Stage 1 - Programming Language - Java

Stage 2 - Selenium (Web Application) & AutoIT (Windows automation)

Stage 3 - Hybrid Framework

Stage 4 - CI/CD - Git and Jenkins

Nov 29, 2021

Selenium - <https://www.selenium.dev/>

* Open Source
* Language independent - Java, C#, Python, Ruby, Javascript, php, perl

Selenium - A suite of tools

1. Selenium IDE
   1. No need programming language
   2. Record and playback feature
   3. Supports - Chrome, Firefox, Edge
   4. Simple scripting or exploratory testing
2. Selenium RC - Depreciated
   1. Programming language is must
   2. Java, C#, Python, Ruby, Javascript, php, perl
   3. Architecture

Source code (Java+Selenium RC) 🡪 RC Server (Turn on/off) 🡪 Browser

1. Selenium WebDriver
   1. Programming language is must
   2. Java, C#, Python, Ruby, Javascript, php, perl
   3. Architecture

Source code (Java+Selenium WebDriver) 🡪 Browser

1. Selenium Grid
   1. Hub and Node

Java Programming -

* Installation
  1. **JDK 8 (Java Development Kit)** 
     1. By default - it will install JRE (Java Runtime Environment)
  2. IDE
     1. Eclipse
        + <https://www.eclipse.org/downloads/packages/>

<https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/2021-12/R/eclipse-java-2021-12-R-win32-x86_64.zip>

* + 1. IntelIJ
    2. Netbean
  1. Browser

<https://ftp.mozilla.org/pub/firefox/releases/94.0.2/win64/en-US/>

* Architecture

Source code (.java) 🡪 Byte code (.class) 🡪 O/P

Compile time vs Runtime

Compile time - Source code (.java) 🡪 Byte code (.class)

Runtime 🡪 Byte code (.class) 🡪 O/P

* UpperCamelCase - MyFirstProject

lowerCamelCase - myFirstProject

* Eclipse Structure

Workspace (lowercase)

Project 1 (UpperCamelCase)

Package 1 - lowercase (com.vfislk.purpose or org.vfislk.purpose)

Class (UpperCamelCase)

Methods & variable (lowerCamelCase)

Package 2

Project 2

* Datatypes
  1. Primitive datatypes / Pre-defined
     1. Byte - 8 bit

7 bit for storing the number

1 bit for sign + or -

* 1. Non-Primitive datatypes / non-predefined - collection of primitive datatypes
     1. String
        + Zero based index
     2. Array
        + Zero based index
        + Size is fixed
* Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5

at com.vfislk.datatypes.DatatypesDemo2.main(DatatypesDemo2.java:17)

* Relational operators -

==, !=, >,<,>=,<=

* Logical operators - And (&&), or (II) not
* Debugging
  1. Breakpoint
  2. Resume
  3. Terminate
  4. Step over
  5. Step into
* Conditional statements
  1. If , if else, if else if
  2. Switch

Nov 30, 2021

* Iterative statement
  1. For
     1. Start point
     2. End point
     3. Iteration
  2. For each/ advance for loop - designed for handling array and collections
  3. While
  4. Do while - minimum once it will run the code
* Access modifier
  1. Private - accessible within the class
  2. Default - accessible within the package
  3. Protected - accessible within the package and also with inheritance
  4. Public - anywhere
* Methods - Building block of the program
  1. Reusability
  2. Maintenance
* Static Methods
  + Create and Call it
  + To call the static method

Classname.methodname()

//accessmodifier static returntype methodname(arguments)

* Non-Static Methods
  + Create and Call it
  + To call the non-static method
    - Create object for that class
    - Use objectref.methodname()
* Object
  1. Declaration (Area obj)
  2. Instantiation - new (allocate memory)
  3. Initialization (Area())
* Variable
  1. Static variable
  2. Non-static variable

Dec 1, 2021

* Class & Object
  1. Real time example
  2. Creating Employee type, student type
* Encapsulation
  1. Process of hiding the internal members.
* this keyword - helps to distinguish between instance variable and local variable.
* Constructor - Anything pre-requisite for the object
  1. Constructor name and class name should be same
  2. When no constructor created then there will be default constructor which will initialize all the non-static variable
  3. Can create constructor with parameter or without parameter
  4. When constructor is created explicitly then you need to call that constructor on creating object
* Import option:

Dec 2, 2021

* Created ChromeDriver object by setting up the selenium webdriver environment.
* Constructor overloading

We can create multiple constructor inside the class

By change

1. Number of parameters
2. Datatypes of parameters
3. Sequence of parameters

The constructor to be called is resolved during compile time.

* Method overloading/ Compile time polymorphism/ early binding

We can create multiple method using same name inside the class

By change

1. Number of parameters
2. Datatypes of parameters
3. Sequence of parameters

The method to be called is resolved during compile time.

