To open A browser

1. **System.setProperty(parameter1, marameter2);**

- parameter1- name of browser (“name of browser”)

  - webdriver.chrome.driver
  - webdriver. edge. driver
  - webdriver. gecko. driver

- parameter2- path of browser (“path of webdriver”)

2. **ChromeDriver driver=new ChromeDriver ();**

  - WebDriver driver=new ChromeDriver ();
  - WebDriver driver=new EdgeDriver ();
  - WebDriver driver=new FirefoxDriver ();

- **Upcasting** of ChromeDriver is done
- Create object of ChromeDriver and give reference of WebDriver.

3. **Method with Return type**

Methods	**Return Type**	**Method Of**
driver.get ("https://www.google.com/");	Void	Webdriver
driver. close ();	Void	Webdriver
driver. quite ();	Void	webdriver
driver. manage (). window (). maximize ();	Void	Window Interface
driver. navigate().to("https://www.google.com/");	Void	Window Interface
String Title = driver.getTitle();	String	Webdriver
String URL = driver.getCurrentUrl();	String	Webdriver
Dimension d = new Dimension (600, 500); driver. manage (). window (). setSize(d);	Void	Window Interface
Dimension d2 = driver. manage (). window (). getSize (); System.out.println(d2);	Dimension	Window Interface
Point p = new Point (100, 50); driver. manage (). window (). setPosition(p);	Void	Window Interface

System.out.println(driver. manage (). window (). getPosition ());	Point	Window Interface

4. List of Method with Return Type

5. **Locator**: Locators are used to find the element present on Webpage.

Types of Locators

A. X Path (Slowest locator to find Element)
B. Id (Fastest locator to find Element)
C. Name
D. Class name
E. Tag name
F. CSS selector
G. Link text
H. Partial link text

6. **Locator using X Path**: X path is again divided into following sub types

a) X path by Attributes @
b) X path by Text
c) X path by Contains
d) X path by Index
e) Absolute X path
f) Relative X path

xpath Expressions: -

```
driver. findElement (By. xpath ("xpath")). sendKeys ("value");
```

1. **xpath by attribute**

```
//tagname [@attributeName = 'attribute value']
```

2. **xpath by text**

```
//tagname [text () = 'text value']
```

3. **xpath by contains with attribute**

```
//tagname [contains (@ attributeName, 'attribute Value')]
```

4. **xpath by contains with text**

```
//tagname [contains (text (), 'text value')]
```

5. **xpath by index**

```
//tagname [@attributeName = ‘attribute value’][1]
```

7. **Id Locator: -**

```
driver. findElement (By. id ("id value")). sendKeys ("Value ");
```

8. **Name Locator: -**

```
Driver. findElement (By.name("abc123")). sendKeys ("Sagar ");
```

9. Class Name Locator: -

```
driver. findElement (By. className ("abc123")). sendKeys ("Sagar");
```

10.CSS Selector: -

11.**Tagname Locator: -**

```
driver. findElement (By. tagname ("input")). sendKeys ("Vaibhav");
```

12.Link Text: -

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

13.Partial Link Text

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

14. **WebElement: -** it is an interface use to perform action on element present on webpage.

Method Name	**Syntax**	**Return Type**
sendKeys	driver. findElement (By. xpath (“xpathExp")). Sendkeys (“”);	WebElement
Click	driver. findElement (By. xpath (“xpathExp")). click ();	Void
getText	String text = driver. findElement (By. xpath("xpathExp")). getText ();	String
isEnabled	boolean result = driver. findElement (By. xpath ("xpathExp ")). isEnabled ();	Boolean
isDisplayed	boolean result = driver. findElement (By. xpath (“xpathExp ")). isDisplayed ();	Boolean
isSelected	boolean value = driver. findElement (By. xpath ("xpathExp ")). isSelected ();	Boolean

15. Collection methods

Collection method	**Object**
ArrayList	ArrayList al=new ArrayList ();
LinkedList	LinkedList ll=new LinkedList ();
Vector	Vector v=new Vector ();
HashSet	HashSet hs=new HashSet ();
LinkedHasSet	LinkedHasSet lsh=new LinkedHasSet ();
TreeSet	TreeSet tr=new TreeSet ();
Iterator	Iterator itr=new object.iterator();
ListIterator	ListIterator litr=new objname.listIterator();
Enumeration	Enumeration enu=v.elements();

