

SELENIUM
\$

- >Automation Introduction
- >Advantages & Disadvantages
- >Selenium WebDriver
- >Selenium RC, IDE
- >Advantages & Disadvantages Of Selenium
- >Selenium Installation
- >Selenium Grid
- > 1.Elements
- > 2.Locators
- > 3.X-Path
- > 4.Absolute X-Path
- > 5.Relative X-Path
- > 6.X-Path By Attribute
- > 7.X-Path By Text
- > 8.Handling non-breakable space[nbsp]
- > 9.Multi Select Listbox
- > 10.X-path Traversing
- > 11.Independent & Dependent X-Path
- > 12.Group Index

Interview Quetions

- > 13.JavaScript In Selenium
- > 14.Handling Multiple Elements
- > 15.15.Handling Dropdown Menu / List Box
- > 16.Handling Keyboard & Mouse Operations
- > 17.Handling Frames
- > 18.Handling Pop-Up's
- > 19.Synchronization
- > 20.Encapsulation In Selenium

---> 21.POM
---> 22.TestNG
---> 23.Capturing ScreenShot In Selenium
---> 24.Framework
 1)Method Driven Framework
 2)Data Driven Framework
 3)Hybrid Driven Framework
---> 25.Log4J
---> 26.Maven
---> 27.Jenkins
---> 28.Git & Github

=====

Automation:

1)What is Automation?

A: Performing any task using a system s/w(for as system s/w) is called as Automation.

2)List of s/w Test automation tool available?

-->Selenium

-->QTP

-->RFT(Rational Function Testing)

-->Test complete

-->Test partner

-->Silk Test

-->Auto IT

-->Roubutium

-->Load Runer

-->J meter

etc

3)What are the advantages of Automation?

- >Saves Time
- >Reduces effort
- >Accurate
- >Reusable
- >Consistent

4)What are the disadvantages of Automation?

- >Initial invest is very high
- >Requires consitent maintain

=====

Selenium:

Selenium is free and open source web application automation testing tool.

1)Free:-we can use selenium without purchasing any licence.

URL:- <https://selenium.dev/downloads>

Section:-Selenium standalone server

Link:-Download version 3.141.5 (we used in our project 3.141.59 version)

FileName:-Selenium -Server-Standalone-3.141.5 jar

Jar->Java archive

2)Open Source: We can see and download and customized the source code of selenium.

Available below website:

<https://www.github.com/seleniumHQ/selenium>

3)Web Application:

Using selenium we can test the web application and it supports all the major browser.

such as Mozilla firefox,Google chrome,Opera,Suffari,Microsoft Internet Explorer etc.

4)What type of testing we should do automated?

A:Regression Testing.

5)Should we automated negative Testing?

A:Yes.

6)Which type of test cases automated first?

Smoke testing -->testing the basic or critical features of an appln

A:Login page,Basical or critical feature.

7)Can we do 100% Automation?

A:No

8)Give the example for which we can't go for automation?

A: a)Captring student attendance by swipping the ID card.

b)Taking the attedance of the employee biometric scanner,
paying the bill by swipping the credit card.

c)Entering the details of the product
ex:-barcode scanner

d)OTP(one time password)

=====

Selenium:

-->Free

-->Open source

-->12 languages support

-->All OS support

Installation

Step to install the selenium required

-->JDK

-->Eclipse

-->Mozilla FirefoxDriver / Chrome driver

-->Selenium server standalone software

Sample program

```
package Selenium;

import org.openqa.selenium.firefox.FirefoxDriver;

public class Test
{
    public static void main(String[] args)
    {
        System.setProperty("webdriver.gecko.driver",
"C://Users//Admin//Downloads//Program
files//geckodriver-v0.30.0-win64//geckodriver.exe");

        //Open the Browser
        FirefoxDriver f=new FirefoxDriver();

        //Close the Browser
        f.close();
    }
}
```

Install actiTime:

Next-Next-Next-Finish

Same available internet actiTime
<http://demo.actitime.com/login.do>
<http://localhost/login.do>

Q: Write a script to print the Title & url of the page?

A: Title --> getTitle() --> return type is String

url --> getCurrentUrl() --> return type is String

```
package Selenium;

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

public class Title_URL
{
    public static void main(String[] args)
    {
        //Write a script to print the Title & URL?

        System.setProperty("webdriver.chrome.driver",
"C://Users//Admin//Downloads//Program
files//chromedriver_win32//chromedriver.exe");
        WebDriver driver=new ChromeDriver();
    }
}
```

```

//Maximize the page
driver.manage().window().maximize();

//Enter the URL
driver.get("https://www.hdfc.com");

//Print the title
String title = driver.getTitle();
System.out.println("Title of the page= "+title);

//Print URL
String Url = driver.getCurrentUrl();
System.out.println("URL of the page= "+Url);

//close the browser
driver.close();
}
}

```

--> setSize method is going to set the width & height

--> setPosition method is going to set the X & Y

Ex:

```
package Selenium;
```

```
import org.openqa.selenium.Dimension;
import org.openqa.selenium.Point;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class Test1
```

```
{
```

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

```
        System.setProperty("webdriver.chrome.driver",
"C://Users//Admin//Downloads//Program
files//chromedriver_win32//chromedriver.exe");
```

```
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
```

```
        Thread.sleep(5000);
```

```
        //Enter the url
```

```
        driver.get("https://in.tradingview.com/");
```

```
        Thread.sleep(5000);
```

```
        //width & height
```

```
        Dimension d1=new Dimension(200, 500);
```

```
        driver.manage().window().setSize(d1); //setSize method is going to set
the width & height
```

```
        Thread.sleep(5000);
```

```
        //X & Y
```

```
        Point p1=new Point(50, 200);
```

```
        driver.manage().window().setPosition(p1); //setPosition method is going
```

to set the X & Y

```
        Thread.sleep(5000);
        //Maximize
        driver.manage().window().maximize();

        Thread.sleep(7000);
        //Close the browser
        driver.close();
    }
}
```

Q)How do you close the browser without using close method?

```
:- driver.close(); //closes the current window
    driver.quit(); //closes the entire window
```

Q)How do you enter the url without using get method?

```
:- driver.navigate().to("http://Localhost/login.do");
```

Q)What is difference b/w 'get' and 'navigate' method?

:- get method is used to only enter the url.

Using navigate method we can following below steps

- 1)Enter the url
- 2)click Back
- 3)click forward
- 4)and refresh the page

Ex:

```
---
package Selenium;

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

public class NavigateTo
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver",
"C://Users//Admin//Downloads//Program
files//chromedriver_win32//chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
    }
}
```

```

        Thread.sleep(5000);
        driver.get("https://www.naukri.com/");

        Thread.sleep(5000);
        driver.navigate().to("https://www.facebook.com/");

        Thread.sleep(500);
        driver.navigate().back();

        Thread.sleep(5000);
        driver.navigate().forward();

        Thread.sleep(5000);
        driver.navigate().refresh();

        Thread.sleep(5000);
        driver.close();
    }
}

```

Q)How selenium perform action all the browser?

:- Selenium performs action on all the browser by calling the native method.
 ex: get(); quit(); getTitle();

Q)What is protocol using by selenium to perform action on the browser?

:-JSON wire Protocol:-Java script object Notation(JSON)

Q)What are the languages supported by selenium ?

:-Total 12 languages supported.

```

Java
  C#
Ruby
Python
  Javascript
    Perl
    PHP
      Hasket
      Objective-C
    R
    Dart
    Tcl

```

Note:-Selenium s/w is develop using java hence it is platform independent.

platform independent --> you can write once, run anywhere

Q)Select the unsupporting operating system by selenium?

- :- a)Windows
- b)Linux
- c)Mac

- d)Unix (No browser no selenium)

->Working with GoogleChrome Browser

Step1:-Go to follwing website

<http://chromedriver.storage.googleapis.com/index.html>

Go to latest folder
[2.21]

Download required zip file
[chromedriver_win32.zip]

Extract it in to the required location

Write the code as shown below and execute

```
package Selenium;
```

```
import org.openqa.selenium.Dimension;  
import org.openqa.selenium.Point;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class DemoChrome1  
{
```

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

```
        System.setProperty("webdriver.chrome.driver",  
"C://Users//Admin//Downloads//Program  
files//chromedriver_win32//chromedriver.exe");  
        WebDriver driver=new ChromeDriver();  
        driver.manage().window().maximize();
```

```
        Thread.sleep(3000);  
        driver.get("https://www.facebook.com/");  
        Dimension d1=new Dimension(200, 300);  
        driver.manage().window().setSize(d1);
```

```
        Thread.sleep(3000);  
        Point p1=new Point(100, 10);  
        driver.manage().window().setPosition(p1);
```

```
        Thread.sleep(3000);  
        driver.manage().window().maximize();
```

```
        Thread.sleep(3000);  
        driver.close();  
    }
```

```
}
```

Runtime polymorphism in selenium:

In order to execute same script on any browser to use runtime polymorphism to achieve runtime polymorphism following steps are mandatory.

1) Inheritance

Ex:-all the features of WebDriver interface are inherited in FirefoxDriver class.

2) Method Overriding:

Ex:-all the abstract methods of WebDriver interface are overridden of in FirefoxDriver class.

3) Up casting:

Ex:-WebDriver driver=new FirefoxDriver();

WebDriver is an interface.
driver is a reference variable.
= is assignment operator.
new is keyword.
FirefoxDriver is no args constructor, of FirefoxDriver class.
; is a statement delimiter.

program:

1) What are the flavours of selenium?

:-a) Selenium Core

b) Selenium IDE

c)Selenium RC(Remote Control)

d)Selenium WebDriver.

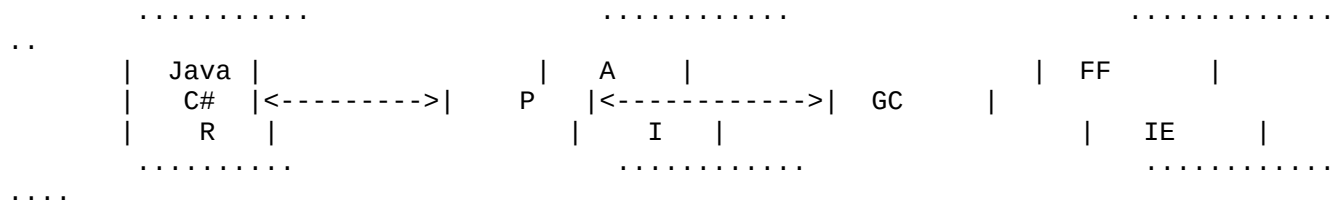
Note:-Selenium RC is also called as selenium1.

Selenium Webdriver is also called as selenium2.

Latest version of selenium webdriver 3.0.1. now latest version is 4.1.0 (imp for interview)

->We also have automation tool develop by same organization to test mobile applications called seledroid

Architechure Of Selenium WebDriver:

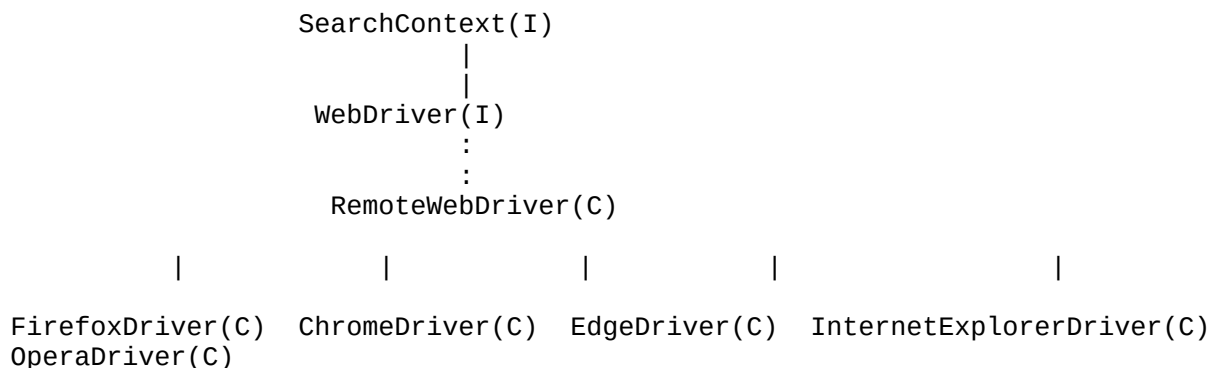


Description:

Selenium software multiple language for each language specific bindings [also called as client bindings] which communicates with webdriver API.

Webdriver API perform the action on the browser using browser specific drivers.

Architechure of webDriver API: (Application Programming Interface)



I - Interface

C - Class

interface to interface --> extends keyword

interface to class --> implements keyword
class to class --> extends keyword

Description:

Searchcontext is super most interface which is extended by webdriver
interface,
all the abstract method or implemented in RemoteWebDriver class which is
inherited to
respective browser classes such as FirefoxDriver, ChromeDriver. etc

=====

Selenium Testing

=====

1.Elements

Anything present on the webpage is called elements.

such as text box,button,radio button,PopUp, Drop-Down,Image,Link, listbox,Checkbox
etc., are called as elements.

There are 3 types

1)TagName --> < lesserthan symbol next word is TagName

2)Attribute --> > till greaterthan Symbol is Attribute

Attribute contains 2 parts --> AN="AV"

1.AttributeName

2.AttributeValue

3)Text --> > after greaterthan symbol there is Text

Ex:

Amazon Sign-in button

```
<input id="continue" tabindex="5" class="a-button-input"
type="submit" aria-labelledby="continue-announce">
```

TagName --> input

Attribute --> id="continue" tabindex="5" class="a-button-input"
type="submit" aria-labelledby="continue-announce"

AttributeName --> id tabindex class type aria-labelledby

AttributeValue --> continue 5 a-button-input submit continue-announce

Text --> No text

=====

2.Locators: --> imp concept

Note: Without locators we are not able to write the script.

--> Before performing the action we should find the required element,
in order to do this we use locator.

--> In selenium there are 8 types of locators, all are static methods
of By.class {it's an abstract class}

--> All this methods takes String as an argument and returns object
of type By, which is used for i/p argument for find element method.

Ex:

abstract class By

{

```

Static void tagName(String tagName)
{
}

Static void id(String id)
{
}

Static void name(String name)
{
}

Static void className(String className)
{
}

Static void linkText(String linkText)
{
}

Static void partialLinkText(String partialLinkText)
{
}

Static void cssSelector(String cssSelector)
{
}

Static void xPath(String xPath)
{
}
}

```

Sample_Automation1

Sample_Automation1.html

Sample Webpage

```

    <html>
    <body>
    <a id="a1" name="n1" class="c1" href="http://Localhost">actitime</a>
    </body>
    </html>

```

Note:Write above code in notepad ,save as demo.html

Scripts

Script-1:

1.Using tagName

```
package Locators;
```

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

```
public class tagName
```

```
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver",
"C://Users//Admin//Downloads//Program
files//chromedriver_win32//chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

```

```
driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation1.html
");
        driver.findElement(By.tagName("a")).click();

        Thread.sleep(2000);
        driver.close();
    }

```

```
}
```

Script-2:

2.Using id

```
package Locators;
```

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

```
public class id
```

```
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver",
"C://Users//Admin//Downloads//Program
files//chromedriver_win32//chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

```

```

driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation1.html
");
        driver.findElement(By.id("a1")).click();

        Thread.sleep(2000);
        driver.close();
    }
}
-----

```

Script-3:

3.Using name

```

package Locators;

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

public class name
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver",
"C://Users//Admin//Downloads//Program
files//chromedriver_win32//chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation1.html
");
        driver.findElement(By.name("n1")).click();

        Thread.sleep(2000);
        driver.close();
    }
}
-----

```

Script-4:

4.Using className

```

package Locators;

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

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

```



```

        {
            System.setProperty("webdriver.chrome.driver",
"C://Users//Admin//Downloads//Program
files//chromedriver_win32//chromedriver.exe");
            WebDriver driver=new ChromeDriver();
            driver.manage().window().maximize();

driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation1.html
");
            driver.findElement(By.className("c1")).click();

            Thread.sleep(2000);
            driver.close();

        }
    }
}
-----

```

Script-5:

5.Using linkText

package Locators;

```

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

```

```

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

```

```

        System.setProperty("webdriver.chrome.driver",
"C://Users//Admin//Downloads//Program
files//chromedriver_win32//chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

```

```

driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation1.html
");
            driver.findElement(By.linkText("actitime")).click();

            Thread.sleep(2000);
            driver.close();

```

```

        }
    }
}

```

--> Note:This Locator can be used only to locate link
(We can't use text box button etc)

Script-6:

6.Using partialLinkText

```
package Locators;
```

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

```
public class partialLinkText
{
```

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

```
        System.setProperty("webdriver.chrome.driver",
"C://Users//Admin//Downloads//Program
files//chromedriver_win32//chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
```

```
driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation1.html
");
```

```
        driver.findElement(By.partialLinkText("time")).click();
```

```
        Thread.sleep(2000);
        driver.close();
```

```
    }
```

```
}
```

-->Note:This locator use to handle dynamic link,where some port of the text is keep on changing.

Ex: Inbox link present in the gmail it will be keep changing.
LinkText is->Inbox(23)

```
driver.findElement(By.partialLinkText("Inbox")).click();
```

-->Note:

1)If specified locator match with multiple element then
findelement method returns address of first matching element.

2)If the specified locator is not matching with any of the element then
find element method will throw NoSuchElementException.

Script-7:

7.Using cssSelector

CSS->Stands for cascading style sheet

Syntax:

```
tagName[Attribute name='Attribute value']
```

Sample_Automation2

Sample_Automation2.html

Sample webpage

```
---
<html>
<body>

    UN:<input type="username">
    PW:<input type="password">

</body>
</html>
```

Note: To identify the pw page present in the above page we can't use id,name,class name,linkText or partialLinkText, because they are not present in the text, we can use tagName but it's duplicate page with the user name field(username).

in this situation we can cssSelector to identify the pw page as show below code.

```
UN
---
input[type='username']
```

```
PW
---
input[type='password']
```

```
package Locators;
```

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

```
public class partialLinkText
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver",
"C://Users/Admin//Downloads//Program
files//chromedriver_win32//chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
    }
}
```

```
driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation1.html
");
    driver.findElement(By.partialLinkText("time")).click();

    Thread.sleep(2000);
    driver.close();
}
```

=====

3.X-Path:

Script-8:

7.Using x-Path

Sample_Automation3

Sample_Automation3.html

Sample webpage

<html>

<body>

 FN:<input type="text">

 LN:<input type="text">

</body>

</html>

To indentify the last name[LN] field,present in the above page we can't use cssSelector also as it's matching with first name field in this situation we go for Xpath.

Xpath:

xpath is path of an element in html tree we use forward(/) which represents immediate child element,if 2 or more elements have same path then we use index in x path index starts from 1.

html Tree:

html[1]

|

body[1]

 |---input[1]

 |---input[2]

X-path for FN TextField:

html[1]/body[1]/input[1]

(OR)

html/body/input[1]

X-path for LN TextField:

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

(OR)

html/body/input[2]

```
package Locators;
```

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

```
public class xpath
```

```
{  
    public static void main(String[] args) throws InterruptedException  
    {  
        System.setProperty("webdriver.chrome.driver",  
"C://Users//Admin//Downloads//Program  
files//chromedriver_win32//chromedriver.exe");  
        WebDriver driver=new ChromeDriver();  
        driver.manage().window().maximize();  

```

```
driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation3.html  
");
```

```
        driver.findElement(By.xpath("html/body/input[1]")).sendKeys("Elon");  
        driver.findElement(By.xpath("html/body/input[2]")).sendKeys("Musk");
```

```
        Thread.sleep(2000);  
        driver.close();  
    }  
}
```

=====

4.Absolute X-Path:

-->From html begin to till element is called Absolute x-Path.

-->Immediate Child (/) -->Single slash means absolute x-Path.

Writing the xpath expression from the beginning of the html tree till element is called as absolute xpath.

