**Firefox Browser:**

driver = new FirefoxDriver();

**Chrome Browser:**

driver = new ChromeDriver();

System.setProperty("webdriver.chrome.driver", "Path/chromedriver.exe");

**IE Browser:**

driver = new InternetExplorerDriver();

System.setProperty("webdriver.ie.driver", "Path/IEDriverServer.exe");

**Headless Browser:**

HtmlUnitDriver unitDriver = new HtmlUnitDriver();

driver.setJavascriptEnabled(true);

**Driver Options:**

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

**Which is the super interface of webdriver:**

SearchContext

**FirefoxDriver is class or an interface and from where is it inherited:**

FirefoxDriver is a class. It implements all the methods of WebDriver interface

**Exceptions:**

NoSuchElementException : FindBy method can’t find the element.

StaleElementReferenceException : This tells that element is no longer appearing on the DOM page.

ElementNotVisibleException: Thrown to indicate that although an element is present on the DOM, it is not visible, and so is not able to be interacted with

ElementNotSelectableException: Thrown to indicate that may be the element is disabled, and so is not able to select.

NoAlertPresentException: Alert not present

NoSuchWindowException: When ever driver tries to switch to the window which is not available

TimeoutException: Thrown to indicate that although an element is present on the DOM, it is not visible, and so is not able to be interacted with

**Waits:**

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

WebDriverWait wait = new WebDriverWait(drv,30);

wait.until(ExpectedConditions.visibilityOfElementLocated(ele));

[presenceOfElementLocated](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/support/ui/ExpectedConditions.html#presenceOfElementLocated-org.openqa.selenium.By-)([By](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/By.html) locator)

[invisibilityOfElementLocated](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/support/ui/ExpectedConditions.html#invisibilityOfElementLocated-org.openqa.selenium.By-)([By](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/By.html) locator)

[frameToBeAvailableAndSwitchToIt](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/support/ui/ExpectedConditions.html#frameToBeAvailableAndSwitchToIt-org.openqa.selenium.By-)([By](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/By.html) locator)

alertIsPresent()

[elementToBeSelected](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/support/ui/ExpectedConditions.html#elementToBeSelected-org.openqa.selenium.By-)([By](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/By.html) locator)

[elementToBeClickable](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/support/ui/ExpectedConditions.html#elementToBeClickable-org.openqa.selenium.By-)([By](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/By.html) locator)

polling Time = 500 MilliSeconds

**Element Identification:**

driver.findElement(By.id("HTMLid"));

driver.findElement(By.name("HTMLname"));

driver.findElement(By.cssSelector("cssLocator"));

driver.findElement(By. className ("CalssName”));

driver.findElement(By. linkText ("LinkText”));

driver.findElement(By. partialLinkText ("PartialLink”));

driver.findElement(By. tagName ("TagName”));

driver.findElement(By.xpath("XPathLocator));

element = (WebElement) ((JavascriptExecutor)driver).executeScript(strJavascript);

**Clear the Content:**

ele.clear();

**SendKeys:**

ele.sendKeys(Keys.RETURN); //For Enter

ele.sendKeys(Keys.ENTER); //For TAB

ele.sendKeys(“Line1\n Line2”); //Multiline

ele.sendKeys(Keys.chord(Keys.SHIFT,Keys.ENTER)); //Shift+Tab

**How to get typed text from a textbox:**

ele.getAttribute("value");

String Text = driver.findElement(ele).getText();

**How to click on element:**

el.click();

**How to enter text into page element:**

el. sendKeys (“User name”);

**How to count total number rows in table:**

List rows = driver.findElements(By.xpath("//table[@id='tableID']/tr"));

int totalRow = rows.size();

**How to get page title:**

String title =  driver.getTitle()

**How to get page source of webpage:**

String pagesource = driver.getPageSource()

**How to get current url :**

String currentURL  = driver.getCurrentUrl()

**How to Refresh Page:**

driver.navigate().refresh();

driver.get(driver.getCurrentUrl());

driver.findElement(By.id("Contact-us")).sendKeys(Keys.F5);

driver.findElement(By.id("Contact-us")).sendKeys("\uE035");

JS.executeScript("history.go(0)");

**How to assert text of webpage:**

String text = ele.getText();

Assert.assertEquals("Element Text", text); //assert text from expected

**How to get element attribute:**

String attributeValue = ele.getAttribute("AttributeName") ;

**Retrieve Element CSS Properties:**

String bgColor = td\_Home.getCssValue("background-color");

getCssValue("text-decoration"); //Text Underline

**How to check if an element is visible on the web page:**

ele.isDisplayed();