Method	**Syntax**	**Return Type**
add	al. add (“value”);	boolean
isEmpty	al. isEmpty ();	boolean
size	al. size ();	int
contains	al. contains (“value”);	boolean
add (index, value)	al. add (index, value);	void
remove	al. remove (index);	Object

set	al. set (index, value);	Object
indexOf	al. indexOf("value");	Int
get	al. get (index);	Object
	TreeSet Methods	
pollFirst	tr. pollFirst ();	Object
pollLast	tr. pollLast ();	Object
First	Tr. first ();	Object
Last	Tr. last ();	Object

- **PollFirst** & **PollLast** used to **remove**

16.Print ArrayList info Using foreach loop

```
For (Object s1:ar)
 {
  System.out.println(s1);
 }
```

17.Print ArrayList info using for loop cursor

```
For (int i=0; i<=ar.size()-1; i++)
 {
    System.out.println(ar.get(i));
 }
```

18. Print ArrayList info using ListIterator cursor

```
ListIterator Litr = ar.listIterator();
   While (Litr.hasNext())
       {
         System.out.println(Litr.next());
       }
```

19.Print ArrayList info using Iterator cursor

```
Iterator itr = lhs. iterator ();
 while(itr.hasNext())
 {
   System.out.println(itr. next ());
 }
```

20.Print vector info using Enumeration cursor

```
Enumeration enu=v. elements ();
While (enu. hasMoreElement ())
{
  System.out.println(enu. nextElement ());
}
```

21.List Box handling

- **WebElement** month = driver. findElement (By. xpath ("xpathExp"))
- **Select** s = new **Select**(month);

Select s = new **Select** (driver. findElement (By. xpath ("xpath")));		Return type
By value	s. selectByValue (String value);	Void
By visible text	s. selectByVisibleText (String Text);	Void
By index	s. selectByIndex (int index);	Void
Deselect by index	s. deselectByIndex ()	Void
Deselect by value	s. deselectByValue ()	Void
Deselect by visible text	s. deselectByVisibleText ()	Void
Deselect All	s. deselectAll ()	Void
Is Multiple	s. isMultiple ()	boolean
Get all selected options	s. getAllSelectedOptions ()	List<WebElement>
Get first selected option	s. getFirstSelectedOption ()	WebElement
Get options	s. getOptions ()	List<WebElement>

22.Screenshot

- File src = ((**TakesScreenshot**)driver). **getScreenshotAs** (**OutputType**.**FILE**);
- **File** dest = new **File**("path"+**RS**+**TestID**+**DatewithTime**".jpg");
- **FileHandler.copy(**src**,** dest**);**

- **Test Case Id: - @Test** mention there **TestID**
- **Random String (RS): -** String RS = RandomString.make(3);
- **Img format: - .png, .jpeg,**
- **Date with time: -**

a. Create object of **SimpleDateFormat class** and decide the format
b. To get current date & time create **object of Date class**
c. Now format the date

- **DateFormat** dateFormat **=** new **SimpleDateFormat** ("MM/dd/yyyy HH: mm: ss ");
- **Date** date **=** new **Date ();**
- **String** DatewithTime **= dateFormat.format(**date**);**

23. Parameterization

- To open Excel Sheet

- **FileInputStream** file=new **FileInputStream** ("Path of Excel");
- **WorkbookFactory** book=WorkbookFactory. create (**file**);

- **To open particular sheet**
  - **Sheet** sh=WorkbookFactory. create (**file**).getSheet ("Sheetname");
- **To identify the desired row use method getRow ();**
  - Row rw =sh. getRow (index);
- **To identify the desired cell, use method getCell ();**
  - Cell c =rw. getCell (index);

#Cell Shortcut: -

Cell c1=WorkbookFactory. create (file). getSheet ("Sheet name"). getRow (index). getCell (index);

#String Shortcut: -

String value = WorkbookFactory. create (file). getSheet ("Sheet1"). getRow (index). getCell (index). **getStringCellValue ();**

- **To get all in one line**

```
String  Value1=Sh. getRow (index). getCell (index). getStringCellValue ();
int     Value1=Sh. getRow (index). getCell (index). getNumericCellValue ();
boolean Value1=Sh. getRow (index). getCell (index). getBooleanCellValue ();
```

Row*Cell

- **To print Full Row**

```
int lastRowIndex = sh. getLastRowNum ();
 for (int i=0; i<=lastRowIndex; i++)
   {
      String value1 = sh. getRow (i). getCell (1). getStringCellValue ();
      System.out.println(value1);
   }
```

- **To print Full Cell only**

```
int lastCellIndex = sh. getRow (1). getLastCellNum ();
    for (int i=0; i<=lastCellIndex; i++)
       {
          String value1 = sh. getRow (1). getCell (i). getStringCellValue ();
           System.out.println(value1)
        }
```

