

```
If (exp.equals("acu"))  
    test.log(logStatus.pass, "Login Pass");  
} else {  
    test.log(logStatus.fail, "Login fail");  
}  
test.endTest(test);  
test.flush(); }
```

@After method
public void logout(){
 driver.quit();
}

→ [8908]

stmt.executeQuery("select * from employee");

) Process the results: results from the executed query are stored in the result set object.

1. Result set rs = stmt.executeQuery("select * from employee");

Method Name	Description
String getString()	Method is used to fetch the String type data from result set.
int getInt()	" " " " " Integer " " "
double getDouble()	" " " " " double " " "
Date getDate()	" " " " " Date type " " "
boolean next()	" " " " " more to the next record in the result set.
boolean previous()	" " is used to move to the previous record in the result set.
boolean first()	" " " " " First " " "
boolean last()	" " " " " last " " "
clear absolute (int rowNumber)	" " " " " Specific " " "

Adding MySQL dependency Jar file:-

Go to mavenrepository.com - SQL-6.0.16

Download XAMPP:- Go to google search - XAMPP -

<https://apachefriends.org/download.html> → download & install.

to start click on XAMPP folder - XAMPP Control Panel → click start for Apache & MySQL.

Open a browser - <http://localhost/phpmyadmin/>

Creating database :- Click on new, - give database name in Create database textbox. click on create.

give the table name as Employee → Select no. of Col. 2 → click on Go

give Col. names as Ename, Eno, ESal, Vender Int Int click on save.

WebTable:-

Table is a kind of HTML data which is displayed with the help of `<table>` tag in conjunction with the `<tr>` & `<td>` tags. Although there are other tags for creating tables these are the basics for creating a table in HTML. Here tag `<tr>` defines the row & tag `<td>` defines the column of the table. Excel sheet is a simple example of table structures. whenever we put some data in to excel we give them some heading as well. In HTML we use `<th>` tag for headings which defines heading of the table. Each cell in the Excel sheet can be represented as `<td>` in the HTML table. The `<td>` elements are the data containers & these can contain all sorts of HTML elements like text, images, list, other tables etc.

Package Tables_Pack;

import java.util.List;

public class Sample_Example{

public static void main (String [] args){

WebDriver driver = new ChromeDriver();

driver.navigate().to ("file:///D%2Fgedgetable.html");

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

//Store completable into webElement class object

WebElement table = driver.findElement(By.name("gedgetch"));

String tabletext = driver.findElement(By.xpath("//html/body/table[tbody/tr[5]/td[1]]")).getText();

//Count no of rows in a table

List<WebElement> rows = table.findElements(By.tagName("tr"));
System.out.println("no. of rows are:: "+rows.size());

for (int i=1; i<rows.size(); i++){

//Count no of columns in rows

List<WebElement> cols = rows.get(i).findElements(By.tagName("td"));

if (i==1) { System.out.println("no. of cols are:: "+cols.size());}

} //Iterate all columns

for (int j=0; j<cols.size(); j++) { //get text of all cols

```
public static void main (String [] args) throws InterruptedException {
    WebDriver driver = new ChromeDriver();
    JavascriptExecutor js = (JavascriptExecutor) driver;
    // to launch url
    js.executeScript ("window.location = 'https://facebook.com'");
    driver.manage().window().maximize();
    // fetching the domain name
    String sDomain = js.executeScript ("return document.domain;"),
    System.out.println ("Domain = " + sDomain);
    // fetching the URL
    String sURL = js.executeScript ("return document.URL").toString(),
    System.out.println ("URL = " + sURL);
    // fetching the title
    String sTitle = js.executeScript ("return document.title;").toString();
    System.out.println ("Title = " + sTitle);
    Thread.sleep (5000);
    // vertical scroll down by 200 pixels
    js.executeScript ("window.scrollBy(0, 200)");
    System.out.println ("successfully did the vertical scroll by 20px");
    Thread.sleep (5000);
    // to scroll down complete page
    js.executeScript ("window.scrollTo(0, document.body.scrollHeight)");
    Thread.sleep (5000);
    // to scroll up complete page
    js.executeScript ("window.scrollTo (document.body.scrollHeight, 0)");
    Thread.sleep (5000); }
```

click on Add JDK. Uncheck Install automatically. In JDK Name text box type JAVA_HOME in caps. Copy the path of JDK & paste under Java_HOME textbox. scroll down. Click on Add Maven.

Uncheck Install automatically. In Maven name as → MAVEN_HOME (caps) → MAVEN_HOME text box provide path of Maven. Copy the Maven path from Maven folder & paste to Maven_Home box. Click on Save.

Email Integration - To receive emails from Jenkins whenever build is unstable. Click on Manage Jenkins. click on Configure System. scroll down to bottom under 'Email Notification' - update as below.

SMTP server → Smtp.gmail.com
check check box Gmail (Email)

Click on Advanced. Use SMTP Authentication. Enter Username & Password.

Check check box use SSL → SMTP Port as 465 (Gmail server).

→ Reply to address → Gmail Id → Check check box Test Configuration by sending test email. Give Email address in Test email recipient.

Click on Test Configuration. Click save.

Creating Project in Jenkins - Click on 'new item / create new jobs'. Enter ur project name. In enter an Item name. Click on Freestyle project. Click on 'OK'.

Giving the path of pom.xml (maven project) for the project! -

Click on 'Build' tab - scroll down expand 'Add Build Step'. In dropdown 'Select 'Invoke top level maven targets''. Select 'maven version' as 'MAVEN_HOME'. Give the goals name as 'test'. Click advanced. Copy the path of pom.xml from Eclipse: Maven Project. (right-click properties) & Paste to pom text box. To receive mails from Jenkins whenever project got fail. Click on 'Post build Actions' tab. Expand 'Add Post build action' → Select 'Email notification' → provide Email id in Recipients. → Click Save. → Click on Jenkins dropdown expand ur project Select 'Build Now'.

To view Results in Jenkins → Click on Build History → Click on icon for view results.

Downloading AUTOIT:-

<https://autoitscript.com/site/autoit/downloads/>
click on download AUTOIT .Install

start menu → all programs → click on AutoIT folder → **open** → AutoIT window info
Open → AutoIT Script editor → to write script
Script:-
 winwaitActive ("open", "")
 ControlSend ("Open", "", "Edit1", "C:/users/John/Desktop/capture.Png")
 ControlClick ("open", "", "Button1")
 click on save button → save type as AUTOIT(AU3) -
 filename as employee - save to desktop.

click on attachment (browse)
 ControlSend ^{wnd} - sending ~~keys~~ screenshot in a textbox
 Control Click ^{wnd} - click the button
 Go to saved file
 Right click mouse compile script - In web browser

Runtime.getRuntime().exec("C:/users/John/Desktop/employee.exe");
 Path of the compiled .

Jenkins :- It is the leading open source continuous integration tool developed by Hudson lab. It is cross platform & can be used on windows, linux, mac, os & solaris environments. It is written in Java.

Why Jenkins & Selenium? Running selenium tests in Jenkins allows you to run your tests every time ur software changes & deploy the software to a new environment when the tests pass → Jenkins can schedule ur tests to run at specific time → You can save the execution history & Test Reports.

Important Features of Jenkins:-

Jenkins provides direct links to Email when deployment is fail / updates of deployment.

```
webdriver driver;  
driver = new RemoteWebDriver(  
    new URL("http://192.168.1.4:5556/wd/hub"), capabilities);
```

↑
IP address & port on machine

DO not close command prompt window. (hub, node)

→ Package Resources;

```
import java.net.MalformedURLException;  
public class AppTest {  
    public static WebDriver dr;  
    // DesiredCapabilities cap;
```

@Test

```
public void verify_login() throws MalformedURLException {  
    Cap = new DesiredCapabilities();  
    Cap.setBrowserName("firefox");  
    Cap.setPlatform(Platform.WINDOWS);  
    dr = new RemoteWebDriver(new URL("http://localhost:5550/wd/hub"),  
                           Cap);  
    dr.navigate().to("http://orangehrm.qedgetech.com/symfony/web/index.php/auth/login");  
    dr.manage().window().maximize();  
    dr.findElement(By.xpath("//*[@id='txtUsername']")).sendKeys("Admin");  
    dr.findElement(By.xpath("//*[@id='txtPassword']")).sendKeys("Admin");  
    dr.findElement(By.id("btnLogin")).sendKeys(Keys.ENTER);  
}
```

