("remeber"));
		String tt = chkBox.getAttribute("title");
		System.out.println("tt");
	}}
```

IQ15)Write a script to find the phone number of mubai in isrtc.com

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Demo01216 {

	public static void main(String[] args)
	{
		WebDriver driver = new FirefoxDriver();
		driver.get("https://demo.vtiger.com/");
		WebElement chkBox=driver.findElement(By.name("remeber"));
		String tt = chkBox.getAttribute("title");
		System.out.println("tt");


	}

}
```

IQ16)What is the difference between getAttribute() & getText()

Ans: getAttribute() get the value of the specified attribute where as getText() is used to get the text of the specified element

---

IQ17)Write a script to print x and y co-ordinates of an element.

```
import org.openqa.selenium.By;
import org.openqa.selenium.Point;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Demo622016 {

	public static void main(String[] args)
	{
		WebDriver driver = new FirefoxDriver();
		driver.get("http://localhost/login.do");
		Point p=driver.findElement(By.id("username")).getLocation();
		int x=p.getX();
		int y=p.getY();
		System.out.println(x);
		System.out.println(y);
		driver.close();

	}

}
```

IQ18)Write a code to print height and width of the code

```
import org.openqa.selenium.By;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.Point;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Demo622016 {

	public static void main(String[] args)
	{
		WebDriver driver = new FirefoxDriver();
		driver.get("http://www.gmail.com");
		Dimension d=driver.findElement(By.id("next")).getSize();
		int h=d.getHeight();
		int w=d.getWidth();
		System.out.println(h);
		System.out.println(w);
		driver.close();

	}

}
```

IQ19)Write a code to print font size and color of the username text box in actiTime application

```
import org.openqa.selenium.By;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.Point;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Demo622016 {

		public static void main(String[] args) {
		WebDriver driver = new FirefoxDriver();
		driver.get("http://localhost/login.do");
		WebElement un = driver.findElement(By.id("username"));
		String fs = un.getCssValue("font-size");
		System.out.println(fs);
		String fc = un.getCssValue("color");
		System.out.println(fc);
		String ff = un.getCssValue("font-family");

		System.out.println(ff);

		driver.close();

	}

}
```

OUTPUT:

14px

Rgba(0,0,0,1)

MS Shell Dlg\32

IQ20)Write a script to verify that login button is enabled

```
import org.openqa.selenium.By;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.Point;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Demo622016 {

	public static void main(String[] args) {
		WebDriver driver = new FirefoxDriver();
		driver.get("http://localhost/login.do");
		WebElement button = driver.findElement(By.id("loginButton"));
		if(button.isEnabled())
		{
			System.out.println("Login Button is Enabled");
		}
		else
		{
			System.out.println("Login Button is not Enabled");
		}

		driver.close();

	}

}
```

IQ21)Write a script to verify that Next button in the gmail login page is visble(hint:using isDisplayed())

IQ22)Write a script to verify whether keep me logged in checkbox present in facebook login page is selected or not?(hint:isSelected())

Note:

1.isSelected() can be also used on radio button

2.Importent mathods of WebElement Interface

1. clear()
2. **click()***
3. **getAttribute()***
4. getCssValue()
5. getLocation()
6. getSize()
7. getTagName()

8. **getText()***
9. isDisplayed()
10. isEnabled()
11. isSelected()
12. **sendKeys()***
13. submit()

---

last page:

InvalidStateElementException(Unchecked Selenium Exception)

EXECUTING JAVA SCRIPT

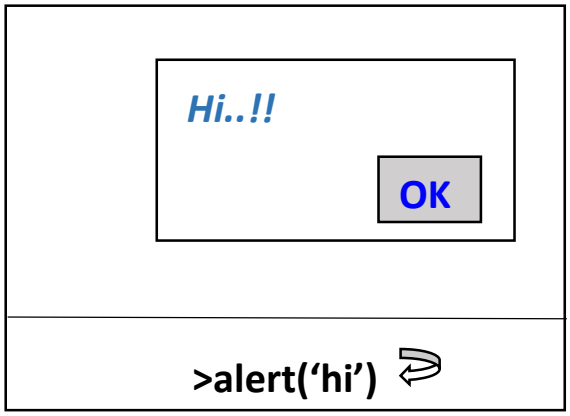
Sometimes Selenium methods such as **click(),sendKeys() etc**.,may not work as an alternative option or work around we can use java script.