Sample_Automation4

Sample_Automation4.html

Sample webpage

```

---
<html>
<body>
<table border="1">
<tbody>

    <tr>
    <td>Selenium</td>
    <td>900</td>
    </tr>

    <tr>
    <td>Java</td>
    <td>2000</td>
    </tr>

</tbody>
</table>
</body>
</html>
-----

```

Elements	Absolute x-Path
Selenium	/html/body/table/tbody/tr[1]/td[1]
900	/html/body//table/tbody/tr[1]/td[2]
Java	/html/body//table/tbody/tr[2]/td[1]
2000	/html/body//table/tbody/tr[2]/td[2]
Selenium, 900	/html/body/table/tbody/tr[1]/td
Java, 2000	/html/body/table/tbody/tr[2]/td
Selenium, 900, Java, 2000	/html/body/table/tbody/tr/td
Selenium, Java	/html/body/table/tbody/tr/td[1]
900, 2000	/html/body/table/tbody/tr/td[2]
Selenium, 2000	/html/body/table/tbody/tr[1]/td[1] /html/body/table/tbody/tr[2]/td[2]
Java, 900	/html/body/table/tbody/tr[2]/td[1] /html/body/table/tbody/tr[1]/td[2]

```

=====
=====

```

5.Relative X-Path:

-->Any child element (//)

Absoulate xpath expression will very lengthy,in order to reduce the length we can

use relative xpath,
to write any relative xpath we use(//) which that present any child element.

Elements	Absolute x-Path
Selenium	-----> //tr[1]/td[1]
900	-----> //tr[1]/td[2]
Java	-----> //tr[2]/td[1]
2000	-----> //tr[2]/td[2]
Selenium, 900	-----> //tr[1]/td
Java, 2000	-----> //tr[2]/td
Selenium, 900, Java, 2000	-----> //td
Selenium, Java	-----> //td[1]
900, 2000	-----> //td[2]
Selenium, 2000	-----> //tr[1]/td[1] //tr[2]/td[2]
Java, 900	-----> //tr[2]/td[1] //tr[1]/td[2]

Q)What's the difference b/w / and //[slash]?

:-

-->Single forward(/) represents immediate child element

-->Double forward(//) represents any child element

double forward slash is also called dependends

Q)Derive xpath expression which matches with all the link present on the webpage?

:-//a

Q)Derive an xpath which matches with all the images present on the webpage?

:-//img

Q)Derive an xpath matches with all this links & images present on the webpage?

:-//a|//img

Q)What is difference b/w //a and //table//a?

:-//a matches with all the links which are present in the entire webpage
//table//a matches with all the links which are present inside the table

Q)Difference b/w //a & //a[1]?

```
:-      1 html
        |
        |__ body
            |
            |__ |_a (A)1
                  |_a (B)2
                  |_table
                      |__tbody
                          |__tr
                              |__td
                                  |
                                  |__td |_a (C)1
                                  |
                                  |__td |_a (D)1
```

Note:- //a matches with all the a links

//a will match --> //(A) //(B) //(C) //(D)

//a[1] matches with all the a 1st links

//a[1] will match --> //(A) //(C) //(D)

```
html
|
body
|
table
|--tr[1]
|--->a Gmail
|--->a Amazon
|--->a Ola
|--tr[2]
|--->a HDFc
|--->a Axis
table
|--tr[1]
|--->a faceBook
|--->a ICICI
```


Questions)	Answers)
---	---
All the links	-----> //a
All the first link	-----> //a[1]
Only HDFC	-----> //tr[2]//a[1]
Only last link	table[1] tr[1] ----> //table[1]//tr[1]//a[3]

X-path expression matches with hidden elements also, if we try to perform any action on the hidden elements we get ElementNotvisibleException.

=====

6.X-Path By Attribute:

In order to identify the required element we can also use attribute it has following syntax.

//tagName[@AttributeName='AttributeValue']

Ex:

//input[@id='username']

//input[@name='username']

//input[@placeholder='username']

//input[@type='email']

//input[@tabindex='2']

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

Interview Question

1)Derive an xpath expression which matches with all the chekbox present on the webpage?

:-//input[@type='checkbox']

2)Derive an xpath which matches with all the buttons present on the webpage?

:-//input[@type='button']

3)Can we use multiple attributes in single xpath expression?

```
: -//a[@id='a1'][@class='c1']  
  //a[@id='a1'and @class='c1']  
  //a[@id='a1'or @class='c1']
```

=====

7.X-Path By Text:

If attribute is not present or with attribute is matching with multiple elements then we can identify it's text the syntax is

```
//tagName[text()='tv'] --> recommended
```

or

```
//tagName[.='tv']
```

Ex: Source of code of login button present on the actitime application
<div>Login</div>

In this example no attribute we can identify by using it's text xpath .
//div[text()='Login']

Ex: Source code of users tab present in home page of actitime
<div class="label">USERS</div>

When we try to identify using attribute
//div[@class='label']->4 matches, so attribute is duplicate

We can identify it using text xpath is
//div[text()='USERS']

Note: In xpath shortcut for text function is(.)dot,
In example
//td[.='Java']

=====

X-Path By Contains Text:

For handling non-breakable space(nbsp)

```
//tagName[contains(text(), 'tv')]
```

```
//tagname[conatains(@AN, 'AV')]
```

8. Handling non-breakable space[nbsp]:

Developer can give space using spacebar or using this keyword

Ex: html code

```
<div> Java </div><br>
<div>&nbsp;Selenium&nbsp;</div>
```

Handling above code

```
Ans: //div[.=' _ Java _']Right
      //div[.=' Selenium ']Wrong
      //div[.='&nbsp;Selenium&nbsp;']Wrong
```

-->X-path by using Text do not support nonbreakable space.
To handle to use Contains function

Syntax:

```
//tag[contains(text(),'textvalue')]
//tag[contains(@AN, 'AV')]
```

Ex: //div[contains(., 'Selenium')]

We can handle normal space also

```
//div[contains(., 'Java')]
```

We can handle dynamic link(any other dynamic element also)

```
//a[contains(text(), 'Inbox')]
```

9. Multi Select Listbox:

Sample html code

```
<html>
<body>
<Select Multiple>

<Option>Idly</Option>
<Option>Dosa</Option>
<Option>Vade</Option>
<Option>Poori</Option>
```

```

<Option>CCBath</Option>

</Select>
</html>
</body>

```

```

-----

      html
      |__body
      |__Select
      |__Option Idly
      |__Option Dosa
      |__Option Vade
      |__Option Poori
      |__Option CCBath

```

Following-Sibling:

This x-path axes will match with all the siblings[Child of same parent] which are present after the given element

```

Ex: //Option[.='Vade']/following-sibling::*
    //Option[.='Vade']/following-sibling::*[1]
    //Option[.='Vade']/following-sibling::*[2]
    //Option[.='Vade']/following-sibling::*[last()]

```

Preceding-Sibling:

This x-path axes will match with all the siblings[Child of same parent] which are present before the given element

```

Ex: //Option[.='Vade']/preceding-sibling::*
    //Option[.='Vade']/preceding-sibling::*[1]
    //Option[.='Vade']/preceding-sibling::*[2]
    //Option[.='Vade']/preceding-sibling::*[last()]

```

10.X-path Traversing:

we can derive xpath expression which navigates from one element to another element which is called as traversing x-path.

Sample html code

```
-----
<html>
<body>

<h1><i><b>Buy Books:</b></i></h1>

<table border="1">

<tbody>

  <tr>
    <td>Selenium</td>
    <td>1000</td>
  </tr>

  <tr>
    <td>Java</td>
    <td>2000</td>
  </tr>

  <tr>
    <td>Python</td>
    <td>3000</td>
  </tr>

</tbody>
</table>
</html>
</body>
```

There are 2 paths of traversing

1]Forward traversing

 Navigating from parent element to child element

Ex: //table/tbody/tr[2]/td[1]

The above xpath is navigating from table to java

2]Backward traversing

 Navigating from child element to parent element

Ex: //td[text()='3000']/../../../../..

 //td[text()='3000']/parent::tr

 //td[text()='3000']/ancestor::body

 //td[text()='3000']/ancestor::table

 //td[text()='3000']/ancestor::tbody

```
//td[text()='3000']/ancestor::html
```

Q)How to handle dynamic element?

: -if dynamic element is link and only some part of the value keep changing
then use partialLinktext

Ex: dynamic link

=====

11.Independent & Dependent X-Path:

If the element is completely dynamic or it is duplicate then we can identify using some other unique element, this concept is called as independent dependent x-path.

To derived x-path we should perform following steps

Step1: Inspect independent element and notedown it's sourcecode

Step2: Place the mouse pointer on the source code of independent element
then move in upwards direction till it highlights both independent and
dependent elements, it's called as common parent add in to html tree.

Step3: Use arrow keys to navigate from common parent to dependent element
add it's path to the tree.

Step4: Derive the x-path from independent element to common parent[backward
traversing]
then derive the x-path from common parent to dependent element [forward
traversing]

Sample html code

```
<html>
<body>

<h1><i><b>Buy Books:</b></i></h1>

<table border="1">

<tbody>

  <thead>
    <th>Subject</th>
    <th>Cost</th>

  </thead>
```

```

<tr>
  <td>Selenium</td>
  <td>1000</td>
</tr>

<tr>
  <td>Java</td>
  <td>2000</td>
</tr>

<tr>
  <td>Python</td>
  <td>3000</td>
</tr>

</tbody>
</table>
</body>
</html>

```

Independent & Dependent xpath

```
//td[.='Python']/../td[2]
```

```
//td[.='Selenium']/../td[2]
```

```
//td[.='Java']/../td[2]
```

Ex:

```
https://www.selenium.dev/downloads/
```

```
//p[text()='Ruby']/../p[2]//a
```

```
//p[.='Ruby']/../p[4]//a
```

```
//p[contains(., 'Latest stable ')]/../p[1]//a
```

```
=====
=====
```

12.Group Index:

If we can't identify the element using add preview text and independent & dependent concept we can use the group index.

--> Retrieving one element from a group of matching elements by using index is called group index.

--> When the xpath expression matches with multiple elements, then we go for group

index.

Sample html code

```
-----
<html>
<body>

<h1><i><b>Group Index in Selenium:</b></i></h1>

    <div>
        <input type="text" value="A">
        <input type="text" value="B">
    </div>

    <div>
        <input type="text" value="C">
        <input type="text" value="D">
    </div>

</body>
</html>
```

Ans:

```
-----
//input[1]----->Normal index A C

(//input)[1]--->A
(//input)[2]--->B
(//input)[3]--->C
(//input)[4]--->D

-----

(//input[1])[1]--->A
(//input[1])[2]--->C
(//input[2])[1]--->B
(//input[2])[2]--->D
```

Ex:

<https://www.google.com/>

(//div[@id='SIVcob']/a[7]

=====
=====

Interview Questions:

-->Derive an xpath which matches with all the checkboxes are present on the page?

:- //input[@type='checkbox']

-->Derive an xpath which matches with first and last check box?

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

-->Derive an xpath which matches with first 3 checkbox?

:- (//input[@type='checkbox'])[position()<=3]

--->Out of 8 locators we use following 4 locators

:- 1)Id

2)Name

3)LinkText

4)Xpath

1)How do you get the text present in the textbox?

:- Using getAttribute(" ")

2)What is difference b/w getText & getAttribute?

:- getText is used to get the value of the inner text.

getAttribute is used to get the value of the Attribute.

getText --> used to print normal text

getAttribute --> used to print text inside the text box

Ex: <input name="email" id="user" value="admin@mail.com">abc@mail.com</input>

getAttribute("name") --> email

getAttribute("id") --> user

getAttribute("value") --> admin@mail.com

getText() --> abc@mail.com

3)The textbox has the text but it is disabled can you change?

:- No,if we try to do that we get InvalidElementException.

Sample_Automation5

Sample_Automation5.html

Sample webpage

UN:<input id="un" value="Anil" disabled>

PW:<input id="pw">

4)How do you print the value present in the disabled username text box?

Script:-

```
package interviewQuestions;
```

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

```
public class getAttribute_value
```

```
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
```

```
driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation5.html
");
```

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

```
        //get text present inside the text box
        String val = unTB.getAttribute("value");
```

```
        System.out.println(val);
```

```
        Thread.sleep(2000);
        driver.close();
```

```
    }
```

```
}
```

5)The textbox has the text but it is disabled can you change?

:-No,if we try to do that we get InvalidElementStateException.

6)The textbox has the text but it is disabled can you send any value?

:-No,if we try to do that we get ElementNotInteractableException.

7)The page has two text box first one has a value but it is disabled the second one is empty, how do you copy paste the text to second textbox?

Script:-

```
package interviewQuestions;

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

public class copyPaste
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);

        driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation5.html
");

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

        String val1 = unTB.getAttribute("value");

        //paste in password text box
        WebElement pwTB = driver.findElement(By.id("pw"));

        pwTB.sendKeys(val1);

        Thread.sleep(2000);
        driver.close();
    }
}
```

8)How do you verify textbox is enabled or disabled?

:-Using isEnabled() method.

Script:-Text box is disabled

```
package interviewQuestions;
```

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

```
public class isEnabled
```

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

```
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
```

```
        Thread.sleep(2000);
```

```
driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation5.html
");
```

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

```
        if (unTB.isEnabled())
```

```
        {
            System.out.println("True: Text box is enabled");
```

```
        }
        else
```

```
        {
            System.out.println("False: Text box is disabled");
```

```
        }

        Thread.sleep(2000);
        driver.close();
```

```
    }
```

Answer:False: Text box is disabled

Script:-Text box is enabled

Sample_Automation5

Sample_Automation5.html

Sample webpage

UN:<input id="un" value="Anil" enabled>

PW:<input id="pw">

```

package interviewQuestions;

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

public class isEnabled
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);

        driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation5.html
");

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

        if (unTB.isEnabled())
        {
            System.out.println("True: Text box is enabled");
        }
        else
        {
            System.out.println("False: Text box is disabled");
        }

        Thread.sleep(2000);
        driver.close();
    }
}

```

Answer:True: Text box is enabled

6)cont...-->text box is enabled now, can you send any value?

:- Yes

Script:-

```

package interviewQuestions;

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

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

```

```

        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);

driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation5.html
");

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

        if (unTB.isEnabled())
        {
            System.out.println("True: Text box is enabled");

            Thread.sleep(2000);
            unTB.clear();

            Thread.sleep(2000);
            unTB.sendKeys("Ani");
        }
        else
        {
            System.out.println("False: Text box is disabled");
        }

        Thread.sleep(2000);
        driver.close();
    }
}

```

Answer:True: Text box is enabled

9)How do you clear the text without using clear method?

:-by passing the shortcut control a and delete.

Script:-

```

-----
package interviewQuestions;

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;

class clearWithoutClearMethod
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");

```

```

        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);

driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation5.html
");

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

        Thread.sleep(2000);
        unTB.sendKeys(Keys.CONTROL+"a"+Keys.CONTROL.DELETE);

        Thread.sleep(2000);
        driver.close();
    }
}

```


10)Write a script to copy paste the value present in the one text box to another textbox without using getAttribute method?

:-
 Script:-

```

package interviewQuestions;

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;

public class copyPaste_value_without_getAttribute
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);

driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation5.html
");

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

        WebElement pwTB = driver.findElement(By.id("pw"));

        unTB.sendKeys(Keys.CONTROL+"ac");

        pwTB.sendKeys(Keys.CONTROL+"v");
    }
}

```

```

        Thread.sleep(3000);
        driver.close();
    }
}

```

11) Write a script to click on the checkbox present in the login page and verify it's selected?

:-

Script:- without clicking check box

```

package interviewQuestions;

```

```

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

```

```

public class checkBox_Verify
{

```

```

    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
        driver.get("http://localhost/login.do;jsessionid=6yv5h1kbrg16");

        WebElement chkBox = driver.findElement(By.id("keepLoggedInCheckBox"));

        if (chkBox.isSelected())
        {
            System.out.println("True: Text box is selected");
        }
        else
        {
            System.out.println("False: Text box is not selected");
        }

        Thread.sleep(3000);
        driver.close();
    }
}

```

Answer: False: Text box is not selected

Script:- clicking check box

```

package interviewQuestions;

```

```

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

```



```

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

public class checkBox_Verify1
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
        driver.get("http://localhost/login.do;jsessionId=6yv5h1kbrg16");

        WebElement chkBox = driver.findElement(By.id("keepLoggedInCheckBox"));

        Thread.sleep(2000);
        chkBox.click();

        if (chkBox.isSelected())
        {
            System.out.println("True: Text box is selected");
        }
        else
        {
            System.out.println("False: Text box is not selected");
        }

        Thread.sleep(3000);
        driver.close();
    }
}

```

Answer: True: Text box is selected

NOTE:

->check box & radio button --> isSelected()

->text box --> isEnabled()
isDisabled()

->isSelected method is also be used for verifying whether radio button selected or not.

Assignment

12)Write script go to facebook.com select the radio button and verify that selected?

:- we can verify radio button by using isSelected method also.

Script:-

13)Write a script to find height & width, x & y and font-size of any textbox?

:- Ex: in facebook login page

Script:-

```
package interviewQuestions;

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.chrome.ChromeDriver;

public class WH_XY_fontSize
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
        driver.get("https://www.facebook.com/");

        WebElement unTB = driver.findElement(By.id("email"));

        //width & height
        Dimension wh = unTB.getSize();
        System.out.println("Width & height: "+wh);

        //x&y
        Point xy = unTB.getLocation();
        System.out.println("X & Y: "+xy);

        //font-size
        String fs = unTB.getCssValue("font-size");
        System.out.println("Font-size: "+fs);

        Thread.sleep(3000);
        driver.close();
    }
}
```

14)What are the different ways of clicking a button?

:-click method

Sendkeys method

Submit method

Java Script

Script:-

```
-----  
package interviewQuestions;  
  
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;  
  
public class diffWaysOfClickingButton  
{  
    public static void main(String[] args) throws InterruptedException  
    {  
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\  
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");  
        WebDriver driver=new ChromeDriver();  
        driver.manage().window().maximize();  
  
        Thread.sleep(2000);  
        driver.get("https://crmaccess.vtiger.com/log-in/");  
  
        WebElement loginButton = driver.findElement(By.xpath("//button[.='Sign  
in']"));  
  
        //1st way - click()  
        loginButton.click();  
  
        //2nd way - sendKeys()  
        loginButton.sendKeys(Keys.ENTER);  
  
        //3rd way - submit()  
        loginButton.submit();  
  
        Thread.sleep(3000);  
        driver.close();  
    }  
}
```

Note:

Submit method works only in type of the element is submit

type="submit" --> then submit() works else it will not.

->If we use submit method on buttons where type is not submit then we get NoSuchElementException throw by submit button and not by findelement method

15)Write a script to verify that email text box and next button present in gmail page are allign?

:-

Script:-

```
package interviewQuestions;

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.firefox.FirefoxDriver;

public class textbox_button_allign {

    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.gecko.driver",
"C://Users/Admin//Downloads//Program
files//geckodriver-v0.30.0-win64//geckodriver.exe");
        WebDriver driver=new FirefoxDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
        driver.get("https://accounts.google.com/ServiceLogin?");

        WebElement emailTB = driver.findElement(By.id("identifierId"));
        int x1 = emailTB.getLocation().getX();
        System.out.println(x1);

        WebElement nextButton = driver.findElement(By.id("identifierNext"));
        int x2 = nextButton.getLocation().getX();
        System.out.println(x2);

        if (x1==x2)
        {
            System.out.println("Pass");
        }
        else
        {
            System.out.println("Fail");
        }

        Thread.sleep(3000);
        driver.close();
    }
}
```

Answer: Fail

16)getText() & getAttribute() program

:- getText is used to get the value of the text.

getAttribute is used to get the value of the Attribute.

getText --> used to print normal text

getAttribute --> used to print text inside the text box.