⇒ How 2 run on multiple browsers:-

Create one XML file & define all ur parameters

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE suite SYSTEM "http://testing.org/Testing-1.0.dtd">  
<suite name="multiple" parallel="tests">  
    <test name="Firefox Test">  
        <parameters name="browser" value="Firefox"/>  
        <classes>  
            <class>  
                <package>resources.AppTest</package>  
            </class>  
        </classes>  
    </test>
```

To execute any testcase via maven make sure that class name should be AppTest.

Executing Testcase via MAVEN: - Right click on pom.xml run as maven test. After executing the ~~class~~ testcase refresh ur project under target folder make sure there Surefire reports.

Selenium Grid: It is a part of the Selenium Suite that specializes on running multiple tests across diff. browsers, operating systems & machines in parallel. Selenium Grid uses a hub node concept where u only run the test on a single machine called a hub but the execution will be done by different machines called nodes.

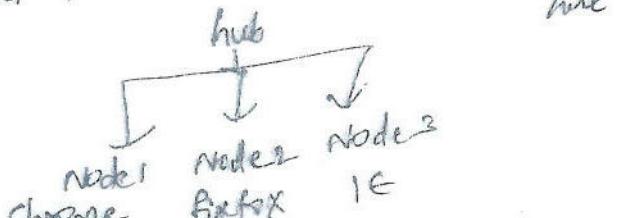
When to use selenium grid: Run ur tests against diff browsers operating systems & machines all at the same time. This will ensure that the appn. u r testing is fully compatible with a wide range of browser OS combinations. Save time in execution of ur test suites. If u setup selenium grid for run, say, 4 tests at a time then u would be able to finish the whole suite around 4 times faster.

The Hub: - Hub is the central point where u load ur tests into. There should be one hub in a grid. The hub is launched only on a single machine says a computer who o's is windows 7 & whose browser is IE. The machine containing the hub is where the tests will be run but u will see the browser being automated on the node.

The Nodes: - Nodes & the Selenium instances that will execute the tests that u loaded on the hub → There can be 1 or more nodes in a grid. → nodes can be launched on multiple machines with different platforms & browsers → The machines running the nodes need not be the same platform as that of the hub.

In D:\ any create a folder Selenium-Grid. In test folder maintained in latest JAR file of stand alone Server.

Running Hub: - Whenever u r practising Grid make sure that Hub should be always running. Select the folder in which u have JAR file. Press 'Shift' key in keyboard right click the mouse select 'open command window'



Creating Maven project: Go to File → New → Other → Expand Maven folder

→ Select Maven project → click next → next → next

✓ Group Id as package name

✓ Artifact Id as Project name

Under package remove projectname → Package = package name → click finish

Maven: - It is a software project management tool which provides a new concept of Object model (POM) Maven allows the developer to automate the process of the creation of the initial folder structure performing the compilation & testing & the packaging & deployment of the final product. It cuts down the good number of steps in build process & it makes it one step process to do a build.

Few glossaries around Maven

Maven Local Repository:- This is the place where Maven stores all the project jars files or libraries of dependencies. By default the folder name is '.m2' & by default the location in windows 7 is 'Libraries/Documents/.m2'.

Maven Central Repository:- It is the default location

<http://mvnrepository.com/> for maven to download all the project dependency libraries. For any library required in the project maven first looks into the '.m2' folder of local Repository if it doesn't find the required library then it looks in central & download the library into local repository.

Dependency Keyword:- Dependencies are the libraries which are required by the project. For example log4j jars, Apache poi jars, Selenium jars etc.

Dependencies are mentioned in the Maven POM.XML like this:

<dependency>

2 <groupId>org.seleniumhq.selenium</groupId>  package name

3 <artifactId>selenium-java</artifactId>  project name

4 <version>3.4.0</version>

</dependency>

TC002	Step 1 Step 2 Step 3 Step 4	Launch appr login as admin Enter all employee details & save Logout	Launch App Admin login User reg Logout
TC003	Step 1 2 3 4	Login as Admin Go to user management & Create new user for Employee Logout	Launch App Admin login User reg Logout
TC004	Step 1 2 3	Launch App Login as Employee Logout	Launch APP User login Logout

- Project / create a package 'Config'.
- Under src / create a package.
- Save xl sheet into Config package.
- Create login under Config package.
- Creating property file for Admin login, User creation & Employee creation.
- Under Config package create Property file as Repository Properties.
- Right click on package → New → File (8) General
- Define locator values for all the modules.
- Developing functional libraries for all the identified scenarios using return type methods.

Launch application (4) Employee creation

- (1) Admin Login (2) User "
- (3) Admin logout (4) Closing application browser. under test package
- Under src → Create a package, as (External) APPTest → Create class as HRM-functional Libraries

, without selecting main method

Package APPTest;

import java.io.FileInputStream;

public class HRM_FunctionalLibraries {

// Global Variables declaration

public static webdriver driver;

public static String url = "http://orangehrm.fedgetech.com/Sitecore/web/";

// u String uid = "Admin";

```
    my.sleep("wait for 5 seconds");
    Thread.sleep(5000);
    log.info("Close browser");
    driver.quit(); } }
```

Framework:- A Framework defines a set of rules or best practices which follow in a systematic way to achieve the desired results.

⇒ There are different types of test automation frameworks & the most common ones are :-

⇒ Modular Testing framework

⇒ Data driven " "

⇒ Keyword driven " "

⇒ Hybrid " "

⇒ Behavior Driven Development " "

Why Framework:- In test automation project we do perform diff. tasks by using diff. types of files. To organize & manage all the files & to finish all the tasks in a systematic approach we use a framework.

⇒ Advantages of using Test automation framework:

1. saves time & money. Automation Testing is faster in execution
2. Reusability of code. Create once & execute multiple times with less or no maintenance
3. easy reporting. It generates automatic reports after test execution
4. Easy for Compatibility testing. It enables parallel execution in combination of diff. OS & browser environments.
5. low cost maintenance - It is cheaper compared to manual testing in a long run.
6. Automated testing is more reliable
7. " " is " powerful & versatile
8. It is mostly used for regression testing. Supports execution of repeated test cases
9. Minimal manual Intervention. Test scripts can be run unattended
10. Max. Coverage. It helps to ↑ the test coverage.

Keyword Driven Framework:-

" " means as a step it is a step by step execution. It is very similar to manual testing format.

The way we have used to implement the test cases in manual testing the same can be followed in keyword driven with the help of JUnit & TestNG below.

```
Reporter.log("alertonerror, true);  
Thread.sleep(2000);  
driver.switchTo().alert().accept();
```

3

@after method
public void logout()
{ driver.quit(); } }

→ Passing Multiple values for Branch Creation using POM!

open excel sheet & pass testdata file, different branches.

BranchName	Address 1	Address 2	Address 3	Area	Zipcode	Country	State	City
Upper 1	Hyderabad	Hyderabad	Ahmedabad	Bangalore	56551	UK	England	London
Results	String style quotation & convert to ^{format} text							

Save the excelsheet into Test-Data folder under ur project.

Under Test Script package create one more TestNG class as DataDriver_Testing

→ Package Test Script

```
public class DataDriver_Testing  
{  
    public static WebDriver driver;  
    public static File xf;  
    public static FileInputStream fi;  
    public static XSSFWorkbook wb;  
    public static XSSFSheet sh;  
    public static FileOutputStream fo;
```

@Test

public void verify_multiple_branchcreation() throws InterruptedException, IOException
{
 String readfile = "D:\\AutomationFramework\\TestNG-Framework\\TestData";
 give path of xl sheet having data

f=new File(readfile);

```
fi=new FileInputStream(f);  
wb=new XSSFWorkbook(fi);  
sh=wb.getSheetAt(0);  
int rc=sh.getLastRowNum();  
int cc=sh.getRow(0).getLastCellNum();  
Reporter.log("no. of rows are :" + rc, true);  
XSSFRow row=sh.getRow(0);  
int cc1=row.getLastCellNum();  
Reporter.log("no. of columns in first row :" + cc1, true);
```

Selenium has built-in class called PageFactory, which they mainly created for Page object purpose, which allows us to store elements in Cache lookup.

Advantages of using Page Object Model:-

- Increases code reusability - Code to work with events of a page is written only once & used in different test cases.
- Improves code maintainability - Any UI changes ~~need~~ leads to updating the code in page object classes only leaving the test classes unaffected.
- makes code more readable & easy to understand.

Developing POM on Primus Bank:-

Develop POM for admin login, branch creation & branch updation, logout.

→ Under your project create a package as 'pageclassmodel'. Under that package create classes for adminlogin, Branch creation, Branch updation & Admin logout without selecting main method. (Go to Java class)

In every class define the locators for the elements & develop methods.

U can take object & String arguments (sendkeys) different.
create class in package & class extends to

→ Adminlogin

```
Package pageclassmodel;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
public class AdminLoginPage {
```

//Storing locators for login