EXECUTING JAVA SCRIPT MANUALLY:

Step#1: Open FireFox browser and open the required web page.press **F12** which opens firebug window.

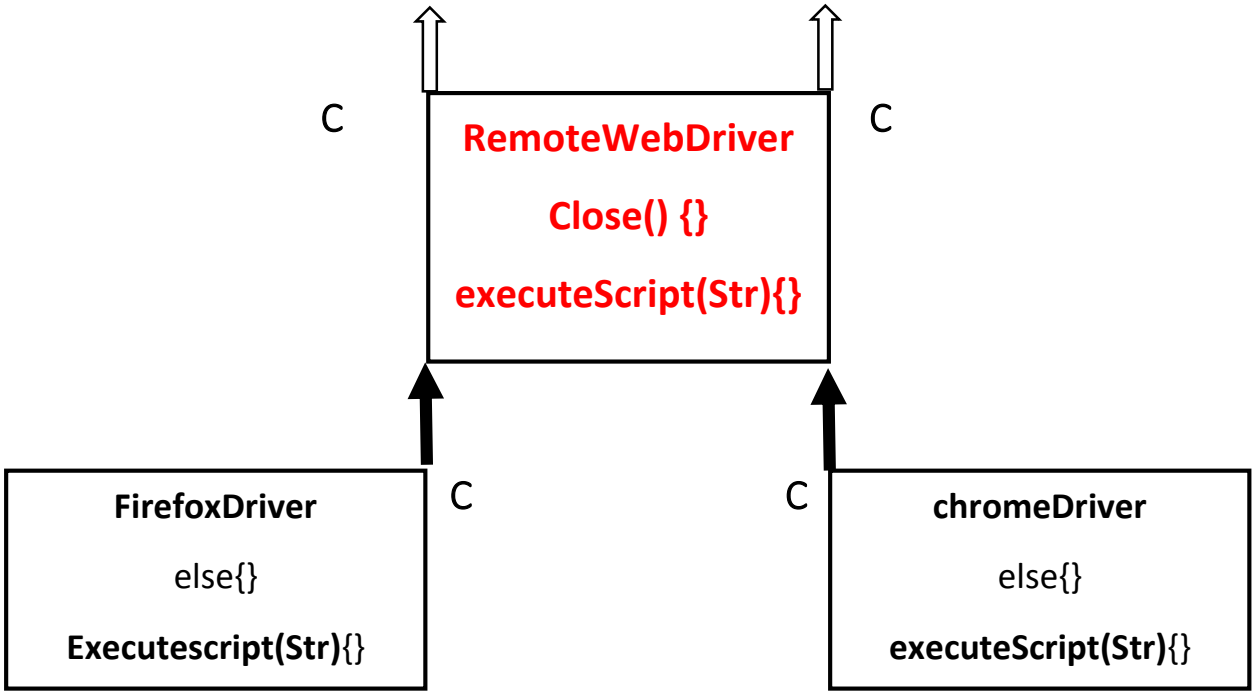
Step#2: Click on console tab

Step#3: Type the javascript statement in the text box which is available at rite the java script statement in the text box which is available at the bottom of the firebug window and press enter



EXECUTING JAVA SCRIPT PROGRAMMATICALLY:

To run the java scripts programmatically in selenium we should use **excecuteScript()** of **JavascriptExcecutor**.Generally the object of the browser will be upcasted to WebDriver interface hence excecuteScript()will be hidden.In order to access this method either we should downcast it to RemoteWebDriver class or we should type cast it to JavascriptExcecuter interface