* Inheritance
  1. Reuse the methods
* Method overriding/ Run time polymorphism/ late binding
  1. Recreate the method in child class with different definition.
  2. Object of the child class should be stored in parent.
  3. Based on object created if the method is overridden then that method will be called.
  4. The method to be called is resolved runtime.
* Abstract class - set of the rules to be followed by child class
  1. If no definition known then we can make the method as abstract.
  2. If any of the method is abstract, you need to make class the abstract.
  3. Cannot instantiate the abstract class
* Interface - - set of the rules to be followed by child class
  1. Contains only method declaration
  2. All methods are by default public and abstract
  3. Cannot instantiate the interface
* Collections
  1. Non-Generic type (not -recommended)
  2. Generic type
     1. ArrayList
* List vs Set
  1. List can contain duplicate records
  2. Set cannot contain duplicate

Designing of WebDriver:- \_\_\_\_\_\_

* Achieve cross browser testing (code should be present in all browser class)
* Lower level module should depends on higher level module
* To provide set of rules for browser class
* == vs string equals method
  1. == 🡪 compare location and then the text
  2. string equals method🡪 compare the text

Dec 3, 2021

Selenium WebDriver

1. Create a java project
2. Download and configure the selenium jar (4.1.0)
   1. Download <https://www.selenium.dev/downloads/>
   2. Right click on project 🡪 Build Path🡪 Configure Build path🡪 Libraries 🡪 Choose classpath🡪 add external jar
3. Exception in thread "main" java.lang.IllegalStateException: The path to the driver executable The path to the driver executable must be set by the webdriver.chrome.driver system property;
4. To fix above error:

Download and configure the driver based on the browser and browser version.

<https://chromedriver.chromium.org/downloads>

Use anyone option:

* Keep the driver in project home directory
* Using System.setProperty
* Using webdrivermanager (automatically download and driver)

1. Chrome, firefox, Edge
2. Internet Explorer
   1. Zoom level - 100 %
   2. Security mode should be same either disable or enable (check img 1.3)
3. Selenium doc - <https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebDriver.html>
4. Inspect 🡪 tagname, attribute or not , text or not
5. Basic Locators (static methods from the class - By)
   1. Id
   2. Name
   3. Classname
   4. Tagname
   5. Linktext
   6. Partial link text

When the locator is pointing to multiple element then findElement will pick the first one.

1. Advance locators
   1. Xpath
   2. CSS
2. Exception in thread "main" org.openqa.selenium.NoSuchElementException: no such element: Unable to locate element: {"method":"link text","selector":"Logout"}
3. driver.findElement(By.*id*("txtUsername")) 🡪 check for presence of element in 0.5s
4. Synchronization
   1. Unconditional wait

Thread.*sleep*(5000); 🡪 wait for 5sec

* 1. Conditional wait (from selenium jar)
* Implicit wait
  + Default implicit wait = 0s
  + Applicable for all findElement and findElements methods
  + Example: Implicit wait = 30s
    1. If element is not present then it throws exception after 30s
    2. If element is present then will do the operation immediately
    3. Polling time (how frequently it check for the element) - 0.5s
* Explicit wait
  + Exact condition
  + Timeout (need to mention)
  + Polling time (how frequently it check for the element) - 0.5s

1. Click, sendKeys, Select
2. Dropdown
   1. Dropdown with select tag (Select class)
      1. selectByIndex
      2. selectByValue
      3. selectByVisibleText
   2. Dropdown without select tag
      1. Just use click and solve

Dec 6, 2021

1. Gettext(), getAttribute(), getTagName()
2. Exception in thread "main" org.openqa.selenium.ElementClickInterceptedException: element click intercepted: Element <a href="/meeting/trial" class="button button--secondary button--large" data-cta="1">...</a> is not clickable at point (329, 570). Other element would receive the click: <div class="trustarc-banner-footer">...</div>
3. Frame, Multiple tabs/windows, alert - swtichTo()
4. Frame
   1. Frame - even though the locator is correct still you get nosuchelement exception
   2. Check for tagname either iframe or frame.
   3. To get into the frame (anyone)
      1. Index

driver.switchTo().frame(0);

* + 1. Name or id as a String

driver.switchTo().frame("login\_page");

* + 1. WebElement

driver.switchTo().frame(driver.findElement(By.*xpath*("//frame[contains(@src,'RSNBLogin.html')]")));

1. To inspect - ctrl+shift+i
2. Alert - javascript alert
3. Exception in thread "main" org.openqa.selenium.NoAlertPresentException: no such alert