Script:-

```
package interviewQuestions;
```

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

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

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

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class getTect_getAttribute_program {
```

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

```
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\  
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
```

```
        WebDriver driver=new ChromeDriver();  
        driver.manage().window().maximize();
```

```
        Thread.sleep(2000);  
        driver.get("http://localhost/login.do;jsessionid=6yv5h1kbrg16");
```

```
        //getText -->no parameter  
        WebElement text = driver.findElement(By.xpath("//nobr[text()='actiTIME  
2016.3']"));
```

```
        String text2 = text.getText();  
        System.out.println("Text: "+text2);
```

```
        //getAttribute --> parameter  
        WebElement attribute =  
driver.findElement(By.xpath("//nobr[text()='actiTIME 2016.3']"));
```

```
        String attribute2 = text.getText();  
        System.out.println("Attribute: "+attribute2);
```

```
        Thread.sleep(3000);  
        driver.close();
```

```
    }
```

```
}
```

17)isDisplayed() program

:- login page/home page displayed or not

Script:-

```

package interviewQuestions;

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

public class isDisplayed
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(1000);
        driver.get("http://localhost/login.do;jsessionId=6yv5h1kbrg16");

        Thread.sleep(1000);
        driver.findElement(By.id("username")).sendKeys("anil");

        Thread.sleep(2000);
        driver.findElement(By.name("pwd")).sendKeys("12345");

        Thread.sleep(2000);
        driver.findElement(By.id("loginButton")).click();

        Thread.sleep(3000);
        WebElement errorMsg =
driver.findElement(By.xpath("(//span[@class='errorMsg'])[1]"));

        if (errorMsg.isDisplayed())
        {
            System.out.println("True: Error message is displayed");
        }
        else
        {
            System.out.println("False: Error message is not displayed");
        }

        Thread.sleep(3000);
        driver.close();
    }
}

```

Answer: True: Error message is displayed

*****Important methods of webelement Interface*****

-->clear -- clears the text

-->click -- click on the lnk,button,check box,radio button etc *****

-->getAttribute -- printing text present inside the text box *****

```

-->getCssvalue -- font-size
-->getLocation -- x & y
-->getSize -- width & height
-->getTagName -- finding the tagName
-->getText -- prints only text *****
-->isDisplayed -- login page/home page displayed or not *****
-->isEnabled -- text box/button is enable or not *****
-->isSelected -- radio button/check box is selected or not *****
-->SendKeys -- sending username, password etc *****
-->submit -- only when button type="submit"

```

```

=====
=====

```

13.JavaScript In Selenium:

```
-----
```

JavascriptExecutor is an interface.

: -Sometime methods of selenium not work.

Ex:

```

-->Sendkeys and clear method will not work if element is disabled.
    in such cases we can use javascript to perform the action.

```

```

-->For scrollDown we use js

```

```
-----
```

1)How do you handle element if it is disabled?

: -Using JavaScript.

2)How do you entered value in to text box if it's disabled?

: -Using JavaScript

3)How do you enter text in to text box without using sendkeys method?

: -Using JavaScript

```
-----
```

```

-->Steps to execute JavaScript Manually:

```

```

    -->Go to console

```

Type the javascript in the textbox present at the bottom box.

Enter

```
alert('Hi')
```

1)How do you execute javascript in selenium?

:-by using executeScript method of JavaScriptExecutor.

By Downcasting:

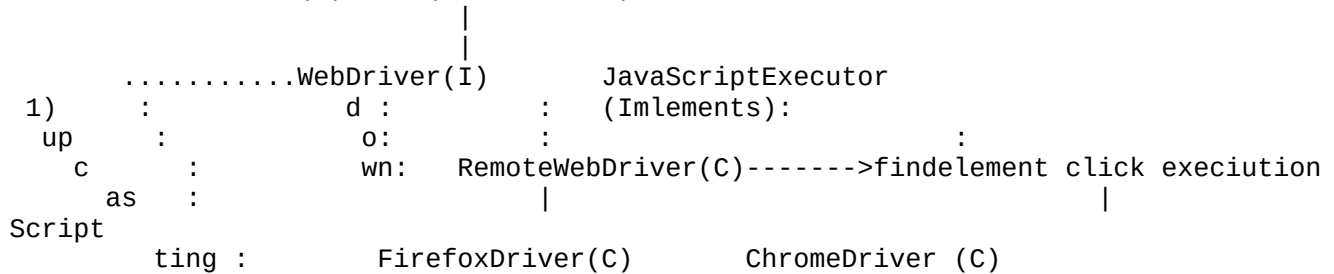
```
1.WebDriver driver=new FirefoxDriver();
```

```
2.driver.get("file:///F:/Program/Rad.html");
```

```
3.RemoteWebDriver r=(RemoteWebDriver)driver;
```

```
4.r.executeScript("alert('Hi')");
```

SearchContext(I)---->(FindElement)



Type casting:

```
A.WebDriver driver=new FirefoxDriver();
```

```
B.driver.get("file:///F:/Program/Rad.html");
```

```
C.JavascriptExecutor js=(JavascriptExecutor)driver;
```

```
D. js.executeScript("confirm('bye')");
```

Q):

1)Write Script enter value a textbox which is disabled?

2)How do you remove value present in the textbox?

or

Without using clear method and shortcut(control a+delete)?

A):

Sample WebPage:

Sample_Automation7.html

UN:<input id="un" value="Anil" disabled>

PW:<input id="pw">

Script:

package Selenium;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class JavaScriptExecutor

{

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

 {

 System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");

 WebDriver driver=new ChromeDriver();

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

 Thread.sleep(2000);

driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation6.html");

 JavascriptExecutor js=(JavascriptExecutor)driver;

 //sendKeys

 // js.executeScript("document.getElementById('un').value='Selenium1'");

 //clear

 js.executeScript("document.getElementById('un').value='');

 Thread.sleep(3000);

 driver.close();

 }

}

3)Write a script to scroll the page?

:-

Script:

```

package Selenium;

import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Js_scrollBy
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
        driver.get("https://www.naukri.com/");

        Thread.sleep(5000);
        JavascriptExecutor js=(JavascriptExecutor)driver;

        js.executeScript("window.scrollTo(0,1000)");

        Thread.sleep(2000);
        driver.close();
    }
}

```

4)Write a script to scroll bottom to the page?

:-

Script:

```

package Selenium;

import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Js_scrollTo_bottom
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
        driver.get("https://www.naukri.com/");

        Thread.sleep(5000);
        JavascriptExecutor js=(JavascriptExecutor)driver;

        js.executeScript("window.scrollTo(0,document.body.scrollHeight)");
    }
}

```

```

        Thread.sleep(3000);
        driver.close();
    }
}

```

Note:

Scroll bar is 2000

1.ScrollBy(0,1000)->get to 3000

2.ScrollTo(0,1000)->get to 1000

5)Write a script to scroll in to sports section present in the google news?

:-
Script:

```

package Selenium;

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

public class Js_scrollTo_sportsPage
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
        driver.get("https://news.google.com/");

        Thread.sleep(3000);
        WebElement sportsPage =
driver.findElement(By.xpath("(//span[.='Sports'])[1]"));

        int y = sportsPage.getLocation().getY();

        System.out.println(y);

        JavascriptExecutor js=(JavascriptExecutor)driver;

        js.executeScript("window.scrollTo(0,"+y+)");

        Thread.sleep(5000);
    }
}

```

```

        driver.close();
    }
}

```

6)Write a script to click on login button by using Javascript?

or

How do you click on login button it's disabled?

:-
Script:

```

WebDriver driver=new FirefoxDriver();
driver.get("https://www.facebook.com/login/");
JavascriptExecutor js=(JavascriptExecutor)driver;
js.executeScript("document.getElementById('loginbutton').click");

```

=====

14.Handling Multiple Elements:

Key points to remember imp

****findElements()

****getText()

****for loop

To handle multiple elements we use findElements method, it's return type is List and should be imported from java.util package.

Q) What are the difference b/w FindElement & FindElements?

A)

FindElement

FindElements

1)Return type is WebElement

1)Return type is List<WebElement>

2)If locator is matching with multiple elements,it returns first matching element.

3)If the locator is not matching with any of the elements, it throws NoSuchElementException

2)If locator is matching with multiple elements,it returns All the matching elements.

3)it will not throw any exception, but it returns an empty list (0 elements)

Sample WebPage:

Sample_Automation7.html

Amazon
Facebook
Flipkart
Google

1)Count all the links in the webpage
Print all the links in the webpage
Click on any one link except Amazon

A)

package Selenium;

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 handlingMultipleElements
{

 public static void main(String[] args) throws InterruptedException
 {
 System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
 WebDriver driver=new ChromeDriver();
 driver.manage().window().maximize();

 Thread.sleep(2000);

driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation7.html
");

 Thread.sleep(2000);
 List<WebElement> allLinks = driver.findElements(By.xpath("//a"));

above script architecture:



: -

Approach:

FindElements and getText() inside for loop.

Pseudo code:

Step1 - Open the browser
Step2 - Enter the google.com
Step3 - Type in Selenium search in the textbox
Step4 - Press Enter
Step5 - Get all the link and print text
Step6 - Close the browser

Script:

```
package interviewQuestions;

import java.util.List;

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;

public class handlingMultipleElements_count_print_allLinks
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
        driver.get("https://www.google.co.in/");

        WebElement searchButton = driver.findElement(By.name("q"));

        searchButton.sendKeys("Selenium" + Keys.ENTER);

        Thread.sleep(2000);

        List<WebElement> allLinks = driver.findElements(By.xpath("//a"));

        //count all the links
        int count = allLinks.size();
        System.out.println("Total links: "+count);

        //print all the links
        for (int i = 0; i < count; i++)
        {
            String text = allLinks.get(i).getText();

            if (text.length()>0)           //to remove hidden links
            {
                System.out.println(text);
            }
        }
    }
}
```

```

    }

    Thread.sleep(3000);
    driver.close();
}
}

```

3) Write a script to search in selenium in google and print all the Autosuggetions?
& select the last one?

:-

Pseudo code:

Step1:-Open the browser
 Step2:-Enter the url
 Step3:-Type in selenium search in the selenium last link
 Step4:-Select all the autosuggestions.
 Step5:-Select the last autosuggestions.
 Step6:-Close the browser

Script:

```

package interviewQuestions;

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 handlingMultipleElements_autoSuggetions
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
        driver.get("https://www.google.co.in/");

        WebElement searchButton = driver.findElement(By.name("q"));

        searchButton.sendKeys("Selenium");

        Thread.sleep(2000);

        List<WebElement> allAutoSugg =
driver.findElements(By.xpath("//span[contains(text(), 'selenium')]"));

        //count
        int count = allAutoSugg.size();
    }
}

```



```

        System.out.println("Total auto suggetions: "+count);

        //print
        for (int i = 0; i <count; i++)
        {
            String text = allAutoSugg.get(i).getText();

            System.out.println(text);
        }

        //click on last autoSuggetion
        allAutoSugg.get(count-1).click();

        Thread.sleep(3000);
        driver.close();
    }
}

```

Sample WebPage:

Sample_Automation7.html

```

<html>
<body>
<table border="1">
<tbody>
    <tr>
        <td>slno</td>
        <td>sub</td>
        <td>cost</td>
    </tr>
    <tr>
        <td>1</td>
        <td>unit</td>
        <td>123</td>
    </tr>
    <tr>
        <td>2</td>
        <td>sql</td>
        <td>156</td>
    </tr>
</tbody>
</table>
</body>
</html>

```

Q)Print all the text?

Script:

```
package interviewQuestions;
```

```
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 handlingMultipleElements_print_allText
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);

driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation8.html
");

        List<WebElement> allElements = driver.findElements(By.xpath("//td"));

        //count
        int count = allElements.size();
        System.out.println("Total elements: "+count);

        //print
        //allElements.get(0).getText(); //--> if only 1 element is there

        for (int i = 0; i < count; i++)
        {
            String text = allElements.get(i).getText(); //--> more than 1
element is there

            System.out.println(text);
        }

        Thread.sleep(3000);
        driver.close();
    }
}

```

```

=====
=====

```

15.Handling Dropdown Menu / List Box:

```
-----
```

Key points to remember imp

```
-----
```

****Select class

****We have 9 methods in Select class

- 1.selectByIndex()
- 2.selectByValue()
- 3.selectByVisibleText()
- 4.deselectAll()
- 5.deselectByIndex()
- 6.deselectByValue()
- 7.deselectByVisibleText()
- 8.isMultiple() --> to check the list is multi list box or not
- 9.getOptions() --> for counting, selecting and deselecting multiple elements

--> We can handle dropdown / list box by using Select class, it should be imported from
org.openqa.selenium.support.ui package

--> Select class is parametrized constructor, we can select the required option present in the listbox based on text,index or value.

NOTE:

1) How can we verify Multi List Box?

:- by using isMultiple()

2) deselectAll() will only work for Multi select box or Multi List Box.

3) When we give duplicate vales then the first method will work.

Note:

-->If specified index ,text or value is invalid we will get NoSuchElementException Excep.

-->If specified text or value duplicate it will select first matching element.

-->To handle multiselect listbox also we use Select class here.the specified option is invalid we get NoSuchElementException Excep.

-->On multiselect listbox we can run use all the three selectby methods and we can also use deselect methods.

-->Deselect method works only on multilevel listbox,on dropdown listbox we get UnsupportedOperationException.

-->in order to check whether listbox isMultiselect or single select we use isMultiple method

1) Write a script to select Month from facebook create account page?
and check whether it is a multi list box.
and answer can we use deselectAll().

:- deselectAll() will only work for multi select box.

Script:

```
-----
package Selenium;

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 Handling_ListBox
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
        driver.get("https://www.facebook.com/r.php");

        Thread.sleep(2000);
        WebElement listBox = driver.findElement(By.id("month"));

        //handling month list --> using Select class
        Select select=new Select(listBox);

        //we have 3 methods in Select class

        Thread.sleep(2000);
        //1st method -- select by index
        select.selectByIndex(5); //June

        Thread.sleep(2000);
        //2nd method -- select by value --> recommended in real time
        select.selectByValue("11"); //Nov

        Thread.sleep(2000);
        //3rd method -- select by visibleText
        select.selectByVisibleText("Jul"); //July

        // select.deselectAll(); //UnsupportedOperationException

        if (select.isMultiple())
        {
            Thread.sleep(2000);
            select.deselectAll();
        }
        else
    }
```

```

        {
            System.out.println("False: It is not a Multi List Box");
        }

        Thread.sleep(3000);
        driver.close();
    }
}

```

Sample WebPage:

Sample_Automation9.html

```

<h1><em>Please find the books:</em></h1>

<select id="book" multiple>

    <option value="s">Selenium</option>

    <option value="j">Java</option>

    <option value="p">Python</option>

    <option value="r">Ruby</option>

    <option value="p1">Perl</option>

    <option value="c">C++</option>

</select>

```

2) Write a script to handle Multi List Box?

:-

Script:

```

package Selenium;

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 Handling_MultiListBox
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
    }
}

```

```
driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation9.html");
```

```
Thread.sleep(2000);
WebElement listBox = driver.findElement(By.id("book"));
```

```
//using Select class
Select select=new Select(listBox);
```

```
//we have 3 methods in Select class
```

```
Thread.sleep(2000);
//1st method -- select by index
select.selectByIndex(3); //Ruby
```

```
Thread.sleep(2000);
//2nd method -- select by value --> recommended in real time
select.selectByValue("j"); //Java
```

```
Thread.sleep(2000);
//3rd method -- select by visibleText
select.selectByVisibleText("C++"); //C++
```

```
if (select.isMultiple())
{
```

```
    Thread.sleep(2000);
    //select.deselectAll();
```

```
    Thread.sleep(2000);
    select.deselectByIndex(3); //Ruby
```

```
    Thread.sleep(2000);
    select.deselectByValue("j"); //Java
```

```
    Thread.sleep(2000);
    select.deselectByVisibleText("C++"); //C++
```

```
    Thread.sleep(2000);
    System.out.println("True: It is Multi List Box");
```

```
}
else
{
```

```
    System.out.println("False: It is not a Multi List Box");
```

```
}
```

```
Thread.sleep(3000);
driver.close();
```

```
}
```

```
}
```


Interview Quetions:

- 1) ->Write a script to count no of content present in the listbox?
- >Write a script to print content of textbox?
- >Write a script select & deselect all the option present in the listbox?

```

1.count
2.print
3.select all
4.deselect all

```

:-

Script:

```
package interviewQuestions;
```

```
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 Handling_MultiListBox1
{
```

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

```
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
```

```
        WebDriver driver=new ChromeDriver();
```

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

```
        Thread.sleep(2000);
```

```
        driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation9.html
");
```

```
        Thread.sleep(2000);
```

```
        WebElement listBox = driver.findElement(By.id("book"));
```

```
        Select select=new Select(listBox);
```

```
        //getOptions()-->for counting, selecting and deselecting multiple
elements
```

```
        Thread.sleep(2000);
```

```
        List<WebElement> options = select.getOptions();
```

```
        //count
```

```
        int count = options.size();
```

```
        System.out.println("No of content present in the listbox: "+count);
```

```
        //print
```

```
        for (int i = 0; i < count; i++)
```

```
        {
```

```
            Thread.sleep(2000);
```

```
            String text = options.get(i).getText();
```

```
            System.out.println("content of textbox: "+text);
```

```
        //select all
    }
```

```

        select.selectByIndex(i);

    }

    //deselect all
    select.deselectAll();

    Thread.sleep(3000);
    driver.close();
}
}

```

2)Write a script to select & deselect all the option present in the listbox in reverse order?

:-

Script:

```

-----
package interviewQuestions;

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 Handling_MultiListBox_reverseOrder
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);

        driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation9.html
");

        Thread.sleep(2000);
        WebElement listBox = driver.findElement(By.id("book"));

        Select select=new Select(listBox);

        //getOptions()-->for counting, selecting and deselecting multiple
elements
        Thread.sleep(2000);
        List<WebElement> options = select.getOptions();

        //count
        int count = options.size();
    }
}

```



```

        System.out.println("Total elements in list box: "+count);

        //print
        for (int i = count-1; i > 0; i--)
        {
            Thread.sleep(2000);
            String text = options.get(i).getText();
            System.out.println(text);

            //select all
            select.selectByIndex(i);

        }

        //deselect all
        select.deselectAll();

        Thread.sleep(3000);
        driver.close();
    }
}

```

3)Write script to print content of listbox in sorted order?

:-
Sccript:

```

package interviewQuestions;

import java.util.ArrayList;
import java.util.Collections;
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 Handling_MultiListBox_sortedOrder
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);

        driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation9.html
");

        Thread.sleep(2000);
        WebElement listBox = driver.findElement(By.id("book"));
    }
}

```

```

        Select select=new Select(listBox);

        //getOptions()-->for counting, selecting and deselecting multiple
elements
        List<WebElement> alloptions = select.getOptions();

        ArrayList<String> allText=new ArrayList<String>();

        //count
        int count = alloptions.size();
        System.out.println("Total elements in list box: "+count);

        //print
        for (WebElement option : alloptions)
        {
            String text = option.getText();
            allText.add(text);
        }

        Collections.sort(allText);

        for (String text : allText)
        {

            System.out.println(text);
        }

        Thread.sleep(3000);
        driver.close();
    }
}

```

```

=====
=====

```

16.Handling Keyboard & Mouse Operations:

```
-----
```

:- We can handle Keyboard & Mouse operations by using Actions class.

```

***Key points to remember*** imp
-----

```

***Actions class

****1.to move to element -->moveToElement()

****2.Right click --> contextClick()

****3.Double click --> doubleClick()

****4.Drag & drop --> dragAndDrop()

```
*****
```

Note:

Action is an interface of Interactions package which has abstract methods such as moveToElement, context click, double click, drag & drop, send keys.

--> perform all these methods are implemented in Actions class.

--> Actions class as parametrized constructor, it takes arguments of WebElement, whenever we call any method of Actions class we must perform() method.

1) how to move to element --> moveToMethod

:-

Script:

```
package Selenium;

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 Handling_dropDownMenu_moveToElement
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
        driver.get("https://www.actimind.com/");

        WebElement moveToElement =
driver.findElement(By.xpath("//a[contains(text(), 'EXPERTISE')]"));

        //Actions class
        Actions action=new Actions(driver);

        //how to move to element --> moveToElement()
        action.moveToElement(moveToElement).perform();

        Thread.sleep(2000);
        driver.findElement(By.xpath("(//a[.='Cloud Applications'])
[1]")).click();

        Thread.sleep(3000);
        driver.close();
    }
}
```

How do you handle context menu?

:- To handle context menu we use Actions class right clicking is called as contextclick.

When we Right click of the element we get list of applications we will get context menu.

To select required option present in context menu we used sendkeys methods on actions class.

2) how to right click element --> contextClick()

:-

Script:

```
package Selenium;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.interactions.Actions;

public class Handling_dropDownMenu_contextClick
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.gecko.driver", "C:\\\\Users\\Admin\\Downloads\\Program files\\geckodriver-v0.30.0-win64\\geckodriver.exe");
        WebDriver driver=new FirefoxDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
        driver.get("http://localhost/login.do");

        WebElement rightClick =
driver.findElement(By.xpath("//a[text()='actiTIME Inc.']"));

        Actions action=new Actions(driver);

        //how to right click -->contextClick()
        action.moveToElement(rightClick).contextClick().perform();

        Thread.sleep(2000);
        action.sendKeys("q").perform(); //to click on Inspect element,we should
pass the underlined letter i.e., "q"

        Thread.sleep(3000);
        driver.close();
    }
}
```

Q)What is difference b/w Findelements sendkeys & Actions Sendkeys? ***imp

:-

Findelements sendKeys -----	Actions sendKeys -----
->Method is present under webDriver Interface.	->Method is present under Action Interface.
->We use to enter the input on a curzer Specific element.	->it is used enter the input in the current position on the specific element.
->Not overloaded,overrided method	->Overloaded method
->No Perform() method	->should call Perform()

Q)What is the use of Actions class?

:-Use Actions class to handle drop down menu context menu and also to perform drag & drop actions.

We can also use for double clicking.

3.Double click --> doubleClick()

:-

Script:

```
package Selenium;

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.firefox.FirefoxDriver;
import org.openqa.selenium.interactions.Actions;

public class Handling_dropDownMenu_doubleClick
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
        driver.get("https://www.facebook.com/");

        WebElement doubleClick = driver.findElement(By.id("email"));
```

```

        Actions action=new Actions(driver);

        //how to double click -->doubleClick()

        action.moveToElement(doubleClick).sendKeys("Anil").doubleClick().perform();


        Thread.sleep(3000);
        driver.close();
    }
}

```

Q)How do you perform drag & drop action in selenium?

:- It is done using drag & drap actions class.

4.Drag & drop --> dragAndDrop()

:-

Script:

```
package Selenium;
```

```
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.firefox.FirefoxDriver;
import org.openqa.selenium.interactions.Actions;
```

```
public class Handling_dropDownMenu_dragAndDrop
```

```
{
```

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

```
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
```

```
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();
```

```
        Thread.sleep(2000);
        driver.get("http://www.dhtmlgoodies.com/submitted-scripts/i-google-
like-drag-drop/index.html");
```

```
        WebElement dragFrom = driver.findElement(By.xpath("//h1[.='Block
1']"));
```

```
        WebElement dragTo = driver.findElement(By.xpath("//h1[.='Block 4']"));
```

```
        Actions action=new Actions(driver);
```

```
        //how to drag and drop -->dragAndDrop()
        action.dragAndDrop(dragFrom, dragTo).perform();
    }
}
```

```

        Thread.sleep(3000);
        driver.close();
    }
}

```

```

=====
=====

```

17.Handling Frames:

```
-----
```

```

***Key points to remember*** imp
-----

```

```

****//handling iframe

```

```

        driver.switchTo().frame(arg0);

```

```

****//going to main page

```

```

        driver.switchTo().defaultContent();

```

```

            or

```

```

        driver.switchTo().parentFrame();

```

```

*****

```

-->Webpage present inside another webpage is called as embedded webpage.

-->Developer uses iframe tag to create embedded webpage.

-->When we right click on the element it will display this frame option in the content menu, if the selected element is inside the frame.

-->If the element is inside the frame then we should transfer the control into the frame using following statement.

