1. **What is Selenium WebDriver?**

Selenium WebDriver is the main component of Selenium

Practically speaking, Selenium WebDriver is a good list of API's for automating the Web Applications.



1. **What is Selenium Grid and when do we go for it?**

* Selenium Grid is one of the four components of Selenium.
* Selenium Grid is used to distribute the automated tests across different machines, having different OS, Browsers and their versions for execution.
* Selenium Grid only distributes the tests across multiple machines having different configurations, but don't have the capability for Parallel Execution.
* Selenium Grid when used with TestNG, we can achieve parallel execution of automated tests across multiple machines.

1. **What are the advantages of Selenium Grid?**

* Distribution of Selenium Automated Tests for execution across the Multiple machines having

different OS, Browsers and Browser Versions.

* Selenium Grid when used with TestNG, can enable Parallel execution of Tests across multiple

machines, which will reduce the time of execution of Tests without reducing the performance of the machines.

1. **What is a Hub in Selenium Grid?**

* As part of Selenium Grid Configuration, we configure a single machine as Hub and other remaining machines as Nodes.
* Hub machine acts as a central point for distributing the Selenium Automated Tests to Node machines.
* Automation Tests will be available in the Hub machine.
* Automation Tests are triggered for execution from Hub machine, but the triggered tests will be executed in different Node machines.

1. **What is a Node in Selenium Grid ?**

* As part of Selenium Grid Configuration, we configure a single machine as Hub and other remaining machines as Nodes.
* Actual execution of the Automated Tests happen in the Node machines
* Node Machines receive the test execution requests from Hub machine
* There can be more than one Node machines in the Selenium Grid configuration
* Node Machines can have different Platforms and Browsers
* Node Machines platform can be different from Hub machine

1. **What are the different types of APIs in Selenium WebDriver?**

* Selenium WebDriver itself is an API
* WebDriver is an Interface in this API
* WebDriver API has the below implementing Classes :
  1. ChromeDriver
  2. ChromiumDriver
  3. EdgeDriver
  4. EventFiringDriver
  5. FirefoxDriver
  6. InternetExplorerDriver
  7. OperaDriver
  8. RemoteWebDriver
  9. SafariDriver

1. **Which WebDriver implementation claims to be the fastest?**

* Earlier HtmlUnitDriver used to be the fastest headless browser.
* As HtmlUnitDriver and PhantomJS are now deprecated, we are left with the Chrome.
* Headless and Firefox Headless
* Hence, we can answer, in the current days Chrome Headless and Firefox Headless are

claimed to the fastest, as they run without rendering GUI.

WebDriverManager.chromedriver().setup();

ChromeOptions options = new ChromeOptions();

Options.addArguments(“--headless”)

WebDriver driver = new ChromeDriver(options);

driver.get(https://google.com);

1. **What are few Open Source Frameworks supported by Selenium WebDriver?**

The below are the list of few open source frameworks supported by Selenium WebDriver:

* Appium
* Protractor
* WebDriverIO
* CodeceptJS
* Robot Framework
* Carina
* Galen Framework
* Serenity

1. **What is the difference between Soft Assertion and Hard Assertion?**

* Hard Assertions when failed will stop the execution and the remaining steps in the

Automation code won't be executed.

* Soft Assertions when failed, will still continue the execution and hence the remaining steps in the Automation code will be executed

//**Hard Assertion**

Assert.assertTrue(driver.getTitle().equals(“xyz”)) //Test execution will stop here and steps after that will noy be executed.

//**Soft Assertion**

SoftAssert soft = new SoftAssert();

soft.assertTrue(driver.getTitle().equals(“xyz”)); // Test execution will not stop here

// code

soft.assertAll(); // Error will be reported else we will not see any failure

1. **What are the different verification points available in Selenium?**

**Selenium IDE**

* Verify and Assert commands

**Selenium WebDriver**

* isDisplayed()
* isEnabled()
* isSelected()
* size() - Element is present or not
* getTitle()
* getCurrentURL()
* getPageSource()

1. **What are the different types of Exceptions you have faced in Selenium WebDriver?**

The below are the different types of Selenium WebDriver Exceptions:

* NoSuchElementException
* NoSuchWindowException
* NoSuchFrameException
* NoAlertPresentException
* InvalidSelectorException- if issue in xpath.
* ElementNotInteractableException – if element is hidden.
* NoSuchSessionException- if trying to access an element after closing the browser.
* ElementNotSelectibleException
* TimeoutException
* StaleElementReferenceException

1. **Why do we create a reference variable ‘driver’ of type WebDriver and what is the purpose of its creation? (Interview Question #12)**

* For writing Selenium Automation code once and running it on different browsers
* Few things to understand:
  + - * + We cannot create an object for WebDriver interface.
        + Objects created for ChromeDriver, FirefoxDriver, InternetExplorerDriver can be upcasted to WebDriver interface.

String browser =”chrome”;

WebDriver driver=null;

If(browser.equalsIgnoreCase(“chrome”))

{

WebDriverManager.chromedriver().setup();

Driver= new ChromeDriver();

}

else if(browser.equalsIgnoreCase(“firefox”))

{

WebDriverManager.firefoxdriver().setup();

Driver= new FirefoxDriver();

}

1. **How to login into any site using Selenium if it is showing an authentication pop-up? (Interview Question #13)**
   * + - * https://the-internet.herokuapp.com/
         * driver.get("http://admin:admin@ the-internet.herokuapp.com/basic\_auth");

//here first admin is username and second admin is password for the authentication pop-up

1. **What is Implicit Wait in Selenium WebDriver? (Interview Question #14)**

* Implicit Wait is a Global Wait which waits for all the Web Elements equally
* Unlike Thread.Sleep, Implicit Wait waits for the Web Elements dynamically
* driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS);

1. **What is Explicit Wait in Selenium WebDriver? (Interview Question #15)**

* Explicit Wait - Instead of waiting for all the Web Elements in the automation script, Explicit wait will wait only for the specific web element - Demonstrate here

WebDriverWait wait = new WebDriverWait(driver, 5);

WebElement element =

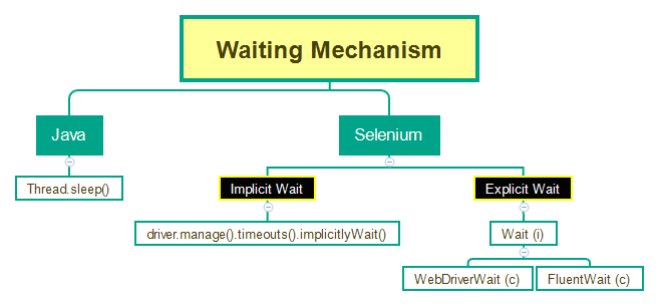
wait.until(ExpectedConditions.visibilityOfElementLocated(By.linkText("Facebook")));

element.click();

* Also demonstrate 'ElementToBeClickable'

1. **What is Fluent Wait in Selenium WebDriver? (Interview Question #16)**

* Fluent Wait is an advanced version of Explicit Wait



* Use Duration.ofSeconds(30) in the deprecated methods
* Polling time can be customized in Fluent Wait whereas WebDriverWait has a default polling time of .5 seconds.