Dec 7, 2021

1. Multiple tabs/windows
   1. How to handle two tabs/windows
   2. Each tab/window will have a session Id (dynamic)
   3. driver.getWindowHandles() 🡪 returns all the session id
2. Exception in thread "main" org.openqa.selenium.NoSuchWindowException: no such window: target window already closed
3. Close vs Quit
   1. Close - Close the current tab/session
   2. Quit - close the browser and also kill the process associated with it.
4. Navigate - back, forward, refresh
5. Web Table
6. isSelected(), isDisplayed(), isEnabled()
7. Selector hub - plugin

<https://chrome.google.com/webstore/detail/selectorshub-xpath-plugin/>

Dec 8, 2021

1. Exception in thread "main" org.openqa.selenium.ElementNotInteractableException: element not interactable
2. Actions class -mouse/keyboard activities
   1. Will not give proper errors
   2. Will not support headless mode
   3. While runtime, you should not disturb the mouse/keywords
3. Keyboard activities
   1. Modifier keys (ctrl, alt, shift) 🡪 KeyDown(), KeyUp()
   2. Remaining keys 🡪 sendKeys()
4. FindElements
   1. Count of the locator
5. Dynamic pages
6. Exception in thread "main" org.openqa.selenium.StaleElementReferenceException: stale element reference: element is not attached to the page document

Dec 10, 2021

1. Upload
   1. Use sendkeys --

driver.findElement(By.*id*("edit-field")).sendKeys("D:\\Profile.docx");

* 1. Using AutoIT

1. Windows authentication
   1. Using url - <http://Username:Password@SiteURL>
   2. Using robot class from java libraries
   3. Using AutoIT
2. Options - ChromeOptions, FirefoxOptions
   1. Headless mode browser
   2. Chrome notifications
   3. Download directory
   4. To provide the directory of chrome browser/firefox browser (binary exception)
3. Javascript
   1. When facing issue with click and type
   2. Click on hidden elements
   3. Scroll to element/ scroll windows
   4. Table with scroll
   5. Get the tool tip details
   6. While typing using sendkeys method will not do validation in some text box. Just try using javascript type.
4. Javascript - click & type

document.getElementById('authUser').value='Balaji'

document.querySelector('#authUser').value='king'

document.querySelector("[type='submit']").click()

document.querySelector("[data-bind='click: logout']").click()

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("document.querySelector(\"[data-bind='click: logout']\").click()");

1. Javascript + webelement (xpath)

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("arguments[0].click()",driver.findElement(By.*xpath*("//li[@data-bind='click: logout']")));

js.executeScript("arguments[0].value='bala'",driver.findElement(By.*id*("authUser")));

Scroll into view

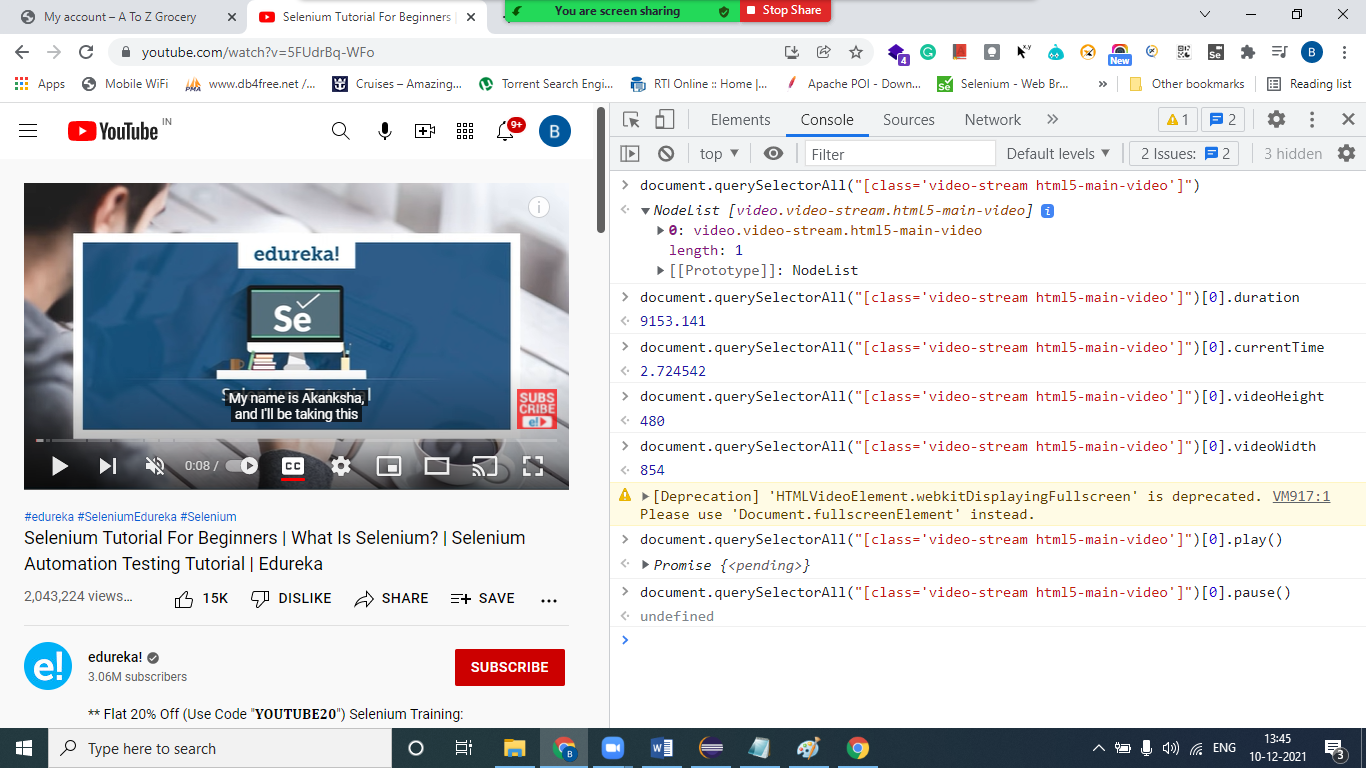
document.querySelector("a[id='gnav\_1165'] span span span span").scrollIntoView()

