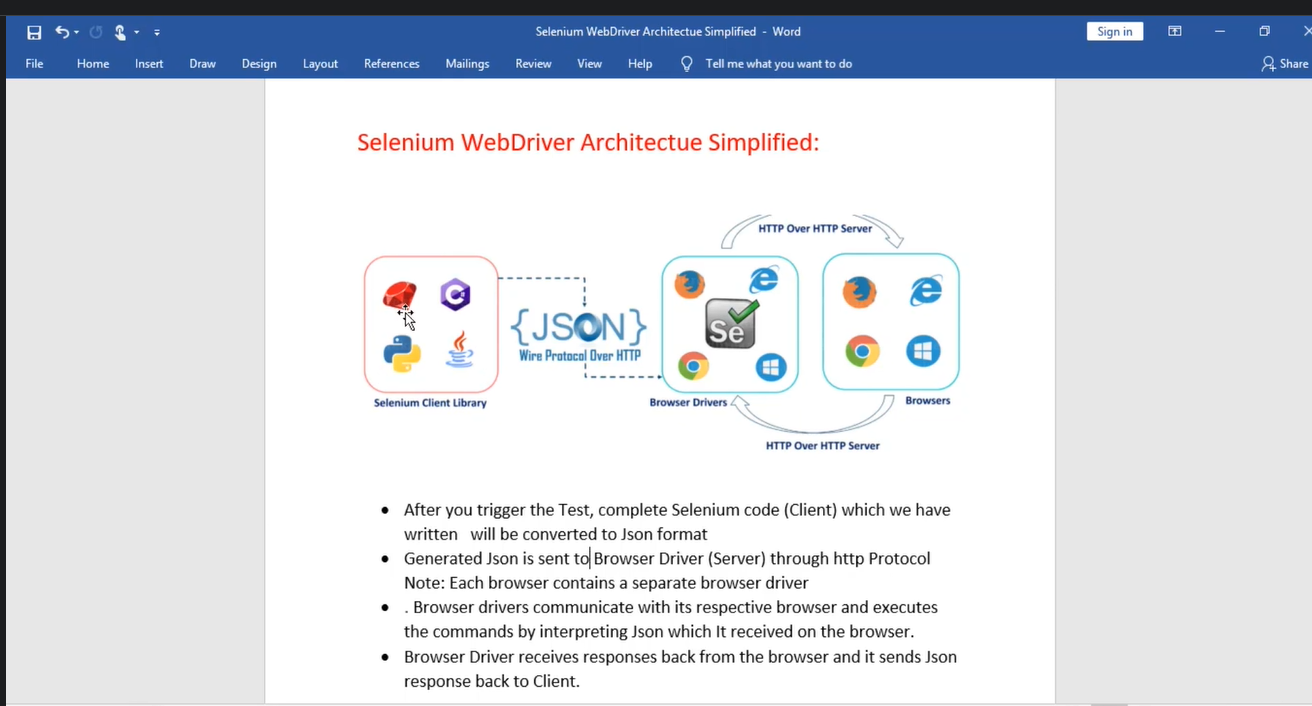
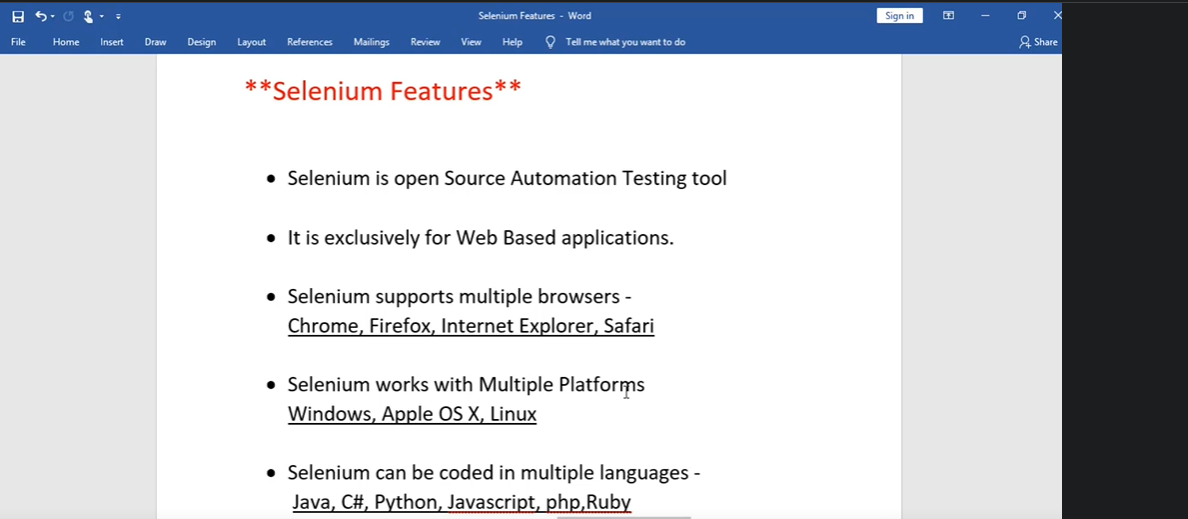
[Mentor@rahulshettyacademy.com](mailto:Mentor@rahulshettyacademy.com) for docs