Wait<WebDriver> wait= new FluentWait<WebDriver>(driver)

.withTimeout(Duration.ofSeconds(30))

.pollingEvery(Duration.ofSeconds(2))

.ignoring(NoSuchElementException.class);

WebElement foo = wait.until(new Function<WebDriver,WebElement>() {

public WebElement apply(WebDriver driver) {

return driver.findElement(By.id("foo"));

}

});

1. **How to enter text into text field without using sendKeys()? (Selenium Interview Question #17)**

JavascriptExecutor can be used for this purpose.

Webdriver driver = new ChromeDriver();

WebElement searchBox = driver.findElement(By.name(“search”));

JavascriptExecutor jse = (JavascriptExecutor) driver;

Jse.executeScript(“arguments[0].value=’iMac’”, searchBox);

1. **How to clear the text inside the text box fields using Selenium WebDriver? (Selenium Interview Question #18)**
   * + Using clear() command of Selenium WebDriver
       - * driver.findElement(By.id("abc").clear();
2. **How to get an attribute value of an element using Selenium WebDriver?**
   * + - Using getAttribute() command of Selenium WebDriver
         * driver.findElement(By.id("abc").getAttribute("attributename");
3. **How to press Enter key using Selenium WebDriver?**

* By passing Keys.Enter into the sendKeys() command
* driver.findElement(By.id("abc")).sendKeys(Keys.Enter);

driver.findElement(By.xpath(“//input[@value=’Login’]”)).sendKeys(Keys.ENTER)

1. **How to pause execution using Selenium WebDriver?** 
   * + - driver.wait(5000) using selenium and thread.sleep using Java.
         * This will pause the execution for 5 seconds.
       - We need to put the above statement inside the synchronized block to work.

Synchronized (driver)

{

driver.wait(5000);

}

Driver.findElement(By.id(“email”)).sendKeys(“abc@gmail.com”)

1. **Is Selenium Server required to run Selenium WebDriver scripts?**

No, its not required, Selenium server was required with selenium RC.

1. **What happens if we run this command driver.get(“www.google.com”);?** 
   * + InvalidArgumentException
       - We need to pass the URL argument properly, by also providing http:// or https://
2. **What is an alternative to get() command in Selenium WebDriver?**

driver.navigate().to("URL");

1. **What is the difference between get(“URL”) and navigate().to(“URL”)?**

Both are same, except one of them takes less time to type than other

1. **What are the different navigation commands in Selenium WebDriver?**
   * navigate().to();
   * navigate().back();
   * navigate().forward();
   * navigate().refresh();
2. **How to fetch the current page URL in Selenium WebDriver?**

driver.getCurrentUrl()

1. **How can we maximize browser window in Selenium WebDriver?**

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

1. **How to delete cookies in Selenium?**

driver.manage().deleteAllCookies();

1. **What are the different ways for refreshing the page using Selenium WebDriver?**
   * driver.navigate().refresh();
   * **Actions class**

Actions actions = new Actions(driver) ;

actions.sendKeys(Keys.F5).build().perform() ;

* + **Javascript Executor**

JavascriptExecutor jse = (JavascriptExecutor) driver;

jse.executeScript("window.location.reload();");

* + driver.get(driver.getCurrentUrl());
  + driver.navigate().to(driver.getCurrentUrl());

1. **What is the difference between getWindowHandle() and getWinowHandles() in Selenium**

**WebDriver?**

* **getWindowHandle()** is used to get the window handle of the window, that is currently in focus
  + - Return type is String
    - Window Handle stores the unique address of the browser window
* **getWindowHandles()** is used to get the widow handles of all the current opened windows
* Return type is Set<String>

Assuming 2 windows are open, and focus is on first window:

String defaultWindow = driver.getWindowHandle() ;

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

Iterator<String> itr = allWindows.iterator();

itr.next(); //first window handle

String childWindow = itr.next(); // Second window handle

driver.switchTo().window(childWindow);

Thread.sleep(3000);

driver.close();

driver.switchTo().window(defaultWindow);

Thread.sleep(3000);

driver.close();

1. **How to handle hidden elements in Selenium WebDriver?**

* By default we cannot handle hidden elements in Selenium WebDriver
* Will result in Exceptions like ElementNotVisible or ElementNotInteractable Exceptions
* But, there is a way, where we can use JavascriptExecutor for handling hidden elements too
* We can execute the JavaScript code to handle hidden elements

// when the text box is in hidden mode still we can enter text into it using below code

JavascriptExecutor jse = (JavascriptExecutor) driver;

jse.execteScript(“document.getElementById(‘displayed-text’).value=’Adnan’;”);

1. **How can you find broken links in a page using Selenium WebDriver?**

* (If we get 4xx or 5xx HTTP Status codes in the response from the Server, when we hit any URL, then they are broken links.
* Here is the Selenium WebDriver code for finding the broken links in the page

package dpack;

import java.io.IOException;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import io.github.bonigarcia.wdm.WebDriverManager;

public class Demo {

public static void main(String[] args) throws InterruptedException {

WebDriverManager.chromedriver().setup();

WebDriver driver = new ChromeDriver();

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

driver.get("http://tutorialsninja.com/demo/");

List<WebElement> elements =driver.findElements(By.tagName("a"));

for(WebElement element : elements) {

String URL = element.getAttribute("href");

System.out.println("-------------------------");

System.out.println(URL);

if(URL==null || URL.isEmpty()) {

System.out.println("URL is empty");

continue;

}

HttpURLConnection huc;

try {

huc = (HttpURLConnection)(new URL(URL).openConnection());

huc.connect();

if(huc.getResponseCode()>=400) {

System.out.println(URL+" is broken");

}

else {

System.out.println(URL+" is a valid link");

}

} catch (MalformedURLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

driver.quit();

}

}

1. **How to find more than one web element in Selenium WebDriver?**

findElements()

List<WebElement> links = driver.findElements(By.tagName("a"));

System.out.println(links.size());

1. **How to read a JavaScript variable in Selenium WebDriver?**

JavascriptExecutor jse = (JavascriptExecutor) driver;

String title = (String) jse.executeScript("return document.title;");

1. **What is JavascriptExecutor and in which case JavascriptExecutor will help in Selenium automation?**

* JavascriptExecutor is a predefined interface in Selenium WebDriver API
* Using JavascriptExecutor, we can run JavaScript code in Selenium.
* We generally use JavascriptExecutor in Selenium WebDriver,

When the normal click() is not working

Selenium WebDriver don't have the capability to scroll the page.