```
@FindBy(xpath = "//*[@id='txtuid'])")
```

WebElement username;

 |
 +-- class
 +-- object

```
@FindBy(xpath = "//*[@id='txtpwd'])")
```

WebElement password;

 |
 +-- class
 +-- object

```
@FindBy(xpath = "//*[@id='login'])")
```

WebElement login;

arguments
 |
 +-- send keys

//develop non static method

```
public void Verify_login(string username, string password) throws InterruptedException
```

```
{ this.username.sendKeys(username);
```

```
    |  
    +-- class  
        +-- object
```

```
        " " (password);
```

```
    |  
    +-- class  
        +-- object
```

```
        login.click();
```

```
    |  
    +-- class  
        +-- object
```

```
    Thread.sleep(1000); }
```

```

} else if (Reporter.log ("Login unsuccess", true),
String text = dr.findElement(By.cssSelector("#spn message")).getText()
Reporter.log (text, true);
dr.quit(); } browser(); } }

```

@DataProvider :- If we use @DataProvider annotation for any method that means we are using that method as a data supplier. Configuration of @DataProvider annotated method must be like it always return Object[][] which we can use in @Test annotated method. The @Test method that wants to receive data from this data provider needs to use a data provider name equal to the name of this annotation.

Write a script to verify login functionality by accepting data from Data provider:-

Create TestNG class by using Data provider store ur data in Data provider.

⇒ Package 'ReTesting';

```

import org.testng.annotations.Test;
public class Logindata {

```

```

@DataProvider
public Object[][] Testdata_login() {
Object login [] [] = new Object [5] [2]; // 5 rows & 2 columns
login [0] [0] = "Admin"; // first row & 1st column
login [0] [1] = "mercury"; // 2nd row & 2nd column
login [1] [0] = "Admin";
login [1] [1] = "mercury";
login [2] [0] = "Admin";
login [2] [1] = "mercury";
login [3] [0] = "Admin";
login [3] [1] = "mercury";
login [4] [0] = "Admin";
login [4] [1] = "mercury";
return login;
}

```

writing text in to notepad from website.

```
Package Petesting;
import java.io.BufferedReader;
public class writingText_from_website {
    public static void main (String [] args) throws IOException {
        WebDriver driver = new ChromeDriver();
        driver.navigate().to ("http://SeleniumHQ.org/");
        driver.manage().window().maximize();
        File f = new File ("e://Notepad.txt");
        f.createNewFile();
        FileWriter fw = new FileWriter (f);
        BufferedWriter bw = new BufferedWriter (fw);
        String para1 = driver.findElement (By.xpath("//html/body"));
        String para2 = driver.findElement (By.xpath("give xpath"));
        bw.write (para1);
        bw.newLine();
        bw.write (para2);
        bw.flush();
        bw.close();
        driver.quit();
    }
}
```

→ write a script to read data from a file from one to last line.

```
Package Petesting;
import java.io.BufferedReader;
" " " .FileReader;
" " " .IOException;
public class Reading_TextFromnotepad {
    public static void main (String [] args) throws IOException {
        FileReader fr = new FileReader ("e://Notepad.txt");
        BufferedReader br = new BufferedReader (fr); → give path
        S.o.p.1 (br.readLine ());
        String str = "";
        while ((str = br.readLine ()) != null) { S.o.p.1 (str); }
    }
}
```

→ writing into Result column:-

```
package TestNG;
import java.io.File;
public class WritingText {
    public static void main (String[] args) throws IOException {
        File f = new File ("D:\\filepath");
        // read file f = data
        FileInputStream fi = new FileInputStream(f);
        // Create WB
        XSSFWorkBook wb = new XSSFWorkbook (fi);
        XSSFSheet ws = wb.getSheetAt (0);
        // Count no. of rows → take program from previous page till S.O.P.1 (username + " " + password)
        // Write into results column
        WS.getRow(1).createCell(2).setCellValue ("Pass");
        FileOutputStream fo = new FileOutputStream (f);
        wb.write (fo);
        fo.close ();
        wb.close ();
    }
}
```

Example! Write a script to Parameterize login functionality with multiple test-data using locators.

open Excel sheet & prepare locators & testdata for login

Username-locators	password-locators	password-testdata	password
/*[@name='username']	/*[@name='password']	admin	mercury
xpath	admin	Result	mercury
Sign-button	/*[@name='login']	Status	

Save excel sheet into Test Data folder.

C: Selenium Automation → TestNG Framework → Testdata project.

- ensure that under Referenced Library all poi jar files will get added (13 files)
- creating folder under project → Right click on project → New → Folder
 Folder name - Test-data → click finish
 any name.
- open XLS → Create 2 columns → use name & password → maintain date
 in diff. rows → save data XLS sheet in to Test-data folder under the project
 Browse → save location → local system.
- ### Ist class
- 1) FileInputStream : To read data from a file (XLS, XLSX sheet) → Eclipse path, FileInputStream fi = new FileInputStream fi = new
- creating class for Workbook.
- XSSF workbook wb = new XSSFWorkbook(fi);
- creating object for Worksheet.
- XSSFSheet ws1 = wb.getSheet("LoginData");
- (any variable) ↑ sheet name of sheet.
- getSheet() :- getting sheet from workbook with the help of sheet name
- XSSF getSheet (Sheet name);
- getSheetAt() :- getting sheet from workbook w/o index number.
 Index starts with (0)
 If no. of row no./col.no./sheet no. will always starts from '0'. which indicates column
- 1st , 2nd , 3rd , ... , nth number
- getLastRowNum() :- To count no. of Rows in a sheet.
- getLastCellNum() :- To count number of columns in a Row.
- getRow() :- To get specific Row from a sheet.
- getLastRow (Row number); reading
- getCell() :- To get specific column from a Row.
- Syntax :- getCell (Column number);
- columns get from Rows
 Rows from Sheet
- getStringCellValue() :- To get string contained data from column.
- getNumericCellValue() :- " " Integer values from cell/column. ↓
 alpha numeric string only.

Create TestNG class & access property file in to that class.

for login & Register functionality

→ Sent Email → ^{testng} package

(Parameters) Whenever u want to access parameter values from Configured XML file into ur test case.

→ Creating XML file & storing TestData for Login & Registered module

→ Right click on project → New → File → Give filename.XML
webtour.xml

→ click finish

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE SUITE SYSTEM "http://testing.org/testng-harness.dtd"
<suite name="Parameter webtour Suite" parallel="classes">
  <test name="Register & login Test">
```

```
    <!-- parameters for login -->
    <parameter name="username" value="admin"/>
    <parameter name="password" value="mercury"/>
```

```
  <classes>
    <class name="property_how25.webtour"/>
```

</classes>

<!-- Register & login -->

Define for Register also <!-- Register & login -->
parameters
webtour
<!-- Parameter webtour -->

→ Create one TestNG class & access located values from Property file & ^{Parameter} values from XML file.

→ When u create property file, parameter, run via XML file.

while executing run via XML file

Primesbank
branch creation
branch

choose ~~pos~~ package & class name in parameter.

switch to the frame by index:-

Index is one of the attributes for the iFrame through which u can switch to it.

Index of the iframe starts with '0'

Suppose if there are 100 frames in page, u can switch to the iframe by using index.

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

```
driver.switchTo().frame(1);
```

switch to the frame by Name or ID:

Name & ID are attributes of iframes through which u can switch to it.

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

```
or " " ("id of the element");
```

switch to the frame by WebElement:- u can even switch to the frame

using WebElement

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

Now to switch back to main frame?

have to come out of the frame

move back to the parent frame, u can either use switchTo().parentFrame()

- if u want to go back to the main (or most parent) frame

```
or use switchTo().defaultContent();
```

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

```
or " " defaultContent();
```

Object Repository

1/17
Object Identification! In selenium u can make it identify the object in the web application with the help of properties like id, name, linkname, class, xpath, tagname etc.

⇒ write a script to close multiple windows by using arraylist class.
open seleniumhq site → right click on downloads by using contextclick method.
→ open a new window by using Robot class → get all windows & store into
array list class by using getwindow handles method → close second window
by switching & close 1st window.