Windows scroll

js.executeScript("window.scrollBy(0,100)");

1. Javascript advance

document.querySelectorAll('#reg\_email')



1. AutoIT
   1. Windows auth
   2. Upload

Dec 13, 2021

Hybrid Framework

* Unit Test Framework - TestNG - <https://testng.org/doc/documentation-main.html>
* Data Driven Framework - excel, properties
* Page Object Model - Design pattern
* Key word driven framework (optional right now - it’s better to implement to reuse the code)

4 Pacakges

1. Package 1 - Test Methods
2. Package 2 - Browser configuration
3. Package 3 - handling excel, properties
4. Package 4 - Page object Model- classes

Maven - Build Management tool

Pom.xml - Project Object Model

* Jars to be configured to the project.

Steps to create framework

1. TestNG Plugin - Refer the document



1. Create Maven Project 🡪Provide groupId and ArtifactId
2. If any error in created maven project

Right click on project name 🡪 Maven 🡪 update project 🡪 click force update

1. Any access issue, we need to add the setting.xml (where you will provide proxy, server details)
2. Create a package for setting up test methods under src/test/java
3. Create a test class
4. Create a test method by providing annotation (@Test)
5. To import the TestNG details - add dependency in the pom.xml

<https://mvnrepository.com/artifact/org.testng/testng/7.4.0>

1. Testng documentation - <https://testng.org/doc/documentation-main.html>
2. Test Method (@Test) trigger in the ASCII key code order.
3. Use priority to order the test method. If we don’t provide priority then that method will be given higher precedence
4. Invocation count
5. Annotation