IQ23)Write a code to click() on the button using java script

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.RemoteWebDriver;

public class Demo062 {

        public static void main(String[] args) {
        Interface  WebDriver driver = new FirefoxDriver();
                driver.get("http://localhost/login.do");
                RemoteWebDriver r=(RemoteWebDriver) driver;
                r.executeScript("document.getElementById('loginButton').click()");

        }}
```

Browser
Constructor
Up-Casting
Down casting

IQ24) Write a script to enter the text into text box without using sendKeys()

```
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.remote.RemoteWebDriver;

public class Demo062 {

                public static void main(String[] args) {
                 WebDriver driver = new FirefoxDriver();
```

```
                driver.get("https://demo.vtiger.com/");
                String c = "document.getElementById('username').value='abc'";
                JavascriptExecutor j = (JavascriptExecutor) driver;
                j.executeScript(c);
        }

}
```

IQ25**) How do you enter the text if the text box is disabled using JavaScript

```
<html>

<body>

UN:<input id=”username” type=”text” disabled>

</body>

</html>
```

```
                WebDriver driver = new FirefoxDriver();
                driver.get("file:///c:/demo.html");
                String c = "document.getElementById('username').value='bhanu'";
                JavascriptExecutor j = (JavascriptExecutor) driver;
                j.executeScript(c);
```

IQ26**) Write a script to scroll to the bottom of the page

```
                WebDriver driver = new FirefoxDriver();
                driver.get("http://news.google.com/");
                String c = "window.scrollTo(0,document.body.scrollHeight)";
                JavascriptExecutor j = (JavascriptExecutor) driver;
                j.executeScript(c);
```

|||ly String c = "window.scrollTo(0,document.body.scrollHeight/2)"; → for half

```
                String c = "window.scrollTo(document.body.scrollWidth,0)";
```
→for complete right

IQ27)Write a script to scroll to the element

(**hint:** find the x and y co-ordinates of the element using getLocation(),pass them as argument for javaScript)

Driver.findElement(By.id(“”)).getLocation()

Window.scrollTo(x,y)

IQ28)Write a script to take the photo of a application

```
import java.io.File;
import java.io.IOException;

import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
```

```
import org.openqa.selenium.support.events.EventFiringWebDriver;


public class Demo062
{

	public static void main(String[] args) throws IOException
	{
		WebDriver driver = new FirefoxDriver();
		driver.get("http://localhost/license.jsp");
		EventFiringWebDriver e=new EventFiringWebDriver(driver);
		File srcFile = e.getScreenshotAs(OutputType.FILE);
		File destFile = new File("c:/abc.png");
		FileUtils.copyFile(srcFile, destFile);
		driver.close();
	}

}
```

NOTE:

Limitation:

*using selenium we can take screen shot in PNG(portable network graphics) format only.we can not take the screenshot of popups,we cannot take screenshot of specific area in the page,we can not take the screenshot of multiple browser or desktop

* If the page is very lengthy it will automatically takes the screenshot of complete page.

ENCAPSULATION

"Process of hiding the data and binding with methods is called as encapsulation."

Example:

```
public class A

{

private int i;

public A()

{

i=10;

}

public int getValue()

{

return i;
```

}

}

Class B

{

Psvm()

{

A a1=new A();

S.o.p(a1.getValue());

}

}

For any given variable in java we should perform following steps;

1.Decleration

2.Initialization

3.Utilization

NOTE:

Usually initialize variables within constructor for encapsulation.

