## 1)What is Selenium ?

- Selenium is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms.
- We can use multiple programming languages like Java, C#, Python, etc. to create Selenium Test Scripts.
- Testing done by using the Selenium testing tool is usually referred to as Selenium Testing.

## 2)What is the Advantages of Selenium?

- 1) Selenium is an Open Source Software.
- 2) Selenium supports various programming languages to write programs (Test scripts)
- 3) Selenium supports various operating systems (MS Windows, Linux, Macintosh etc...)
- 4) Selenium supports various Browsers (Mozilla Firefox, Google Chrome, IE, Opera, Safari etc.
- 5) Selenium supports Parallel Test Execution.

## 3)What is the Limitation of Selenium?

- We can't test Video, Audio, OTP, Captcha, n/w failure errors
- 100% automation is not possible
- To use Selenium Automation Framework , we need Any Programming Languages like Java, Python ,C# etc.

## 4)Explain Flavours(Components) of Selenium?

Selenium is not just a single tool but a suite of software, each with a different approach to support automation testing. It comprises of four major components which include:

- Selenium Integrated Development Environment (IDE)
- Selenium Remote Control (Now Deprecated)
- WebDriver
- Selenium Grid

5)What is Selenium WebDriver?

- Selenium WebDriver is a collection of open source APIs which are used to automate the testing of a web application.
- Selenium WebDriver tool is used to automate web application testing to verify that it works as expected. It supports many browsers such as Firefox, Chrome, IE, and Safari.

6)what are all the WebDrivers available in Selenium?

- ChromeDriver
- FirefoxDriver
- EdgeDriver
- SafariDriver
- InternetExplorerDriver

# Selenium 3 Architecture


Note: In Selenium 4 Only Changes is Instead of JSON , W3c (world wide Web Consortium ) is there.