```

        driver.switchTo().frame(arg0);

```

```

            frame(int);
            frame(String);
            frame(WebElement);

```

```

            //all these 3 are overloaded methods.

```

Where, argument can be index of the frame(int) or id of the frame(String) or element of the frame(WebElement).

-->In order to go to the main page we should use following statement.

```

        driver.switchTo().defaultContent();

```

```

            or

```

```
driver.switchTo().parentFrame();
```

```
-----  
Sample WebPage:
```

```
-----  
Sample_Automation10.html
```

```
-----  
<html>
```

```
<body>
```

```
FN:<input id="t3" type="text"></br>
```

```
C1:<input id="c1" type="checkbox" name="n1"></br>
```

```
C2:<input id="c2" type="checkbox" name="n1">
```

```
</body>
```

```
</html>
```

```
Sample_Automation11.html
```

```
-----  
<html>
```

```
<body>
```

```
UN:<input id="t1" type="text"><br>
```

```
PW:<input id="t2" type="password"><br>
```

```
<iframe id="f1" class="c1" src="Sample_Automation10.html"/>
```

```
</body>
```

```
</html>
```

```
-
```

```
-----  
:- handling iframe in the above sample webpage
```

```
Script:
```

```
-----  
package Selenium;
```

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

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

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

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class Handling_Frames
```

```
{
```

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

```
    {
```

```
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\  
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
```

```
        WebDriver driver=new ChromeDriver();
```



```

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

        Thread.sleep(2000);

driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation11.htm
l");

        //handling iframe
        WebElement frames = driver.findElement(By.id("f1"));
        driver.switchTo().frame(frames);

        driver.findElement(By.id("t3")).sendKeys("Ani"); //sending value to FN
        driver.findElement(By.id("c1")).click(); //clicking C1
        driver.findElement(By.id("c2")).click(); //clicking C

        //going to main page

        driver.switchTo().defaultContent();
//    driver.switchTo().parentFrame();

        driver.findElement(By.id("t1")).sendKeys("Anil"); //sending value to UN
        driver.findElement(By.id("t2")).sendKeys("123456789"); //sending value
to PW

        Thread.sleep(3000);
        driver.close();
    }
}

```

```

=====
=====

```

18.Handling Pop-Up's:

```
-----
```

Key points to remember imp

```
-----
```

In selenium we categorised the pop-up into following types.

- 1.Alert Pop-up or JavaScript pop-up
- 2.Confirmation pop-up
- 3.Hidden division pop-up or Calender pop-up
- 4.Child browser pop-up
- 5.Window pop-up or File upload pop-up

1.Alert pop-up

Characterstics

- >We can move the pop-up
- >We can't inspect the element
- >We have only OK button

Handling:

Alert class-->accept()
 -->getText

Q)How do we handle alert & confirmation pop-up?

: -To handle this pop-up we use

driver.switchTo.alert();and then we use any of the following methods

- 1)Alert class
- 2)getText [get the present on the pop-up]
- 3)accept [To click ok]

Note:

-> If the pop-up is not present then we get NoAlertPresentException

-> If we try to action on the page without closing the pop-up we get
UnhandledAlertException.

Script:

```
-----  
package Selenium;  
  
import org.openqa.selenium.Alert;  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.firefox.FirefoxDriver;  
  
public class Handling_PopUp_Alert  
{  
    public static void main(String[] args) throws InterruptedException  
    {  
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\  
Downloads\\Program files\\geckodriver-v0.30.0-win64\\geckodriver.exe");  
        WebDriver driver=new FirefoxDriver();  
        driver.manage().window().maximize();  
  
        Thread.sleep(2000);  
        driver.get("https://ksrtc.in/oprs-web/");  
  
        WebElement searchBtn =  
driver.findElement(By.xpath("//button[contains(text(),'Search for Bus')]"));  
        searchBtn.click();  
  
        Alert alert = driver.switchTo().alert();  
  
        String text = alert.getText();  
        System.out.println(text); //printing alert pop-up text  
  
        alert.accept();  
  
        //send From location  
        driver.findElement(By.id("fromPlaceName")).sendKeys("Bangalore");  
  
        Thread.sleep(3000);  
        driver.close();  
    }  
}
```

2. Confirmation pop-up

Characterstics

->We can move the pop-up

->We can't inspect the element

->We have OK & Cancel button

Handling:

Alert class-->accept() & dismiss()
 -->getText

Q)How To Handle Confirmation Pop Up In Selenium WebDriver

:- Confirmation pop up is also a type of alert pop up.

The difference between:

-> alert pop up and confirmation pop up is that alert pop up contains single button whereas

-> confirmation pop up contains two buttons **OK** and **CANCEL**.

-> To click on OK button, use accept() method and if you want to click on CANCEL button, use dismiss() method.

Sample WebPage:

Sample_Automation10.html

<html>
<body>
<p>click on button</p>
<button onclick="myFun()">Click Me</button>
<p id="demo"></p>
<script>
function myFun()
{
var a;
if(confirm("Press a button")==true)
{
a="you pressed ok.";
}
else
{
a="you pressed cancel.";
}
document.getElementById("demo").innerHTML=a;
}
</script>
</body>
</html>

Script:

```

package Selenium;

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 Handling_PopUp_Confirmation
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);

driver.get("file:///C:/Users/Admin/Desktop/Anil/htmlScripts/Sample_Automation12.htm
l");

        WebElement clickMeBtn = driver.findElement(By.xpath("//button[.='Click
Me']"));

        //click
        clickMeBtn.click();

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

        String text = alert.getText();
        System.out.println(text);

        Thread.sleep(2000);
        // alert.accept(); //clicking on OK btn in pop-up

        alert.dismiss(); //clicking on Cancel btn in pop-up

        Thread.sleep(3000);
        driver.close();

    }
}

```

Note:

- 1.If you want to click on OK button, you need to use the accept() method.
- 2.But if you want to click on Cancel button, you have to use dismiss() method.

3.Hidden division pop-up or Calender pop-up

Characterstics

- >We can't move the pop-up
- >We can inspect the element
- >it is colorful pop-up

Handling:

findElement()

Scrppt:

```
package Selenium;

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

public class Handling_PopUp_HiddenDivisionOrCalender
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(2000);
        driver.get("https://www.cleartrip.com/");

        driver.findElement(By.xpath("//div[@id='root']/div/div/div/div/div[2]/
div/div[3]/div[3]/div/div/div/div/button")).click();;

        WebElement departBtn =
driver.findElement(By.xpath("(//div[@id='root']/div/div/div/div/div[2]/div/div[3]/
div[3]/div/div/div/div/div/ul/div/div/div[2]/div/div[3]/div[3]/div[7]/div/div)
[1]"));

//body/div[@id='root']/div[1]/div[1]/div[1]/div[1]/div[2]/div[1]/div[3]/div[3]/
div[1]/div[1]/div[1]/div[1]/div[1]/ul[1]/div[2]/div[1]/div[1]

//(//div[text()='20'])[1]
        departBtn.click();

        Thread.sleep(3000);
        driver.close();
    }
}
```

4.Child browser pop-up

Characterstics

- >We can move the pop-up
- >We can inspect the element
- >it is colorful pop-up
- >we can minimize & maximize the pop-up

Handling: We handle this pop-up using SwitchTo window statement.

getWindowHandles()

driver.switchTo().window()

--> How can we handle child browser Pop-up ---> getWindowHandles()

--> How many windows

--> How can we print all the URL

--> How can we print all the title

--> How can we close current window ---> close()

--> How can we close specified child window

Script:

Quetions:

1)What is difference b/w getWindowhandle and getWindowhandles?

:-

getWindowhandle --> returns windowhandle of current browser, it's return type is String.

getWindowHandles --> returns windowhandle of all the browser, it's return type is Set<String>

2) What is the difference between driver.close() and driver.quit() Method?

:-

close() --> WebDriver's close() method closes the web browser window that the user is currently working on or we can also say the window that is being currently accessed by the WebDriver. The command neither requires any parameter nor does it return any value.

quit() --> Unlike close() method, quit() method closes down all the windows that the program has opened. Same as close() method, the command neither requires any parameter nor does it return any value.

3) How do you count no of browser opened by selenium?

:

```
WebDriver driver=new FirefoxDriver();  
driver.get("https://www.naukri.com/");  
driver.manage().window().maximize();  
int count = driver.getWindowHandles().size();  
System.out.println(count);
```

4) Write a script to print window handle of all the browser?

or
Write a script to print title of all the browser?
or
Write a script to close all browser without using quit method?

:-

```
WebDriver driver=new FirefoxDriver();  
driver.get("https://www.naukri.com/");  
driver.manage().window().maximize();  
Set<String> allWH = driver.getWindowHandles();  
int count = driver.getWindowHandles().size();  
System.out.println(count);  
for(String allw:allWH)
```



```

{
    System.out.println(allw);

    driver.switchTo().window(allw);

    String title = driver.getTitle();
    System.out.println(title);
    driver.close();
}

```

5) Write a script to close specified browser?
 and
 Write a script to close only child browser?
 and
 Write a script to close only parent browser?
 :-

6) How do you handle new tab in selenium?

:- Not possible.

In selenium no option switchTo one tab to another tab

--> but when selenium clicks on a link it forcefully opens the page in new window
 so that we can use child browser concept.

5.Window pop-up or File upload pop-up

Characterstics

->We can move the pop-up

->We can't inspect the element

->close button will not be there

NOTE: It is not alert, child browser, confirmation, Hidden division pop-up

Handling:

Third party automation tools:

AutoIt Automation Tool

Download &
 Installation

--> Pop-up displayed on the application is not alert ,hidden division,file upload
file download or child browser then we call it has window pop-up.

--> Using selenium we can't handle window pop-up hence we use some other
tool such as AutoIT.

AutoIT: is free window based application automation tool.

Download setup file from following website and install:

URL: <https://www.selenium.dev/downloads/>

File:-

Autoit-v3-setup.exe

-->Writing script using AutoIT:-

Step1:- Go to start

```
    |
  All programs
    |
  AutoIT
    |
  Select Script Editor
    |
  Write the code save it's required loc
    |
  Source code extention will be .Au3
```

Q)Write a script to perform following steps?

:-

- 1)Open the browser
- 2)Go to download page of selenium
- 3)Click on Selenium server standalone link
- 4)Activate the window pop-up
- 5)Press left Arrow-key so that control is transfered to save file button
- 6)press enter so that it clicks on save button

-->Open the AutoIt and write the following code:

WinWaitActive("Opening selenium-server-4.1.2.jar")

```
Send("(LEFT)")
```

```
Sleep('5000')
```

```
Send("(ENTER)")
```

```
-----
```

```
Script:
```

```
-----
```

```
package Selenium;
```

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

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

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

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

```
public class Handling_PopUp_Window
```

```
{
```

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

```
    {
```

```
        System.setProperty("webdriver.gecko.driver", "C:\\Users\\Admin\\  
Downloads\\Program files\\geckodriver-v0.30.0-win64\\geckodriver.exe");
```

```
        WebDriver driver=new FirefoxDriver();
```

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

```
        Thread.sleep(2000);
```

```
        driver.get("https://www.selenium.dev/downloads/");
```

```
        Thread.sleep(2000);
```

```
        driver.findElement(By.xpath("//a[.='4.1.2']")).click();
```

```
        Runtime.getRuntime().exec("C:\\Users\\Admin\\Desktop\\Anil\\autoIT  
Scripts\\WindowPopUp1");
```

```
        Thread.sleep(3000);
```

```
        driver.close();
```

```
    }
```

```
}
```

```
=====
```

```
19.Synchronization:
```

```
-----
```

Process of matching selenium speed with application is called
synchronization.

--> During runtime if the application is slow then we get NoSuchElementException or
ElementNotInteractableException,

even though specified locator is correct, because FindElement method will not wait till the element to load.

--> In order to handle this we should match speed of the selenium with the application.

One of the way is:-

-> 1) Using sleep() *****imp

Ex: Thread.sleep(10000);

Script:

```
package Selenium;
```

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

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

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class Synchronization_sleep
```

```
{
```

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

```
    {
```

```
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
```

```
        WebDriver driver=new ChromeDriver();
```

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

```
        Thread.sleep(2000);
```

```
        driver.get("https://webdriveruniversity.com/Popup-Alerts/index.html");
```

```
        driver.findElement(By.xpath("//a[.='CLICK ME!']")).click();
```

```
        Thread.sleep(10000); //using Thread.sleep()
```

```
        driver.findElement(By.xpath("//p[.='CLICK ME!']")).click();
```

```
        Thread.sleep(3000);
```

```
        driver.close();
```

```
    }
```

```
}
```

Ex:- driver.findElement(By.id("Loginbutton")).click();

```
    try
```

```
    {
```

```
        Thread.sleep(30000);
```

```
    }
```

```
    catch(InterruptedException e)
```

```
    {
```

```
    }
```

```
driver.findElement(By.id("Logoutbutton")).click();

driver.close();
```

--> sleep method, it's always wait for the specified duration that is if the duration is 20 sec even though element is loaded in the 5 sec.

--> sleep method will also drastically increase no of statements in the script.

--> In order to overcome this problem we use ImplicitlyWait.

Other of the way is:-

-> 2)implicitlyWait() *****imp

Ex: driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

-->implicitlyWait applicable for only 2 methods:

- 1.findElement
- 2.FindElements

Script:

```
package Selenium;
```

```
import java.util.concurrent.TimeUnit;
```

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

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

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class Synchronization_implicitlyWait
{
```

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

```
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
```

```
        WebDriver driver=new ChromeDriver();
```

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

```
        driver.get("https://webdriveruniversity.com/Popup-Alerts/index.html");
```

```
        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
```

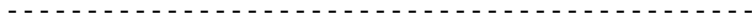
```
        driver.findElement(By.xpath("//a[.='CLICK ME!']")).click();
```

```
        driver.findElement(By.xpath("//p[.='CLICK ME!']")).click();
```

```
        Thread.sleep(3000);
```

```
        driver.close();
```

- a) TimeUnit.DAYS
- b) TimeUnit.SECONDS
- c) TimeUnit.HOURS
- d) TimeUnit.MICROSECOND-S
- e) TimeUnit.MILLISECONDS
- f) TimeUnit.MINUTES
- g) TimeUnit.NANOSECONDS



```
-->explicitly wait applicable for all the methods other than findElement &
findElements methods.
```

Script:

```
package Selenium;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedCondition;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class Synchronization_explicitlyWait
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        driver.get("http://localhost/login.do");

        WebDriverWait wait=new WebDriverWait(driver, 20);

        String loginPageTitle = driver.getTitle();
        System.out.println("Login page title: "+loginPageTitle);

        driver.findElement(By.id("username")).sendKeys("admin");

        driver.findElement(By.name("pwd")).sendKeys("manager");

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

        wait.until(ExpectedConditions.titleContains("Time")); //or Time or

        String homePageTitle = driver.getTitle();
        System.out.println("Home page title: "+homePageTitle);

        Thread.sleep(3000);
        driver.close();
    }
}
```

1)Write a script login to the application with valid credentials and display title of the homepage?

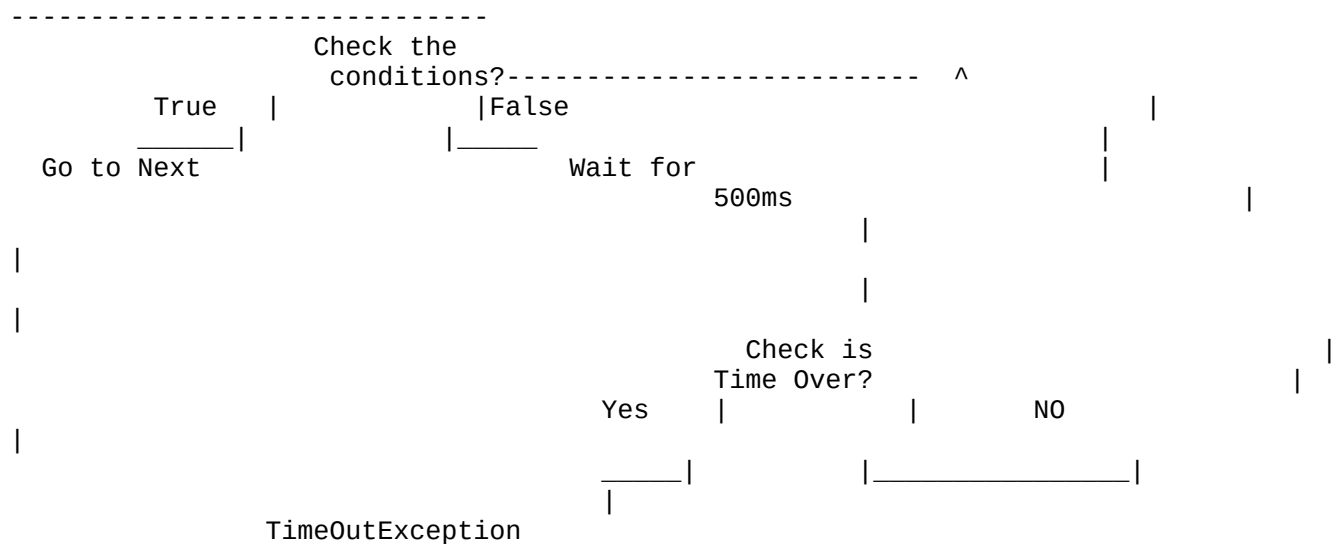
:-

After login if we use getTitle it returns title of the curent page if the

pplication is slow we get title of loginpage instead of homepage. in this scenarious we can't use implicitlyWait beacuse getTitle method do not uses implicitlyWait. to handle this scenarious we use ExplicitWait.

- a)WebDriverWait is called ExplicitWait beacuse of it's specified waiting condition Explicitly.
- b)The conditions are present in a class called expected conditions, they are also called as predicates.
- c)If in ExplicitWait if condition is satisfied it's goes to next step if not satisified even after specified duration it will throw TimeOutException.
- d)WebDriver extends fluentwait class that is uses pooling period.

->Flow diagram explain usage of explicitlyWait:-



*****imp question

Q) What are the difference between ImplicitlyWait and ExplicitlyWait?