```
public class LoginPage
{
private WebElement unTextBox;
public LoginPage(WebDriver driver)
{
unTextBox=driver.findElement(By.id("username"));
}
public void setUserName()
{
unTextBox.sendKeys("admin");
}
}
```

Example:

```
package qspiders;
```

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;

public class LoginPage {

	private WebElement unTextBox;
	private WebElement pwTextBox;
	private WebElement loginButton;

	public LoginPage(WebDriver driver) {
		unTextBox = driver.findElement(By.id("username"));
		pwTextBox = driver.findElement(By.id("pwd"));
		loginButton = driver.findElement(By.id("loginButton"));
	}

	public void login(String un, String pw) {
		unTextBox.sendKeys(un);
		pwTextBox.sendKeys(un);
		loginButton.sendKeys(un);
	}

}
```

```
package qspiders;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Demo
{

	public static void main(String[] args)
	{
		WebDriver driver=new FirefoxDriver();
		driver.get("http://localhost");
		LoginPage l=new LoginPage(driver);
		l.login("admin","manager");

	}

}
```

Lastpage:

StaleElementReferenceException(unchecked Selenium Exception) We get this when the page is reloaded

---

```
package qspiders;
```

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;

public class Demo2 {

	public static void main(String[] args) throws InterruptedException {
		WebDriver driver = new FirefoxDriver();
		driver.get("http://localhost");
		LoginPage l = new LoginPage(driver);
		l.login("abc", "xyz");
		Thread.sleep(3000);
		l.login("admin", "manager");
	}

}
```

StaleElementReferenceException:

When we run the following code it will perform following steps;

Step#1: Open the browser

Step#2: Open the login page

Step#3: Create the object of Login page class by initializing its elements such as username (a1), password (b1), login (c1).

Where a1, b1, c1 are the address of the current address of the respective elements.

Step#4: Go to address a1 and type abc

Step#5: Go to address b1 and type xyz

Step#6: Go to address c1 and click

Step#7: Since invalid username and password are not entered after clicking on login button, it will display error message by reloading the complete page

Step#8: After reloading elements will be having different address i.e,username(x1),password(y1) and loginButton(z1)

Step#9: It waits for 3 seconds, it will try to enter “admin”in a1 and it is old address of username which does not exist anymore hence we get StaleElementReferenceException

PAGE OBJECT MODEL

*It is one of the java design pattern.

*It is used to develop and test webpages.

*In selenium we use page object model to avoid staleElementReferenceException and to improve the performance of the code.

*In page object model we declare the element using FindBy Annotation(@FindBy) should be imported from following package;

Import org.openqa.selenium.support.FindBy;

It has the following syntax;

@FindBy(locator=”locatorValue”)

private webElement elementName;

We use initElement() [it is a static method] of pageFactory class to initialize all the elements of current class which are declared using @FindBy,it takes 2 arguments

1. WebDriver

2. Current object of the class

Example:

IQ29)What happens if we do not use initElements()

→We get null pointer NullPointerException

IQ30)How do you develop Pageobject model class without writing constructors

→In the POM class we include only declration and utilization and before calling any method of page object class in main method we use initElement statement.

Example:

@FindBy(id=”username”)

Private WebElement unTextBox;

//no constructor

Public void setUserName(String un)

{

unTextBox.sendKeys(un);

}