# Selenium WebDriver Architecture

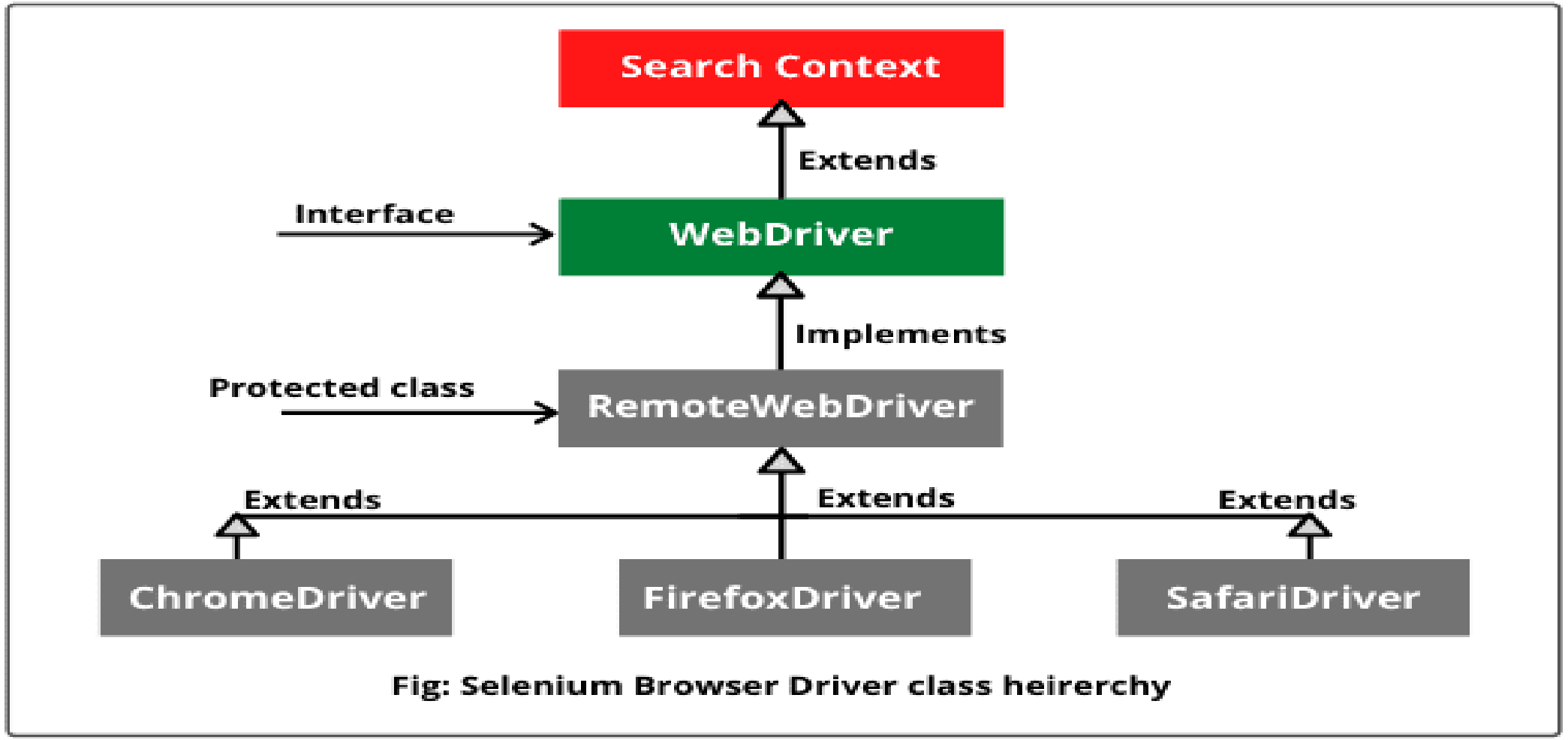


Fig: Selenium Browser Driver class heirerchy

# Selenium WebDriver

- Selenium WebDriver is a collection of open source APIs which are used to automate the testing of a web application.
- Selenium WebDriver tool is used to automate web application testing to verify that it works as expected. It supports many browsers such as Firefox, Chrome, IE, and Safari.

# WebDriver Methods in Selenium

| WebDriver Method | Description |
|---|---|
| close() | Closes the current active window if there are multiple windows. The browser quits if only one window is active |
| findElement() | Find the first WebElement based on the locator type |
| findElements() | Find all elements within the current page based on the locator type |
| get() | Loads a new web page in the current browser window |
| getCurrentUrl() | Gets a string defining the current web page URL |
| getPageSource() | Gets the complete page source of the loaded web page. |
| getTitle() | Gets the current page title |
| getWindowHandle() | Handles a browser after switching a specific window |
| getWindowHandles() | Handles all browser windows and permits the user to switch control between the parent window and child window |
| manage() | Receives the option interface |
| navigate() | Navigates to a specific URL |
| quit() | Stops/Quits the driver instance and close all open browser windows |
| switchTo() | Switch from one browser window to another browser window |

# What is WebElement in Selenium

- Anything which is Present in  our Website is Called as WebElement

- WebElement in Selenium is essentially an HTML element on a website. HTML documents consist of HTML elements. Each HTML element consists of a start tag and an end tag.

# 3 Important Things to Understand in HTML

In HTML

**1)TagName-**----------> is nothing but, anything which is comes after the <  Symbol .

Ex: <input>,<body>,<Span>,<button>,<head>,<p>,<li> etc.

**2)Attribute(AN=“AV”).** Attribute Name="Value“

for ex: class=“Value”, type="Text", id="Email", name="Password".

**3)Text-**--------> is Nothing but a Sequence of Character or Group of word which is Present Between the >Text< Symbol

1)what are all the web Element Methods?

- Clear();
- Click();
- getAttibute();
- getCssValue();
- getRect();
- get Locators();
- get Size();
- get tag();
- get text();
- is displayed();
- is enabled();
- is selected();
- submit();
- send Keys();

2)what is Locators ?

- Locators is a way which is used to find the web element in our webpage .
- 8 types available
- And these 8 types all of them are static methods which is Present in BY class

  Syntax :
  BY.Locators();

3)what are all types of locator?

- Tagname()
- Id()
- Name()
- ClassName()
- LinkText()
- partialLinkText()
- css Selector();  tagname[AN=AV];
- xPath.

4)different between get() and Navigate().

| **Get()** | **Navigate()** |
|---|---|
| It is used to enter into the webpage | It is also used to enter into web page and it also having some special feature like forward, backward, and refresh |

5)what is getPageSource() Method?