:-

ImplicitlyWait	ExplicitlyWait
1)We can handle the synchronization FindElement & FindElements methods	1)Any Method
2)We do not specify waiting condition	2)We should specify conditions
3)After the duration we get NoSuchElementException or ElementNotInteractableException	3)TimeOutExceptions

4)TimeUnit can be SECONDS,HOURS,DAYS
etc.

4)Only Seconds

5)Implicit Wait time is applied to all
only to those
the elements in the script

5)Explicit Wait time is applied
elements which are intended by us

=====

20.Encapsulation In Selenium:

****Realtime project, we are not going to use Encapsulation concept **imp**

****Problem is --> if page/app gets refresh/load, address will change/expiry hence we will get **imp**

--> NoSuchElementException or

--> StaleElementReferenceException (Reference - address, stale-
invalid/expiry)

To overcome this problem, in reaal time we should go for POM ****imp**

--->Selenium code to enter username.

```
driver.findElement(By.id("Username")).sendKeys("Prasanna");
```

--->Selenium code to enter username using encapsulation concept:-

1)Binding a data member (variable) with member function is called encapsulation.

it contains final variables, getters() & setter ()

2)For any given data member(variable) we perform following 3 steps:-

->Declaration Ex: private int empNo; private WebElement unTB;

->Initialization Ex: empNo=1001;
unTB=driver.findElement(By.id("Username")); //Constructor

->Utilization unTB.sendKeys("Anil"); //Method

Ex:

class Encapsulation

{

```

//Declaration
private WebElement unTB;

//Initialization [Inside the Constructor]
public Encapsulation(WebDriver driver)
{
    unTB=driver.findElement(By.id("username"));
}

//Utilization [Inside the Method]
public void setUserName(String un)
{
    unTB.sendKeys(un);
}

```

//Execute Script

Class ExecuteScript1

```

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

    }

}

```

Script-1:

```

-----
package EncapsulationInSelenium;

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

public class FaceBookAppInSelenium_Encapsulation
{
    //Declaration

    private WebElement unTB;

```

```

private WebElement pwTB;
private WebElement loginButton;

//Initialization [Inside the Constructor]

public FaceBookAppInSelenium_Encapsulation(WebDriver driver)
{
    unTB=driver.findElement(By.id("email"));

    pwTB=driver.findElement(By.id("pass"));

    loginButton=driver.findElement(By.name("login"));

}

//Utilization[Inside the Method]

public void setUsername(String un)
{
    unTB.sendKeys(un);
}

public void setPassword(String pw)
{
    pwTB.sendKeys(pw);
}

public void clickOnLoginButton()
{
    loginButton.click();
}
}

```

Note: We can't execute in above class directly beacuse no main method.
for execution purpose we develop another class.

```

package EncapsulationInSelenium;

import java.util.concurrent.TimeUnit;

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

import POMInSelenium.FaceBookAppInSelenium_POM;

public class ExecuteScript1_Encapsulation
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
    }
}

```

```

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

        Thread.sleep(5000);
        driver.get("https://www.facebook.com/");

        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

        FaceBookAppInSelenium_Encapsulation f1=new
        FaceBookAppInSelenium_Encapsulation(driver);

                f1.setUserName("Anil");

                f1.setPassword("1234567");

                f1.clickOnLoginButton();

                Thread.sleep(5000);
                driver.quit();
    }
}

```

Script-2:

```

package EncapsulationInSelenium;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class ActiTimeAppInSelenium_Encapsulation
{
    //Declaration

    private WebElement unTB; //private-->within class

    private WebElement pwTB;

    private WebElement chkBox;

    private WebElement loginButton;

    private WebElement logoutButton;

    private WebElement actiTimeInc;

    //Initialization [Inside the Constructor]

    public ActiTimeAppInSelenium_Encapsulation(WebDriver driver)
    {
        unTB=driver.findElement(By.id("username"));

        pwTB=driver.findElement(By.name("pwd"));
    }
}

```

```

        chkBox=driver.findElement(By.id("keepLoggedInCheckBox"));

        loginButton=driver.findElement(By.id("loginButton"));

        logoutButton=driver.findElement(By.xpath("//a[text()='Logout']"));

        actiTimeInc=driver.findElement(By.xpath("//a[text()='actiTIME
Inc.']]"));
    }

```

//Utilization[Inside the Method]

```

public void setUsername(String un)
{
    unTB.sendKeys(un);
}

public void setPassword(String pw)
{
    pwTB.sendKeys(pw);
}

public void clickOnChkBox()
{
    chkBox.click();
}

public void clickOnLoginButton()
{
    loginButton.click();
}

public void clickOnLogoutButton()
{
    logoutButton.click();
}

public void clickOnActiTimeInc()
{
    actiTimeInc.click();
}
}

```

Note: We can't execute in above class directly beacuse no main method.
for execution purpose we develop another class.

```

package EncapsulationInSelenium;

import java.util.concurrent.TimeUnit;

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

```

```

import POMInSelenium.ActiTimeAppInSelenium_POM;

public class ExecuteScript2_Encapsulation
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(5000);
        driver.get("http://localhost/login.do");

        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

        ActiTimeAppInSelenium_Encapsulation a1=new
ActiTimeAppInSelenium_Encapsulation(driver);

        //driver.navigate().refresh();

        Thread.sleep(2000);
        a1.setUserName("admin");

        Thread.sleep(2000);
        a1.setPassword("manager");

        Thread.sleep(2000);
        a1.clickOnChkBox();

        Thread.sleep(2000);
        a1.clickOnLoginButton();

        Thread.sleep(5000);
        a1.clickOnLoginButton();

        Thread.sleep(5000);
        a1.clickOnActiTimeInc();

        Thread.sleep(5000);
        driver.quit();

    }
}

```

NOTE: If we execute the above Script, we get StaleElementReferenceException or NoSuchElementException.

If we uncomment this line --> //driver.navigate().refresh(); --> we will get StaleElementReferenceException

If we don't uncomment also --> page gets loaded --> we will get NoSuchElementException

****coz the page gets loaded once it opens, so that address of the elements will change/expiry. **imp**

Reference --> address of the element
Stale --> invalid or expired

Ex:

Script-1:

public class Demo {

private WebElement unTB;

public Demo(WebDriver driver)
{
unTB=driver.findElement(By.id("username"));
}

public void setUsername()
{
unTB.sendKeys("Anil");
}
}

Note: We can't execute in above class directly beacuse no main method.
for execution purpose we develop another class.

Ex:

Script-2:

public class Demo1 {

public static void main(String[] args) {

WebDriver driver=new FirefoxDriver();

driver.get("http://localhost/login.do");

Demo d=new Demo(driver);
d.setUsername();

-->
public class Demo
{
private WebElement unTB;
private WebElement pwTB;
private WebElement login;

public Demo(WebDriver driver)
{
unTB=driver.findElement(By.id("username"));

```

pwTB=driver.findElement(By.name("pwd"));
login=driver.findElement(By.id("loginButton"));
    }
    public void setUserName(String un,String pw)
    {
        unTB.sendKeys(un);
        pwTB.sendKeys(pw);
        login.click();
    }
}

```

```

-----

-->
public class Demo1
{
    public static void main(String[] args)
    {

        WebDriver driver=new FirefoxDriver();

        driver.get("http://localhost/login.do");

        Demo d=new Demo(driver);

            d.setUserName("admin", "manager");

    }
}

```

```

-----

--> Modified code:
-----

```

```

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

    driver.get("http://localhost/login.do");

    Demo d=new Demo(driver);

    driver.navigate().refresh();
    d.setUserName();
}

```

```

-----

```

Note: If we execute above code we get StaleElementReferenceException ; coz we are using refresh().

Reference --> address of the element
 Stale --> invalid or expired

->In the above example selenium is finding the username textbox and starting its address in unTB is [@a1]

->Then page is getting refreshed username text box will be reloaded but into a different address [@x1]

Address changed from @a1 to @x1 hence we will get
StaleElementReferenceException.

->In order to handle this we use one of the Java design pattern called POM

=====

21.POM (Page Object Model):

1)What is Page Object Module[POM]? **imp

:- it is java desgin pattern used to develop text webpages.

POM stands for Page Object Model. It is a design pattern for creating an Object Repository for web UI elements.

Page Object Model, also known as POM, is a design pattern in Selenium that creates an object repository for storing all web elements.

It is useful in reducing code duplication and improves test case maintenance. In Page Object Model, consider each web page of an application as a class file.

2)What is Annoation? **imp

:- We use in pom class to declare elements.

```
Syntax:-    FindBy Annotation
             @FindBy(locator="value")

             private WebElement elementName;
```

3)What is the use of Initelements? **imp

:- it is used to intialize pom class.

Syntax:- it is used as object for pom class

POM is similar to encapsulation concept

--->Selenium code to enter username.

```
driver.findElement(By.id("Username")).sendKeys("Prasanna");
```

--->Selenium code to enter username using POM concept:-

For any given data member(variable) we perform following 3 steps:-

```
->Declaration Ex: private int empNo; private WebElement unTB;

->Initialization Ex: empNo=1001;
    unTB=driver.findElement(By.id("Username")); //Constructor

->Utilization unTB.sendKeys("Anil"); //Method
```

Ex:

class POM

```
{

    //Declaration

    @FindBy(locators="value")
    private WebElement unTB;

    //Initialization [Inside the Constructor]

    public POM(WebDriver driver)
    {
        Pagefactory.initElements(driver,this);
    }

    //Utilization [Inside the Method]

    public void setUsername(String un)
    {
        unTB.sendKeys(un);
    }

    //Execute Script
```

Class ExecuteScript1

```
{

    public static void main(String []args)
    {

    }

}
```

```
}
```

Script-1:

```
-----  
package POMInSelenium;  
  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.support.FindBy;  
import org.openqa.selenium.support.PageFactory;  
  
public class FaceBookAppInSelenium_POM  
{  
    //Declaration  
  
    @FindBy(id="email")  
    private WebElement unTB;  
  
    @FindBy(id="pass")  
    private WebElement pwTB;  
  
    @FindBy(name="login")  
    private WebElement loginButton;  
  
    //Initialization [Inside the Constructor]  
  
    public FaceBookAppInSelenium_POM(WebDriver driver)  
    {  
        PageFactory.initElements(driver, this);  
    }  
  
    //Utilization[Inside the Method]  
  
    public void setUsername(String un)  
    {  
        unTB.sendKeys(un);  
    }  
  
    public void setPassword(String pw)  
    {  
        pwTB.sendKeys(pw);  
    }  
  
    public void clickOnLoginButton()  
    {  
        loginButton.click();  
    }  
}  
  
package POMInSelenium;
```

```

import java.util.concurrent.TimeUnit;

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

public class ExecuteScript1_POM
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(5000);
        driver.get("https://www.facebook.com/");

        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

        FaceBookAppInSelenium_POM f1=new FaceBookAppInSelenium_POM(driver);

        Thread.sleep(3000);

        driver.navigate().refresh();

        Thread.sleep(3000);

        f1.setUserName("AnilKumar");

        f1.setPassWord("1234567");

        f1.clickOnLoginButton();

        Thread.sleep(5000);
        driver.quit();
    }
}

```

Script-2:

```

package POMInSelenium;

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

public class ActiTimeAppInSelenium_POM
{
    //Declaration

    @FindBy(id="username")
    private WebElement unTB; //private-->within class

    @FindBy(name="pwd")

```

```

private WebElement pwTB;

@FindBy(id="keepLoggedInCheckBox")
private WebElement chkBox;

@FindBy(xpath="//div[.='Login ']")
private WebElement loginButton;

@FindBy(id="logoutLink")
private WebElement logoutButton;

@FindBy(xpath="//a[.='actiTIME Inc. ']")
private WebElement actiTimeInc;

//Initialization [Inside the Constructor]

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

//Utilization[Inside the Method]

public void setUsername(String un)
{
    unTB.sendKeys(un);
}

public void setPassword(String pw)
{
    pwTB.sendKeys(pw);
}

public void clickOnChkBox()
{
    chkBox.click();
}

public void clickOnLoginButton()
{
    loginButton.click();
}

public void clickOnLogoutButton()
{
    logoutButton.click();
}

public void clickOnActiTimeInc()
{
    actiTimeInc.click();
}
}

```

```

package POMInSelenium;

```

```

import java.util.concurrent.TimeUnit;

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

public class ExecuteScript2_POM
{
    public static void main(String[] args) throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
        WebDriver driver=new ChromeDriver();
        driver.manage().window().maximize();

        Thread.sleep(5000);
        driver.get("http://localhost/login.do");

        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

        ActiTimeAppInSelenium_POM a1=new ActiTimeAppInSelenium_POM(driver);

        Thread.sleep(3000);

        driver.navigate().refresh();

        Thread.sleep(3000);

        a1.setUserName("admin");

        a1.setPassword("manager");

        a1.clickOnChkBox();

        a1.clickOnLoginButton();

        a1.clickOnLogoutButton();

        a1.clickOnActiTimeInc();

        Thread.sleep(5000);
        driver.quit();
    }
}

```

Note: The class which will test the application by executing the POM methods is called

as "Test Class"

In selenium we develop two types of classes

1)POM class

2)Test class

```
=====
=====

22.TestNG:
```

```
-----
    Test next generation is a unit testing frame work.
```

```
Adv: ***vvvvimp
-----
```

1)For every manual testcase we develop testclass that is if we have 500 manual tetcases we will develop 500 test class. --> TestNG class -->package com.acti_time.script;

2)Executing 500 test clases individually is a monotonous job to execute multiple test class and generate execution result we use TestNG. --> TestNG Suite

3)Even though if the execution fails while running any class, the rest of the code will execute. --> assert

5)TestNG is availble as plugin for eclipse.

Features of Using TestNG are: *****vvvvimp

- 1.Different Types of Assertions
- 2.Run tests in Parallel
- 3.Make tests dependent on one another
- 4.Prioritizing tests
- 5.Grouping of tests
- 6.Data-Driven Testing
- 7.Reporting
- 8.Parameterization

```
-----

-->Steps to install TestNG:
```

Step-1:

```
-----
->In eclipse go to help
      |
Select eclipse marketplace
      |
Search for TestNG
      |
Click on Install button of TestNG
      |
Follow the default instructions till finish
      |
After installation it re opens the eclipse
```

```
-----

Step-2:
```

```
-----
-->Associate TestNG:
```

- 1)Right click on the java project and select properties.
- 2)Click on java Build Path
- 3)Go to Libraries tab
- 4)Add Library
- 5)Click on TestNG
- 6)Next-Next-Finish-Ok

TestNG Class:

->It is a java class which has test method instead of main method.

->Any method developed using @Test [Test Annotation] is called as test method.

->While developing the TestNG class we do not use default package, main method and S.o.p

Instance of main method --> @Test
s.o.p --> Reporter

Ex:-package corejava;

```
import org.testng.Reporter;
import org.testng.annotations.Test;
public class Test10
{
    @Test
    public void testDemo()
    {
        Reporter.log("test Result",true);
    }
    @Test
    public void testDemo1()
    {
        Reporter.log("Hello",false);
    }
}
```

-->After the execution, TestNG creates 'test-output' inside the java project
[Refresh java project]

-->Where the output stored in html format [emailble-report.html]

Right click
|
Open with
|
Web browser

TestNG Suite:

It's an xml file which is used to run multiple TestNG classes in order to create it right click on java project go to TestNG .

TestNG select to convert to TestNG and Finish.

Execute TestNG .xml

Go to Run As

1 TestNG Suite

-->Content of TestNG .Suite in xml:-

```
<suite name="Suite" parallel="none">
<test name="Test">
  <classes>
    <class name="corejava.Test10"/>
  </classes>
</test>
</suite>
```

Interview Questions:

1)Can we develop multiple test methods in same TestNG class?

:-Yes.

2)What is the order of execution?

:-If TestNG class multiple test methods present in TestNG class
->Alphabetical order

3)How do you execute test method with the required order?

:-Using Priority

Note: Default priority is zero[0]

->We can use positive or negative numbers

->but variables and decimals are not allowed

->It always execute ascending order of Priority

4)How do you execute test method multiple times?

:-using invocationCount

Ex:-@Test(invocationCount=2)

Note:

->Default invocationCount is 1. If it is 0 or negative number it will not execute the respective TestMethod.

->Variable & decimals numbers are not allowed.

5)If there is a method in TestNG method without any Annotation, will it executed?

:-No

Scripts:

Script-1:

```
package Selenium_TestNG;

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

public class Demo1
{
    @Test
    public void testCase1()
    {
        Reporter.log("TestCase-1", true);
    }

    @Test
    public void testCase2()
    {
        Reporter.log("TestCase-2", true);
    }

    @Test
    public void testCase3()
    {
        Reporter.log("TestCase-3", true);
    }
}
```

Script-2:

```
package Selenium_TestNG;

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

```

public class Demo2
{
    @Test
    public void createTestCase1()
    {
        Reporter.log("CreateTestCase-1", true);
    }

    @Test
    public void editTestCase2()
    {
        Reporter.log("EditTestCase-2", true);
    }

    @Test
    public void deleteTestCase3()
    {
        Reporter.log("DeleteTestCase-3", true);
    }
}

```

Script-3:

```

package Selenium_TestNG;

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

public class Demo3
{
    @Test(priority = 0)
    public void createTestCase1()
    {
        Reporter.log("CreateTestCase-1", true);
    }

    @Test(priority = 1)
    public void editTestCase2()
    {
        Reporter.log("EditTestCase-2", true);
    }

    @Test(priority = 2)
    public void deleteTestCase3()
    {
        Reporter.log("DeleteTestCase-3", true);
    }
}

```

Script-4:

```

package Selenium_TestNG;

```

```

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

public class Demo4
{
    @Test(priority = 1)
    public void createTestCase1()
    {
        Reporter.log("CreateTestCase-1", true);
    }

    @Test(priority = 2)
    public void editTestCase2()
    {
        Reporter.log("EditTestCase-2", true);
    }

    @Test(priority = 0)
    public void deleteTestCase3()
    {
        Reporter.log("DeleteTestCase-3", true);
    }
}

```

Script-5:

```

package Selenium_TestNG;

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

public class Demo5
{
    @Test(priority = 1) //priority starts from 0
    public void createTestCase1()
    {
        Reporter.log("CreateTestCase-1", true);
    }

    @Test(priority = 1)
    public void editTestCase2()
    {
        Reporter.log("EditTestCase-2", true);
    }

    @Test(priority = 1)
    public void deleteTestCase3()
    {
        Reporter.log("DeleteTestCase-3", true);
    }
}

```

Script-6:

```
-----  
package Selenium_TestNG;  
  
import org.testng.Reporter;  
import org.testng.annotations.Test;  
  
public class Demo6  
{  
    @Test(priority = 1)  
    public void createTestCase1()  
    {  
        Reporter.log("CreateTestCase-1", true);  
    }  
  
    public void editTestCase2()  
    {  
        Reporter.log("EditTestCase-2", true);  
    }  
  
    @Test(priority = 1)  
    public void deleteTestCase3()  
    {  
        Reporter.log("DeleteTestCase-3", true);  
    }  
}
```

Script-7:

```
-----  
package Selenium_TestNG;  
  
import org.testng.Reporter;  
import org.testng.annotations.Test;  
  
public class Demo7  
{  
    @Test(priority = 1, enabled = false)  
    public void createTestCase1()  
    {  
        Reporter.log("CreateTestCase-1", true);  
    }  
  
    @Test(priority = 1) //enabled = true  
    public void editTestCase2()  
    {  
        Reporter.log("EditTestCase-2", true);  
    }  
  
    @Test(priority = 1, enabled = false)  
    public void deleteTestCase3()  
    {  
        Reporter.log("DeleteTestCase-3", true);  
    }  
}
```

Script-8:

```
package Selenium_TestNG;

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

public class Demo8
{
    @Test(priority = 0, invocationCount = 100) //invocationCount starts from 1
    public void createTestCase1()
    {
        Reporter.log("CreateTestCase-1", true);
    }

    @Test(priority = 1, invocationCount = 20)
    public void editTestCase2()
    {
        Reporter.log("EditTestCase-2", true);
    }

    @Test(priority = 2, invocationCount = 50)
    public void deleteTestCase3()
    {
        Reporter.log("DeleteTestCase-3", true);
    }
}
```

Script-9:

```
package Selenium_TestNG;

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

public class Demo9
{
    @Test
    public void createTestCase1()
    {
        Reporter.log("CreateTestCase-1", true);
    }

    @Test(priority = 1)
    public void editTestCase2()
    {
        Reporter.log("EditTestCase-2", true);
    }

    @Test(priority = 1)
    public void deleteTestCase3()
    {
        Reporter.log("DeleteTestCase-3", true);
    }
}
```

Script-10:

```
-----  
package Selenium_TestNG;  
  
import org.testng.Reporter;  
import org.testng.annotations.Test;  
  
public class Demo10  
{  
    @Test  
    public void loginCreateTestCase1()  
    {  
        Reporter.log("LoginCreateTestCase-1", true);  
    }  
  
    @Test(dependsOnMethods = {"loginCreateTestCase1"})  
    public void composeEditTestCase2()  
    {  
        Reporter.log("ComposeEditTestCase-2", true);  
    }  
  
    @Test(dependsOnMethods = {"loginCreateTestCase1","composeEditTestCase2"} )  
    public void sentMailDeleteTestCase3()  
    {  
        Reporter.log("SentmailDeleteTestCase-3", true);  
    }  
}
```

Script-11:

```
-----  
package Selenium_TestNG;  
  
import org.testng.Assert;  
import org.testng.Reporter;  
import org.testng.annotations.Test;  
  
public class Demo11  
{  
    @Test  
    public void loginCreateTestCase1()  
    {  
        Reporter.log("LoginCreateTestCase-1", true);  
    }  
  
    @Test(dependsOnMethods = {"loginCreateTestCase1"})  
    public void composeEditTestCase2()  
    {  
        Reporter.log("ComposeEditTestCase-2", true);  
        //Assert.fail();  
    }  
  
    @Test(dependsOnMethods = {"loginCreateTestCase1","composeEditTestCase2"} )  
    public void sentMailDeleteTestCase3()  
    {
```

```

        Reporter.log("SentmailDeleteTestCase-3", true);
    }
}

```

-

Important Annotation Of TestNG --> imp

1)@Test:-

This represents Test method of TestNG class.

2)@BeforeMethod:-

The annotated method will be run before each test method.

3)@AfterMethod:-

The annotated method will be run after each test method.

4)@BeforeClass:-

The annotated method will be run before the first test method in the current class is invoked.

5)@AfterClass:-

The annotated method will be run after all the test methods in the current class have been run.

Script-1:

```

-----
package Selenium_TestNG;

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 TestNG_AnnotationEx1
{
    @BeforeClass
    public void openApp()
    {
        Reporter.log("OpenApp", true);
    }

    @AfterClass
    public void closeApp()
    {
        Reporter.log("CloseApp", true);
    }
}

```



```

    }

    @BeforeMethod
    public void login()
    {
        Reporter.log("Login", true);
    }

    @AfterMethod
    public void logout()
    {
        Reporter.log("logout", true);
    }

    @Test
    public void testCase1()
    {
        Reporter.log("TestCase-1", true);
    }

    @Test
    public void testCase2()
    {
        Reporter.log("TestCase-2", true);
    }

    @Test
    public void testCase3()
    {
        Reporter.log("TestCase-3", true);
    }
}

```

Script-2:

```

package Selenium_TestNG;

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 TestNG_AnnotationEx2
{
    @BeforeClass
    public void openApp()
    {
        Reporter.log("OpenApp", true);
    }

    @AfterClass
    public void closeApp()
    {
        Reporter.log("CloseApp", true);
    }
}

```

```

@BeforeMethod
public void login()
{
    Reporter.log("Login",true);
}

@AfterMethod
public void logout()
{
    Reporter.log("logout",true);
}

@Test
public void createTestCase1()
{
    Reporter.log("CreateTestCase-1", true);
}

@Test
public void editTestCase2()
{
    Reporter.log("EditTestCase-2", true);
}

@Test
public void deleteTestCase3()
{
    Reporter.log("DeleteTestCase-3", true);
}
}

```

Note:

->We can have multiple before method,after method,before class,after class methods but they will be executed alphabetical order.

->Actual testing steps should be written inside test method, other annoation should be used to achieve pre-conditions & post-conditions.

Inheritance In TestNG --> imp

@BeforeSuite:-

The annotated method will be run before all tests in this suite have run.

@AfterSuite:-

The annotated method will be run after all tests in this suite have run.

@BeforeTest:-

The annotated method will be run before any test method belonging to the classes inside the tag is run.

@AfterTest:-

The annotated method will be run after all the test methods belonging to the classes inside the tag have run.

Hierarchy of the TestNG Annotations: **vvimp

@BeforeSuite
@BeforeTest
@BeforeClass
@BeforeMethod
@Test
@AfterMethod
@AfterClass
@AfterTest
@AfterSuite

Ex:

-->BaseTest.Java

```
import org.testng.Reporter;
import org.testng.annotations.AfterClass;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.AfterSuite;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.BeforeSuite;
import org.testng.annotations.BeforeTest;

abstract public class BaseTest
{
    @BeforeSuite
    public void bs()
    {
        Reporter.log("@BeforeSuite", true);
    }
    @AfterSuite
    public void as()
    {
        Reporter.log("@AfterSuite", true);
    }

    @BeforeTest
    public void bt()
    {
        Reporter.log("@BeforeTest", true);
    }
    @AfterTest
    public void at()
    {
```

```

        Reporter.log("@AfterTest", true);
    }

    @BeforeClass
    public void bc()
    {
        Reporter.log("@BeforeClass", true);
    }
    @AfterClass
    public void ac()
    {
        Reporter.log("@AfterClass", true);
    }

    @BeforeMethod
    public void bm()
    {
        Reporter.log("@BeforeMethod", true);
    }
    @AfterMethod
    public void am()
    {
        Reporter.log("@AfterMethod", true);
    }
}

----
package Selenium_TestNG;

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

public class InheritanceEx_In_TestNG extends BaseTest_Ex
{
    @Test
    public void testCase1()
    {
        Reporter.log("testCase1", true);
    }
}

```

o/p:

```

----
@AfterSuite
@BeforeTest
BeforeClass
@BeforeMethod
testCase1
@AfterMethod
@AfterClass
@AfterTest
PASSED: testCase1

```

=====

Default test
Tests run: 1, Failures: 0, Skips: 0

=====

@Aftersuite

=====

Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0

=====

Script:

```
-----  
package Selenium_TestNG;  
  
import org.testng.Reporter;  
import org.testng.annotations.AfterClass;  
import org.testng.annotations.AfterMethod;  
import org.testng.annotations.AfterSuite;  
import org.testng.annotations.BeforeClass;  
import org.testng.annotations.BeforeMethod;  
import org.testng.annotations.BeforeSuite;  
  
public abstract class BaseTest  
{  
    @BeforeSuite  
    public void bs()  
    {  
        Reporter.log("@BeforeSuite",true);  
    }  
  
    @AfterSuite  
    public void as()  
    {  
        Reporter.log("@AfterSuite", true);  
    }  
  
    @BeforeClass  
    public void openApp()  
    {  
        Reporter.log("OpenApp",true);  
    }  
  
    @AfterClass  
    public void closeApp()  
    {  
        Reporter.log("CloseApp",true);  
    }  
  
    @BeforeMethod  
    public void login()  
    {  
        Reporter.log("Login",true);  
    }  
  
    @AfterMethod  
    public void logout()  
    {  

```

```

        Reporter.log("logout",true);
    }
}

-----

package Selenium_TestNG;

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 Inheritance_In_TestNG extends BaseTest
{
    @Test
    public void createTestCase1()
    {
        Reporter.log("CreateTestCase-1", true);
    }

    @Test
    public void editTestCase2()
    {
        Reporter.log("EditTestCase-2", true);
    }

    @Test
    public void deleteTestCase3()
    {
        Reporter.log("DeleteTestCase-3", true);
    }
}

```

o/p:

```

@BeforeSuite
OpenApp
Login
CreateTestCase-1
logout
Login
DeleteTestCase-3
logout
Login
EditTestCase-2
logout
CloseApp
PASSED: deleteTestCase3
PASSED: createTestCase1
PASSED: editTestCase2

```

```

=====
Default test
Tests run: 3, Failures: 0, Skips: 0

```

=====

@AfterSuite

=====

Default suite

Total tests run: 3, Passes: 3, Failures: 0, Skips: 0

=====

Note:

In realtime every test will have some common steps.

->Open the browser

->Instead of writing code in every class we use inheritance concept that is Parent class and all the class will extends it.

->We can't run directly the parent class [BaseTest] beacuse it is incomplete [needs test method] hence make this class as abstract class.

1)How to create dependency in TestNG?

: -Using dependsOnMethods

Note:

-> Dependence means execution of a test method only after successfully execution of some other Test method

-> If test method has both dependency & priority testNG uses dependency

2)What happens if two methods are dependent on each other?

: -We get TestNG Exception error is cyclic dependence Exception.

-->

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

```
public class DemoTest
{
    @Test
    public void createUser()
    {
        Reporter.log("CreateUser",true);
        Assert.assertEquals("abc", "xyz");
    }

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

```
}  
}
```

O/p:

CreateUser

FAILED: createUser

java.lang.AssertionError: expected [xyz] but found [abc]

SKIPPED: deleteUser

java.lang.Throwable: Method DemoTest.deleteUser()[pri:0, instance: BloodBank.DemoTest@2eeee9593] depends on not successfully finished methods.

Note:

->In the above example delete user method depends on status of create user method, If create user method is pass then only it will execute delete user method.

->If createUser method is failed it will not execute deleteuser method[skip]

->If we use both priority & dependency that is always consider dependency. it can create multiple dependency using comma as a separator.

dependsOnMethods={"createUser","login"}

The dependent method will be executed only if all the specified methods are successfully executed.

Note:

Priority ****imp

If multiple methods have same priority then those method will be executed in alphabetical order.

-->

@Test

public void createProduct()

```
{  
    Reporter.log("Product to Create?",true);  
    Scanner sc=new Scanner(System.in);  
    String expected = sc.nextLine();  
    Assert.assertEquals("Pen", expected);  
}
```

@Test(dependsOnMethods={"createProduct"})

public void deleteProduct()

```
{  
    Reporter.log("DeleteProduct",true);  
}
```

--> Scenario-1:

Product to Create?

Pen

DeleteProduct

PASSED: createProduct

PASSED: deleteProduct

```
=====
      Default test
```

```
      Tests run: 2, Failures: 0, Skips: 0
=====
```

-->Scenario-2:

Product to Create?

Idly

FAILED: createProduct

java.lang.AssertionError: expected [Idly] but found [Pen]

SKIPPED: deleteProduct

java.lang.Throwable: Method DemoTest.deleteProduct()[pri:0,
instance:BloodBank.DemoTest@2eeee9593] depends on not successfully
finished methods

Interview Quetions: **vvimp

tesng.xml --> heart of TestNG

1)Running all classes in a package in TestNG: ***imp

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="Test">
    <classes>
      <class name="Selenium_TestNG.TestNG_AnnotationEx2"/>
      <class name="Selenium_TestNG.Demo8"/>
      <class name="Selenium_TestNG.Demo7"/>
      <class name="Selenium_TestNG.InheritanceEx_In_TestNG"/>
      <class name="Selenium_TestNG.Demo9"/>
      <class name="Selenium_CapturingScreenshot.VerifyLoginPage_Screenshot"/>
      <class name="Selenium_TestNG.Demo4"/>
      <class name="Selenium_TestNG.Demo3"/>
      <class name="Selenium_TestNG.Demo6"/>
      <class name="Selenium_TestNG.Demo5"/>
      <class name="Selenium_TestNG.Demo2"/>
      <class name="Selenium_TestNG.Demo1"/>
      <class name="Selenium_TestNG.Demo11"/>
      <class name="Selenium_TestNG.Demo10"/>
      <class name="Selenium_TestNG.VerifyLoginPage_SoftAssert"/>
      <class name="Selenium_TestNG.VerifyLoginPage_Assert"/>
      <class name="Selenium_TestNG.Inheritance_In_TestNG"/>
      <class name="Selenium_TestNG.TestNG_AnnotationEx1"/>
    </classes>
  </test>
</suite>
```

```

    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->

```

2)Running more than 1 package in TestNG: **imp

```

    <suite name="Suite1">
      <test name="Regression1">
        <packages>
          <package name="Selenium_TestNG"/>
          <package name="Selenium_CapturingScreenshot"/>
        </packages>
      </test>
    </suite>

```

3)Running required packages & classes in TestNG: **imp

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="Test">
    <classes>

      <class name="Selenium_TestNG.TestNG_AnnotationEx2"/>
      <class name="Selenium_TestNG.InheritanceEx_In_TestNG"/>
      <class name="Selenium_TestNG.VerifyLoginPage_Assert"/>
      <class name="Selenium_TestNG.Inheritance_In_TestNG"/>
      <class name="Selenium_TestNG.TestNG_AnnotationEx1"/>
      <class name="Selenium_CapturingScreenshot.VerifyLoginPage_Screenshot"/>

    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->

```

4)Inside specific class/classes, specific method/methods should run: **imp
(Ex: Inside the two different class, one one method should run???)

:- include method --> including
exclude method --> excluding

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
  <test thread-count="5" name="Test">
    <classes>
      <class name="Selenium_TestNG.Demo1"/>
        <methods>
          <include name="testCase1"/>
        </methods>
      </class>

      <class name="Selenium_CapturingScreenshot.VerifyLoginPage_Screenshot"/>

```

```

        <methods>
        <include name="signInSubmit"/>
        </methods>
    </class>

</classes>
</test> <!-- Test -->
</suite> <!-- Suite -->

```

Verification vvimp

->Comparing actual result with expected result is called as verification.

->In TestNG we do this using Assert class

->Assert class has the static methods such as:-

- 1)Assert.assertEquals
- 2)Assert.assertNotEquals
- 3)Assert.assertTrue
- 4)Assert.assertNull
- 5)Assert.assertNotNull
- 6)Assert.assertSame
- 7)Assert.assertNotSame
- 8)Assert.fail
- 9)Assert.assertFalse

Interview Questions:**imp

1)Write script to open the browser and enter the url, verify the loginpage displayed or not using TestNG?

:-

-->Assert / Hard Assert

```
package Selenium_TestNG;
```

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.Assert;
import org.testng.annotations.Test;
import org.testng.asserts.SoftAssert;
```

```
public class VerifyLoginPage_Assert
{
```

```
    @Test
    public void verifyValidLoginPage() throws InterruptedException
    {
```

```
System.setProperty("webdriver.chrome.driver", "C:\\Users\\Admin\\Downloads\\Program
files\\chromedriver_win32\\chromedriver.exe");
```

```

        WebDriver driver=new ChromeDriver();

        Thread.sleep(3000);
        driver.get("http://localhost/login.do");

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

        Thread.sleep(3000);

        String loginPageTitle = driver.getTitle();

                                //actual - Selenium, expected - Manual Test Case

        Assert.assertEquals(loginPageTitle, "aaaaaaaaaaaaa"); //static method

        Thread.sleep(2000);
        driver.close();

    }

    @Test
    public void verifyValidLoginPage1() throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");

        WebDriver driver=new ChromeDriver();

        Thread.sleep(3000);
        driver.get("http://localhost/login.do");

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

        Thread.sleep(3000);

        String loginPageTitle = driver.getTitle();

                                //expected - Manual Test Case, actual -
Selenium

        Assert.assertEquals(loginPageTitle, "actiTIME - Login"); //static
method

        Thread.sleep(2000);
        driver.close();

    }

}

```

Note:

- > If comparision fails then TestNG will not execute remaining statements of current test methods, other methods executed as it is.
- > In order to continue the execution even if comparision fails we can use soft Assert, which has the nonstatic method.

-->Soft Assert

```
package Selenium_TestNG;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.Assert;
import org.testng.annotations.Test;
import org.testng.asserts.SoftAssert;

public class VerifyLoginPage_SoftAssert
{
    @Test
    public void verifyValidLoginPage() throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");

        WebDriver driver=new ChromeDriver();

        Thread.sleep(3000);
        driver.get("http://localhost/login.do");

        SoftAssert soft=new SoftAssert();

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

        Thread.sleep(3000);

        String loginPageTitle = driver.getTitle();

        //expected - Manual Test Case, actual - Selenium

        soft.assertEquals(loginPageTitle, "aaaaaaaaaaaa"); //non-static method
or instance method

        Thread.sleep(2000);
        driver.close();

        soft.assertAll();
    }

    @Test
    public void verifyValidLoginPage1() throws InterruptedException
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");

        WebDriver driver=new ChromeDriver();

        Thread.sleep(3000);
        driver.get("http://localhost/login.do");

        SoftAssert soft=new SoftAssert();

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

```

        Thread.sleep(3000);

        String loginPageTitle = driver.getTitle();

        //expected - Manual Test Case, actual - Selenium

        soft.assertEquals(loginPageTitle, "actiTIME - Login"); //non-static
method or instance method

        Thread.sleep(2000);
        driver.close();

        soft.assertAll();

    }
}

```

Note:

-> It is mandatory to call assertAll(), this method updates the status on the console window (PASS or FAIL).

-> If any failure then here also TestNG will not execute any statements which are present after assertAll method. Hence assertAll() is the last statement.

*****vrimp*****

Hard Assert

- 1) Remaining statements of current Test method will not be executed if comparison fails.
- 2) All the methods are static instance methods
- 3) We do not call assertAll()
- 4) We use it for verification of basic or critical features.
[Smoke Testing]

Soft Assert

- | | |
|--|---|
| <ol style="list-style-type: none"> 1) Remaining statements of current Test method will be executed if comparison fails. 2) All the methods are Non Static / instance methods 3) We should call assertAll() 4) Use to verify non critical features. | <ol style="list-style-type: none"> 1) Remaining statements of current Test method will be executed if comparison fails. 2) All the methods are Non Static / 3) We should call assertAll() 4) Use to verify non critical features. |
|--|---|

=====

23. Capturing ScreenShot In Selenium:

```
package Selenium_CapturingScreenshot;
```

```
import java.util.concurrent.TimeUnit;
```

```

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

public class VerifyLoginPage_Screenshot
{
    public static WebDriver driver;

    @Test(priority = 0)
    public void loginPage()
    {
        System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");

        driver=new ChromeDriver();

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

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

        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
    }

    @Test(priority = 1)
    public void moveToAccountsAndLists()
    {
        Actions action=new Actions(driver);

        WebElement move = driver.findElement(By.id("nav-link-accountList"));

        action.moveToElement(move).perform();

        Assert.assertEquals("abc", "xyz");
    }

    @Test(priority = 2)
    public void clickOnSignInButton()
    {
        driver.findElement(By.xpath("//div[@id='nav-al-signin']/div/a//span")).click();

        Assert.assertEquals("abc", "xyz");
    }

    @Test(priority = 3)
    public void enterLoginCredentials()
    {
        driver.findElement(By.id("ap_email")).sendKeys("anil.krtg@gmail.com");

        driver.findElement(By.id("continue")).click();
    }
}

```

```

        driver.findElement(By.id("ap_password")).sendKeys("1234567");

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

    }

}

-----

package Selenium_CapturingScreenshot;

import java.io.File;
import java.io.IOException;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.testng.ITestResult;
import org.testng.TestListenerAdapter;

public class CapturingScreenshot extends TestListenerAdapter
{
    public void onTestFailure(ITestResult tr)
    {
        File scrFile =
        ((TakesScreenshot)VerifyLoginPage_Screenshot.driver).getScreenshotAs(OutputType.FILE);
        //Now you can do whatever you need to do with
        it, for example copy somewhere

        try
        {
            FileUtils.copyFile(scrFile, new File("C:\\Users\\Admin\\Desktop\\
Amzon_Failed_Test_Script_Screenshot\\"+tr.getName()+".png"));
        }
        catch (IOException e)
        {

            e.printStackTrace();

        }

    }
}

```

TestNG Suite:-

```

-----
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">

<listeners>

<listener
class-name="Selenium_CapturingScreenshot.CapturingScreenshot"></listener>

```



```

</listeners>

<test thread-count="5" name="Test">
  <classes>

    <class name="Selenium_CapturingScreenshot.VerifyLoginPage_Screenshot"/>

  </classes>

</test> <!-- Test -->
</suite> <!-- Suite -->

```

```

=====
=====

```

24.Framework:

There are different Framework

```

-----
--> Method driven Framework
--> Module driven Framework
--> Data driven Framework
--> BDD Framework
--> Hybrid driven Framework

```

Automation Project Using Framework

It is a standard rules, guidelines and best practices which should be followed by all the automation engineers in the team, while converting manual testcases in to automation scripts, we should use automation framework to have consistency.

We have 3 states

```

-----
-->1.Framework Design
-->2.Framework Implementation
-->3.Framework Execution

```

1.Framework Design

```

=====

```

This is the initial stage where automation lead or manager designs the specified files & folder structure based on the experience.

EX:

```

---
E-Commerce_Domain
|
ActiTime_Automation_Project
|
src
|
  com.acti_time.pom
  com.acti_time.script
  com.acti_time.generics
jars
  |--selenium, poi, Log4j etc
driver
  |--chrome, firefox, ie, edge, opera etc
TestNG
Test_OutPut

```

Ex:

Step1:-Go to required location