```
public static void main(String[] args) throws InterruptedException {
WebDriver driver = new FirefoxDriver();
driver.get("http://localhost");
```

Log inPage l=new LoginPage();

pageFactory.initElements(driver,l);

l.setUserName("admin");

We develop 2 types of classes

1. POM class: It is used to store the elements and its objects

2. Test class: It is used to execution purpose

Note:POM class is also called as Page object Repository because we use it to store elements(object of the page)

TestNG: (Test Next Generation)

It is an unit testing framework.

It is a tool used by developers to run unit test cases.It can also be used by automation engineers to run multiple automation scripts and to generate the execution results.

TestNG is available as plugin for eclipse.To install it perform the following steps;

#Step1: Go to help in eclipse

#Step2: Select eclipse marketplace

#Step3: Search for TestNG

#Step4: Click install button of TestNG for eclipse

#Step5: Follow the default instructions till finish

After restarting the eclipse right click on java project→Properties→click on java build path→

Library tab→click Add library→Select TestNG→Click next→Click Finish→click OK

13/02/2016

TestNG class:

1.It is a java class which contains Test method.

2.Any method written using Test annotation(@Test).

3.When we run the TestNG class it automatically generates execution result in HTML format.The name of the file is 'emailable-report.html' which is present inside a folder called'test-output'

4.If the folder is not visible refresh the java project.

Example:

Note:

*To print the message in HTML report we use log reporter class to print same in console also we specify second argument as True

*While developing TestNG class do not use default package, main method and s.o.p

TestNG Suite

It is an xml file which contains list of TestNG classes which are to be executed,In order to create it →right click on java project→goto TestNG →select convert to TestNG→click Finish.It creates a file with the name testing.xml inside java project folder

Example:

<suite name="Suite"parallel="none">

<test name="Test">

<classes>

<class name="qsp.DemoA"/>

<class name="qsp.DemoB"/>

</classes>

</test>

</suite>

Important tags:

1) <suite>
2) <test>
3) <classes>
4) <class>

In order to execute it right click on the xml file→goto runAs→select TestNG Suite

Note: We can directly execute multiple classes by right clicking on the package or java project but it execute them in the random order

IQ31)Can we have more than one Test method in TestNG class

→Yes

IQ32)If multiple Test methods are present,In which order they will be executed?

→In the alphabetical order

IQ33) How do you execute Test methods in required order

→Using 'priority'

Note:

*It always executes in ascending order.priority value need not be in continuous order it can be any integer values (+,-) and variables and decimals are not allowed.

*The default priority value is "0"

*If the priority is duplicate then those methods will be executed in alphabetical order

IQ34) How do you write run a test method multiple times

→ using invocationCount

IQ35) What is the default invocationCount

→1

Note: *If invocationCount is less than or equal to 0(<=0), it will not execute the test method.

*For invocationCount we cannot use variables and decimal numbers.

IMPORTANT ANNOTATIONS OF TestNG

1. @Test→indicates Test method
2. @BeforeMethod→indicates that the method should be executed before the execution of every @Test method
3. @AfterMethod→This method is executed after every @Test method

4. @BeforeClass→this method is executed at the beginning of the TestNG class
5. @BeforeClass→this method is executed at the ending of the TestNG class

Example Program:

```
package Demo;

import org.testng.Reporter;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

public class Demo1 {

	public class DemoA
	{
		@BeforeClass
		public void openApp()
		{
			Reporter.log("open App",true);
		}
		@AfterClass
		public void closeApp()
		{
			Reporter.log("close App",true);
		}
		@BeforeMethod
		public void login()
		{
			Reporter.log("login",true);
		}
		@AfterMethod
		public void logout()
		{
			Reporter.log("logout",true);
		}
		@Test(priority=1)
		public void deleteuser()
		{
			Reporter.log("deleteuser",true);
		}
		@Test(priority=0,invocationCount=3)
		public void edituser()
		{
			Reporter.log("edituser",true);
		}
		@Test(priority=-1)
		public void registeruser()
		{
```

```
                Reporter.log("registeruser",true);
            }


      }

}
```

OUTPUT:

```
open App
login
registeruser
logout

login
edituser
logout

login
edituser
logout

login
edituser
logout

login
deleteuser
logout

close App
```

IQ36)How do you create dependency in TestNG

→Using “dependsOnMethods”

NOTE:

*If both priority and dendency are specified it will consider the dependency

*If both methods are independent then we will get TestNGException(unchecked TestNG exception)

Example program:

```
package Demo;

import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.Test;

public class Demo2 {

            @Test
            public void createUser()
            {
```

```
		Reporter.log("createUser",true);
		Assert.fail();
	}
	@Test(dependsOnMethods={"createUser"})
	public void deleteUser()
	{
		Reporter.log("deleteUser",true);

	}
}
```

OUTPUT:

```
FAILED: createUser
```

Skipped: deleteUser

IQ37)How do you fail TestNG test

→using Assert.fail()

IQ38)How do we compare actual value with expected value without using ifelse statement

→Using assertEquals() of Assert class

Example: Assert.assertEquals(actual,expected);

IQ39)what are the methods available under Assert class

→ 1) asserEquals()

2) asserNotEquals()

3) assertTrue()

4) assertFalse()

5) assertNull() (used to check whether object is initialized or not)

6) assertNotNull()

7) assertSame()

8) assertNotSame()

9) fail()

Note:

*All the above methods are static methods of Assert class.

***If comparison fails then remaining statements of the current Test method will not be executed

*To continue the execution even after the comparision fails we should use softAssert which has non-static methods.

Example:

```
package Demo;

import org.testng.Reporter;
import org.testng.asserts.SoftAssert;

public class Demo3 {

	{
		@Test
		public void create()
		{
			SoftAssert soft=new SoftAssert();
			Reporter.log("Step1", true);
			soft.assertEquals("abc", "xyz");
			Reporter.log("Step2", true);
			soft.assertAll();
			Reporter.log("Step3", true);

		}

	}

}
```

IQ40)What are the difference between Assert and SoftAssert

Assert	SoftAssert
If comparision fails,it will not execute the remaining statements of current method.	If comparision fails,it will execute the remaining statements of current method.
All the methods are static	All the methods are non-static
We do not call assertAll()	We should call assertAll()

- To verify major and critical features we use assert statements and to verify minor statements we use SoftAssert Statements

IQ41) How do you rerun only failed TestNG classes?
→ Using "testing-failed.xml" which is autogenerated by TestNG and it will be present inside test-output folder

Important link:
http://testing.org/doc/documentation-main.html

AUTOMATION FRAMEWORK

*It is the standard guideline,best practices and rules which should be followed while automating the application.

*We should use automation framework to have consistency.

*In automation framework we have three stages:

1)Framework design

2)Framework Implimentation

3)Framework Excecution

1. FRAMEWORK DESIGN

This is the initial stage where Automation lead or manager will specify the folder structures,naming conventions, types of files used etc., based on their past experience and project need.

Example:

Types of files used in the framework with their location

File type	Location
.java	Javaproject/src
.class	Javaproject/bin
.html	Javaproject/test-output
.xml	Javaproject
.jar	Javaproject/jarfiles
.exe	Javaproject/exefiles
.xlsx	Javaproject/TestData
.bat	
.war	

STEPS TO CREATE FOLDERS AND CONFIGURE THE FRAMEWORK:

#Step1: Goto required drive example:D drive and create a folder with the name BSSW5.

#Step2:I n eclipse goto file switch work space other.Browse and select newly created folder(D://BSSW5)→click OK

#Step3: After restarting the eclipse create a java project with the name Automation

#Step4: Under the java project create a folder with the name jarfiles

#Step5: copy selenium jar file and paste it inside the above folder.

#Step6: Right click selenium.server.standalone file→go to build path→select add to build path.It will associate jar file with the java project

#Step7: Create a folder with the name exefiles under java project→copy chromedriver.exe and iedriversever.exe files and paste the exefiles folder

#Step8: Create a folder with the name testdata inside javaproject folder,it will be used to store excel files

#Step9: Right click on java project→goto properties→click on java build path→click on Add library→under library tab select TestNG→next→finish→ok

#Step10: Under src create 2 packages with a name pom and test scripts used to store pom class and testscripts of TestNG class

2. FRAMEWORK IMPLIMENTATION

In this stage each automation engineer will convert the assigned test cases into automation scripts by developing 2 types of classes

1)POM class

2) TestNG class

First we should develop POM class

Automation team will select the test case for automation based on following 2 criterias;

1)It should be part of regression testing,this information will be provided by manual testing team

2)Test case shoud not have any manual intervensions;

a) CAPTCHA(Completely Automated Public Turing test tell Computers and Humans Apart)

b) Bar Code Scanning

c) Bio metric scanning, access cards,OTP,Credit cards etc.,

Because of the above reason 100% automation is not possible

SAMPLE TEST CASES

TESTCASE 1: Valid Login;

Precondition: Login page should be present

Post Condition: Application should be closed

#step1: Enter valid user name

#step2: Enter valid password

#step3: Click on login button

#step4: Click on logout button

TESTCASE 2: Invalid Login:

#Step1:Enter invalid username

#Step2:Enter invalid password

#Step3:click on login button

#Step4:Verify that error message is displayed

TESTCASE 3: Verify build number:

#step1: Login to the application using valid user name and password

#step2: click on help icon

#step3: click on about actiTime

#step4: verify build number

#step5: click on close

#step6: click on logout

STEPS TO DEVELOP A POM CLASS:

#step1: Number of POM class Should be same as number of web pages present on the application i.e.,for each web page there should be a POM class

#step2: Name of the POM class should be same as Title of the respective Web page ending with the word "Page"

#step3:I n each POM class we should declare the elements using @FindBy and initialized using PageFactory class

#step4: The action which should be performed by the elements should be developed of methods

#step5: First execute test cases manually which gives more clarity on the steps,which are to be automated

#step6: While executing Test cases note down title of the page,elements present on the page and actions which s ould be performed on the elements

Example:

PAGE 1:

Title→Login

Elements→username text box,password textbox,Login button,error message.

Actions→ #step1: Enter a value in the UN

#step2: Enter a value in password text field

#step3: click on Login button.

#step4: verify error message is displayed or not.

PAGE 2:

Title→enter Time-Track

Elements→Logout link,Help,about actiTime,Build number,close.

Actions→

#step1: Click on logout link

#step2: click on help

#step3: click on about actiTime

#step4: click on close

#step5: verify Build number

POM class for login Page:

```
package pom;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.testng.Assert;

public class LoginPage {
	@FindBy(id="username")
	private WebElement unTextBox;

	@FindBy(name="pwd")
	private WebElement pwTextBox;

	@FindBy(id="loginButton")
	private WebElement loginButton;

	@FindBy(xpath="//span[contains(text()='invalid')]")
	private WebElement errMsg;
	public LoginPage(WebDriver driver)
	{
		PageFactory.initElements(driver, this);
	}
	public void setUserName(String un)
	{
		unTextBox.sendKeys(un);
	}
```

```
	public void setPassword(String pw)
	{
		unTextBox.sendKeys(pw);
	}
	public void clickLoginButton()
	{
		 loginButton.click();
	}
	public void verifyErrMsg()
	{
		Assert.assertTrue(errMsg.isDisplayed());
	}
```

```
}
```

POM class for **Enter Time Track Page**:

```
package pom;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.testng.asserts.SoftAssert;

public class EnterTimeTrackPage {
	@FindBy(id = "logoutlink")
	private WebElement logoutLink;
	@FindBy(xpath = "(//div[@class='popup_menu_arrow')")
	private WebElement help;

	@FindBy(linkText = "About actiTIME")
	private WebElement aboutActiTime;

	@FindBy(xpath = "//img[@title='close']")
	private WebElement close;

	@FindBy(xpath = "//span[contains(text(),'build')]")
	private WebElement buildNumber;

	public EnterTimeTrackPage(WebDriver driver) {
		PageFactory.initElements(driver, this);
	}

	public void clicklogoutlink() {
		logoutLink.click();
	}

	public void clickhelp() {
		help.click();
	}
```

```
	public void clickAboutActiTime() {
		aboutActiTime.click();
	}

	public void clickClose() {
		close.click();
	}

	public void verifyBuildnumber(SoftAssert s, String eBuildNumber) {
		String aBuildNumber = buildNumber.getText();
		s.assertEquals(aBuildNumber, eBuildNumber);
	}