- It is a Method which is present in our Webdriver.
- it is used fetch a source code of an Webpage.
- Syntax : driver.getPageSource();

6)what is getCurrentURL()?

- It is a method which is present in our Webdriver
- It is used to fetch our current url of our Webpage.
- Syntax : driver.getCurrentUrl().

7)what is getTitle()?

- It is a method which is present in webdriver
- It is used to fetch our Title of our current webpage.
- Syntax: driver.getTitle().

1)What is Xpath?

- **XPath in Selenium** is an XML path used for navigation through the HTML structure of the page.
- It is a syntax or language for finding any element on a web page using XML path expression.
- In Selenium automation, if the elements are not found by the general locators like id, class, name, etc. then XPath is used to find an element on the web page.

3)Types of Xpath?

There are two Types

- Absolute Xpath
- Relative Xpath

4)what is Absolute Xpath?

- Specifying the Complete path of the Element is Called as Absolute Xpath.
- To find the WebElement ,we have to come from Root of the Path called as HTML
- It is represented as /

5)Disadvantages of Absolute Xpath?

- It is very lengthy to Write
- Prone to Human error while Writing the path
- It is not possible to write path for Bigger Web Application with complex Hierarchy
- And, if there are any changes made in the path of the element then that XPath gets failed. And we will get **NoSuchElementException**.

6)what is Relative Xpath?

- **Relative Xpath** starts from the middle of HTML DOM structure.
- It is represented as //
- It can search elements anywhere on the webpage, means no need to write a long xpath and you can start from the middle of HTML DOM structure.

7)what are all the important Types of relative Xpath?

There Five Important Types

- Xpath by Attribute
- Xpath by Text Function
- Xpath by Contains Function
- Independent and Dependent Xpath
- Xpath by Index

8)Syntax for Xpath By Attribute

- //TagName[@AN='AV'];

9)Syntax for Xpath by Text Function?

- //TagName[text()='Text Value'];

10)Syntax for Xpath by Contains Function?

- //TagName[contains(text(),'Text Value')];

11)Syntax for Group of Index?

- (Any Xpath)[index];

12)why we Need to Go for Xpath contains function?

- When having an extra Spaces and some Special Characters in a Text at that Time we go for Contains Function.
- The main feature of contains function is that we can find elements with partial text.

13) why we Need to Go for independent and dependent Xpath?

Independent and Dependent xpath
--->when the text value of the web element is completely changing , at that time we can't use xpath like text function or contains function
---> to handle these kind of situation , we go for Independent and dependent xpath

Steps to fallow
---> we need to find the independent element first
---> after ,we have to traverse to the common parent for both independent and dependent webElement
--->last ,we have to traverse to the targeted dependent webElement

14)How to traverse between Siblings?
--->for moving forward
following-sibling::tagName
--->for backward
preceding-sibling::TagName .

Shortcuts.
--->to parent /..
---> to child /TagName

1)Difference Between FindElement and FindElements .

| FindElement | FindElements |
|---|---|
| Its is used to handle Single Element | Its is used to handle Multiple Elements |
| Return type is Web Element | Return type is List of Web Element |
| If the locator is not matching ,at that time we will get NoSuchElementException | If the locator is not matching ,at that time we will get Empty List |
| If the Locator is Matching multiple elements ,then it will return the first matching element | It will return all matching Elements . |

2)Difference Between Close() and Quit() Method

| Close() | Quit() |
|---|---|
| It is used to close the current Window in the browser | It is used to close all the windows in the browser.(child and Parent window) |

3)What is ClassName Locator in Selenium?

- ClassName is a One of the Locator in Selenium which is used to find Web Element.
- To use ClassName Locator ,we need to attach class attribute value inside the ClassName locator .
- Syntax :driver.findElement(By.className("value"));

Note:

When to go--→if it is not separated with space for example: class="ico-login" .

When not to go -→if it is separated with for example: class="text-button button-text";

4)What is Name Locator in Selenium?

- Name is a One of the Locator in Selenium which is used to find Web Element.
- To use Name Locator ,we need to attach the Name attribute value inside Name locator .
- Syntax :driver.findElement(By.Name("value"));

5)What is Id Locator in Selenium?