And many more cases, where Selenium WebDriver's default things are not going to help.

JavascriptExecutor jse = (JavascriptExecutor) driver;

Jse.executeScript(“alert(‘Adnan’)”); // Javascript code for displaying an alert

// vertical scrolling downward, give -600 if need to scroll up

js.executeScript("window.scrollBy(0,600)", "");

// Scrolling down the page till the element is found

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

// scroll down to the bottom of the webpage

js.executeScript("window.scrollBy(0,document.body.scrollHeight)");

WebElement button=driver.findElement(By.id(“alert1”));

jse.executeScript(“arguments[0].click()”,button); //clicking button using Javascript

1. **How to handle Ajax calls in Selenium WebDriver?**

* Web Pages make Ajax calls, to retrieve small amount of data from server without the need for reloading the page.
* Selenium WebDriver handles Ajay calls using Waiting mechanism
  + Implicit Wait
  + Explicit Wait
  + Fluent Wait

1. **List some scenarios which we cannot automate using Selenium WebDriver?**

The below are few scenarios which we cannot automate using Selenium:

* Captcha (Demo)
* We cannot read bar code or QR code (Demo)
* OTP
* Video Controls
* Desktop Applications

1. **How you build object repository in your project framework?**

I have used Page Object Model and Page Factory to build Object Repository in my Project framework.

According to Page Object Model all the pages in the application should have separate Java file created where objects related to the particular page are kept.

Page factory design purpose is to support page object model.

Just need to create an object for the particular page and we can access all the element of that page.

@FindBy(xpath=”//a[@title=’My Account’]”)

WebElement myAccount;

1. **What is Page Object Model (POM) ?**

* Page Object Model is a design pattern.
* According to the Page Object Model:
  + - Each page in the Application is required to have its own corresponding Java class in the Project framework.
    - Web UI elements on a page, will be stored as Objects in the corresponding Java classes in the framework

1. **What is Page Factory?**

* PageFactory is a built-in class in Selenium WebDriver.
* Page Factory supports Page Object Model Design Pattern, by initializing Objects.

public class landing Page

{

public WebDriver driver;

public LandingPage(WebDriver driver)

{

this.driver=driver;

PageFactory.initElements(driver,this); // When object of a page is created, this is invoked and all the object of this page is initialized

}

@FindBy(xpath=”//a[text()=’Login’]”)

WebElement login;

@FindBy(xpath=”//a[text()=’Register’]”)

WebElement register;

}

1. **What is the difference between Page Object Model and Page Factory?**
   * + Page Object Model is a design pattern
       - * Each Web Page has a separate java class file, where Web UI Elements of pages
         * are stored as objects in their respective Java class files.
     + Page Factory is a built-in Selenium Class
       - * Supports Page Object Model
         * By initializing Page Objects automatically on creating instances.
       - Without using Page factory we might get stale element reference exception and we have to initialize element again and again.
2. **What are the advantages of Page Object Model?**

* Code Readability
* Best way to create Object Repository
* Easy to find Objects, as they are created in respective pages.

1. **How can we use Recovery Scenario in Selenium WebDriver?**

* Recover Scenario is a concept of QTP/UFT
* In Selenium Java, we use try catch block to create Recover scenarios.

1. **How to upload a file in Selenium WebDriver?**

* sendKeys() can be used to upload a file in Selenium WebDriver
* Uploading the files from a workspace folder instead of local folder
* driver.findElement(By.id("upload")).sendKeys("Path of the file");

String projectPath = System.getProperty(“user.dir”);

driver.findElement(By.id("upload")).sendKeys(projectPath + "\\Files\\Adn.jpg");

**46. How to download a file in SeleniumWebDriver?**

1. Create an object for ChromeOptions

* ChromeOptions options = new ChromeOptions();

2. Write the below ChromeOptions code for setting the download path

* HashMap<String, Object> chromePrefs = new HashMap<String, Object>();
* chromePrefs.put("profile.default\_content\_settings.popups", 0);
* chromePrefs.put("download.default\_directory", downloadFilepath);
* options.setExperimentalOption("prefs",chromePrefs);

1. WebDriver driver = new ChromeDriver(options);

package dpack;

import java.io.File;

import java.util.HashMap;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.testng.Assert;

import org.testng.annotations.Test;

import io.github.bonigarcia.wdm.WebDriverManager;

public class Demo {

@Test

public void downloadFile() throws InterruptedException {

ChromeOptions options = new ChromeOptions();

String downloadPath = System.getProperty("user.dir");

HashMap<String, Object> chromePrefs = new HashMap<String, Object>();

chromePrefs.put("profile.default\_content\_settings.popups", 0);

chromePrefs.put("download.default\_directory", downloadPath);

options.setExperimentalOption("prefs",chromePrefs);

WebDriverManager.chromedriver().setup();

WebDriver driver = new ChromeDriver(options);

driver.get("http://omayo.blogspot.com/p/page7.html");

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

driver.findElement(By.linkText("ZIP file")).click();

Thread.sleep(5000);

String filePathUpdated = downloadPath+"\\DownloadDemo-master.zip";

File file = new File(filePathUpdated);

Assert.assertTrue(file.exists());

if(file.exists()) {

file.delete();

}

}

}

1. **How to run Selenium WebDriver tests from command line ?**

* Export the Java file having Selenium Automation Test as Runnable Jar file using Eclipse IDE
* Run from the Command line using java -jar filename.jar

1. **How to switch to frames in Selenium WebDriver?**

driver.switchTo().frame()

WebElement frame1 = driver.findElement(By.id(“ifr1”));

driver.switchTo().frame(frame1);

1. **How to connect to a database in Selenium?**

Connection con = DriverManager.getConnection(dbUrl,username,password);

// mysql connector dependency must be added in pom.xml

Connection con= DriverManager.getConnection(“jdbc:mysql://localhost:3306/dbname”,”root” ,”root”);

If(connection.isCLosed())

{

System.out.println(“We have not connected to the database”)

}

else

{

System.out.println(“We have successfully connected to the database”)

}

1. **How to resize browser window using Selenium WebDriver?**

* setSize() command
* driver.manage().window().setSize(d);