}
```

DEVELOPING TestNG CLASS (automation script)

1)For every manual TestCase we shpold develop TestNG class inside scripts package.

2)For all the Test cases there will be common steps such as preconditions and postconditions.instead of writing the code repetitively we use inheritance as shown below which increases code reusability;

BaseTest Class:

```
package scripts;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;

public class BaseTest {
	public WebDriver driver;

	@BeforeClass
	public void preCondition() {
		driver = new FirefoxDriver();
		driver.get("http://localhost");
		driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

	}

	@AfterClass
	public void postCondition() {
		driver.close();
	}
}
```

STEPS TO WRITE AUTOMATION SCRIPTS:

#step1: create a class under scripts package and the name of the class should be same as Respective Test case ID.

#step2: Extend it from BaseTest class

#step3: Create a Test Method and the name of the method should start with “test” and end with the class name

#step4: inside the test method write test case steps as in-line comments so that we will not skip any of the steps and it also helps the reviewer

#step5: After each in-line comment call the required method of POM class.

Example:

TEST SCRIPT 1:

```
package scripts;

import org.testng.annotations.Test;

import pom.EnterTimeTrackPage;
import pom.LoginPage;

public class ValidLogin extends BaseTest {
	@Test
	public void testValidLogin() {
		// enter valid un
		LoginPage l = new LoginPage(driver);
		l.setUserName("admin");
		// enter the password
		l.setPassword("manager");
		// click on login button
		l.clickLoginButton();
		EnterTimeTrackPage e = new EnterTimeTrackPage(driver);
		e.clicklogoutlink();

	}

}
```

HIDING METHODS OF OBJECT CLASS

In eclipse goto window →preferences→java→appearance→Type filters→Add→java.lang.Object→OK→OK

TEST SCRIPT 2:

```java
package scripts;

import org.testng.annotations.Test;

import pom.LoginPage;

public class InvalidLogin extends BaseTest
{

	@Test
		public void testInvalidLogin()
		{
			// enter invalid un
			LoginPage lp = new LoginPage(driver);
			lp.setUserName("abc");
			// enter invalid password
			lp.setPassword("xyz");
			// click on login button
			lp.clickLoginButton();
			lp.verifyErrMsg();

		}

	}
```

TEST SCRIPT 3:

```java
package scripts;

import org.testng.annotations.Test;
import org.testng.asserts.SoftAssert;

import pom.EnterTimeTrackPage;
import pom.LoginPage;

public class VerifyBuildNumber extends BaseTest {
	@Test
	public void testverifyBuildNumber() {
		SoftAssert s = new SoftAssert();
		// enter valid un
		LoginPage lp = new LoginPage(driver);
		lp.setUserName("admin");
		// enter valid password
		lp.setPassword("manager");
		// click on login button
		lp.clickLoginButton();
		// click on help
		EnterTimeTrackPage e = new EnterTimeTrackPage(driver);
		e.clickAboutActiTime();

		// verify Build Number
		e.verifyBuildnumber(s, "build 27261");
		//click on close
```

```
            e.clickClose();
            //click on logout
            e.clicklogoutlink();
            s.assertAll();

      }

}
```

NOTE: In order to open the login page and close the application only one time we can ude @BeforeSuite and @AfterSuite annotations.

3. FRAMEWORK EXECUTION

To run all the scripts present in the frame work we use TestNG suite file.

Example:

Right click on the scripts package go to TestNG→Select convert to TestNG→Finish

Right click on testing.xml file→select testing suite→Refresh the java project which will display test-output folder→open emailable-report.hyml file in the browser to see the execution purpose

```
<suite name="Suite" parallel="none">
  <test name="Test">
    <classes>
     <class name="scripts.ValidLogin"/>
      <class name="scripts.InvalidLogin"/>
      <class name="scripts.VerifyBuildNumber"/>


    </classes>
  </test>
</suite>
```

URL:https://poi.apache.org/download.html

Section:29 September 2015-POI 3.13 available

SubSection:Binary Distribution

File:poi-bin-3.13-20150929.zip