```

      |
      | Example D drive
      |
      | Create folder ERP
      |
      | In eclipse File ->Switch workspace ->Other
      |
      | Browser selected above created folder
      |
      | Click on ok
      |
      | It will restart the eclipse
      |
      | Create Java project with the name Amazon
      |
      | Associate TestNG follow old steps
      |
      | Create two packages under src
      |
      | with the name->pom(small letter)-<com.amazon.pom
      |                                     ->script(small letter)-<com.amazon.script
      |
      | Create folder with name jars under the java project
      |
      | copy paste selenium jar file and build path and add build
      |

```

create folder with name driver under the java project
|
copy paste chromedriver.exe

```
-----  
D  
|-->ERP  
    |-->Amazon[Java project]  
        |-->src  
            |-->pom  
            |-->script  
        |  
        |-->generics  
        |  
        |-->jar  
            |-->selenium server standalone software  
        |-->driver  
            |-->chromedriver.exe  
        |  
        |-->geckodriver.exe  
            |-->IEDriverServer.exe  
-----
```

2.Framework Implementation =====

This is the second stage where we convert manual testcase in to automation scripts by using the framework it's designed by lead or the manager.

Note:-

- > In automation the reference document is manual testcase.
- > We only automate Regression Testcases, which are provided by ManualTeam.
- > We may not be able to automated all the regression testcases, present in the Regression suite.

Following feautres we can't automated

- ```

->Email notification
->SMS notification
->OTP
->Audio,Video[Multimedia Testing]
->Animation
->Gaming application
->Printing
->CAPTCHA[Credit card,Debit card, Axxess card,Swipping card]

```

Sample TestCase:-

-----

1)PreCondition:-

Login page should be displayed.

2)PostCondition:-  
     Application should be closed.

3)TestCaseNo1:-  
     Verify version.  
     Step no1:-Verify that version of the application actiTime 2016.1

4)TestCaseNo2:-  
     Verify valid login.  
     Step no1:-Enter valid UN  
                 Enter valid PW  
                 Click on Login  
                 Verify that home page is displayed  
                     [actiTime Enter Time-Track]  
                 Click Logout  
                 Verify that login page displayed.  
                     [actiTime Login]

5)TestCase No 3:-  
     Verify Invalid Login  
     Step no1:-Enter Invalid UN  
                 Enter Invalid PW  
                 Click on Login  
                 Verify that error is displayed

-----  
 Developing POM class:-

-----  
     Guideline:-develop pom class

- 1)Execute the Testcase manually atleast once which gives more clarity on the steps.
- 2)While executing the testcase notedown the page element at the actions.

-----  
 Page No 1:-actiTime Login  
     ->Elements  
     ->Version  
     ->UN  
     ->PW  
     ->Login Button  
     ->Error msg

-----  
 Actions:- Verify the version  
           Enter the un  
           Enter the pw  
           Click on login  
           Verify error msg is displayed.  
           Verify title.

-----  
 Page no-2:-Enter actiTime Enter Time track  
     ->Element  
     ->Logout

-----  
 Actions:- Verify the home page title  
           click

-----  
\*\*\*Rules To Develop the POM Class\*\*\*

- 1) For every page present in the application, we should create a POM module inside the pom package.
- 2) Name of the class should be same as title of the page ending with word 'page'.
- 3) Inside the pom class we should declare elements using FindBy Annotation, Initializing using "InitElements" and utilizing use methods.
- 4) If the method is common in more than one page then it should be developed in BasePage(Parent Class). ->inheritance
- 5) All the pom class should extend BasePage class.
- 6) BasePage class is abstract, because it is incomplete and does not present in real world.

-----  
package com.acti\_time.pom;  
  
import org.openqa.selenium.WebDriver;  
import org.testng.Assert;  
  
public abstract class BasePage  
{  
 public WebDriver driver;  
  
 public BasePage(WebDriver driver)  
 {  
 this.driver=driver;  
 }  
  
 public void verifyTitle(String eTitle)  
 {  
 String aTitle = driver.getTitle();  
  
 Assert.assertEquals(aTitle, eTitle);  
  
 System.out.println("True: Title displayed successfully");  
 }  
}

-----  
Refer com.acti\_time.pom package for scripts in eclipse:  
-----

Developing TestNG Class:-  
-----

For every TestCase we develop test class and each TestCase has some common steps such as:-

-->Opening the browser  
-->Entering the URL  
-->Closing The application etc.

Note: Instead of writing this code multiple times in TestClass we use Inheritance concepts as shown below.

```

package com.acti_time.script;

import java.util.concurrent.TimeUnit;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;

public abstract class BaseTest implements AutomationConstant
{

 public WebDriver driver;

 //PreCondition
 @BeforeClass
 public void preCondition()
 {
 System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\Admin\\
Downloads\\Program files\\chromedriver_win32\\chromedriver.exe");
 //Open the browser
 driver= new ChromeDriver();

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

 //Enter the URL
 driver.get("http://localhost/login.do");

 //Wait for 20 secs if application slow
 driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
 }

 //PostCondition
 @AfterClass
 public void postCondition()
 {
 //Closes the current window
 driver.close();

 //Closes all the windows
 //driver.quit();
 }
}
```

-----  
The above class will be created while desgining the framework.

GuideLine:-For writing TestClass.

-----

-->Before writing any TestClass, execute the given TestCase manually which gives more clarity on the steps which are to be automated.

-->Some times test data may not be printed in the TestCase this will help us to gather testcase also.

-----

Rules to develop TestNG Class:-

-----

- 1)For every TestCase, we should create TestNG class inside script package.
- 2)Name of the class should be same as respective TestCase ID or Name.
- 3)Every TestNG class extends BaseTestNG class inside the TestNG class we should create Test method[@Test] and the name of the method start with small case.
- 4)Inside the Test Method we should write Test case steps in line comment [so that we do not miss any steps] to review.
- 5)Below the line comment call the respective method of POM class.

-----

Refer com.acti\_time.script package for scripts in eclipse:

-----

--> While writing TestNG class if we find any mistakes in the POM method inform to lead or manager,we should not change the pom class we are not authorized persons.

Ex: Method verifyeTittle page is not wait till page is completed.verification is fail then we inform this author the mistakes will be corrected.

-----  
-----

\*\*\*\*\* Data Driven FrameWork \*\*\*\*\*vvimp

-----

Before going to Data Driven FrameWork, we have to understand Data Driven Testing[DDT].

Data Driven Testing[DDT] \*\*\*vvimp

-----

--> Testing the application multiple times this is called as data driven testing[Thorough testing].

--> We derived multiple inputs using testcase design techniques such as error guessing,equalence partitioning, boundry value analysis etc.,

--> Commonly manual team store those data in organize way inside an excel file.

--> To take the data from excel file, we use API or APACHE POI:-  
[Poor Obfuscation Implementation]

-----> apache poi --> To handle data driven testing\*\*\*\*\*imp

-----  
--> It can be download from following website:-  
<http://poi.apache.org/download.html>

File Name:-poi-bin-5.2.0-20220106zip

Extract the downloaded zip file it creates poi-bin-5.2.0

copy all the Jar files present inside the above folder and paste.  
Then inside jar folder of the framework add to buildpath.

-----  
Reading Data From Cell: Of an Excel File:-  
-----

Ex: dummy package

---  
package com.data\_driven\_testing;

import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;  
import org.apache.poi.ss.usermodel.Cell;  
import org.apache.poi.ss.usermodel.Row;  
import org.apache.poi.ss.usermodel.Sheet;  
import org.apache.poi.ss.usermodel.Workbook;  
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Excel  
{

    public static void main(String[] args) throws EncryptedDocumentException,  
    IOException  
    {

        String path="C:\\Users\\Admin\\Desktop\\Anil\\Poo\\E\_Commerce\_Domain\\  
Acti\_Time\_Automation\_Project\\data\\Book1.xlsx";

        FileInputStream fis=new FileInputStream(path);

        Workbook wb = WorkbookFactory.create(fis);

        //Sheet

        Sheet sheet = wb.getSheet("Emp\_Details");

        //Row --> starts from 0th index

        Row row = sheet.getRow(4);

                //(9)-->NullPointerException

        //Cell --> starts from 0th index

        Cell cell = row.getCell(4);



```

 //(6)-->NullPointerException

 //toString() --> for printing the value
 String printValue = cell.toString();

 System.out.println(printValue); //Bangalore
 }
}

-----Modified Code-----:-

package com.data_driven_testing;

import java.io.FileInputStream;
import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class ExcelModifiedCode
{
 public static void main(String[] args) throws EncryptedDocumentException,
 IOException
 {
 //modify code --> Rel filepath
 String path1="./data//Book1.xlsx";

 FileInputStream fis=new FileInputStream(path1);

 Workbook wb = WorkbookFactory.create(fis);

 //modify excel sheet
 Cell sh1 = wb.getSheet("Emp_Details").getRow(4).getCell(4);

 //toString --> for printing the value
 String printValue = sh1.toString();

 System.out.println(printValue); //Bangalore
 }
}

```

-----

Note:

- > While using path of excel ,start with .(dot) which represents path of current java project folder (i.e, uses relative filepath)
- > If sheet name, Row Index or cell Index is invalid we get NullPointerException.
- > If sheet path is invalid we get FileNotFoundException

-----

Interview Quetions:

-----

1)Write a script to find how many no of Rows present in excel sheet and  
How many cells are there in each row?

:- By using getLastRowNum() & getLastCellNum()

```
package com.data_driven_testing;

import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Excel_Total_RowsColumns
{
 public static void main(String[] args) throws EncryptedDocumentException,
 IOException
 {

 //modify code --> Rel filepath
 String path1="//data//Book1.xlsx";

 FileInputStream fis=new FileInputStream(path1);

 Workbook wb = WorkbookFactory.create(fis);

 //Total rows in Emp_Details sheet -->getLastRowNum()
 int totalRows = wb.getSheet("Emp_Details").getLastRowNum();

 System.out.println("Total rows: "+totalRows); //8

 //How many cells are there in each row -->getLastCellNum()

 for(int i=0; i<=totalRows; i++)
 {
 short eachRowtotalCells =
wb.getSheet("Emp_Details").getRow(i).getLastCellNum();

 System.out.println("Each Row Total Cell:
"+eachRowtotalCells); //6
 }
 }
}
```

-----

2)Write a script to print contents of the excel sheet in table format where has  
data in three rows and three columns?

:- We should go for nested for loop  
rowCount-->starts from 0th index  
columnCount-->starts from 0th index

```
package com.data_driven_testing;
```

```

import java.io.FileInputStream;
import java.io.IOException;

import org.apache.poi.EncryptedDocumentException;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Excel_PrintContents_3Rows3Columns
{
 public static void main(String[] args) throws EncryptedDocumentException,
 IOException
 {

 //modify code --> Rel filepath
 String path1="//data//Book1.xlsx";

 FileInputStream fis=new FileInputStream(path1);

 Workbook wb = WorkbookFactory.create(fis);

 for(int i=0;i<=2;i++) //3 rowCount-->starts from 0th index
 {
 for(int j=0;j<=2;j++) //3 columnCount-->starts from 0th index
 {
 String printValue =
wb.getSheet("Emp_Details").getRow(i).getCell(j).toString();
 System.out.print(printValue+" ");
 }
 System.out.println(); //to go to next line
 }
 }
}

```

-----  
-----

Developing generic methods:-

-----

\*\*\*\*Generic Methods\*\*\*\*

-----

The methods which can be used in any project are called as Generic methods.

--> In company, by default generic package will be there,  
inside this package several generic methods will be available.

Ex: Methods to handle excel file:- these are readymade methods

-----

```

package com.acti_time.generics;

import java.io.FileInputStream;

import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;

```

```

public class Excel
{
 //row count

 public static int getRowCount(String path,String sheet) //Return type is int
 {
 int rc=0;

 try
 {
 FileInputStream fis=new FileInputStream(path);

 Workbook wb = WorkbookFactory.create(fis);

 rc=wb.getSheet(sheet).getLastRowNum();
 }

 catch(Exception e)
 {
 System.out.println("We are not able to count last row number");
 }

 return rc;
 }

 //.....

 //cell count

 public static int getCellCount(String path,String sheet, int r) //Return type
is int
 {
 int rc=0;

 try
 {
 FileInputStream fis=new FileInputStream(path);

 Workbook wb = WorkbookFactory.create(fis);

 rc=wb.getSheet(sheet).getRow(r).getLastCellNum();
 }

 catch(Exception e)
 {
 System.out.println("We are not able to count last cell number");
 }

 return rc;
 }

 //.....

 //printing purpose

 public static String getCell(String path,String sheet,int r, int c) //Return
type is String
 {

```

```

 String getSheet="";
 try
 {
 FileInputStream fis=new FileInputStream(path);

 Workbook wb = WorkbookFactory.create(fis);

 getSheet=wb.getSheet(sheet).getRow(r).getCell(c).toString();
 }

 catch(Exception e)
 {
 System.out.println("We are not getting cell");
 }

 return getSheet;
 }
}

```

-----

\*\*\*\*\* Data Driven Framework \*\*\*\*\*vvimp

-----

To test the application with multiple inputs we take the data from excel file.

If the framework has the future then it is called data driven framework.

--> Data driven framework will have additional methods to handle the excel file such as reading the data, counting the row, modify etc.,

--> This framework will also have separate folder, where excel file to be stored.

Ex: While designing the framework we create folder with name data and within that we create an excel file and store the data as [Input.xlsx]

Manual team will provide the excel file.

--> Since it's path [Input.xlsx] will never change we will make it as constant and as per java

coding practices we should store it as an interface in script package.

EX:

----

```

package com.acti_time.script;

public interface AutomationConstant
{
 public static final String PATH=".//data//Input.xlsx";
}

```

NOTE: All the constants should be capital.

Whenever final keyword comes, variable name should be in uppercase.

-----

vvimp\*\*\*\*

Since this constant is used in all the TestNG classes, to access we are using Inheritance concepts, instead of making all the TestNG classes to inherit from Interface we can make BaseTest class to inherit from the Interface because all the TestNG classes inherit from BaseTest.

Ex: We should update as shown below in BaseTest

-----

```
public abstract class BaseTest implements AutomationConstant
{
 Other lines/codes already written beginning
}
```

-----

Rules:-

-----

- > Data should be present inside the Input.xlsx file.
- > Class name should be same as respective TestCase name or ID.
- > No of columns in the Excel file should be same as no of requirement Input.
- > No of columns should not be change, only we can change no of Rows.
- > First row always use the specific header.

-----

Ex: IN EXCEL Sheet --> VerifyVersionE

-----

```
Sl No Version
1 actiTIME 2016.3
```

Script-1:

-----

```
package com.acti_time.script;

import org.testng.annotations.Test;

import com.acti_time.generics.Excel;
import com.acti_time.pom.Version;

public class VerifyVersionE extends BaseTest
```

```

{
 @Test
 public void verifyVersionE()
 {
 String sheet="VerifyVersionE";

 String version = Excel.getCell(PATH, sheet, 1, 1);

 Version v1=new Version(driver);

 v1.verifyVersion(version);

 }
}

```

-----

Ex: IN EXCEL Sheet -->VerifyValidLoginPageE

----

| Sl No | Username | Password | LoginPage Title  | HomePageTitle               |
|-------|----------|----------|------------------|-----------------------------|
| 1     | admin    | manager  | actiTIME - Login | actiTIME - Enter Time-Track |

Script-2:

-----

```

package com.acti_time.script;

import org.testng.annotations.Test;

import com.acti_time.generics.Excel;
import com.acti_time.pom.HomePage;
import com.acti_time.pom.LoginPage;

public class VerifyValidLoginPageE extends BaseTest
{
 @Test
 public void verifyValidLoginPageE() throws InterruptedException
 {
 String sheet="VerifyValidLoginPageE";

 String un = Excel.getCell(PATH, sheet, 1, 1);

 String pw = Excel.getCell(PATH, sheet, 1, 2);

 String lpt = Excel.getCell(PATH, sheet, 1, 3);

 String hpt = Excel.getCell(PATH, sheet, 1, 4);

 LoginPage l1=new LoginPage(driver);
 l1.verifyTitle(lpt);
 l1.setUsername("admin");
 l1.setPassword("manager");
 l1.clickOnLoginButton();

 HomePage h1=new HomePage(driver);
 }
}

```

```

 Thread.sleep(3000);
 h1.verifyTitle(hpt);

 Thread.sleep(3000);
 h1.clickOnLogoutButton();
 }
}

```

-----

Ex: IN EXCEL Sheet --> VerifyInvalidLoginPageE

----

We have to apply Test Case Design Techniques.

--> Error guessing technique for Invalid Login Page feature.

--> Manual team is responsible.

| Sl No | Username | Password        |
|-------|----------|-----------------|
| 1     | aaaa     | bbbb            |
| 2     | admin    | Tiger           |
| 3     | Tiger    | manager         |
| 4     |          | nnnn            |
| 5     | dddd     |                 |
| 6     |          |                 |
| 7     | am2707   | ma2105          |
| 8     | am#27    | ma21\$          |
| 9     | Mamtha   | Happy Birthday! |
| 10    | !@#\$%   | ^&*()           |

:- using for loop we can achieve this

Script-3:

-----

```

package com.acti_time.script;

import org.testng.annotations.Test;

import com.acti_time.generics.Excel;
import com.acti_time.pom.InvalidLoginPage;
import com.acti_time.pom.LoginPage;

public class VerifyInvalidLoginPageE extends BaseTest
{
 @Test
 public void verifyInvalidLoginPageE() throws InterruptedException
 {
 String sheet="VerifyInvalidLoginPageE";

 int totalRowCount = Excel.getRowCount(PATH, sheet);

 for(int i=1; i<=totalRowCount; i++)
 {
 String un = Excel.getCell(PATH, sheet, i, 1);

 String pw = Excel.getCell(PATH, sheet, i, 2);

```



```

LoginPage l1=new LoginPage(driver);
 l1.setUsername(un);
 l1.setPassword(pw);
 l1.clickOnLoginButton();

InvalidLoginPage i1=new InvalidLoginPage(driver);

 Thread.sleep(3000);

 i1.verifyErrorMsg();

 }
}

```

-----  
 \*\*\*\*\*Architecture Of FrameWork\*\*\*\*\* imp  
 -----

Description:-  
 -----

->It is a hybrid framework which is a combination of data driven and method driven framework.

->This framework is develop using TestNG.

->For every TestCase have develop TestClass which is extends BaseTest. Base Test has before method and after method it also implements and interface class Automationconst.

->First it's executes BeforeMethod[PreCondition] which opens the browser and enters the url.

->After it's executing beforemethod it will stop the execution of test method, Test method will take the data from excel file using generic library. then it performs it action on the application by calling the methods present in the parent called BasePage it is an abstract class this class is inherited by all the POM classes.

->After executing Test method, it is executed after method postcondition. process the browser at the end output is generated in html format.

Architecture:- imp  
 -----

Ref diagram....

-----  
 -----  
 3.Framework Execution

=====

1)How do you execute framework?

:-Using TestSuiteFile

-----

2)How do you create TestNG suiteFile?

:-Right click on Java Project

```

 |
Go to TestNG
 |
Convert TestNG
 |
Click
 |
Finish
 |
it generates TestNG.xml
 |
Right click on TestNG.xml
 |
RunAS
 |
TestNG Suite

```

TestSuite:-

-----

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
 <test thread-count="5" name="Test">
 <classes>
 <class name="com.acti_time.script.VerifyVersion"/>
 <class name="com.acti_time.script.VerifyHomePage"/>
 <class name="com.acti_time.script.VerifyValidLoginPage"/>
 <class name="com.acti_time.script.VerifyReports"/>
 <class name="com.acti_time.script.VerifyVersionE"/>
 <class name="com.acti_time.script.VerifyCheckBox"/>
 <class name="com.acti_time.script.VerifyLoginPageTitle"/>
 <class name="com.acti_time.script.VerifyTasks"/>
 <class name="com.acti_time.script.VerifyInvalidLoginPage"/>
 <class name="com.acti_time.script.VerifyInvalidLoginPageE"/>
 <class name="com.acti_time.script.VerifyValidLoginPageE"/>
 <class name="com.acti_time.script.VerifyHomePageTitle"/>
 </classes>
 </test> <!-- Test -->
</suite> <!-- Suite -->

```

-----

3)Can manual Testing team executing your framework they don't know  
Java or eclipse?

:-Yes

=====

Log4j Introduction

-----

## 25.Log4J:

-----  
print statement --> save, print & time

Why logging is important in any application?