Dimension d = new Dimension(500,900);

driver.manage().window().setSize(d);

1. **How to scroll web page up and down using Selenium WebDriver?**

JavascriptExecutor jse = (JavascriptExecutor) driver;

jse.executeScript("window.scrollBy(0,300);");

jse.executeScript("window.scrollBy(0,-300);");

1. **How to perform right click (Context Click) action in Selenium WebDriver?**

* contextClick() command of Actions class

WebElement searchBox = driver.findElement(By.name(“search”));

Actions actions = new Actions(driver) ;

actions.contextClick(searchbox).build().perform() ;

1. **How to perform double click action in Selenium WebDriver?**

* doubleClick() command of Actions class

WebElement button = driver.findElement(By.name(“Double Click”))

Actions actions = new Actions(driver) ;

actions.doubleClick(button).build().perform() ;

1. **How to perform drag and drop action in Selenium WebDriver?**

* dragAndDrop() command of Actions class

WebElement source = driver.findElement(By.name(“box1”));

WebElement target = driver.findElement(By.name(“box2”));

Actions actions = new Actions(driver) ;

Actions.dragAndDrop (source,target).build().perform() ;

1. **How to highlight elements using Selenium WebDriver?**

* JavascriptExectuor
* jse.executeScript("arguments[0].setAttribute('style','background: yellow; border:2px

solid red')", searchBox);

WebElement searchBox = driver.findElement(By.name(“search”))

JavascriptExecutor jse = (JavascriptExecutor) driver;

Jse.executeScript(“arguments[0].setAttribute(‘style’,’background: yellow;border:2px solid red;’);”, searchBox);

1. **Have you used any cross browser testing tool to run Selenium Scripts on cloud?**

I have used SauceLabs to run our Selenium Scripts on Cloud

Write the below code in main method

1. WebDriver driver = new RemoteWebDriver(new java.net.URL(),caps);

2. Search for 'SauceLabs Platform Configurator' and get the auto- generated capabilities code

3. Paste the auto-generated code

4. Search for 'saucelabs getting started with selenium website testing'

5. And copy paste three lines of code directly inside the class

6. Modify the username in the code

7. Go to Account > User Settings and copy the Access Key and paste into the

code

8. Write some sample selenium code

9. View the code here

10. Execute the code and watch under SauceLabs > Automated > Test Results

package dpack;

import java.net.MalformedURLException;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.openqa.selenium.remote.RemoteWebDriver;

public class Demo {

public static final String USERNAME = "arunmotoori";

public static final String ACCESS\_KEY = "0bb431c9-0900-46af-8e12-3c7e7d8f910b";

public static final String URL = "http://" + USERNAME + ":" + ACCESS\_KEY + "@ondemand.saucelabs.com:80/wd/hub";

public static void main(String[] args) throws MalformedURLException, InterruptedException {

DesiredCapabilities caps = DesiredCapabilities.chrome();

caps.setCapability("platform", "Windows 7");

caps.setCapability("version", "latest");

WebDriver driver = new RemoteWebDriver(new java.net.URL(URL),caps);

driver.get("http://tutorialsninja.com/demo/");

Thread.sleep(5000);

driver.close();

}

}

1. **What are the DesiredCapabitlies in Selenium WebDriver and their use?**
   * + - * DesiredCapabilities is a predefined class in Selenium WebDriver
         * We use DesiredCapabilities when we are running our tests on a remote node machine or remote cloud machine

Using DesiredCapabilities we can specify in which browser and OS the Test should run, when the tests are running in remote node machine or cloud machine.

import java.net.MalformedURLException;

import java.net.URL;

import org.openqa.selenium.By;

import org.openqa.selenium.Platform;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.remote.DesiredCapabilities;

import org.openqa.selenium.remote.RemoteWebDriver;

public class Demo {

public static void main(String[] args) throws MalformedURLException {

DesiredCapabilities dc = new DesiredCapabilities();

dc.setBrowserName("chrome");

dc.setPlatform(Platform.WINDOWS);

WebDriver driver = new RemoteWebDriver(new URL("http://192.168.0.106:4444/wd/hub"),dc);

driver.get("http://omayo.blogspot.com/");

driver.findElement(By.name("q")).sendKeys("Arun");

driver.findElement(By.xpath("//input[@class='gsc-search- button']")).click();

driver.close();

}

}

1. **What is Continuous Integration?**
   * + - * Continuous Integration is a development practice.
         * One of the most effective Agile Practices
         * In the Projects, where Continuous Integration development practice is followed, developers check-in their code frequently into the shared repository, several times a day.
         * Each and every check-in or integration, are generally verified by the automated tests.
2. **How to achieve database testing in Selenium?**

package dpack;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class Demo {

public static void main(String[] args) throws SQLException {

Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/demo","root","root");

if(connection.isClosed()) {

System.out.println("Not connected to database");

}else {

System.out.println("Successsly connected to database");

}

Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery("Select \* from employees where id=3;");

while(resultSet.next()) {

System.out.println(resultSet.getInt("id"));

System.out.println(resultSet.getString("name"));

System.out.println(resultSet.getString("place"));

System.out.println(resultSet.getInt("experience"));

}

}

}

1. **What is TestNG?**
   * + TestNG is a powerful unit testing framework.
     + The below are the advantages of using TestNG:
     + Controls the flow of test case execution.
     + Grouping the test cases in to separate groups is possible
     + Prioritizing the test cases is possible to prioritize which test needs to be executed

first and which next.

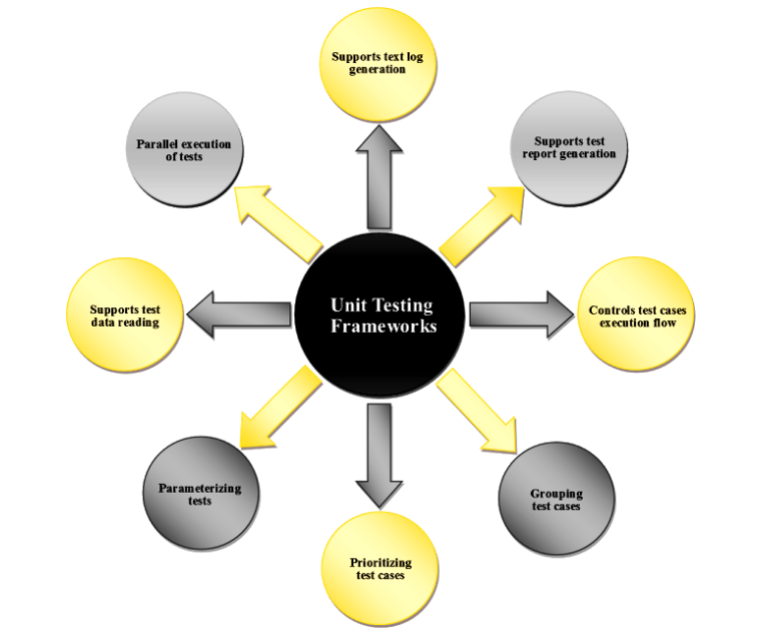
* + - Parameterizing the tests in such a way that the same tests can run multiple times

using the different sets of data

* + - Supports reading the data from the external sources like Excel files etc.
    - Parallel execution of tests to save time of test execution by executing multiple tests at the same time
    - Supports generating text logs to later find-out what are the different things that

happened while the tests were executing.