- Id is a One of the Locator in Selenium which is used to find Web Element.
- Id is Unique value, each and every web element having a unique Id value .
- To use Id Locator ,we need to attach the Id attribute value inside Id locator .
- Syntax :driver.findElement(By.Id("value"));

6)What is TagName Locators in Selenium?

- TagName is a One of the Locator in Selenium which is used to find Web Element.
- To use TagName Locator ,we need to attach the particular Tag Name inside TagName locator .
- Syntax :driver.findElement(By.tagName("Tag name"));
- For Example : <input>

  driver.findElement(By.tagName(“input”))

7)What is linkedText Locator in Selenium?

- linkedText is a One of the Locator in Selenium which is used to find Web Element.
- To use linkedText Locator ,we need to attach the Text of that Web Element inside linkedText locator .
- Syntax :driver.findElement(By.linkedText("text"));

8)What is partialLinkedText Locator in Selenium?

- partialLinkedText is a One of the Locator in Selenium which is used to find Web Element.
- To use partialLinkedText Locator ,we need to attach the partial Text of that Web Element inside partialLinkedText locator .
- Syntax :driver.findElement(By.partialLinkedText("partial text"));

Note :Partial text is nothing but some portion of the text

9)What is cssSelector Locator in Selenium?

- cssSelector is a One of the Locator in Selenium which is used to find Web Element.
- Syntax :TagName[AN='AV'];
- when there is no direct locator to find the WebElement , at the time we go for cssSelector
- for example :value="value-Name",type="submit",ariel-style="style-Name";
  ----> a[type='submit']
  ----> div[value='button-type']
  ----->input[id='searchField']

Note:

- Shortcuts for using id and Class

1)for Id

- id is represented by #,for Example : id=”Email”;
- Syntax: driver.findElement(By.cssSelector(“#Email”)).

2)for Class

- Class is represented by dot ( . ),for Example class=”ico-login”.
- Syntax: driver.findElement(By.cssSelector(“.ico-login”)).

1) What is Select Class?

- Select class is a Selenium class .
- Which having some predefined methods and it is used to perform select and deselect actions on drop down menu.
- Syntax : Select ref_var=new Select(webElement);

Types of dropdown menu

→single select : which we can perform only select action

→multi select : which we can perform both select and deselect

2) Methods in Select class?

- selectByIndex();
- selectByValue();
- selectByVisibleText();
- deselectByIndex();
- deselectByValue();
- deselectByVisibleText();
- deSelectAll();
- isMultiple();
- getOptions();
- getFirstSelectOption();
- getAllSelectedOptions();
- getWrappedElement();
- hashCode();
- equals();

3) what is Actions Class?

- Actions class is a Selenium class
- Which is having some predefined methods and it is used to perform keyboard and mouse action on the browser.
- Syntax : Actions ref_var=new Action(driver);

4) Methods for Mouse Actions?

- moveToElement();
- doubleClick();
- dragAndDrop();
- contextClick();
- clickAndHold();
- click();

5) Methods in KeyBoard Actions ?

- sendKeys();
- KeyDown();
- KeyUp();

6) Types of PopUp?

- Child window PopUp
- JavaScript PopUp/Alert PopUp
- Notification PopUp
- File Upload PopUp
- File Download PopUp

- Calendar PopUp

7) How to handle Child Window PopUp?

- To handle , first we need to switch the controller to the particular child window
- To switch , we have to fallow a syntax
- Syntax : driver.switchTo.window(window handle);
- Window handle is a unique one, each and every window having their own handle, that handle can be find by using getWindowHandle() and getWindowHandles();

8) how to handle Alert Pop?

- To handle alert pop , first we need to switch the controller to the alert
- To switch , we have to fallow a syntax
- Alert ref_var=driver.switchTo.alert();

9) Types of alert PopUp?

- There are 3 types
- Simple alert
- Confirmation alert
- Prompt alert

10) how to handle or avoid notification PopUp?

To handle

- We need to use Robot class

To avoid

- We need to use ChromeOptions

11) how to handle or avoid file Upload PopUp?

To avoid

- By using sendKeys Methods
- Syntax : WebElement.sendKeys("path of the file").

To handle

- By using 3rd part tool called AutoId.

12) Methods present in Alert ?

- accept();
- dismiss();
- getText();
- senKeys();

13)What is iframe?