**How to check if a button is enabled on the page:**

ele.isEnabled();

**How do u get the width of the textbox:**

ele.getSize().getWidth();

ele.getSize().getHeight();

**Dropdown:**

Select sel= new Select(ele);

sel.selectByVisibleText("Pass your desire Text");

sel. selectByValue( ("Pass your desire Text");

sel. selectByIndex (0);  // 0 is first index

Boolean status = sel. isMultiple()

int  totalOption  = sel.getOptions().size()

getOptions

getFirstSelectedOption()

getAllSelectedOptions

deselectAll( )

deselectByVisibleText(String arg0)

**Radio Button:**

el.click() //to perform check operation

el.isSelected() //verfiy to radio button is check it return true if selected else false

**CheckBox:**

el.click() //to perform check operation

el.isSelected() //verfiy to check box it return true if selected else false

**XPATH:**

 //input[@name=’q’]

//input[@id=’lst-ib’]

//input[@class=’lst’]

//input[contains(@id,’ lst’)]

//a[contains(text(),’click me’)]

//input[@id=’lst-ib’]/parent::div[//a[text()='SELENIUM']]

//input[@id=’lst-ib’]/following-sibling::div[@class='rt-grid-2 rt-omega']

//input[@id=’lst-ib’]/preceding-sibling::button[@name='settings']

//input[@id=’lst-ib’]/ancestor::div[@class='rt-grid-2 rt-omega']

//input[@id=’lst-ib’]/descendant::div[@class='rt-grid-2 rt-omega']

**CSS:**

input[name=’q’]

input[id=’lst-ib’] or input#lst-ib

input[class=’ lst’] or input.lst

 input[id\*=' lst-ib ')] //contains

a:contains(‘log out’) //InnerText

div > p //Child Element

div p //Inside elements

**JavaScript Executor:**

JavascriptExecutor js = (JavascriptExecutor)driver;

Js.executeScript("alert('hello world');");

js.executeScript("arguments[0].click();", element);

string sText = js.executeScript("return document.documentElement.innerText;").toString(); //InnerText of whole page

string sText = js.executeScript("return document.title;").toString(); //Title of web page

js.executeScript("$('ul.menus.menu-secondary.sf-js-enabled.sub-menu li').hover()"); //Hover on

js.executeScript("window.location = 'https://www.facebook.com/uftHelp'"); //Navigate to new Page

js.executeScript("window.scrollTo(0, document.body.scrollHeight)"); // Page Scroll

js.executeScript( "arguments[0].scrollIntoView();", element);

js.executeScript("window.scrollBy(0,500)");

**Multi Window:**

String handle= driver.getWindowHandle();

Set<String> handles = driver.getWindowHandles();

for (String handle1 : driver.getWindowHandles()) {

driver.switchTo().window(handle1);

String windowTitle = driver.getTitle();

System.out.println("Current Window , Title: " + windowTitle + " Handle: " + handle1);

Thread.sleep(5000);

driver.close();

}

**Alert:**

String mainPage = driver.getWindowHandle();

Alert alt = driver.switchTo().alert(); // to move control to alert popup

alt.accept(); // to click on ok.

alt.dismiss(); // to click on cancel.

alt.getText(); // to get content

alt.sendKeys(“sample”); // to set text

driver.switchTo().window(mainPage); // to switch back to main page.

**Frames:**

List <WebElement> framesList = driver.findElements(By.xpath("//iframe"));

int numOfFrames = frameList.size();   
driver.switchTo().frame(“ID of the frame“); // Select iframe by id  
driver.switchTo().frame(driver.findElements(By.tagName(“iframe”).get(0));  
driver.switchTo().frame(0); // Locating iframe using index

driver.switchTo().frame(“name of the frame”); //By Name  
driver.switchTo().defaultContent(); // Parent Window

**AutoIt:**

Runtime.getRuntime().exec(".exe file path");

**Actions Class:**

Actions builder = new Actions(driver);

Action mouseOverHome = builder.moveToElement(ele).build();

mouseOverHome.perform();

builder.doubleClick(el).build().perform();

builder.dragAndDrop(source, destination ).perform();

act.moveToElement(webElement).perform();

act.contextClick().perform();

click(WebElement element) – Clicks an element

void mouseDown(WebElement element) - Holds down the left mouse button on an element.

mouseUp(WebElement element) - Releases the mouse button on an element.

mouseMove(WebElement element) - Moves element form current location to another element.