* + - Supports generating reports to find out the test results after tests execution.



1. **What are TestNG Annotations and name different annotations available in TestNG?**
   * + TestNG annotations are specified before methods in the Test Automation code and they are used to:
       - * Control the flow of execution.
         * Parameterization

Receive Data from a centralized location.

Data Driven Testing

* + - Different Annotations available in TestNG are:
* testng.org > Javadocs
* @Test - Represents the methods as Tests
* @BeforeMethod - Executes before every @Test annotated method
* @AfterMethod - Executes after every @Test annotated method
* @BeforeClass - Executes before executing the Class
* @AfterClass - Executes after executing the class
* @BeforeTest - Executes before <test> tags in TestNG.xml file
* @AfterTest - Executes after <test> tags in TestNG.xml file
* @BeforeSuite - Executes before <suite> tag in TestNG.xml file
* @AfterSuite - Executes after <suite>tag in TestNG.xml file
* @Parameters - For parameterizing the test and need to be provided before the test method which receives the data from the testng.xml file
* @DataProvider - Need to be provide before the method which supplies multiple sets of data as part of Data Driven Testing

1. **What is TestNG Assert and list out some common TestNG assertions?**

* TestNG Assertions
  + We perform testing to verify whether a particular test is passed or failed.
  + In the similar way, TestNG provides a predefined Class 'Assert' and its predefined methods assertEquals(), assertNotEquals(), assertTrue(), assertFalse() and fail() to verify whether @Test annotated methods are passed or failed.

**assertEquals()**

Assert.assertEquals(9,5) // Fail

Assert.assertEquals(9,5) // Pass

**assertTrue()**

Assert.assertTrue(9<5) // Fail

Assert.assertTrue(9>5) // Pass

**assertFalse()**

Assert.assertFalse(9<5) // Pass

Assert.assertFalse(9>5) // Fail

**Fail**

int age = 3; //this will fail but with age greater than 5 it will pass

if (age<=5)

{

Assert.fail(“Age should be greater than 5”)

}

1. **How to create and run TestNG.xml?**

Easiest way to create TestNG.xml file in the Eclipse IDE and run

Right click on the project, go to TestNG and click Convert to TestNG

1. **How to set test case priority in TestNG?**

Using priority attribute of TestNG, we can prioritize the test cases.

public class Demo

{

@Test(priority=2)

Public void login

{

System.out.println(“Executing Login Test”)

}

@Test(priority=1)

Public void register

{

System.out.println(“Executing Login Test”)

}

}

1. **What is parameterized testing in TestNG ?**
   * + - Parameterization in TestNG can be achieved in two ways using:
         * Parameters Annotation

Receives the data from a centralized location say TestNG.xml file

**Testing.xml**

<suite name=”Suite1”>

<parameter name=”URL” value =” <http://google.com>” />

<test name= “Test”>

<classes>

<class name=”dpack.One” /”>

<classes>

</test>

</suite>

**One.java file**

public class One

{

@Parameter({“URL”})

@Test

public void testOne(String url)

{

WebDriverManager.chromedriver().setup();

WebDriver driver = new ChromeDriver();

driver.get(url);

}

}

* + - * + DataProvider Annotation

Supplies multiple sets of data to a Test Method

public class Demo

{

@Test(dataProvider= “getData”)

public void loginTest(String user, String Pass)

{

System.out.println(“Username is” + user+ “and password is”+ pass);

}

@DataProvider

public Object[][] getData()

{

Object[][] data = new Object[3][2];

data[0][0] = “Adnan”;

data[0][1] =”1234”;

data[1][0] = “Abc”;

data[1][1] =”12345”;

}

}

1. **How to run a group of test cases using TestNG?**

Using **groups** predefined attribute of TestNG and by mentioning groups tags in TestNG.xml file.

**testing.xml**

<suite name=”Suite1”>

<test name= “Test”>

<groups>

<run>

<include name=”smoke” />

</run>

</groups>

<classes>

<class name=”dpack.One” /”>

<class name=”dpack.Two” /”>

<classes>

</test>

</suite>

**One.java**

public class One

{

@test(groups = {“smoke”})

Public void testOneA()

{

System.out.println(“Inside testOneA”)

}

}

// Same for class Two.java

1. **What is the use of Listeners annotation in TestNG?**

We use Listeners annotation ,when we are implementing Listeners at Class level.

**ListenerDemo.java**

public class ListenersDemo implements ITestListener

{ // for getting method to override in ITestListener interface, click on Source in eclipse and select Override/Implement Methods and select the required methods

@Override

public void onTestStart(ITestResult result)

{

System.out.println(result.getName()+ “Test started”);

}

@Override

public void onTestSuccess(ITestResult result)

{

System.out.println(result.getName()+ “Test is successfull”);

}

@Override

public void onTestFailure(ITestResult result)

{

System.out.println(result.getName()+ “Test has failed”);

}

}

**Demo.java**

@Listeners(ListenerDemo.class)

public class Demo

{

@Test

public void testOne(String url)

{

WebDriverManager.chromedriver().setup();

WebDriver driver = new ChromeDriver();

driver.get(“http://www.google.com”)

Assert.assertEquals(driver.getTitle(),”Google”)

}

}

//According to the status of the test, the particular listener will be called.

1. **How can we implement Data Driven Testing for Data Driven Framework using TestNG?**

We can achieve this by using @DataProvider annotation and dataProvider predefined annotation of TestNG.

public class Demo

{

@Test(dataProvider= “dsm”)

public void loginTest(String user, String Pass)

{

System.out.println(“Username is” + user+ “and password is”+ pass);

}

@DataProvider(name=”dsm”)

public Object[][] dataSupplierMethod()

{

Object[][] data = {{“Adnan”,”123”},{“ABC”,”1234”},{“DEF”,”456”}};

return data;

}

}

1. **Where you have applied Java OOPS (Object Oriented Programming) concepts in Automation Framework?**
   * + **Inheritance**

Initialize browser in Base class and instead of implementing in test classes we can directly call it.