⇒ package pack1;
*import java.awt.AWTException;
public class UsingArrayListClass{
public static WebDriver driver;
@Test
public void verifyDownloads() throws InterruptedException, AWTException {
driver = new ChromeDriver();
driver.navigate().to("http://seleniumhq.org/");
driver.manage().window().maximize();
Actions ac = new Actions(driver);
String downloadLink
WebElement downloads = driver.findElement(By.xpath("//*/@id='J'"));
To moveToElement(downloads).contextClick().perform();
Thread.sleep(5000);
Robot robot = new Robot();
robot.keyPress(KeyEvent.VK_DOWN);
robot.keyRelease(KeyEvent.VK_DOWN);
Thread.sleep(5000);
robot.keyPress(KeyEvent.VK_DOWN);
robot.keyRelease(KeyEvent.VK_DOWN);
Thread.sleep(5000);
robot.keyPress(KeyEvent.VK_ENTER);
Get all windows
ArrayList<String> windows = new ArrayList<String>(driver.getWindowHandles());
driver.switchTo().window(windows.get(1)); // switch to 2nd window
System.out.println(driver.getTitle());
Thread.sleep(3000);

Iterator:-

Syntax:- `Iterator<datatype> iteratorreference = Setname.iterator();`

Note! here iterator's datatype & setname's datatype must be same.

Iterator: Iterator is an interface belongs to the Collection API which is used to handle sets, list & so on. Simply Iterator reference acts like a pointer.

useful methods in Iterator interface for doing testing as follows.

hasNext(): It checks whether Iterator reference has next element or not returns true if it has next element otherwise returns false.

next(): It gets element's data from the location which is pointed by Iterator reference.

→ Set avail

→ ArrayList class

It is a ~~Collection~~ of predefined class of collection API in java

Syntax:- `ArrayList<datatype> arrayname = new ArrayList<datatype>();`

It is used for ~~storing~~ dynamically growing Array.

For retrieving v will use index. Index starts with 0.

→ Add:

To add data ~~to~~ to this array v need to use below syntax.

Syntax:- `arrayname.add(data);`

→ Here data is of same data type's data by the time of declaration.

→ Get(1):

To access data from array list

Syntax:- `arrayname.get(indexno);`

→ Send Email:

write a script to close all advertisement child windows which are opened by webdriver one by one & switch to main window. click on Register link Click on I am First user & fill the form. & close the browser.

```
String myalert = driver.switchTo().alert().getText(),  
String message = alert.getText();
```

To store alert text

```
driver.switchTo().alert().accept();  
alert.accept();
```

To accept alert

```
driver.switchTo().alert().dismiss();  
alert.dismiss();
```

To dismiss Confirmation

```
driver.switchTo().alert().sendKeys("This is a John");  
alert.sendKeys("");
```

to type text in Textbox of prompt pop-up -

→ Handling Alert using Expected Conditions

```
Package Pak;  
import org.openqa.selenium.By;  
public class Using_Expected_Conditions {  
    public static WebDriver dr;  
    @Test  
    public void Verify_Alert() throws InterruptedException {  
        dr = new ChromeDriver();  
        dr.navigate().to("https://www.airtel.co.in/IndianRailways/");  
        dr.manage().window().maximize();  
        // click on search button without filling form  
        findElement(By.xpath("//*[@id='showdomestic']/div[6]")).click();  
        WebDriverWait wait = new WebDriverWait(dr, 40);  
        // wait for alert  
        wait.until(ExpectedConditions.alertIsPresent());  
        // getting text of alert  
        String alerttext = dr.switchTo().alert().getText();  
    }
```

```

// Take screenshot & store into variable
file src = ((TakeScreenshot) driver).GetScreenshotAs(OutputType.FILE);
// Copy screenshot into local system
FileUtils.CopyFile(src, new File("e://100clock/" + date + "homepage.jpg"));
↓
path - local system.
33

```

→ Executing multiple classes in one TestNG.xml file!

Right click on package which has TestNG class.

TestNG → Context to TestNG

In Location TextBox rename XML filename
 location: /TestNG-Framework/Testing.xml
 .. /suite.xml
 ↓ Class

SuiteName: Jagan

TestName: Result

Class Selection - classes

Select parallel mode as classes (methods - multiple methods
 Test - " " Test)
 click finish

XML file:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testing.org/testng-1.0.dtd">
```

S <suite name = "Jagan" parallel = "true" >

T <test name = "Result" >

Classes <classes>

<class name = "pack1.Second-Class" /> ↗ Can add multiple
 package classes here

< " " = "pack1.First-Class" /> ↗ package name & class

</classes> ↗ package name.

</test> <!-- Result -->

</suite> <!-- Jagan -->

Object creation for Robot class

Robot robot = new Robot();

~~key press ()~~

* f9! robot. Key.Press (KeyFront, VK_Down) - This method with press down arrow key of keyboard

→ mousePressEvent():

7. robot.mousePress() (Input Event, BUTTON_3_DOWN_MASK) - This method will press the right click of ur mouse.

→ mouseMove()!

→ mouseMove():
Eg. `robot.mouseMove(point.getX(), point.getY())` - This will move
mouse pointer to the specified X & Y coordinates.

- KeyRebels () -

KeyRelease() -
Eg. robot.KeyRelease(KeyEvent.VK_DOWN) - This method will release
down arrow key of Keyboard

- mouseRelease()

mouseRelease()
eg. robot.mouseRelease(InputEvent.BUTTON3_DOWN_MASK);
..... is rightclick ur mouse.

This method will release the right click of our mouse.

- To press down arrow key of Keyboard N use

press down arrow. →
C#shot, key press (key Event, VK_Down))

- (robot.keypress(KeyEvent.VK_DOWN))
- To press TAB key of keyboard (Use robot.keyPress(KeyEvent.VK_TAB))
- If " " ENTER " " " "

→ Sent emails :-

Take screenshot & store into variable

// Take screenshot
class fixture
{
public:
 void TakeScreenshot() {
 // Create a new instance of the WebDriver class
 WebDriver driver = new FirefoxDriver();

 // Set the URL of the website you want to capture
 driver.get("http://www.google.com");

 // Capture the screenshot and save it as a file
 driver.getScreenshotAs(OutputType.
 File);
 }
};

1 Copy variable screenshot into local system

```
1 Copy screenshot into local system  
fileutils.CopyFile(screenshot, newFile("D:\1\screenshot.jpg"));
```

① // To click on the element

action: moreToElement(element).click(); perform();

Note: we need to use perform() to execute the action.

(Using Action API,

② The following are the regularly used mouse & Keyboard events

Method: clickAndHold()

Purpose: clicks without releasing the current mouse location

③ M: contextclick()

P: performs a right click at the current mouse location

Actions action = new Actions(driver).contextClick(element);
action.build().perform();

④ Method: doubleclick()

P: performs a double click at the current mouse location

M: dragAndDrop(source, target)

Parameters: Source & Target

Purp: performs click and hold at the location of the source element & moves to the location of the target element then releases the mouse.

⑤ M: dragAndDropBy(source, x-offset, y-offset)

Param: Source, x-offset - horizontal move, y-offset - vertical move offset

Purp: performs click & hold at the location of the source element moves by a given offset, then releases the mouse.

M: release()

P: It releases the left mouse button at the current mouse location.

Keystrokes Methods

:=

Method: keyDown(modifier-key)

Param: Modifier-key | Keys ALT & Keys SHIFT & Keys CONTROL

Purp: performs a modifier key press, doesn't release the modifier key Subsequent interactions may assume its kept pressed.

sure for Vacations link with Priority 1

//
}

(2) after test (always Run - true) → It will execute even if test case pass/fail.

Public void logout() throws InterruptedException
Thread.sleep(4000);
Reporter.log("running logout", true);
driver.quit();

}

Explicit Wait :-

The explicit wait is used to tell the WebDriver to wait for certain conditions (Expected Conditions) or the maximum time exceeded by throwing an "Element not visible exception" exception.

The explicit wait is an intelligent kind of wait, but it can be applied only for specified elements. Explicit wait gives better options than that of an implicit wait as it will wait for dynamically loaded Ajax elements.

Once we declare explicit wait we have to use "ExpectedConditions" & we can configure how frequently we want to check the condition using FluentWait. These days while implementing we are using Thread.sleep() generally it is not recommended to use.

In the below example we are creating reference wait for "webElementWait" class & instantiating using "WebDriver" reference & we are giving a max-time frame in seconds.

Syntax:-

WebDriverWait wait = new WebDriverWait(WebDriverReference, Timeout);
→ object → to seconds.

Sathish
Prinamya

(64)

→ Find location
Right click on project, project
of the folder Select properties
Find location.

How to prioritize a Test Case:-

@Test (Priority = 0)

public void TC 1()

}

@Test (Priority = 1)

public void TC 2()

}

@Test (Priority = 2)

public void TC 3()

}

(*) (*) will have highest Priority. Whenever we write more than one test case in testNG it is always better to prioritize test case below
as it executes according to alphabetical order.

How to enable/disable test case:-

@Test(enabled = true)

public void TC1()

}

@Test(enabled = false)

public void TC2()

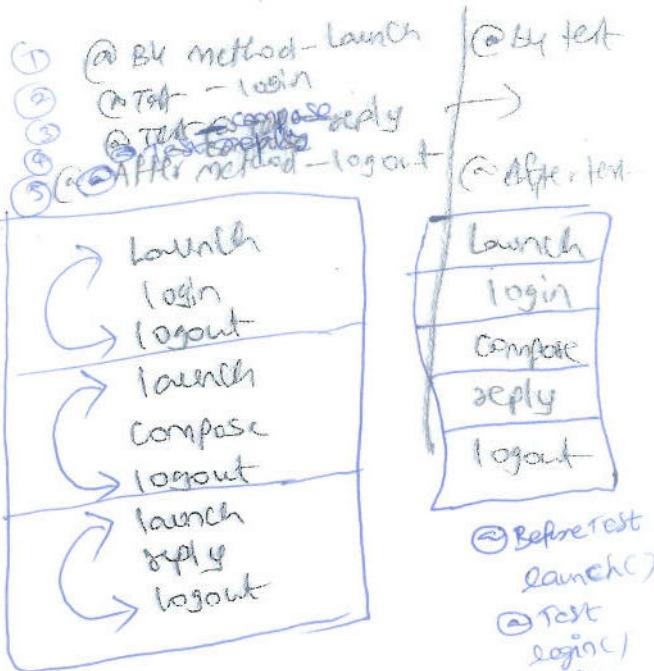
}

@Test(enabled = true)

public void TC3()

}

In above syntax TC 2 will avoid from execution.



@Before test
launch()
@Test
login()
@Test
compose()
@Test
reply()
@After test
logout()

Before Suite! A method which is marked with this annotation will run only once before all tests in the suite have run.

After Suite :- " " " "
" " after execution of all tests in the suite have run.

Before Groups! This annotated method will run before the first run of that specific group.

After Groups! " " " after all test methods of that group completes its execution.

Creating TestNG class :-

Right click on the project
Select TestNG at the bottom - Create TestNG class.

source folder < /TestNG/src
package - Testing - Examples
class name - first class
* src is compulsory for source folder
class should be under src

Select project name/src - Source folder
Give package name, class name

@ Before, @ After method annotation.

- click finish