- An iFrame (Inline Frame) is an HTML document embedded inside the current HTML document on a website.
- Frame or iFrame HTML tag will be used to insert one html page in another html page.
- iFrame HTML element is used to insert content from another source, such as an advertisement, into a Web page.
- iFrame is defined by an <iFrame></iFrame> tag in HTML. With this tag you can identify an iFrame while inspecting the HTML tree.
- In order to embed (include) a web page inside another web page developer uses 'iFrame' or 'Frame' html tag.
- But, if an element is inside the frame then web driver cannot find the element.
- Webdriver can't perform an action on web element automatically when object or web element are inside the frame.

There are three ways we can pass driver control to frame.

Int (0, 1, 2)
String (ID/Name)
Web Element (Address)

**Syntax :**

- Iframes - Switch by index

```
driver.switchTo().frame(0); // # frameid starts from index zero
```

- Iframes - Switch by name

```
driver.switchTo().frame("frame-name");
```

- //Iframes - Switch by WebElement

```
driver.switchTo().frame(driver.findElement(By.tagName("")));
```

In order to give the control back to the main page, i.e. Change the control from frame or to exit from the frame we use the following methods.

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

1)What is Select Class?

- Select class is a Class , which is present in selenium
- Which having an predefined methods and it  is used to perform select and deselect actions on drop down menu.
- Syntax : Select ref_var=new Select(webElement);

2)Methods in Select class?

- selectByIndex();
- selectByValue();
- selectByVisibleText();
- deselectByIndex();
- deselectByValue();
- deselectByVisibleText();
- deSelectAll();
- isMultiple();
- getOptions();

3)what is Actions Class?

- Actions class is a class which is present in selenium
- And it is having some predefined methods which is used to perform keyboard and mouse action on the browser.
- Syntax : Actions ref_var=new Action(driver);

4)Methods for Mouse Actions?

- moveToElement();
- doubleClick();
- dragAndDrop();
- contextClick();
- clickAndHold();

5)Methods in KeyBoard Actions ?

- sendKeys();
- KeyDown();
- KeyUp();

6)Types of PopUp?

- Child window PopUp
- JavaScript PopUp/Alert PopUp
- Notification PopUp
- File Upload PopUp
- File Download PopUp
- Calendar PopUp

7)How to handle Child Window PopUp?

- To handle , first we need to switch the controller to the particular child window
- To switch , we have to fallow a syntax
- Syntax : driver.switchTo.window(window handle);
- Window handle is a unique one, each and every window having their own handle, that handle can be find by using getWindowHandle() and getWindowHandles();

8)how  to handle Alert Pop?

- To handle alert pop , first we need to switch the controller to the alert
- To switch , we have to fallow a syntax
- Alert ref_var=driver.switchTo.alert();

9)Types of alert PopUp?

- There are 3 types
- Simple alert
- Confirmation alert
- Prompt alert

10)how to handle or avoid notification PopUp?

To handle

- We need to use Robot class

To avoid

- We need to use ChromeOptions

11)how to handle or avoid file Upload PopUp?

To avoid

- By using sendKeys Methods
- Syntax : WebElement.sendKeys(“path of the file”).

To handle

- By using 3rd part tool called AutoId.

12)Methods present in Alert ?

- accept();
- dismiss();
- getText();
- senKeys();

## 13)What is iframe?

- An iFrame (Inline Frame) is an HTML document embedded inside the current HTML document on a website.
- Frame or iFrame HTML tag will be used to insert one html page in another html page.
- iFrame HTML element is used to insert content from another source, such as an advertisement, into a Web page.
- iFrame is defined by an <iFrame></iFrame> tag in HTML. With this tag you can identify an iFrame while inspecting the HTML tree.
- In order to embed (include) a web page inside another web page developer uses 'iFrame' or 'Frame' html tag.
- But, if an element is inside the frame then web driver cannot find the element.
- Webdriver can't perform an action on web element automatically when object or web element are inside the frame.

There are three ways we can pass driver control to frame.