* + - **Polymorphism**

In ExcelReader class, getCellData one method is having String colName and other is having int colNum.

public String getCellData(String sheetName, **String colName**, int rowNum)

public String getCellData(String sheetName, **int colNum**, int rowNum)

* + - **Encapsulation**

private WebElement chngPass;

this element can be accessed only in the class where it is defined.

if we want to access it outside the class we have to use a public method.

public boolean chngPassDisp()

{

Return chngPass.isDisplayed();

}

* + - **Abstraction**

enterEmail() is implemented in Login Page.java and it is being hidden in login.java

**Login.java**

public class Login{

@Test

public void login1(){

LoginPage loginPage = new LoginPage(driver);

loginPage.enterEmail(prop.getProperty(“ValidEmail”));

}

}

**LoginPage.java**

public void enterEmail(String emailAddress)

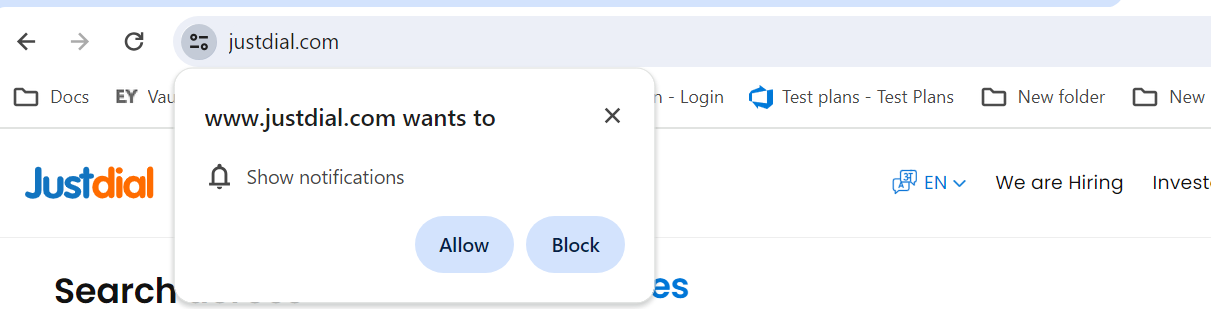
{

emailAddessTextField.sendKeys(emailAddress);

}

1. **How to handle Chrome Browser notifications in Selenium?**

Notification such as Chrome browser wants to know your location or wants to send notification.



public class Demo

{

public static void main(String[] args)

{

WebDriverManager.chromedriver().setup();

ChromeOptions options = new CHromeOptions();

Options.addArguments(“—disable-notifications”);

WebDriver driver = new ChromeDriver(options);

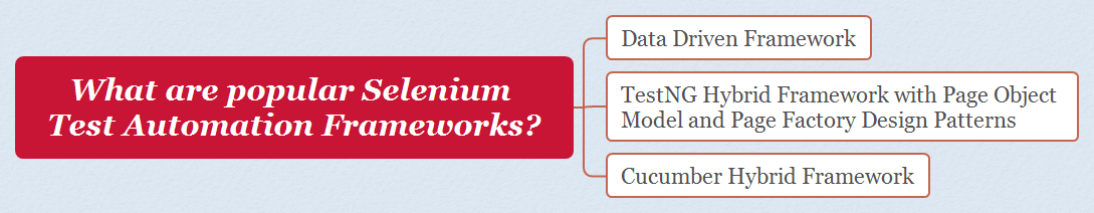
driver.manage().wiindow.maximize();

driver.get(“https://www.justdial.com/”)

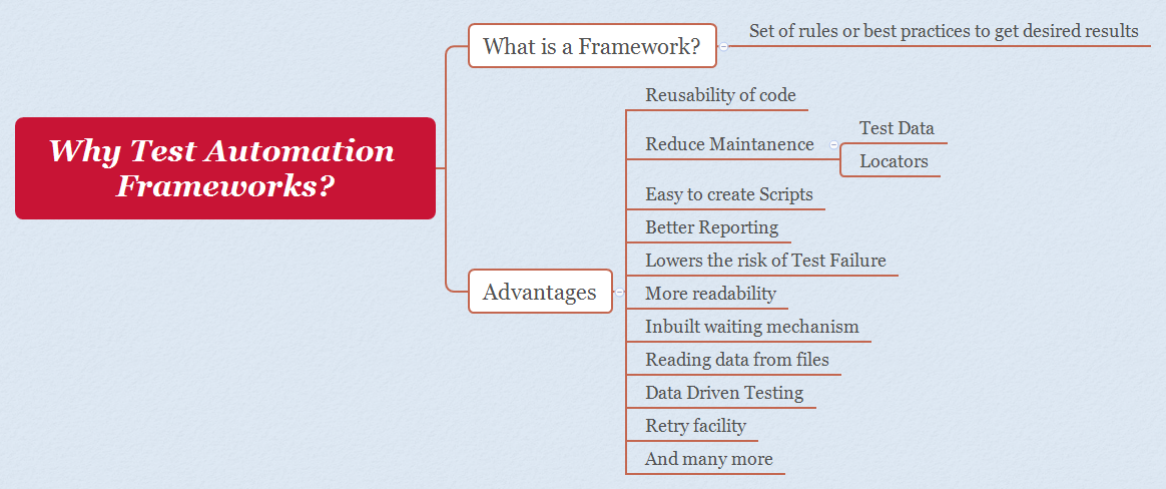
}

}

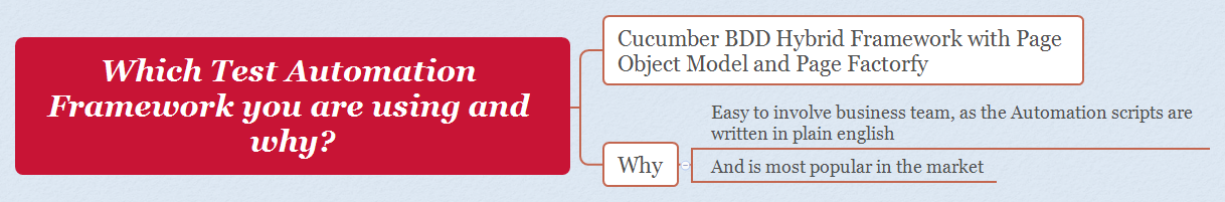
1. **What are popular Selenium Test Automation Frameworks?**