```
Package TestNG - Examples;  
* import org.junit.Test, annotations.Test;  
public class FirstClass {
```

```
@ Before Method  
public void launch() {
```

```
Reporter.log("running launch", true);
```

True - show off in console
false - frontline

```
}
```

@ Test

1. TestNG, JUnit, Nunit are separate testing framework which is freely available in the market.
2. TestNG, JUnit mainly v can integrate into Java whereas Nunit works with C# only.
3. JUnit comes by default with Eclipse but TestNG v have to install an Addons.

4. JUnit has fewer annotations as compared to TestNG.
5. JUnit automatically does not create HTML report so v have to use ant build tool, but TestNG has rich HTML report generations even for single test case it creates 3 HTML reports.

⇒ What is TestNG & v have to use in Selenium
 → TestNG is a testing framework which is designed to cover all categories of tests! unit, functional, end-end, integration etc. we can combine TestNG with selenium & v can write Test cases in Eclipse using Java.

- Some of the TestNG feature as listed below
 1. **Annotation** - It supports multiple annotations at a various level
 2. Support for **data driven testing** (with @ Data provider)
 3. Support for **parameters**
 4. Generate **automatic reports**
 5. v can run failed test case only using testing.xml no need to run the full test suite in case of failure.
- b) Supported by a variety of tools & plugins (Eclipse, IDEA, Maven etc.)
- By using TestNG v can prioritise test cases. v can maintain dependency of test cases.
- we can execute no. of classes, Methods, Packages, groups under one single TestNG.XML file.
- There is no need to select main method for TestNG.
- All Annotations should be denoted with '@' symbol.

Select By value

Select listbox = new Select(driver.findElement(By.xpath("//")));
listbox.selectByValue("Italy");
It will select value by value = "Italy"

Select By Index

Select listbox = new Select(driver.findElement(By.xpath("//")));
listbox.selectByIndex(0);
It will select value by Index = 0 (first option).

Deselect by Visible Text

Select listbox = new Select(driver.findElement(By.xpath("//")));
listbox.deselectByVisibleText("Russia");
It will deselect option by visible Text = Russia from listbox

Deselect By Value

Select listbox = new Select(driver.findElement(By.xpath("//")));
listbox.deselectByValue("Mexico");
It will deselect option by value = Mexico from listbox

Deselect by Index

Select listbox = new Select(driver.findElement(By.xpath("//")));
listbox.deselectByIndex(5);
It will deselect option by Index = 5 from listbox

Deselect All

Select listbox = new Select(driver.findElement(By.xpath("//")));
listbox.deselectAll();
It will remove all selections from listbox.

isMultiple()

Select listbox = new Select(driver.findElement(By.xpath("//")));
boolean value = listbox.isMultiple();
It will return true if select box is multiselect else it will return
false if it is single item.

```
Scanner s = new Scanner (System.in);
for(int i=1; i<=5; i++) {
    driver = new ChromeDriver();
    driver.navigate().to("https://mail.rediff.com/cgi-bin/login.cgi");
    driver.manage().window().maximize();
    driver.manage().cookies().implicitlyWait(20, TimeUnit.SECONDS);
    System.out.println("enter username");
    String user = s.nextLine();
    System.out.println("enter password");
    String pass = s.nextLine();
    WebElement username = driver.findElement(By.xpath("//input[@name='login']"));
    boolean isEnabled = username.isEnabled();
    System.out.println(isEnabled);
    username.sendKeys(user);
    Thread.sleep(4000);
    WebElement password = driver.findElement(By.xpath("//input[@name='password']"));
    boolean passEnabled = password.isEnabled();
    System.out.println(passEnabled);
    password.sendKeys(pass);
    Thread.sleep(4000);
    WebElement checkbox = driver.findElement(By.xpath("//input[@name='remember_me']"));
    boolean chkEnabled = checkbox.isEnabled();
    System.out.println(chkEnabled);
    if(checkbox.isSelected())
        checkbox.click();
    Thread.sleep(4000);
```

```
Thread.sleep(3000);  
driver.findElement(By.xpath("//input[@name='password']"))  
Thread.sleep(3000);  
driver.findElement(By.xpath("//input[@name='login']")).SendKeys  
(" "));  
Thread.sleep(3000);  
driver.quit();
```

↳

Download **Xpath checker** for google chrome to identify perfect Xpath.

site customize Xpath value for login of facebook.

newtours.demoaut.com

Customized - .^{no} (dot)
Not customized - . (dot)

Condition Methods:-

Condition Methods are used to verify boolean type.

isDisplayed Command(method) :-

object.isDisplayed() - This method checks the visibility of a web element & doesn't allow any parameters but its return type is a boolean value.

This method will return value if the element is present & visible on the page. It'll throw a NoSuchElementException exception if the element is not available. If it is available but kept as hidden then the method will return false.

```
WebElement user = driver.findElement(By.id("user"));  
boolean isDisplayed = user.isDisplayed();
```

of path based

Text(): This mechanism is used to locate an element based on the text available on a webpage

//*[text() = 'Gmail']

List(): selects the first element (of mentioned type) out of all input element present

To identify the element (in text field) "four current email addresses" we could use the below Xpath.

find Element (By: xpath ("(//input[@type = 'text']) [last()]"))

position(): selects the element out of all input element present

depending on the position number provided

In below given Xpath, `[@type = 'text']` will locate text field & `[position () = 2]` will locate text field which is located on 2nd position from the top.

find Element (By: xpath ("(//input[@type = 'text']) [position () = 2]"))

06

find Element (By: xpath ("(//input[@type = 'text']) [2]"))

Finding elements using index! Index starts with '0' zero
By providing the index position in the square brackets, we could move to the nth element. Based on the below Xpath we could identify the last name field.

find Element (By: xpath ("//label[2]"))

Following! By using this we could select every field on the webpage after the closing tag of the current node

Xpath of the First Name field is as follows

//*[@id = 'First Name']

How to write Dynamic Xpath in Selenium

① using Single slash:

Absolute X-path

This mechanism is also known as find elements using Absolute Xpath.
Single slash is used to create Xpath with absolute path ie the Xpath would be created to start selection from the document node/ start node/ parent node.

Syntax:-

html/body/div[1]/div[2]/div[2]/div[3]/form/div[1]/input[1]

② Double slash:

Relative X-path

This mechanism is also known as finding element using Relative Xpath.
Double slash is used to create Xpath with relative path ie the Xpath would be created to start selection from anywhere within the document - search in a whole page (Ans) for the preceding string.

Syntax:-

// form/div[1]/div/div[1]/div/div[1]/input[1]

③ Single Attribute:

You could write the syntax in 2 ways as mentioned below:
Including or excluding HTML Tag. If u want to exclude HTML Tag then u need to use *

Syntax:-

tagname[@attribute-name='attribute-value']

//* [@attribute-name='attribute-value']

Note: * after double slash is to match any tag with the desired text - Xpath based on above HTML:

// input[@id='Email'] or //*[@id='Email']

④ Multiple Attributes:-

(To avoid duplicates)

Syntax: Tagname[@attribute-name1='attribute-value1'][@attribute-name2='attribute-value2'] (Ans)

//* [@attribute-name] = 'attribute-value'] ["