Int (0, 1, 2)
String (ID/Name)
Web Element (Address)

**Syntax :**

- Iframes - Switch by index

  driver.switchTo().frame(0); // # frameid starts from index zero

- Iframes - Switch by name

  driver.switchTo().frame("frame-name");

- //Iframes - Switch by WebElement

driver.switchTo().frame(driver.findElement(By.*tagName*("")));

In order to give the control back to the main page, i.e. Change the control from frame or to exit from the frame we use the following methods.

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

## 14 )JavaScriptExecutor

- Javascript Executor is an interface which provides the mechanism to write the JavaScript code in selenium WebDriver. It is having 2 abstract methods:

  executeScript
  executeAsyncScript

- executeScript() : is used to write synchronous JavaScript code.
- executeAsyncscript(): is used to execute asynchronous JavaScript code.

- In the JavaScript code we can use arguments variable to access the second input.
- Arguments in an array variable in JavaScript and we use the inputs.
- We should use the normal array syntax i.e., arguments[0] or args[1] etc.

**Why do we use Javascript code in selenium webdriver?**

Sometimes selenium WebDriver commands will not work as expected. For ex: click() may not work on the identified element. So in that case we can use JavaScript code as a work around.

Also when there is no direct method to perform some operation on the webpage we can use JavaScript code in selenium webdriver.

For ex: There is no direct method to scroll the window. So here we can use Javascript code.

**Note**: Synchronous means one task at a time or single threaded. And Asynchronous means multiple task at a time or multi threaded task.

- These two methods are implemented in RemoteWebDriver class. To use these two method we should cast the WebDriver object to Javascript Executor type i.e.,

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

- The executeScript() receives two inputs

  Javascript code in String form

  Generic type which acts as input to JavaScript code.

15)Methods we are using in javaScriptExecutor

1)ScrollBy

Syntax : ref.executeScript(“window.scrollBy(x,y);”);

2)ScrollTo

Syntax : ref.executeScript(“window.scrollTo(x,y);”);

3)ScrollIntoView

Syntax : ref.executeScript(“arguments[0].scrollIntoView(Boolean value);”);

4)Click

Syntax : ref.executeScript(“arguments[0].click();”,WebElement);

5)Value

Syntax : ref.executeScript(“arguments[0].value=’values’;”,WebElement);

16)Selenium Exceptions

Below are the common exceptions that you may encounter while working with WebDriver.

- NoSuchElementException
- StaleElementReferenceException
- NoSuchAttributeException
- NoAlertPresentException
- ElementNotVisibleException
- ElementNotInteractableException
- TimeoutException
- NoSuchFrameException

**NoSuchElementException**

- This exception is raised when the element is not found in DOM.
- The exception is raised by find_element method.
- You may need to check the selector that you are using in find_element method to rectify the issue.

**StaleElementReferenceException**

- Thrown when a reference to an element is now "stale" or Lost.
- The possible cause for this exception is that you are no longer on the same page, or the page may have refreshed since the element was located.
- The element may have been removed and re-added to the web page, since it was located.
- Element may have been inside an iframe or another context which was refreshed.

**NoSuchAttributeException**

- Thrown when the attribute of element could not be found.
- An attribute could be anything that you are trying to access after dot operator. It can be a method, property, variable etc.

**NoAlertPresentException**

- Thrown when switching to no presented alert.
- This can be caused by calling an operation on the Alert() class when an alert is not yet on the screen.

**ElementNotVisibleException**

- Thrown when an element is present on the DOM, but it is not visible.
- Most commonly encountered when trying to click or edit or read text of an element that is hidden from view.

**ElementNotInteractableException**

- Thrown when an element is present on the DOM but can not interaction with the element.
- Possible cause may be the element is disabled.

**TimeoutException**

- Usually thrown by until method of WebDriverWait class.
- Possible cause would be when the command does not complete with in specified timeout period.

**NoSuchFrameException**

- Thrown when frame target to be switched doesn't exist.

## 17)What is TestNg?

- TestNG stands for test next generation
- TestNG is a testing framework developed in the lines of JUnit and NUnit, however it introduces some new functionalities that make it more powerful and easier to use. TestNG is designed to cover all categories of tests: unit, functional, end-to-end, integration, etc.,
- We use testNG in Automation to perform
  - Batch execution
  - Parallel execution
  - Group execution
  - Perform Parameterization
  - Priorities test case
  - Skip test case
  - To create dependency between test cases
  - To generate reports and many more.