@BeforeMethod - browser launch

@Test - valid credential test

@AfterMethod - close the browser

@BeforeMethod - browser launch

@Test - invalid credential test

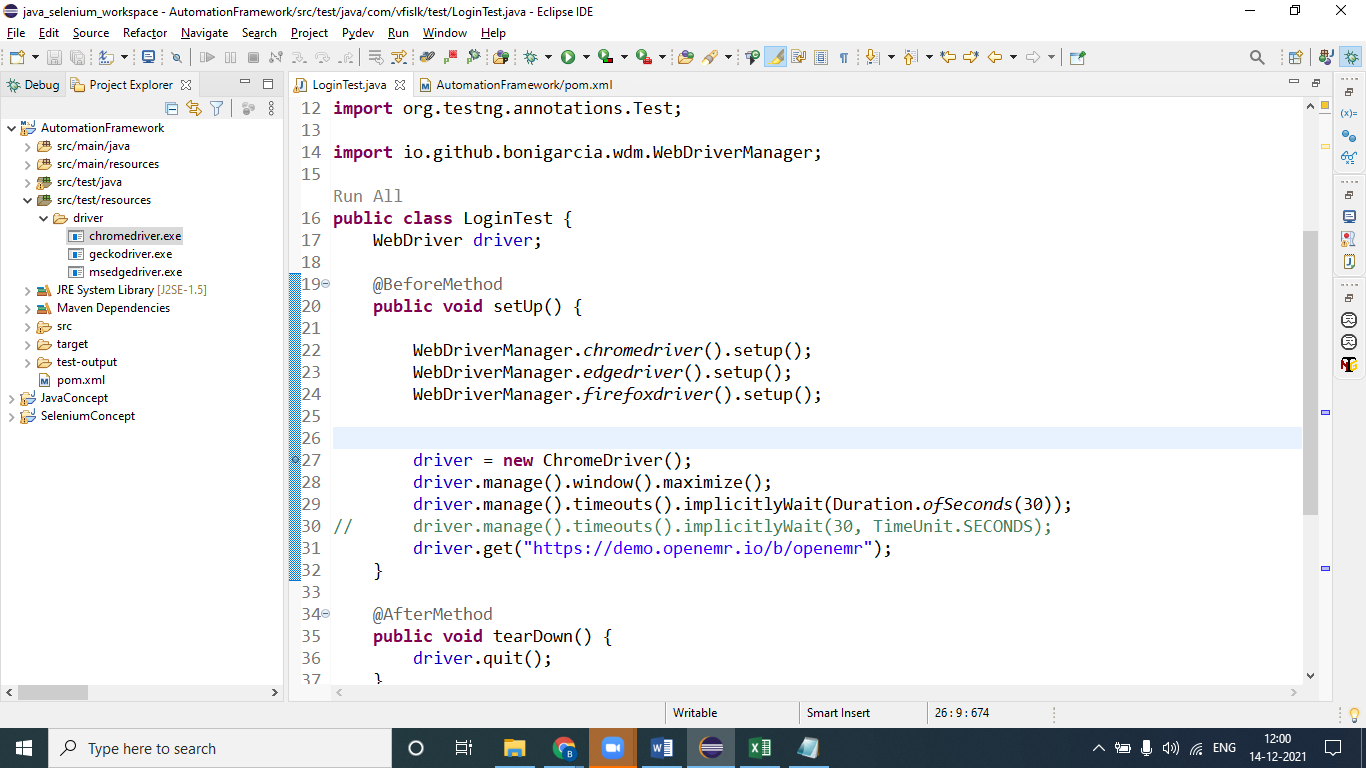
@AfterMethod - close the browser

1. Add Selenium dependencies

<https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium-java/4.1.0>

1. Setting up driver - using webdrivermanager - add dependency in the pom.xml and refer image

<https://mvnrepository.com/artifact/io.github.bonigarcia/webdrivermanager/5.0.3>



1. If point 15 is not working, use System.setproperty
2. Minimum one verification should be added to the test method (@Test)

Assert.*assertEquals*(actualTitle,"OpenEMR");

1. Ctrl+shift+i+o 🡪 to remove unwanted imports
2. Dataprovider & parameterization
3. Dataprovider
   1. Create a test method (@Test) with arguments and use it inside the method
   2. Create a method, provide @Dataprovider and should return two dimensional array.
   3. Connect the dataprovider with the test method
4. Excel (.xls or .xlsx) - <https://poi.apache.org/>
   1. add excel dependency in the pom.xml

POI:

<https://mvnrepository.com/artifact/org.apache.poi/poi/5.1.0>

POI OOXML:

<https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml/5.1.0>

xls - HSSFWorkBook

xlsx - XSSFWorkBook

Excel:

1. Location (read/write)
2. Open
3. Sheet
4. Row
5. Cell
6. Excel with Dataprovider
   1. Create a test method (@Test) with arguments and use it inside the method
   2. Create a method, provide @Dataprovider and should return two dimensional array by calling the