```
// maximize the browser  
driver.manage().window().maximize();  
// Implicit wait for 10 seconds  
driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);  
// To launch url in a browser  
driver.get("http://flipkart.com/");  
// wait for 5 seconds - not needed.  
Thread.sleep(5000);  
// used tagname method to collect the list of items with tagname "a"  
// findElements - to find all the elements with in the current page.  
// It returns a list of all webelements or an empty list if  
nothing matches  
List<WebElement> links = driver.findElements(By.tagName("a"));  
// To print the total no. of links.  
System.out.println("Total links are "+links.size());  
// used for loop to count no. of links  
for(int i=0; i<links.size(); i++) {  
    WebElement element = links.get(i);  
    // By using "href" attribute, we can get the url of the required  
    // link.  
    String url = element.getAttribute("href");  
    // calling verifyLink() method here. Putting the parameter as url  
    // which we collected in the above link  
    // See the detailed functionality of the verifyLink(url) method  
    // below // calling static method with method name  
    verifyLink(url);
```

3
3

Write a script by using above commands in order to perform backward, forward, refresh.

Package WebDriver - Examples;

```
#import org.openqa.selenium.*;  
public class Navigation - Commands {  
    public void verifyNavigation() {
```

try {

```
    WebDriver driver = new ChromeDriver();  
    driver.navigate().to("http://newtours.demoaut.com");
```

```
    System.out.println("first page title [1]:" + driver.getTitle());
```

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

1 click on Register link.

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

```
    System.out.println("first page title [2]:" + driver.getTitle());
```

```
    Thread.sleep(4000);
```

```
    driver.navigate().back();
```

```
    System.out.println("first page title [3]:" + driver.getTitle());
```

```
    Thread.sleep(4000);
```

```
    driver.navigate().forward();
```

```
    System.out.println("first page title [4]:" + driver.getTitle());
```

```
    Thread.sleep(4000);
```

```
    driver.navigate().refresh();
```

```
    Thread.sleep(4000);
```

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

```
} catch(Throwable t) {
```

```
    System.out.println(t.getMessage());
```

```
}
```

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

```
    Navigation - Commands n = new NavigationCommands();
```

```
    n.verifyNavigation();
```

```
} }
```

444

List! It is a Java Collection of API (Appn prog's Interface) Library.

Write a script to get the collection of all links in a page by
package webdriver Examples;

```
import java.util.List;
import org.openqa.selenium.By;
import " " , WebDriver;
" " , WebElement;
" " , chrome.ChromeDriver;
" " , Using - findElements();
public class Using - findElements {
    public WebDriver driver;
    public void verifyLinks() {
        driver = new ChromeDriver();
        driver.get ("http://rediff.com");
        driver.manage().window().maximize();
        //get all links & store into webElement
        List<WebElement> links = driver.findElements (By.tagName ("a"));
        System.out.println ("no. of links are :" + links.size ());
        // iterate all links
        for (WebElement eachLink : links) {
            String linkName = eachLink.getText();
            System.out.println (linkName);
        }
    }
    public static void main (String [] args) {
        Using - findElements obj = new Using - findElements ();
        obj.verifyLinks();
    }
}
```

```
WebElement Name = driver.findElement(By.id("Name"));
Point point = name.getLocation();
String strLine = System.getProperty("line.separator");
System.out.println("X coordinate #" + point.x + strLine + "Y coordinate #" +
                    point.y);
```

```
package WebDriver_Examples;
import org.openqa.selenium.*;
import org.openqa.selenium.chrome.ChromeDriver;
public class Get	RuntimeMethods {
    public static WebDriver dr;
    public void launch(){
        dr = new ChromeDriver();
        dr.get("http://facebook.com");
        dr.manage().window().maximize();
    }
    public void verify - GetText(){
        String text = dr.findElement(By.xpath("//div[@id='content']//div[" +
            "div[@id='text']")].getText();
        System.out.println(text);
        String visibleText = dr.findElement(By.tagName("body")).getText();
        if(visibleText.contains("birthday"))
            System.out.println("text is within html page");
        else
            System.out.println("text is not within html page");
        System.out.println("-----");
    }
    public void verify - GetAttribute(){
        // store first name textbox
        WebElement firstname = dr.findElement(By.id("u_0_g")).get
            TagName();
    }
}
```

For Chrome browser → ^{more} Internet options → security
Uncheck -

Inherit force using static

Package web_driver - chapter 1;
import org.openqa.selenium.By;

public class FB extends GetCommands {
 public void verify_login(String username, String password)
{
 driver.findElement(By.name("input-username")).sendKeys(username);
 driver.findElement(By.id("password")).sendKeys(password);
 driver.findElement(By.id("u_0_2")).click();
 }
}

public static void main(String[] args) throws Exception {
 GetCommands abc = new GetCommands();
 abc.geturl("http://facebook.com");
 Facebook f = new Facebook();
 f.verify_login("test@gmail.com", "123");
}

method
GetText() :-
GetText() - Returns ^{visible} Text of an element in a webpage.

syntax:-
String text = driver.findElement(By.id(" ")).getText();

Get attribute("Value") :-

It returns value of the attribute passed as a parameter.

HTML:
 named <input>
tag

```
WebElement user = driver.findElement(By.id("user"));
user.sendKeys("abc");
```

(8)

```
driver.findElement(By.id("user")).clear();
```

click()

void click() - This method performs click action on the web element like links, check boxes, buttons, radio button etc.

```
WebElement link = driver.findElement(By.linkText("Gmail"));
link.click();
```

(or)

```
driver.findElement(By.linkText("Gmail")).click();
```

Package webdriver - Examples:

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

```
public class GetCommands{
```

```
public WebDriver driver;
```

```
public void launch(String url){
```

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

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

```
    driver.get(url);
```

}

```
public void Verify_GetTitle(){
```

```
String pageTitle = driver.getTitle();
```

```
if (pageTitle.equalsIgnoreCase("Gmail"))
```

```
{ System.out.println("title is matching"); }
```

}

```
else {
```

```
System.out.println("title is not matching"); }
```

```

    // filling login forms
    driver.findElement(By.id("txt_username")).sendKeys("Admin");
    //           (By.name("txt_Password")).sendKeys("Admin");
    //           (By.cssSelector("#btalogin")).click();
    Thread.sleep(3000);
    System.out.println("Second Page title [" + driver.getTitle());
    System.out.println(" " + driver.getCurrentUrl());
    driver.close();
}

```

 You can use above script for Firefox or IE browser by changing class name.

 2.5.3 does not work for below html script
 3.53 only works.

Package WebDriver_ Examples;

```

import org.openqa.selenium.*;
//           " "           ". htmlUnitDriver ";
public class HtmlUnit_Browser {
  public WebDriver obj;
  public void verify() throws InterruptedException {
    obj = new HtmlUnitDriver();
    // launch url
    obj.get("http://Gmail.com");
    System.out.println("title of page is !! " + obj.getTitle());
    System.out.println("length title of page is !! " + obj.getTitle().length());
    System.out.println("current url is !! " + obj.getCurrentUrl());
}

```

added in btrn for this path will no longer work.

Ex:- If the path v defined as

html/head/body/table/tbody/tr/th

If tree is a btr that has added btrn body & table as below

html/head/body/foor/table/tbody/tr/th

The first path will not work as 'foor' tag added in btrn.

Relative Xpath:-

A " " is one whole the path starts from the node of our choice - it doesn't need to start from the root node.

It starts with Double forward slash(//)

start!:-

//table/tbody/tr/th

Advantage of using relative xpath is we don't need to mention the long xpath & we can start from the middle tag in btrn.

Disadvantage of using xpath is it will take more time in identifying the element as we specify the partial path not (exact path) if there are multiple elements for the same path it will select the 1st element that is identified.

BROWSER COMMANDS:-

① Get Command:-

Purpose:- This command is used to open a new web page in the current browser.

```
driver.get("http://google.com")
```

② Get Title Command:-

Purpose:- This command is used to get the title of the current page.