**Robot Class:**

Robot robot = new Robot();

robot.keyPress(KeyEvent.VK\_CONTROL);

robot.keyPress(KeyEvent.VK\_V);

robot.keyRelease(KeyEvent.VK\_V);

**Screenshot:**

File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

FileUtils.copyFile(scrFile, new File(workingDir + "google.jpg"));

**Http Authentication:**

driver.get(“<http://username:password@applicationURL>”)

HKEY\_CURRENT\_USER\Software\WOW6432Node\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE\_HTTP\_USERNAME\_PASSWORD\_DISABLE

**Environment variables:**

String workingDir = System.getProperty("user.dir"); //My Documents

System.getProperty("user.home") + "/Desktop";  //Desktop

String value = System.getenv("username");

**Grid:**

String nodeURL = "http://localhost:4444/wd/hub";

driver = new RemoteWebDriver (new URL(nodeURL),desiredcapability) ;

<http://192.168.0.2:4444/grid/register/>

java -jar selenium-server-standalone-2.44.0.jar -role hub

java -jar selenium-server-standalone-2.52.0.jar -role webdriver -browser "browserName=firefox,maxinstance=3,version=23.0.1,platform=WINDOWS" -hub http://192.168.0.2:4444/grid/register -port 5566

java -jar selenium-server-standalone-2.52.0.jar -role webdriver -browser "browserName=chrome,maxinstance=3,platform=MAC" -hub http://192.168.0.2:4444/grid/register -port 5577 -Dwebdriver.chrome.driver="path\chromedriver.exe"

java -jar selenium-server-standalone-2.52.0.jar -role webdriver -browser "browserName=internetexplorer,maxinstance=3,platform=WINDOWS" -hub http://192.168.0.2:4444/grid/register -port 5588 -Dwebdriver.ie.driver="path\IEDriverServer.exe"

default port for node:5555

**Desired Capability:**

DesiredCapabilities capability=null;

capability= DesiredCapabilities.firefox(); //.chrome(),internetExplorer()

capability.setBrowserName("firefox"); //"chrome",”internetExplorer”

capability.setPlatform(Platform.WINDOWS); // [LINUX](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/Platform.html#LINUX) , [MAC](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/Platform.html#MAC) , [XP](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/Platform.html#XP), [WIN8](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/Platform.html#WIN8), [WIN8\_1](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/Platform.html#WIN8_1) , [VISTA](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/Platform.html#VISTA), [UNIX](https://selenium.googlecode.com/git/docs/api/java/org/openqa/selenium/Platform.html#UNIX),ANY

capability.setVersion("version Number");

capability.setCapability("chrome.switches", Arrays.asList("--start-maximized"));

capability.setCapability(InternetExplorerDriver.INTRODUCE\_FLAKINESS\_BY\_IGNORING\_SECURITY\_DOMAINS,true); //for protected mode in all zones

**Configuring Browser Profile:**

Firefox-profiles

Chrome- options

IE-DesiredCapability

FirefoxProfile profile = new FirefoxProfile();

WebDriver driver = new FirefoxDriver(profile);

profile.setPreference("browser.download.dir","e:\\SampleExcel");

**Jenkins:**

-Browser will not be visible if run Jenkins as standalone ,run from command prompt by extracting the war file

**Maven:**

**TestNG:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| <!DOCTYPE suite SYSTEM "<http://testng.org/testng-1.0.dtd>"> | | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| <suite name="Suite1" verbose="1" > | | | | | | | | | | | |
| <test name="Nopackage" > | | | | | | | | | | |
| <classes> | | | | | | | | | |
| <class name="NoPackageTest" /> | | | | | | | | |
| </classes> | | | | | | | |
| </test> | | | | | | |
|  | | | | | |
| <test name="Regression1"> | | | | |
| <classes> | | | |
| <class name="test.sample.ParameterSample"/> | | |
| <class name="test.sample.ParameterTest"/> | |
| </classes> |
| </test> | | | | | | | | | | | | | | |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | <test name="Regression1"   > | | | | | | | <packages> | | | | | | <package name="test.sample" /> | | | | | | | | </packages> | | | | | </test> | | <test name="Regression1"> | | |  |  |  | | --- | --- | | <groups> | | | <run> | | <exclude name="brokenTests"  /> | | |  |  |  |  |  | | --- | --- | --- | --- | | <include name="checkinTests"  /> | | | | | </run> | | | | </groups> | | |  | | <classes> | | | | | | | | | | | | | | | <class name="test.IndividualMethodsTest"> | | | | | | | | | | | | | | | | <methods> | | | | | | | | | | | | | | <include name="testMethod" /> | | | | | | | | | | | | | </methods> | | | | | | | | | | | | </class> | | | | | | | | | | | </classes> | | | | | | | | | | </test> | | | | | | | | <test name="T1"> | | | | | | | | | <parameter name="number-of-times" value="10"/> | | | | | | | | | | | | | | | | | <class name= "TestWebServer" /> | | | | | | | </test> | | | | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | |

|  |
| --- |
| </suite>  <suite name="My suite" parallel="methods/tests/classes/instances" thread-count="5"> |
| Running TestNG: java org.testng.TestNG testng1.xml [testng2.xml testng3.xml ...]   |  | | --- | | @Test(groups = { "functest", "checkintest" }) |  |  | | --- | | @Parameters({ "db",}) | | @Test (@Optional("mysql") String db) | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | @DataProvider(name = "test1") | | | | | | | | public Object[][] createData1() { | | | | | | | | | return new Object[][] { | | | | | | | { "Cedric", new Integer(36) }, | | | | | | | | { "Anne", new Integer(37)}, | | | | | | }; | | | | | } | | @Test(dataProvider= "test1") | | |  |  |  |  |  | | --- | --- | --- | --- | | public void verifyData1(String n1, Integer n2) { | | | | | System.out.println(n1 + " " + n2); | | | | } | | | |

@BeforeSuite  
@AfterSuite  
@BeforeTest  
@AfterTest  
@BeforeGroups  
@AfterGroups  
@BeforeClass  
@AfterClass  
@BeforeMethod  
@AfterMethod

**Properties File:**

Properties prop = new Properties();

FileOutputStream output = new FileOutputStream("config.properties");

prop.setProperty("user", "krishna");

prop.store(output, null);

FileInputStream input = new FileInputStream("config.properties");

prop.load(input);

prop.getProperty("user"));

**Reflection API:**

public String callFunction(HashMap<String,String> hm,String strKeyword) throws Exception {

String blnFunStatus= "";

//no paramater

Object noparams[] = {};

Object[] obj = { driver,hm};

Class<?>[] argtype = { WebDriver.class,HashMap.class };

try{

Class<?> c=Class.forName("com.proj.selenium.Keywords");

Constructor<?> cons = c.getConstructor(argtype);

Object t = cons.newInstance(obj);

Method[] allMethods = c.getDeclaredMethods();

for (Method m : allMethods) {

String mname = m.getName();

if (mname.equalsIgnoreCase(strKeyword)) {

Object o = m.invoke(t, noparams);

blnFunStatus = o.toString();

System.out.println(blnFunStatus);

}

}

}catch(Exception e){System.out.println(e);}

return blnFunStatus;

}

**Apache POI:**

File file = new File(inputFile);

boolean fileExists = file.exists();

if (fileExists) {

FileInputStream inputWorkbook = new FileInputStream(file);

//Get the workbook instance for XL file

XSSFWorkbook workbook = new XSSFWorkbook (inputWorkbook);

for (int intSheets = 0; intSheets < workbook.getNumberOfSheets(); intSheets++) {

//Get first sheet from the workbook

XSSFSheet sheet = workbook.getSheetAt(intSheets);

SheetName = sheet.getSheetName();

if (SheetName.indexOf("#",0) != 0) { intMaxRowCnt = sheet.getPhysicalNumberOfRows(); //Determine number of Used Rows in spreadsheet

XSSFRow row = sheet.getRow(0);

intColCnt = row.getLastCellNum(); //Determine number of Used columns

for (int col=0;col<intColCnt;col++){

strData = getCellData(sheet,0,col).trim();

}

}

}

public String getCellData(XSSFSheet objSheet,int rowNo,int columnNo) { //To retreive the value of given cell

int intType;

Object result="";

XSSFRow row = objSheet.getRow(rowNo);

Cell cell = row.getCell(columnNo);

if (cell!=null) {

intType = cell.getCellType();

switch (intType)

{

case Cell.CELL\_TYPE\_NUMERIC:

result = cell.getNumericCellValue();

break;

case Cell.CELL\_TYPE\_STRING:

result = cell.getStringCellValue();

break;

case Cell.CELL\_TYPE\_BOOLEAN:

result = cell.getBooleanCellValue();

break;

case Cell.CELL\_TYPE\_BLANK:

result = "";

break;

default:

//System.out.println(intType);

}

} else {

result = "";

}

return result.toString();

}