- **To print Both Row & Cell**

```
int lastRowIndex = sh. getLastRowNum ();
for (int i=0; i<=lastRowIndex; i++)
  {
    int lastCellIndex = sh. getRow (i). getLastCellNum ()-1;
    for (int j=0; j<=lastCellIndex; j++)
      {
    String value1 = sh. getRow (i). getCell (j). getStringCellValue ();
        System.out.print(value1+"_")
      }
   System.out.println();
    }
```

- **To print data type present in Excel sheet**

```
Index a=Row;
Index b =Cell;
```

```
CellType dataType = sh. getRow (a). getCell (b). getCellType ();
    System.out.println(dataType);
```

```
    If (dataType == CellType.STRING)
    {
     String value = sh. getRow (a). getCell (b). getStringCellValue ();
     System.out.println(value);
    }
    else if (dataType == CellType.NUMERIC)
    {
    double value = sh. getRow (a). getCell (b). getNumericCellValue ();
    System.out.println(value);
```

```
}
else if (dataType == CellType.BOOLEAN )
{
boolean value = sh. getRow (a). getCell (b). getBooleanCellValue ();
System.out.println(value);
}
```

24.Synchronisation

- To match browser speed with driver speed

Type	Syntax
Wait	Thread. sleep (1000);
Page Load Time Out	driver. manage (). timeouts (). pageLoadTimeout (500, SECONDS);
Implicitly Wait	driver. manage (). timeouts (). implicitlyWait (1000, TimeUnit.SECONDS);
Explicitly Wait For WebElement	**WebDriverWait** wait = new **WebDriverWait** (driver, 10); **WebElement** element = wait. until (ExpectedConditions.visibilityOfElementLocated (By. xpath ("xpathExp")));
Explicitly Wait For Alert to appear	**WebDriverWait** wait = new **WebDriverWait** (driver, 10); **WebElement** element = wait. until (ExpectedConditions. alertPresent);
Fluent Wait	**Wait** wait = new **FluentWait** (WebDriver reference) . withTimeout (**Duration**. ofSeconds (SECONDS)) . pollingEvery (**Duration**. ofSeconds (SECONDS)) . ignoring (**Exception**. class);
Fluent Wait **Exp.**	**Wait**<**WebDriver**> wait = new **FluentWait**<**WebDriver**>(**driver**) . withTimeout (**30**, TimeUnit.SECONDS) . pollingEvery (**5**, TimeUnit.SECONDS) . ignoring (**NoSuchElementException**.class);

Type	Use
Wait	Normal wait **waits** till given time
Page Load Time Out	To set the time for a page to load
Implicitly Wait	to set a wait time before searching and locating a web element. Only condition req.
Explicitly Wait	Condition and xpath req.
Fluent Wait	Condition, xpath, frequency of checking req.

25.Action class and its method

1. Identify dropdown element
2. create an object of **Actions** class with **Webdriver object** as an input
3. call **Actions class** methods

- **WebElement login**= driver. findElement (By. xpath ("xpathExp"));
- **Actions** act = new **Actions**(**driver**);
- act. moveToElement (**login**). perform ();
- act. **methodGoesHere.** perform ();

Method	Use	Return type
moveToElement	Move one element	
contextClick	Click on Right Button mouse	Void
perform	To perform the Action	Void
build	When multiple action perform/call in single statement then use build.	
click	click on the element	
doubleclick	double click on element	
dragAndDrop	drag one element and drop the same element at another place	
clickAndHold	click and hold	
release	release the click and hold element.	

1. act. moveToElement (cart). contextClick (). build (). perform ();
2. act. moveToElement (cart). click (). build (). perform ();
3. act. moveToElement (ClickDouble). doubleClick(). build (). perform ();
4. act. moveToElement (src). clickAndHold (). moveToElement (dest). release (). build (). perform();
5. act. dragAndDrop (src, dest). perform ();

26.Drag and Drop

- TO drag an element and drop it to another element location.

WebElement src = driver. findElement (By. xpath ("xpathExp"));

WebElement dest = driver. findElement (By. xpath ("xpathExp"));

Actions act = new **Actions**(driver);

1. act. **dragAndDrop** (src, dest). **perform** ();
2. act. moveToElement (src). clickAndHold (). moveToElement (dest). release (). build (). perform ();

27.Customised List box

- **WebElement** month= driver. findElement (By. xpath ("xpathExp"));
- **Actions** act = new **Actions**(**driver**);

Method	**Use**	**Return Type**
Click on Element	**act. click (month). perform ();**	
To move 1 step UP	act. sendKeys (Keys.ARROW_UP). perform ();	
To move 1 step Down	act.sendKeys(Keys.ARROW_DOWN).perform();	
To return to home	act. sendKeys (Keys.HOME). perform ();	
To Move Arrow at the Top	act. sendKeys (Keys.ENTER). perform ();	
To move arrow at the end	act. sendKeys (Keys.END). perform ();	

✓ **To use multiple ARROW_UP & ARROW_DOWN key**

```
For (int i=1; i<=4; i++)
{
   act. sendKeys (Keys.ARROW_UP). perform ();
}
```

Note: To run program any time need to go first down by using End keys then from that point go to req. dest by using ARROW_UP.