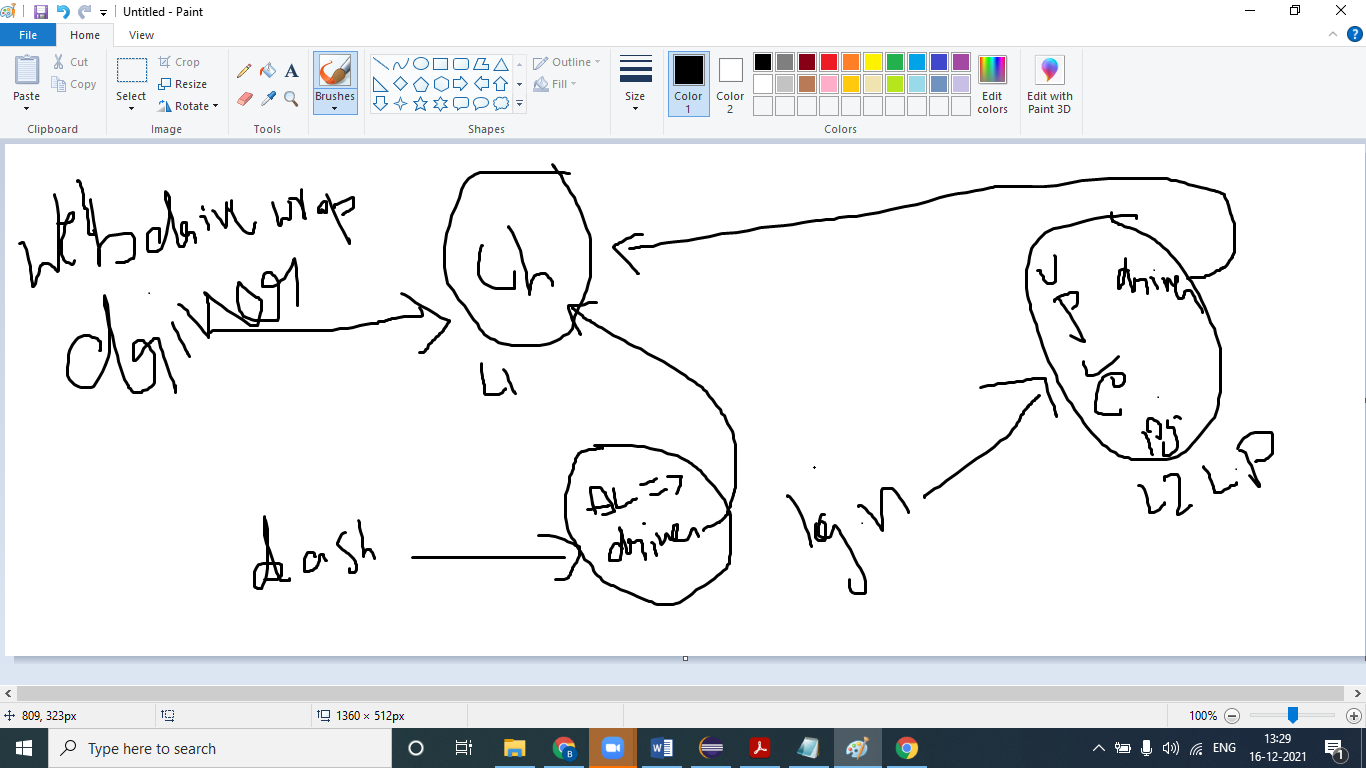
ExcelUtils. getSheetIntoObjectArray(“”,””)

* 1. Connect the dataprovider with the test method

1. Page Object Model - design pattern
   1. Reusability
   2. Maintenance

Steps: -

1. For each web page in the application, we will create a class - Page class
2. Operation will happen through the methods - Page Methods
3. Locators will be collected and kept at class level or in different file - Object repositories



Notes:

1 bit - 0 or 1

1B - 8 bit

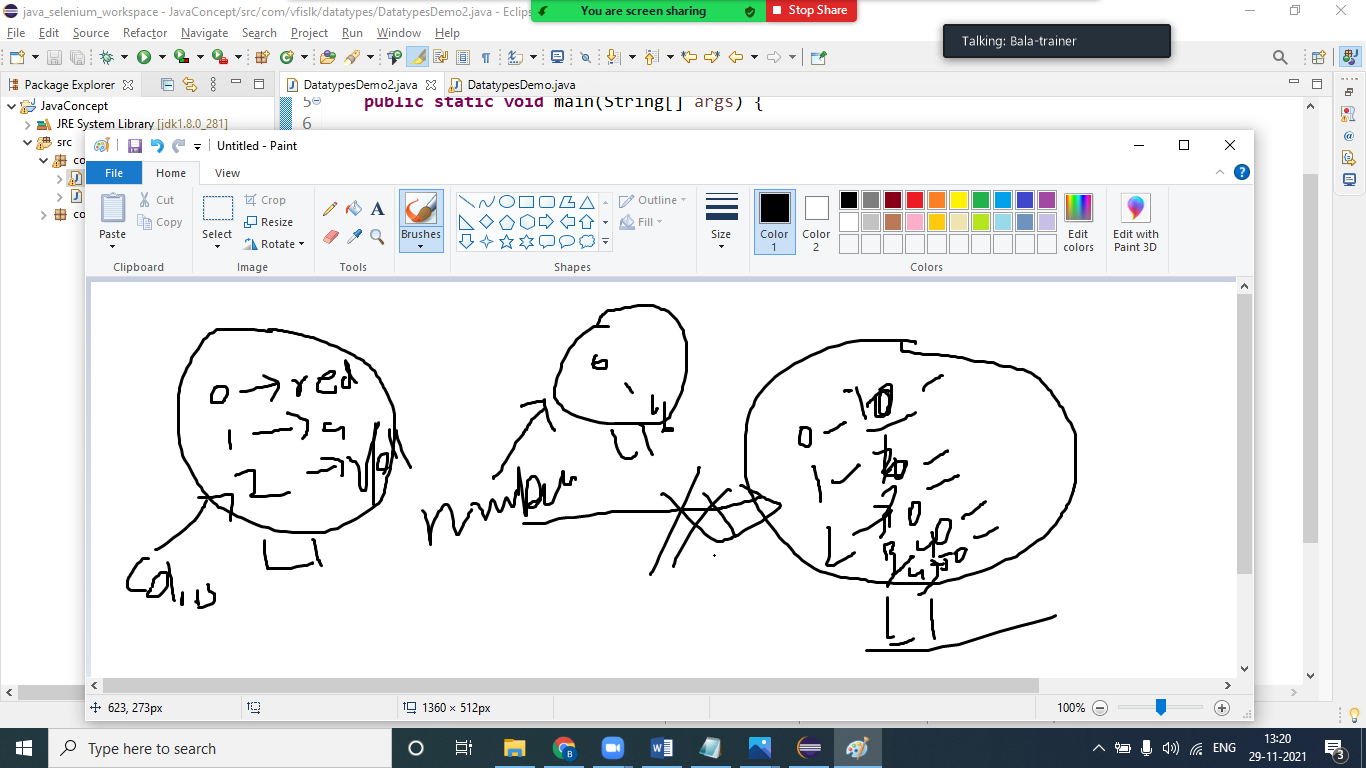
1KB - 1024B

1 MB - 1024KB

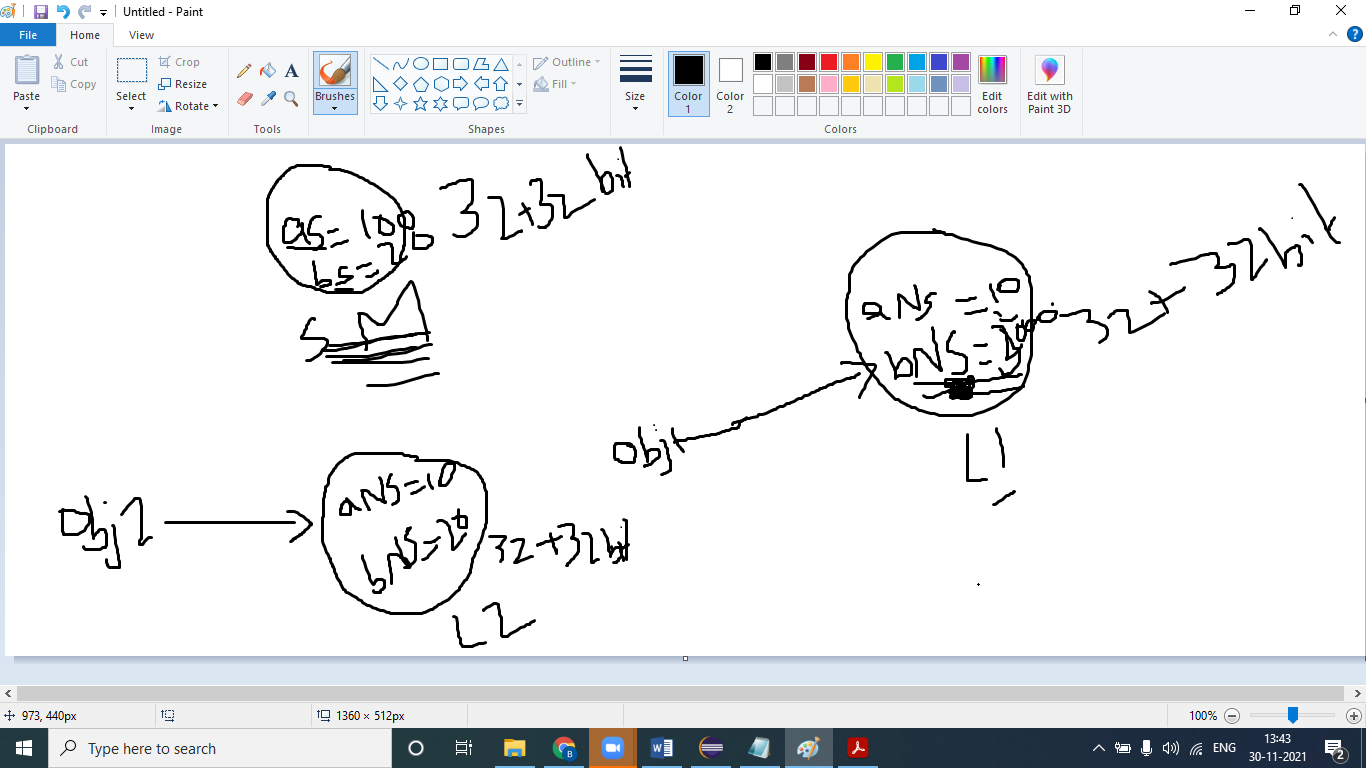
1 GB - 1024 MB

Ref

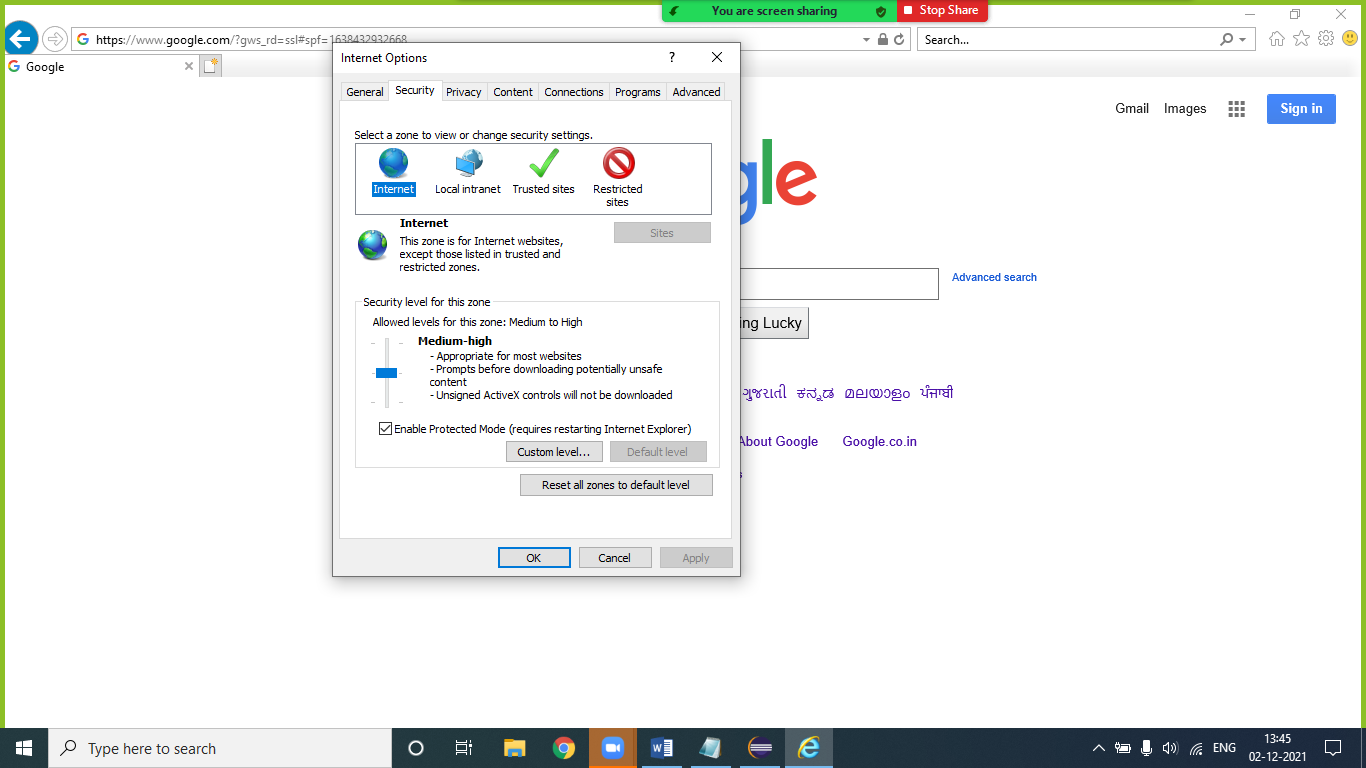
Img 1.0



Img 1.2



Img 1.3



Selenium

By locator= By.*id*("txtUsername");

WebElement element= driver.findElement(locator); element.sendKeys("Admin");

--------------------------------------

WebElement element= driver.findElement(By.*id*("txtUsername"));

element.sendKeys("Admin");

---------------------------------------

WebElement element= driver.findElement(By.*name*("languageChoice"));

Select selectLang=new Select(element);

selectLang.selectByVisibleText("English (Indian)");

https://github.com/balaji-githubstore/VFISLK3.git

Task2:

1. Create Student type with below details (decide between static and non-static variable)

**Student**

* **studentRollno (public)**
* **studentName (public)**
* **studentMailid (public)**
* **studentPercentage (public)**
* **schoolName (public)**

1. **Create 3 different instance for storing below values**

**1001,"jack",jack@gmail.com,45.2, Global school**

**1002,"peter",peter@gmail.com,85.2, Global school**

**1003,"mark",mark@gmail.com,56.5, Global school**

1. Create a method for printing Student details

// public static void printEmployeeDetails(Employee emp)

// {

// System.out.println(emp.empId);

// System.out.println(emp.empName);

// System.out.println(emp.empSalary);

// System.out.println(Employee.companyName);

// System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

// }

// Employee.printEmployeeDetails(emp1);

// Employee.printEmployeeDetails(emp2);

// Employee.printEmployeeDetails(emp3);

**for**(**int** i=1;i<=10;i++)

{

System.***out***.println("i value = "+i+" kkkk");

System.***out***.println("//table[@id='example']/tbody/tr["+i+"]/td[2]");

}