```
driver.getTitle()
```

① By Partial Link Text	locates elements that contain the given link text	find element (By.partialLinkText("RTGn"))
② By tag Name	locates elements by their tag name	find element (By.tagName("a"))
③ By . X Path	locates elements via Xpath	find element (By.xpath(".//"))

Selenium path allows only below:-

Single forward
Double backward //

store driver in environment variables or project folder.

**
firefox - 55 version - standalone 3.15.3.1

3.15.3 - Java - Selenium

gecko driver - 19 - 0.19.0

Send Path
GetPath
Forward Path
Backward Path
Refresh Right
Refered Web

Inspecting By - X Path values in chrome browser :-

Open a chrome browser - Open application (FB, Gmail etc) - select the element
right click the mouse - select inspect window
Right click on highlighted blue word - copy Xpath - a
copy selected -

copy & paste
Copy to Eclipse

in chrome - download Xpath helper! -

- go to google.com - google search - type

- download Xpath helper

click very first link

- Xpath helper page - click on add to chrome

- click add extension

- click add to firefox

- " Install

- Restart now

How to Inspect Xpath & CSS selector using firepath.

Inspecting Xpath & CSS selector "

Open on current

Select element

Right click mouse

Select inspect in firepath →

Go to firepath - Select Generate absolute Xpath " [@id='emp1']
(8) .. Show with Dom

→ ~~Xpath~~ HTML ---

Xpath ↴

css

Xpath checker for Google chrome

Locating GUI elements in Webpage:-

To find elements in webpage we can use the below 2 methods:

① Find element!

② " Elements"

Find element or (1) method:

You can use this command to access any single element on the web page.

It returns the object of the first matching element of the specified location.

It throws a NoSuchElementException exception when it fails to find if the element. Its certain cases as follows:-

= driver.findElement(By.id("user"));

↳ Value
Proxy type

- * One can implement one or more interfaces at a time.
 - we can't create an object for an interface
 - we can create an object for a class which is implementing the interface & then give the reference of the interface.

Syntax: interface Interface name = new class name()

Eg: WebDriver **wd** = new FirefoxDriver();

- By declaring like this one can access all the webdrivers stubs as well as only overridden stub from class name firefox driver.

14/11/17 version
Firefox - 47.0.2 - good (st-st) - uncheck the updates

/firefox 50 version - 3.0.0 - selenium server Jar file standalone

/ " 47 " { 2.5.3 - "
2.53 - "
2.53.1 - "

Downloading Drivers:

Go to selenium HQ site. click on download. click on previous releases.
Internet respect to jar file version. download IDE server (32/64bit)

3.0.0.1
2.53.1 { 3.0.0.1
2.53.1 }

Go back to HQ site. click on broad.
under third party browser drivers! click on
mozilla @ gecko driver version - 0.19.1
download gecko driver by version - 14 (2.53.1)

download for
→ Google chrome - 2.33
chrome win 32 zip - supports 64bit

Project	→ jar file standalone
Class	chromedriver.exe
	gecko " .exe
	IE Driver servise.exe

Inheritance :-

Inheritance (parent-child) is very useful concept of Java object oriented programming language by which we can reuse the code of parent class. Inheritance is providing us a mechanism by which we can inherit the properties of parent class into a child class.

Example:- Audi class Is child class of car class then Audi class can access / use all the non private properties (methods, variables etc) of car class.

e.g:- extends keyword is used to inherit child class from parent class using inheritance we can reuse the code of parent class in to child class so that size of code will ↓. maintenance of code will be also very easy.

Under package pack1 create parent class as 'Car' w/o selecting main method.

```
Package pack1;
public class Car { // Car class is parent class of audi class
    private string type = "vehicle";
    public static int wheels = 4;
    public string color = "blue";
    string fuel = "petrol";
    public string getFuel() {
        return fuel;
    }
}
```

```
protected void seats() {
    int seat = 4;
    System.out.println("Car seats = " + seat);
}
```

Creating child class under same package create child class as Audi by using main method.

```
Package pack1;
public class Audi extends Car { // Audi is child class of Car class.
    public int speed = 150;
```

Diff. b/w static & non static method!

- We can call static methods directly while we cannot call non static methods directly. You need to create & instantiate an object of class for calling non static methods.
- ⇒ Non static stuff (methods, variables) cannot be accessible inside static methods means we can access only static stuff inside "static methods" opposite to it, non static method do not have any such restrictions. We can access static & non static both kind of stuffs inside non static methods.
- static methods is associated with the class while non static method is associated with an object.

Static method:

```
package java - example;  
public class static - method { // class name  
    public static int a, b, c; // global declaration  
    public static void add(int a, int b) // static method  
    {  
        c = a + b;  
        System.out.println(c);  
    }  
    public static void dis(int a, int b)  
    {  
        c = a * b;  
        System.out.println(c);  
    }  
    public static void mul()  
    {  
        a = 30; b = 5;  
        c = a * b;  
        System.out.println(c);  
    }  
    public void sub sub() // non static method  
    {  
        a = 10;  
        b = 5;  
        c = a - b;  
        System.out.println(c);  
    }  
}
```

(2) int x = 15;
 int y = 10;
 int z = x - y;
 System.out.println(z);

mul(); // calling static stuff in non static method

for repeatability, reusability or modularity.

⑨ split :- splits strings into array variable.

String str = "I wait for You";

String var[] = str.split(" ");

System.out.println();

for (String each : var) {

System.out.println(each);

}

System.out.println();

String s = "HCL@DELOITTE@IBM@OPTUM@KPMG";

String array var[] = s.split("@");

System.out.println(Arrays.toString(var));

System.out.println();

for (String eachCompany : arrayVar) {

System.out.println(eachCompany);

}

⑩ charAt(): Returns character for a given position
in a string. Starts with 0
Index no.s " " or n^o

String c = "i am good";

System.out.println(c.charAt(6)); // g

⑪

⑫ It converts Integer to String

Value of : int i = 75;

String val1 = String.valueOf(i);

System.out.println("Value of String val1 is " + val1);

Input = Integer.
Output = String

⑬ It converts String to int

String val1 = "50";

int i = Integer.parseInt(val1);

System.out.println("Value of int i is " + i);

I/P = String

O/P = Integer

called exception: while developing the code if Java itself is giving the hint that there may be an exception then such type of exceptions is known as checked exception.
In called ?? During the execution if Java can't do something then such type of exceptions are known as unchecked exception.

try{ } - catch{ } : Whenever we are developing a program if at all we have a doubt in any block of program which may lead to exception then it is better to keep that block in try & if exception occurs then what should happen we will write in catch block.

```
try {  
    block of code  
} catch (Throwable t) {  
    t.printStackTrace();  
    t.printStackTrace();  
}
```

finally{ } : the script present in the finally block will be executed at any cost irrespective of try block & catch execution is successful or not.

e.g.: finally {System.out.println("! oops! (finally) executed");}
Throwable: It is a super class of both Exception & Error class for catching an exception or error.
we can simply write "Throwable t" in catch block instead of writing exception or error.

```
try {  
    catch (Throwable t) {  
        System.out.println(t.getMessage()); // it will print only else got in block of code  
        t.printStackTrace(); // " " " both else subtype of exception  
        t.printStackTrace();  
    }  
}
```

```
T try {  
    int x=5, y=0, z;  
    z=x/y;  
    System.out(z);  
} catch (Throwable t) {  
    System.out(t.getMessage()); // t.printStackTrace();  
}
```

Multi-dimension array! to store more no. of multiple values of same datatype.

Syn: datatype [][] arrayname = new datatype [rowsize][columnszie];
datatype " [] = "

Object Array! It is used for storing multiple values of diff. data types

Syn: Object [][] arrayname = new object [rowsize][colszie];
object array name [] = "

ArrayList class! - It is used for simulating dynamically growing array

Syn: ArrayList <datatype> arrayname = new ArrayList <datatype>();

API - App Programming Interface

single dimension array:-

```
int num [] = new int [5];  
num [0] = 100;  
num [1] = 456;  
num [2] = 789;  
num [3] = 212;  
num [4] = 313;
```

* columns are in rows.
Rows are in tables.