:-Logging is very important to any application. It helps us collect information about how the application is running and also helps us debug if any failure occurs.

.....  
->What is Log4j?

Log4j is a brilliant logging API available both on Java and .net framework.

Advantages are:

① >Log4j allows you to have a very good logging infrastructure without putting in any efforts.

① >Log4j gives you the ability to categorize logs at different levels (Trace, Debug, Info, Warn, Error and Fatal).

① >Log4j gives you the ability to direct logs to different outputs. For e.g. to a file, Console or a Database.

①> Log4j gives you the ability to define the format of output logs.

① >Log4j gives you the ability to write Asynchronous logs which helps to increase the performance of the application.

①>Loggers in Log4j follow a class hierarchy which may come handy to your applications.

.....  
\*\*\*Log4j\*\*\*:-

Log4j consists of four main components

① LogManager

① Loggers

① Appenders

① Layouts

With these comes some additional components which will be covered in the individual headings.

.....  
.  
LogManager:-

This is the static class that helps us get loggers with different names and hierarchy. You can consider LogManager as a factory producing logger objects.

A sample code will be:-

.....  
package Log4jSample;

import org.apache.log4j.BasicConfigurator;

import org.apache.log4j.LogManager;

import org.apache.log4j.Logger;

public class SampleEntry {

//mainLogger is a logger object that we got from LogManager. All loggers are

//using this method only. We can consider LogManager as a factory to create

//Logger objects

static Logger mainLogger = LogManager.getLogger(SampleEntry.class.getName());

```

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

BasicConfigurator.configure();
mainLogger.info("This is just a logger");

}
}

```

\*\*\*\*\*Logger\*\*\*\*\*;

This is a class which helps you log information at different logging levels.

In the above sample code you can see that we have created a logger named mainLogger using the LogManager static class. Now we can use it to write logs. As you can see we have mainLogger.info("Comments that you want to log") statement which logs the string.

\*\*\*\*\*Appenders\*\*\*\*\*:-

Append means adding in last.

Appenders are objects which help Logger objects write logs to different outputs. Appenders can specify a file, console or a data base as the output location.

In this code sample you will see that we have used a console appender to print logs like we would do using System.out or System.err.

```

package Log4jSample;

```

```

import org.apache.log4j.BasicConfigurator;
import org.apache.log4j.ConsoleAppender;
import org.apache.log4j.LogManager;
import org.apache.log4j.Logger;

```

```

public class SampleEntry {

```

```

 //All the loggers that can be used

```

```

 static Logger mainLogger = LogManager.getLogger(SampleEntry.class.getName());

```

```

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

```

```

 BasicConfigurator.configure();

```

```

 //Create a console appender and attach it to our mainLogger

```

```

 ConsoleAppender appender = new ConsoleAppender();

```

```

 mainLogger.addAppender(appender);

```

```

 mainLogger.info("This is just a logger");

```

```

 }

```

```

}

```

\*\*\*\*\*Layouts\*\*\*\*\*:-

Layout class help us define how the log information should appear in the outputs. Here is a sample code which uses PatternLayout Class to change the formatting of logs:

```

package Log4jSample;

import java.util.Enumeration;

import org.apache.log4j.Appender;
import org.apache.log4j.BasicConfigurator;
import org.apache.log4j.ConsoleAppender;
import org.apache.log4j.Layout;
import org.apache.log4j.LogManager;
import org.apache.log4j.Logger;
import org.apache.log4j.PatternLayout;

public class SampleEntry {

 //All the loggers that can be used
 static Logger mainLogger = LogManager.getLogger(SampleEntry.class.getName());

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

 BasicConfigurator.configure();
 ConsoleAppender appender = new ConsoleAppender();
 appender.activateOptions();
 PatternLayout layoutHelper = new PatternLayout();
 layoutHelper.setConversionPattern("%-5p [%t]: %m%n");
 layoutHelper.activateOptions();
 //mainLogger.getAppender("ConsoleAppender").setLayout(layoutHelper);
 appender.setLayout(layoutHelper);
 mainLogger.addAppender(appender);
 //Create a console appender and attach it to our mainLogger
 mainLogger.info("Pattern 1 is displayed like this");
 layoutHelper.setConversionPattern("%C %m%n");
 mainLogger.info("Pattern 2 is displayed like this");

 }

}
.....
.....

```

Expected output on the console will be

```

INFO [main]: Pattern 1 is displayed like this
Log4jSample.SampleEntry

```

```

Pattern 2 is displayed like this

```

This is brief explanation of the different components inside Log4j APIs.  
 Lets understand each component individually.

-----

\*\*\*\*\*Downloading Log4j Jar\*\*\*\*\*:-  
<https://logging.apache.org/log4j/2.x/download.html>

Extract the downloaded zip file and add in buildpath.

\*\*\*\*\*Log4j property file:-\*\*\*\*\*:

```
log4j.rootLogger=INFO, CONSOLE, R, HTML, TTCC
```

```
#here we define the appender
```

```
log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
log4j.appender.R=org.apache.log4j.RollingFileAppender
log4j.appender.TTCC=org.apache.log4j.RollingFileAppender
log4j.appender.HTML=org.apache.log4j.FileAppender
```

```
#Here we define log file location
```

```
log4j.appender.R.File=./log/testlog.log
log4j.appender.TTCC.File=./log/testlog1.log
log4j.appender.HTML.File=./log/application.html
```

```
#Here we define the layout and pattern
```

```
log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern=%5p [%t] [%F:%L] -%m%n
log4j.appender.R.layout=org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern=%d - %c - %p - %m%n
log4j.appender.TTCC.layout=org.apache.log4j.TTCCLayout
log4j.appender.TTCC.layout.Dateformat=ISO8601
log4j.appender.HTML.layout=org.apache.log4j.HTMLLayout
log4j.appender.HTML.layout.Title=Application log
log4j.appender.HTML.layout.LocationInfo=true
```

```

Log4j.properties:-
```

```
.....
log4j.logger.devpinoLogger=DEBUG, dest1
log4j.appender.dest1=org.apache.log4j.RollingFileAppender
log4j.appender.dest1.maxFileSize=5000KB
log4j.appender.dest1.maxBackupIndex=3
log4j.appender.dest1.layout=org.apache.log4j.PatternLayout
log4j.appender.dest1.layout.ConversionPattern=%d{dd/MM/yyyy HH:mm:ss}%c %m%n
log4j.appender.dest1.File=f:\\Logs.log
log4j.appender.dest1.Append=false
```

```

Script:
```

```
=====
=====
MAVEN [Project Management Tool]

```

26.Maven:

-----

--> Maven is used to define project structure,dependencies build and test management.

--> In Maven ,artifact is a jar file and is identified by a co-ordinate system of groupId,artifectId and version

--> Maven uses the groupId,artifectId and Version to identify dependencies(other jar files)  
needed to build and run your code

--> groupId:-Identify of the group of the project

--> artifactid:-identity of the project snapshot

--> version:-version used by the project

--> Maven plugin for eclipse \*\*\*\*vimp

--> Maven plugin for command prompt

-----

\*\*\*\*\*Maven [Project Management Tool]\*\*\*\*\*vvimp

A build tool is a tool that automates everything related to building the software project.

Advantages \*\*imp

-----

Maven is a Project Management Tool

-->In testing, Maven used to develop the framework,because Maven gives the folder structure itself

-->It can also used by development team & testing team

-->We can add or modify the dependencies very easily\*\*\*imp

Building a software includes:

-----

- 1)Generating source code.
- 2)Generating documentation.
- 3)Compiling codes.
- 4)Packing compiled code into jar or war or zip files.
- 5)Installing the packed code on a server, in a repository or somewhere else.

## Maven Installation

[Non-installable file]

### Step 1

- >Go to google
- >Search apache maven
- >Search for binary zip archive
- >Download and extract the zip file

### Step 2

- >Open the maven -3.8.4
- >Copy the path
- >Open advanced system setting
- >Environment variable

User variable->New

### Step 1

Variable Name:-M2\_HOME

Variable Value:-C:\Users\Admin\Downloads\Program files\apache-maven-3.8.4-bin\apache-maven-3.8.4

### Step 2

Variable Name:-MAVEN\_HOME

Variable Value:-C:\Users\Admin\Downloads\Program files\apache-maven-3.8.4-bin\apache-maven-3.8.4

System variable->New

### Step 1 - can skip this step

Variable Name:-M2\_HOME

Variable Value:-C:\Users\Admin\Downloads\Program files\apache-maven-3.8.4-bin\apache-maven-3.8.4

### Step 2

Variable Name:-path

Variable Value:-C:\Users\Admin\Downloads\Program files\apache-maven-3.8.4-bin\apache-maven-3.8.4\bin



### Step 3

-----

Variable Name:-path

Variable Value:-C:\Users\Admin\Downloads\Program files\apache-maven-3.8.4-bin\apache-maven-3.8.4

-----  
--

How to Cross check Maven Installed or not

-----

->Go to command prompt

->Enter mvn --version

\*\*\*\*\*imp\*\*\*\*\*

Before configuring maven make sure Java class path is done successfully

User variable->New

### Step 1

-----

Variable Name:-JAVA\_HOME

Variable Value:-C:\Program Files\Java\jdk-17.0.2

Note: Do not use semicolon

-----

Now creating Maven Project

-----

1)Using command prompt -- not so imp

2)Using Eclipse -- vvimp

-----

1)Using command prompt - not so imp

-----

Go to below folder in cmd

C:\Users\Admin\Desktop\Anil\Poo\E\_Commerce\_Domain

\*\*Archetype is a maven project templating toolkit

\*\*Archetype is defined as an original pattern or model command to create  
Maven Project :-From which all other things of same kind are made.

D:\mvn archetype:generate

It will ask you for few properties and then it will create a project

Choose a number:8-[Leave empty Enter]

Group-Id:-Unique identification name can be a package name

Ex:com.ibm.gmail.test

artifactId:-can be any name can give the project name

Ex: com.ibm.gmail.test

Version: Leave empty it will take default .jar

-----

\*Go and check the C:\Users\Admin\Desktop\Anil\Poo\E\_Commerce\_Domain for project

Two files:-     1)src---->1)main  
                  2)test  
  
                  2)pom--->Xml file

\*TestNG works with the help of TestNG.xml like that Maven works through pom.xml

\*main-->development code

\*test--->unit testing code(test cases)

-----

Now you need to import this project in to eclipse

->Right click on eclipse  
->click on import  
->General->Existing project in to workspace  
->Next->Browse our project->Locate the project and select.

\*There is a working :No project are found to import error

Beacuse there is no .classpath and no .project

[If you want check your different project inside the project .class and .project is available]

\*Now you need to generate .project

-----

Go to command prompt

C:\Users\Admin\Desktop\Anil\Poo\E\_Commerce\_Domain>mvn eclipse:eclipse

It will generate eclipse understanding .class path and .project

\*Now on your c:\Users\\_\_\_\_ your .class path and .project is added

\*Now you can easily import the project

\*By default we can see the folder structure

\*This is how to create project in command prompt

-----  
----

2)Using Eclipse -- \*\*\*\*\*vvimp

-----

->If you are using an eclipse version after Luna or latest Neon-version,  
by default you will be getting it.

->In case you do not have you can go to  
Help-Eclipse Marketplace and then search Maven and then Install.

-----

-->Now coming to creating a project:

File->New ->Others->MavenProject

It will ask for groupId and artifactId

-----

\* Maven mainly helps in organizing the jars files.

\* Now come to pom.xml

\* To add any jar files we need this code:-

```
<dependency>

<groupId>
 <artifactId>
 <version>

</dependency>
```

\* Add dependencies of selenium, testNG & Webdrivermanager(i.e. add jar files).

-----

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
 <modelVersion>4.0.0</modelVersion>
 <groupId>Maven_Demo_Project</groupId>
 <artifactId>Maven_Demo_Project</artifactId>
 <version>0.0.1-SNAPSHOT</version>

<properties>
 <maven.compiler.source>1.8</maven.compiler.source>
 <maven.compiler.target>1.8</maven.compiler.target>
</properties>

<build>
 <pluginManagement>
 <plugins>
 <plugin>
 <groupId>org.apache.maven.plugins</groupId>
 <artifactId>maven-surefire-plugin</artifactId>
 <version>3.0.0-M5</version>
 </plugin>
 </plugins>
 </pluginManagement>
</build>

<dependencies>
```

```

 <!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java
-->
<dependency>
 <groupId>org.seleniumhq.selenium</groupId>
 <artifactId>selenium-java</artifactId>
 <version>4.1.2</version>

</dependency>

 <!-- https://mvnrepository.com/artifact/org.testng/testng -->
<dependency>
 <groupId>org.testng</groupId>
 <artifactId>testng</artifactId>
 <version>7.5</version>
 <scope>test</scope>

</dependency>

 <!-- https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager
-->
<dependency>
 <groupId>io.github.bonigarcia</groupId>
 <artifactId>webdrivermanager</artifactId>
 <version>5.1.0</version>

</dependency>

</dependencies>

</project>

```

-----

Q. How to resolve following errors when running your Maven Project(Particularly When JDK Version is greater than 8 )

-->[ERROR 1] No compiler is provided in this environment. Perhaps you are running on a JRE rather than a JDK?

1. Set JDK as run time environment

2. Update Maven Project

-->[ERROR 2] Source option 6 is no longer supported. Use 7 or later.  
[ERROR 2] Target option 6 is no longer supported. Use 7 or later.

By default Maven Compiler uses 1.6 as Source and Target. So we need to force it to use 1.8 version(Java 8 version) no matter what JDK version u have Set in Installed JREs.

1. Add Properties to Set Source and Target as 1.8 in POM.XML

2. Configure Maven Compiler plugin to Set Source and Target as 1.8 in POM.XML

3. Best way: Install JDK 1.8 and Set path in Environmental Variables

-----  
\*\*I have written two test classes

\*\*I want execute one class after the other

\*\*There is a naming convention to run --> classname mudt be end with Test  
Ex: Demo1\_Test  
FirstTest

Script-1:

```

package com.practice.scripts;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;

import io.github.bonigarcia.wdm.WebDriverManager;

public class Demo1_Test
{
 public WebDriver driver;

 @Test
 public void testCase1() throws InterruptedException
 {
 WebDriverManager.chromedriver().setup();
 //-->not required for Selenium version 2.53.0
 //-->FirefoxDriver()-->it should be below 39
version
 driver=new ChromeDriver();
 Thread.sleep(3000);
 driver.get("https://www.amazon.in/");
 driver.manage().window().maximize();
 }

 @Test
 public void testCase2() throws InterruptedException
 {
 Thread.sleep(3000);
 driver.close();
 }
}
```

Script-2:

```

package com.practice.scripts;

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

```

import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.Test;

import io.github.bonigarcia.wdm.WebDriverManager;

public class Demo2_Test
{
 public WebDriver driver;

 @Test
 public void testCase1() throws InterruptedException
 {
 WebDriverManager.chromedriver().setup();
 //-->not required for Selenium version 2.53.0
 //-->FirefoxDriver()-->it should be below 39
version

 driver=new ChromeDriver();

 Thread.sleep(3000);

 driver.get("http://localhost/login.do/");

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

 }

 @Test
 public void testCase2() throws InterruptedException
 {
 Thread.sleep(3000);

 driver.close();

 }
}

```

-----

```

**Now go to pom.xml -> Run As -> Maven clean
-->BUILD SUCCESS

```

```

**Now go to pom.xml -> Run As -> Maven test
-->BUILD SUCCESS
-->TestSuite will run

```

\*\*How to run the above test classes in command prompt

:- Go to Maven package folder location & type

```

mvn clean compile test

```

```

=====
=====

```

We can run the test cases in following ways: \*\*\*vvimp

-----

## 1.Maven pom.xml

```
-->Maven clean
-->Maven test
```

## 2.Maven CLI(Command Line Interface)

```
--> cd C:\Users\Admin\myworkspace\Automation Project\Maven_Demo_Project
mvn clean install
```

## 3.run.bat

```
--> cd C:\Users\Admin\myworkspace\Automation Project\Maven_Demo_Project
mvn clean install
```

## 4.Jenkins(Continous Integration)

- 1.using Freestyle Project(run.bat)
- 2.using Maven Project(pom.xml) -- real time approach

```
=====
=====
```

## 27.Jenkins:

-----

**\*\*jenkins is installed in Devops environment, not in QA environment.**

**\*\*In jenkins execution is seamless (we cannot see browser opening & performing actions).**

**\*\*Open source**

Download jenkins: <https://www.jenkins.io/download/>

```
--> cd C:\Program Files\Jenkins
java -jar Jenkins.war
```

open browser and enter: <http://localhost:8080/>

password: 9dab386689f94ab1b2bf4f339ecacd9d  
[C:\Users\Admin\.jenkins\secrets initialAdminPassword]

copy & paste the password in Administration password and click on continue

We can run the test cases using Jenkins by following 2 ways

-----

- 1)Execute project using Freestyle Project(run.bat)
- 2)Execute project using Maven Project(pom.xml) -- real time approach

## 1)Free style project(run.bat file)

-----

Steps to create Freestyle Project

-----

New item--->Item name(project name)--->Freestyle project--->Build--->

cd C:\Users\Admin\myworkspace\Automation Project\Maven\_Demo\_Project  
run.bat

--->Apply--->Save

Build now (To run the automation project)

2)Maven Project(pom.xml file) -- real time approach

-----

Pre-requisites

-----

1.Install Maven Plugins in Jenkins

- >Manage Jenkins
- >Manage Plugins
- >Available
- >Select Plugins related to Maven
- >Click on Install without restart button

2.imp step:

Configure JAVA\_HOME

->Manage Jenkins->Global Tool Configuration->JDK-> JDK installations->Click on Add JDK

Name: myJDK (anything we can give)

JAVA\_HOME: C:\Program Files\Java\jdk-11.0.13

->Apply->Save

Configure M2-HOME

->Manage Jenkins->Global Tool Configuration->Maven - Maven installations->Click on Add Maven

Name: myuMaven (anything we can give)

MAVEN\_HOME: C:\Program Files\apache-maven-3.8.4

->Apply->Save

Steps to create Maven Project

-----

New item--->Item name(project name)---Maven project--->Build--->

while configuring your project, in Build section...

Root pom: C:\Users\Admin\myworkspace\Automation Project\Maven\_Demo\_Project\pom.xml  
Goals & options: clean install

--->Apply--->Save

Build now (To run the automation project)

=====



=====

## 28.Git & Github:

-----

- 1.Local repository (git)
- 2.Global/Remote repository (github)

### Pre-requisites

-----

- 1)Install git in your local system  
<https://git-scm.com/download/win>
- 2)Create an account in github  
<https://github.com/>

Remote Repository URL: [https://github.com/anil2105/Maven\\_Demo\\_Project.git](https://github.com/anil2105/Maven_Demo_Project.git)

### GIT Commands \*\*vvimp

-----

- 1)git init  
-->create an empty git repository (Local repository)
- 2)git remote add origin "[https://github.com/anil2105/Maven\\_Demo\\_Project.git](https://github.com/anil2105/Maven_Demo_Project.git)"  
--> associate local repository to remote repository

NOTE: We have to execute the above 2 commands(1 & 2) only one time.

- 3)git status  
--> what are files ready to commit in git repository

- 4)git add -A  
-->Add all your files from working directory into staging area

1.git config --global user.name "anil2105"

2.git config --global user.email "anil.krtg@gmail.com"

NOTE: We have to execute the above 2 commands(1 & 2) only one time.

- 5)git commit -m "This is my first commit"  
-->commit all the files from staging area into git repository

- 6)git push -u origin master  
-->//Ask you un & pw, pushing the files from git-----to----->github

7)git pull origin master  
-->pull the files from github to git repository

-----  
From the 2nd round onwards, only the following commonads will be repeated \*\*imp

git status  
--> what are files ready to commit in git repository

git add -A  
-->Add all your files from working directory into staging area

git commit -m "This is my first commit"  
-->commit all the files from staging area into git repository

git push -u origin master  
-->Ask you un & pw, pushing the files from git-----to----->github

git pull origin master  
-->pull the files from github to git repository

-----  
Github integration with Jenkins \*imp

-----  
New item--->Item name(project name)---Maven project--->  
Source Code Management--->Git-->Repository  
URL([https://github.com/anil2105/Maven\\_Demo\\_Project.git](https://github.com/anil2105/Maven_Demo_Project.git))--->  
Build--->Root POM(pom.xml)--->Goals and options(clean install)  
Apply--->Save

Build now (To integrate github with jenkins)

-----  
--

=====

=====

===== THE END =====

=====

=====