1. **Why do we have to create and use Test Automation Frameworks?**

****

1. **Which Test Automation Framework you are using and why?**



1. **Explain the Framework you are using in your Project?**

**Feature files:**

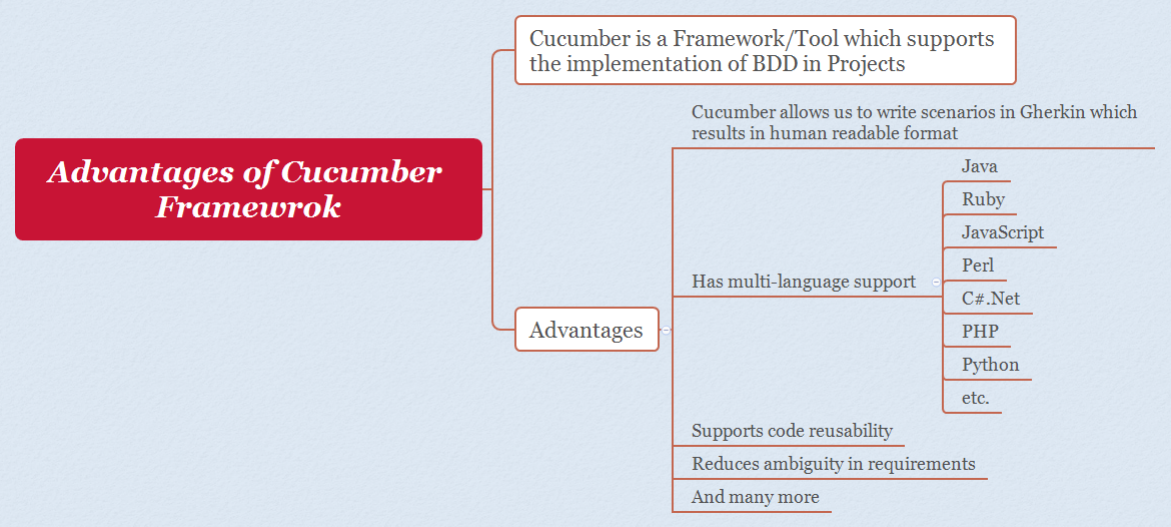
Scenarios are written here

1. **What is Cucumber?**



Cucumber eclipse ide plugin must be installed and dependency has to be added under dependencies in pom.xml file.

1. **What are the advantages of using Cucumber?**

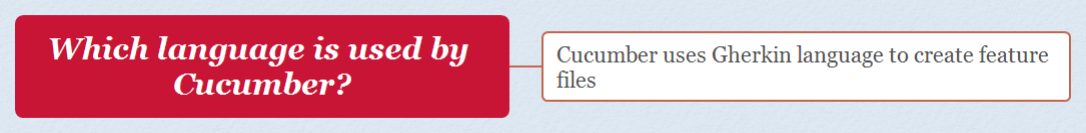
****

1. **What are the two files required for executing Cucumber Test Scenarios?**

****

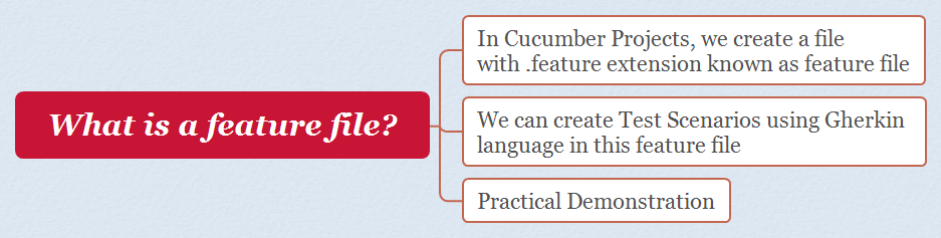
If feature file and step definition file are under different package , runner file is required to connect feature files with step definition files.

1. **Which language is used by Cucumber?**

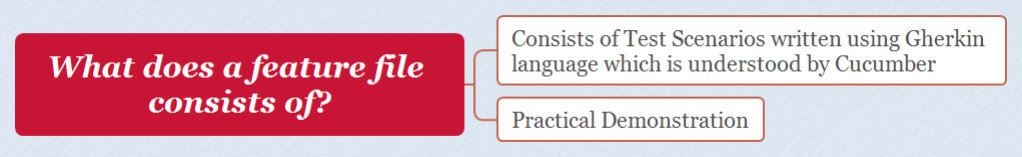
****

Give, when , then, and ,but etc

1. **What is meant by a feature file?**

****

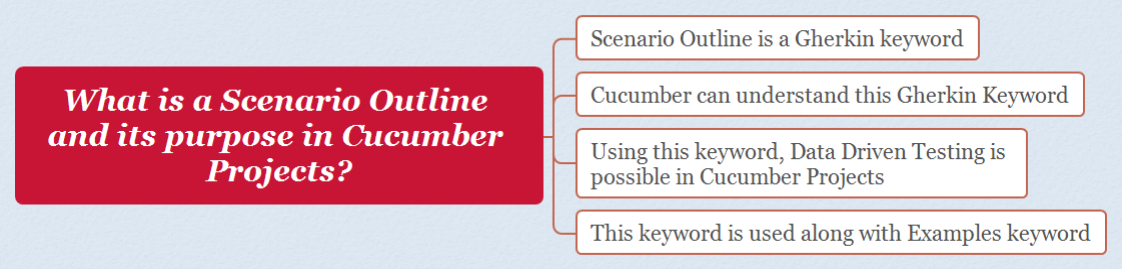
1. **What does a feature file consists of?**