```
System.out.println ("length of array is :" + num.length);  
System.out.println ("5th index value is :" + num[4]);
```

```
for (int eachnum : num)
```

```
{  
    System.out.println (eachnum);
```

syn (" --- using object array ");
Object str [] = new Object [6];

```
str [0] = 1000;
```

```
str [1] = "Hello";
```

```
str [2] = 'd';
```

```
str [3] = 123.456;
```

```
str [4] = "Selenium";
```

```
str [5] = 456;
```

```
System.out.println ("length of array is :" + str.length);  
System.out.println ("4th value is :" + str[4]);
```

```
for (Object obj : str) {
```

```

import java.util.Scanner;
Scanner s = new Scanner(System.in);
for (int i=1; i<=5; i++) {
    System.out.println("enter marks");
    int marks = s.nextInt();
    if (marks >= 35) {
        System.out.println("Pass : " + marks);
    } else {
        System.out.println("Fail : " + marks);
    }
}

```

Nested if :-

when u want to check block of code with multiple conditions

```

if (Condition)
{
    stmt
}
else if (Condition)
{
    stmt
}
else if (Cond){stmt}
else {stmt}

```

Syntax
switch (expression) {
 case 1:
 stmt;
 break;
 case 2:
 stmt;
 break;
 case 3:
 stmt;
 break;
 default:
 stmt;
 break;
}

Switch stmt.
It is alternate to If
Condition. It checks our
given data with multiple

Cases. Default always

executes when data is not matching with given cases. Every case should end
with ; . Every case will have a break stmt to come out of the case.



→ Point sum of values from 1 to 20.

```
{ int sum = 1;
for( int i=1; i<=20; i++ ) {
    sum = sum + i;
    cout << sum;
}
```

while loop :- It repeats block of statements while condition is true.
Point even nos from 1 - 20

Syntax:-
Initialization
while (condition) {
 starts
 = increment / decrement
}

g! int i = 0;
 while(i<=20)
 {
 cout << i;
 i += 2;
 }

Do while :- It repeats block of starts while condition is true &
irrespective of the condition it executes a block of starts atleast once.
Point odd nos from 1 - 20.

g! int i = 1;
 do
 {
 cout << i;
 i += 2;
 } while(i<=20);
 }

) For each loop :- When we want to return anything in collection

array list
for (datatype variable : arrayname)
{
 starts
}

Types of variables

Instance variable - A variable that is declared inside the class but outside the method.

Local variable - A variable that is declared inside the method.

Static variable - " " " as static, it cannot be local.

Operators in JAVA -

Operators are used to perform mathematical comparison & logical operations. In JAVA below are the operators.

1. Assignment operators $=$
2. Arithmetic $+$ $-$ $*$ $/$ $\%$
3. Relational / Comparison $<$ $>$ $=$ \neq
4. Logical $\&$ $\|$ \neg

) Assignment operators $=$: This are used to assign values

int i=10;

i=10;

it=20;

i-=20;

int i=10;

i+=10;

i*=10;

i/=10;

i\=10;

) Arithmetic operators $+$ $-$ $*$ $/$ $\%$ (These operators are used for performing mathematical operations.)

1. Addition $+$ (for addition & string concatenation)
2. Subtraction $-$ (for int & $real$)
3. Multiplication $*$
4. Division $/$
5. modulus $\%$ Return Remainder value.
6. Increment $++$ ↑ by one value.
7. Decrement $--$ ↓ by one value.
8. $+=2$ ↑ by two value.

Ex: public static void main (String [] args) {

int i=10;

i++;

System.out.println ("increase by one line : " + i); // 11

i+=2;

System.out.println ("two time : " + i); // 13

i--;

// ("decrease by one line : " + i); // 12

i-=5;

// (" five line : " + i); // 33

⑩

Creating JAVA class:-

- Select JRE
- Go to file menu → new → class
- Package name in **Package**
- Class name in **Name** Textbox
- Select ~~public~~ main method public static void main
- Finish

Syntax Rules for Java Class:-

1. Every class should start with alphabet.
2. first letter of class name should be in Capital
3. class name should not contain any special characters except underscore.
4. " " " " be unique.
5. Every start in a class should contains {
6. " " class should contain open, close } braces.

Comments!

Single line use double forward slash - //abcd
 ; multiple " /* abc.cde.cfg
 bed
 cfg */

Select data
 New - Source → Add / Remove block comment

DATA TYPES in JAVA!

1. " " Represents type of data or memory allocation, etc.
2. Java there are 2 types of data types
3. primitive Data types
4. Non " "

primitive Data types (8 types)!

Integer types → memory allocation (size)

) byte (8 bit)
 data type variable;

int value;
 data type variable = value;

byte (8bit)

byte a = 100;

) short (16 bit)

short a = 10000;

3) int (32 bits)

int a = 1000000;

B ① byte (8 bit)

B ② boolean (undefined, depends on JVM)

C ③ char (16 bits)

④ short (16 bit)

⑤ int (32 bit)

⑥ float (32 bit)

⑦ double (64 bit)

⑧ Long (64 bit)

⑨ char (16 bits)

char a = 'Z';

→ to store single character.

Rational types (numbers with decimal places)

⑩ float (32 bits)

float a = 1.2f;

6) double (64 bits)

double a = 19.3456f;

⑪

boolean a = true;

⑫ ✓

Drawbacks:-

- webdriver is not having built-in Reports generation facility.
- It won't support new browsers readily.

Required Software :-

- System bit - My Computer → Properties → System type 32-bit / 64-bit
- Downloading JDK: Java development kit
<http://oracle.com/technetwork/java/javase/downloads/jdk8-downloads-213315.html>
- Accept T&C
- Windows X86 = 32bit
- " X64 = 64bit
- After downloading install that in ur system.

Command prompt
 Cmd ➔ java -version

Eclipse ?

It is a platform & editor of workspace which is used for developing web driver scripts
 RC scripts, GRID scripts.

Downloading Eclipse:-
<https://eclipse.org/downloads/packages/releases/neon/3> - Neon

- https://eclipse.org/downloads/packages/releases/neon/3
- click on System bit (32/64bit) → download.

After downloading extract it into a folder.
Setting the Class Path for JAVA :-

Right click on Computer → Properties → Advanced System Settings → Environment variables

under System variables click on New →
 Variable name = JAVA_HOME (in caps)
 Variable value = C:\Program Files\Java\JDK

(JDK path) → copy the path of
 JDK
 Paste to variable

Click OK → JDK = 1.8 version.

under System variables' edit
 don't delete existing path, (if needed)
 End of path add C:\Program Files\Java\JDK\Bin
 Click OK, OK, OK → Binding of JDK & paste it.

click OK, OK, OK → Copy & Paste Eclipse
 Creating Folder structure!

Automation - Framework
 In C:\ or D:\ driver create automation-Framework folder → Copy & Paste extracted eclipse
 folder. into ur " " Create another folder library in Selenium Jar files.
 Library is used for storing

Selenium IDE - ~~firefox 47 version~~
 Supports

** Supports only firefox
 IDE code to
 Export to Java Testing

Selenium IDE:-

IDE stands for Integrated development environment

It is to record & run the script

It is an add-on for Firefox (we can install & record in Firefox)

It is a recorder for user actions (for each & every action it will know what exactly).

It is a recorder for user actions (for each & every action it will generate the script).

If By default it will generate script in html language. If required we can convert to other supported languages like Java, c#, etc.

The recorded scripts can be run against other browsers also with the help of RC or web driver.

Installing Selenium IDE:

→ Open Firefox browser (47 version). Enter below url

→ <http://seleniumhq.org/>

→ click on download

→ <http://addons.mozilla.org>

→ Selenium IDE - download from addons.mozilla.org

→ click on Add to Firefox

→ click on Install

→ click on Install

→ click - Restart.

→ Show on Tools

way to open How to open S.IDE)

1) Open Firefox enter <http://newtours.demoaut.com/>

2) Tools → Selenium IDE.

3) By default it will be in recording

4) Record - perform some action

username - admin
 password - mercury

File → file → save
 File - new test case

- Large & critical projects
- Projects that require testing the same areas frequently
- Requirements not changing frequently
- Accepting the application for load & performance with many virtual users
- stable software with manual testing
- Availability of time

Suite: Combination = Group of - Set of

What is Automation strategy / approach / automation life cycle to automate one project

- > check first whether the application is compatible or not
- Execute or review all the test cases atleast once to get the functional knowledge on the app & also to identify the test cases which are feasible for automation
- Provide the automation estimation for all the identified test cases
- Define & identify the framework for the project
- Develop the scripts for all the identified test cases
- Once the scripts are stable then integrate in a hierarchical order to execute
- If any new build released update the scripts as per the new build
- Execute the scripts & capture the results in a result file
- Analyse the results & identify the bugs
- Report the bugs manually to the developers by any reporting tool like bugzilla, etc, Jira etc

Selenium Components!

- Selenium is built on below components
- Selenium IDE (Integration development environment)
 - 2) " RC
 - 3) Selenium Grid
 - 4) Selenium Web driver
 - 5) Selenium droid & Appium (mobile testing)