**SELENIUM:**

**HOW AUTOMATION WORKS?**

1.object identifications (Web Element) (extra concept in Selenium)

2.Perform actions (input, click, select from dropdown)

3.verification Point (Output)

**Locators: (Commands) -to locate any element on the webpage**

**Id**

**Name**

**ClassName:** - if we have spaces in class means that field contains that many classes, we can use any one of the part to write script

**LinkText –** used to identify only links

**Partial LinkText --** used to identify only links

**CSS ->**

**Way 1:**  **Syntax: Tagname.className**

Ex: Facebook wrong password message

div.\_9ay7(tagname.class)

**Way 2:** **We can build Css with Tagname#id (Tagname+Id)**

**driver.findElement(By.cssSelector("#inputUsername")).sendKeys("rafi");**

**Customized Syntax for CSS:**   
TagName [Attribute=’value’] (Note: same like xpath but no need to give // or @ in css)

**CSS Indexing: input[type='text']: nth-child (3)**

Technically not recommended to use indexing

Note: If we want to validate from console

Xpath: $x (‘’)

Css: $(‘’)

String s2=driver.findElement(By. cssSelector ("div.\_9ay7")). getText ();

getText (); method used to fetch error message or any current response text

css tags:- Tagname space tagename input p

Xpath tags: //input/p (in css no need yo give slashes)

**Xpath ->** it is an expression, using the expression we can find any web element on the page

**1.Absolute Xpath**

**2.Relative Xpath** (In real-time we will use)

//tagName[@attribute=’Name’]

//input[@title=’Search’]  
Note: Selector Hub plugin used in Realtime to find relative xpath

**Using contains () in Xpath:**

**//span[contains(text(),'Rafi Syed')]**

**Tagname**

Example:

Step 1 : Open browser

Step 2 : enter username

Step 3 : Enter password

findElement- It is used to find any element on webpage

sendKeys-used to give input values

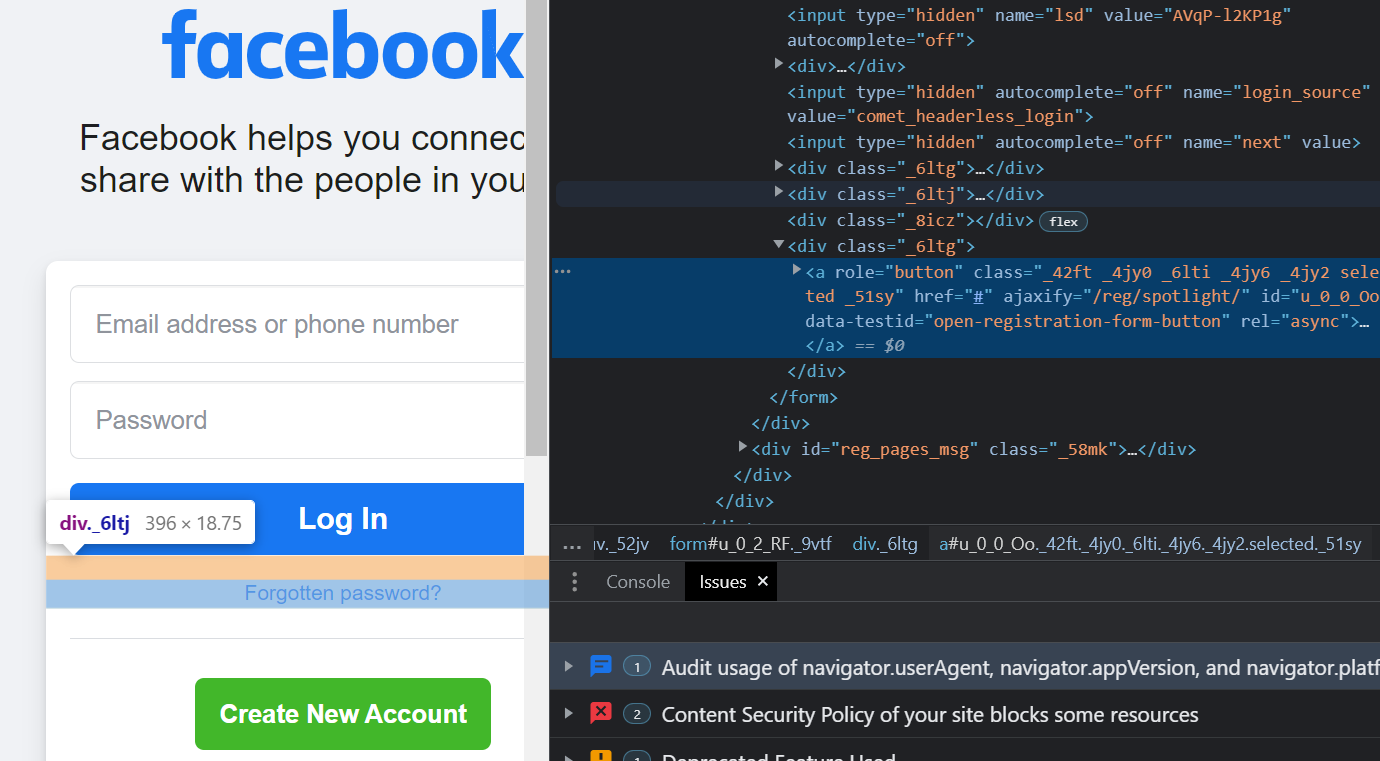
Scenario name will be class name in Automation.

\* Is a wild character

**Note :-** in real time we don’t use tag name simply we will write //\* for xpath

**Dynamic Xpath:**

**Ex: some times elements are dynamic ex : Facebook create Facebook account button**

**Here id ending values are changing means values or elements are dynamic** **id gets changing dynamically ending values**

**AUTOMATE DYNAMIC XPATH SCENARIO**

**NoSuchElement find means values are changed dynamically**

**Reason’s: 1. elements property dynamic**

**2. May be button might be missing (report this issue to dev team)**

**Dynamic Xpath methods:**

**1.starts-with (starting values should not change)**

**2.ends-with ((ending values should not change)**

**3.contains() (Note: real time only this method will use) (starts+ends both will consider)**

**4.Xpath using OR & AND**

**If we are providing more than one property, we will go for AND**

**Ex: [@Name: Sumanth AND LastName: xyz]**

**// [@role=’button’ and @id=’u\_0\_0\_CM’]**

**5. index: - whenever attribute values are duplicate we will use index, we should close xpath while writing index** ” (//\*[@id, ‘u\_0\_0\_’])[2]”