- act. sendKeys (**Keys.END**). perform ();

28.Multiple Element Handling

- Used to get same type of element in bulk and perform other action it.
- Use "**findElements**"
- Return type **List<WebElement>**
- for finding all links **always use //a**

```
List<WebElement> AllLinks = driver. findElements (By. xpath ("//a"));
System.out.println(AllLinks.size());

for (WebElement Links: AllLinks)
{
   System.out.println(Links.getText());
}
```

Method	Return Type
findElement	WebElement
findElements	List<WebElement>

29.Exception Handling

```
WebElement saveButton = driver. findElement (By.id("Save"));
```

```
try
  {
   if (saveButton.isDisplayed())
      {
         saveButton.click();
       }
  }

catch (NoSuchElementException e)
  {
     e. printStackTrace ();
  }
```

30.Pop-Ups

Popups are small or separate window which will be displayed when we perform action on any components present in a webpage.

- **Types of popups:**
  1. Hidden-Division Popup
  2. **Alert popup**
  3. **Child browser Popup**
  4. Authentication Popup
  5. File Upload Popup
  6. File Download Popup

Parameters	**Hidden-Division Popup**	**Alert popup**	**Child browser Popup**
Inspect	Yes	No	Yes
Colourful	Yes	No	New window open
Switch	No Need	Yes	Yes
Action Element Present	Close, Login page,	ok button or cancel button & Text. ? or! Symbol	address field (URL), maximize, minimize and close options.

Alert popup

- Need to change focus to alert popup
-

Alert alt = driver**. switchTo (). alert ();**
driver. switchTo (). alert **(). accept ();**

Method	**Use**	**Syntax**
accept	use to click on ok button	alt**. accept ();**
cancel	use to click on cancel button	alt**. dismiss ();**
getText	use to get text present in an alert popup	alt**. getText ();**

Child wind Pop up

- Need to change Selenium focus from main page to child window
- Sequence of ids are in **Reverse order** of **opening window**

Type	**Syntax**
Main page Id	**String** mainPageId = **driver. getWindowHandle ();** System.out.println**(mainPageId);**
Child Page ID	**Set<String>** ids = driver. **getWindowHandles** (); **ArrayList** al=new **ArrayList(ids);** System.out.println**(al. get (1));**

Set<String> AllIDs = driver**. getWindowHandles ();**

ArrayList<String> al = new **ArrayList<String>(**AllIDs**);**

- Switch to child Window

  driver. switchTo (). window (al. get (1));

- Switch to Main Window back

  driver. switchTo (). window (al. get (0));

31.Autosuggestion

- Collecting all Autosuggested text by google and find Req. one out of all

```
String expText="Redmi note 10";
driver. findElement (By. xpath ("xpathExp")). sendKeys ("Redmi");
```

```
List<WebElement> MultiElement
                              =driver. findElements (By. xpath ("xpathExp"));

for (WebElement singleElement: MultiElement)
 {
    String actText = singleElement.getText();

     if (actText. equalsIgnoreCase (expText))
        {
           singleElement. click ();
           break;
         }
 }
```

32.Test case Pass/Fail

String **actUserID** = driver.findElement (By.xpath("xpathExp")).getText();
String **ExpUserId** = "DV1510";

```
boolean result = actUserID. equals (ExpUserId);
      if (result=true) //Test Case Pass
        {
          System.out.println("Test Case Pass");
        }
    else
       {
          System.out.println("Test Case Fail");
       }
```

33.iframe handling

- Displaying webpage which is a part of another webpage is known as iframe.
- iframe is created by using Tagname iframe.

Method	Syntax
Select iframe by id	driver.switchTo().frame(“ID of the frame“);
Select iframe by Name	
Select iframe by WebElement	driver.switchTo (). Frame (driver. findElements (By. tagname (“iframe”). get (0));
Select iframe by id	driver.switchTo().frame(0);

Method	Syntax
Select Parent Window	driver. switchTo (). defaultContent ();
To go back by 1 frame	driver. switchTo (). parentFrame ();

34.Dynamic xpath handling

- Select full frame and come to respective req. element
- Use both Relative & Absolute xpath.