**SYNTAX:-**

//\*[starts-with(@id, ‘u\_0\_0\_’)]

**//\*[contains (@id,’constant value’)] (we need to provide any constant value from the value)**

**Ex:- (01test02)**

**(03test04)**

**Note: - for duplicates we can use indexing to get single match** ” (//\*[@id, ‘u\_0\_0\_’])[2]”

**Scenario name will be class name**

**Data Types**

**Methods**

**Access specifiers**

**Objects: Instance of a class**

**IMP: sendKeys(Keys.ENTER);**

**Keys.method name like ADD or ESC system buttons to perform actions**

**Diff between implicitwait () vs Thread.sleep()**

**Implicitwait (): - it will wait to load object data which is not present on UI**

**Ex: if we click on forget password it will take some time to load error message this time has to be maintained by using implicitwait()**

**Thread.Sleep(): -**

**It will stop execution for milli seconds to Load or render which are already present in the UI**

**Ex: in Rahul Shetty academy page if we click on forgot password it will re arranging page but selenium immediately looking for data but mean time it is InProgress to arrange page**

**By Using Regular Expressions also, we can build a Xpath or CSS:**

**CSS Ex: input[type=’password’]**

**If last 4 letter of the type is dynamic then we can search field by first 4 letters if those are static by using Regular expression for this we need to use \* it represents regular expressions**

**Syntax: //TagName [attribute=\*’value’]**

**Css Ex: input [type=\*’pass’]**

**For Xpath Little bit complex to write regular expressions:**

**Ex: //input[contains(@class,’submit’)]**

**Parent to Child Traversing using xpath:**

**driver. findElement (By. xpath("//div[@class='forgot-pwd-btn-conainer']/button[1]")).click();**

**TagName Locator:**

**Syntax: Tagname(“p”);**

**If we have only single tag without any attribute, we can go for TagName but it is not recommended because single page can have multiple same tags but if page contains only one tag with that name we can use**

**Assertions:**

**TestNg frame work give lot of Assertions so make it sure dependency should be present in Pom.xml**

**Assert is a class if we click . then we will get related method use appropriate method**

**<scope>test</scope>**

**Note: If we don’t remove scope tag from pom.xml then testNg jars wont Load**

****

**IMP: Xpath writing using button Text: //button[text(),’Log Out’]**

**IMP: We can write like this by using star //\*[text(),’Log Out’]**

**FB Log out : //span[text()='Log Out']**

**Note: There is no css locator to find elements based on button text**

**Xpath: Parent to Sibling Traversing (Not applicable for CSS)**

**//header/div/button[1]/following-sibling::button[1]**

**Xpath: Sibling to Parent Traversing (Not applicable for CSS)**

**//header/div/button[1]/parent::div**

**Dropdowns:**

**Static Dropdown’s:**

**If any Select tag contains for dropdown then we have a special class in Selenium Select and that dropdown should be static dropdown and we need to pass argument WebElement instance of dropdown**

**3 Ways Are there to get Static dropdown values, please refer below code**

****

**Dropdown : 2 (Without having select tag, for this we no need to use Select class, directly we can call methods by using locators**

**EX:**

**Dynamic Dropdown with indexes**

**Ex: driver.findElement(By.xpath("(//a[@value='DEL'])[2]")).click();**

**Parent Child Relationship xpath**

**Ex: driver.findElement(By.xpath("//div[@id='ctl00\_mainContent\_ddl\_destinationStation1\_CTNR'] //a[@value='MAA']")).click();  
  
**

**AutoComplete Dropdown Feature Script**

**We can use Find Elements method along with return type WebElement for AutoComplete dropdown**

****

**How to Find or count Number of Check boxes in page?**

**Find common text in all the check boxes and print it**

**Ex: driver.findElements(By.xpath(“//input[@type=’checkbox’]”).size();**

**Abstraction: Interface**

**Hiding the implementation, providing necessary information to the user, Hiding the unnecessary information to the user**

**Interface: it is also similar to class but it contains incomplete methods only**

**Class: It contains with body**

**//In selenium WebELement is a one of the data type to store to Locator address**

**WebElement googleSearchBox = driver.findElement(By.name("btnK"));**

**Assert.assertEquals(actualData, "Hello World");**

**ListBox: Dropdown multiple fields selection**

//other way to click on notifications ex:fb notifications

driver.findElement(By.*xpath*("//\*[name()='path' and contains(@d, 'M7.847 23.488C9.207 23.488 11.443 23.363')]")).click();

**NoSuchElementException:**

**=======================**

**1. Locator is wrong**

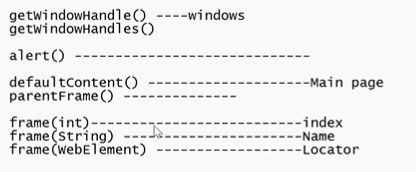
**2. Element is Missing.**

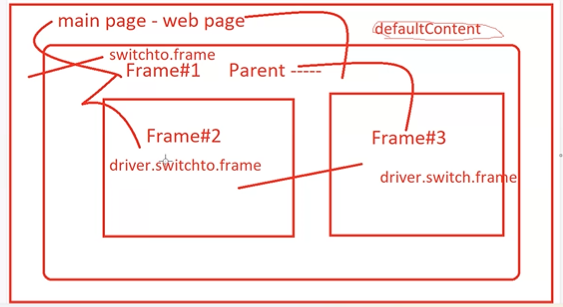
**3. Sync --- slow ---implicitwait(10)**

**4. Iframe**

**IFRAMES**

**Directly we can’t switch one frame to another frame , First we need to go to Default page or parent page then try to switch to targeted frame then it will work**





driver.switchTo().defaultContent();

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



**How to Perform Mouse Hover in Selenium?**

**In Selenium by using Actions class we can achieve hover actions**

Actions action = new Actions(driver);



MouseHover

doubleclick

DragAndDrop

ContextClick will be used for right click

pagesource()

**TESTNG FEATURES**

**Majorly used for execute multiple test cases at a time**

**---Annotations**

**@Test (It is going to replace main method)**

These two are optional

**@BeforeMethod**

**@AfterMethod**

**XML  
a) By class**

**b) By package**

SEQUENCING

PRIORITY

DEPENDS ON METHODS

SKIP TEST SCRIPTS (enabled= false)

PARLLE EXECUTION

Note: Most of the companies’ class means 1 test Case

Note: Some companies class means 1 feature that contains multiple testcases

*@Test* priority = 0)

public void login()

1.Priority

2.dependsonMethods"

3.skip) (enable= false to skip for reenable its true) or @Ignore



Hover DragANDDrop RightCLick DoubleClick



@Test

**public** **void** test() **throws** IOException {

// Code to capture the screenshot

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

// Code to copy the screenshot in the desired location

FileUtils.copyFile(scrFile, **new** File("C:\\CaptureScreenshot\\google.jpg"))

}

}

**Parameterization (pass data from different sources)**

Source:

1.testNG.xml

2. Excel (Preferred mechanism In RealTime)

3. Properties File (Notepad File)

Ex: @Test

Login (username, password)



**Parallel execution (TestNg recommends thread-count=”5”)**TestNg : it will run testcases one by one like synchronized way but here we are executing test cases parallelly, so we need to specify in xml in test Tag parallel =”methods”, max thread-count =”5” recommended  
Note: No need to use any tag in Classes to perform parallel execution

<test name="Test" thread-count="5" parallel="methods">

<suite name="KosmiK" thread-count="5" parallel="tests">

****

**Cross Browser Testing**

**To perform cross browser testing we can add multiple tests in XML   
1 test for Chrome Browser, 1 test for edge browser**

****

**Assertions**