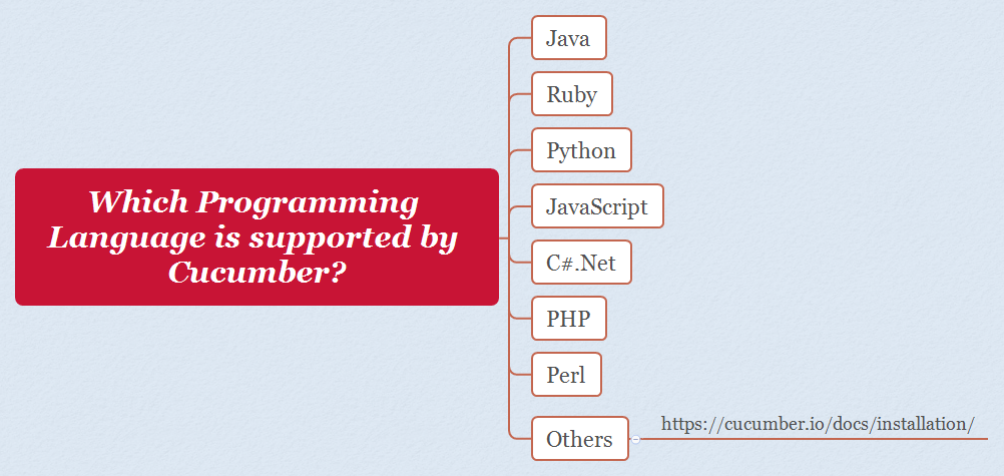
****

1. **What are various Gherkin keywords used in Cucumber Projects?**

****

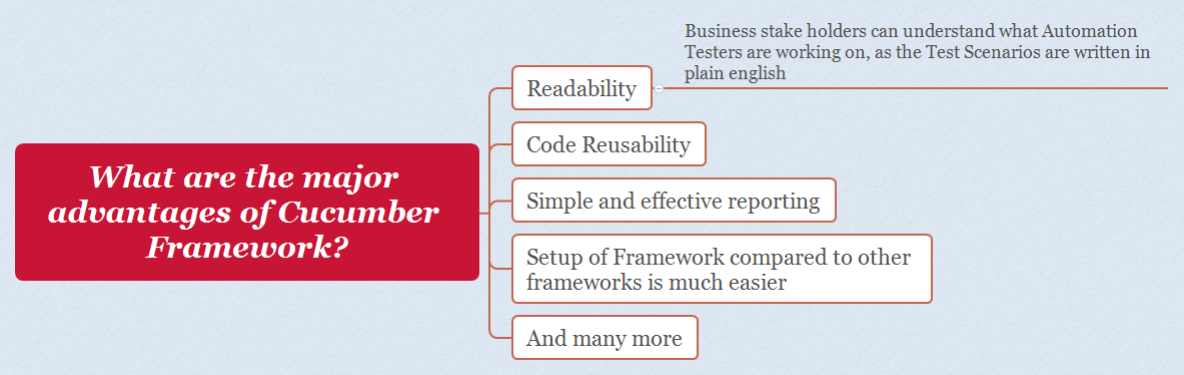
1. **What is a Scenario Outline in Cucumber and its purpose?**

****

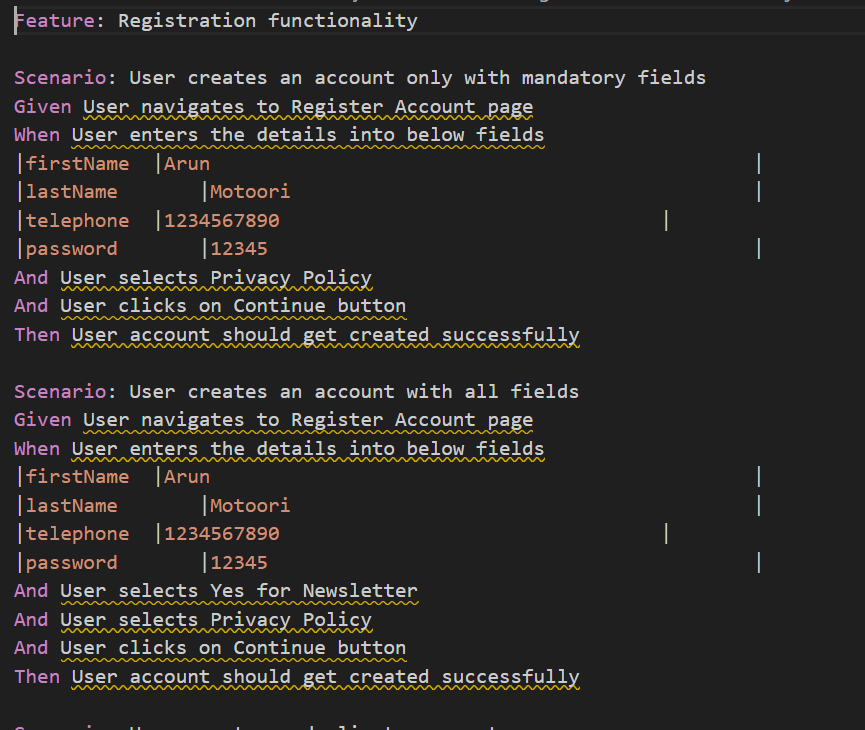
1. **Which progamming language is supported by Cucumber?** 
2. **What is the purpose of Step Definition file in Cucumber?**

****

1. **What are the major advantages of Cucumber Framework?**

****

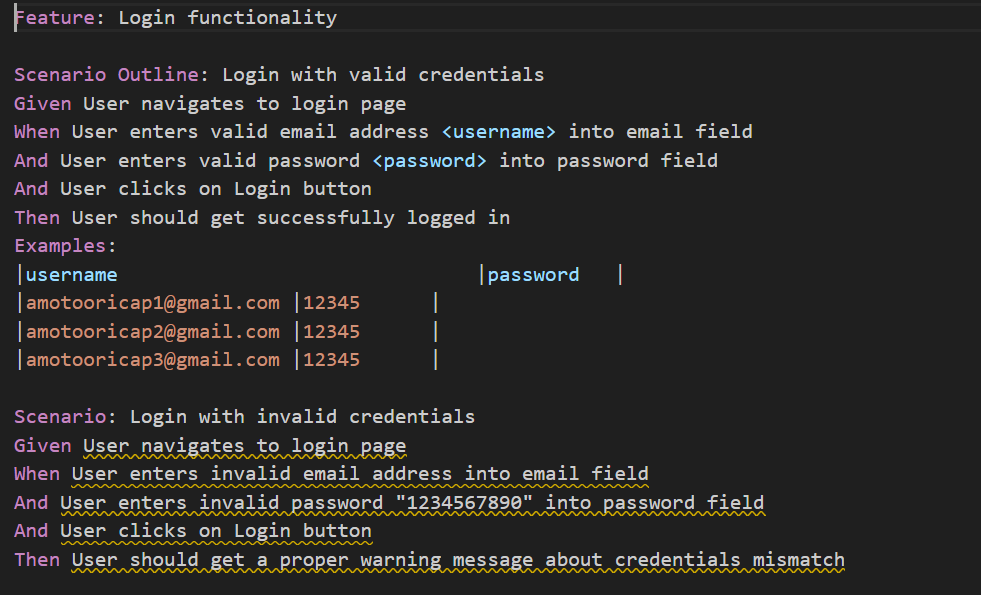
1. **Provide an example of Feature File in Cucumber Framework?**

****

1. **Provide an example of Scenario Outline in Cucumber framework?**

Scenario outline is generally used to implement data driven testing, when you want to

execute same test with different set of data.

****

1. **What is the purpose of Behaviour Driven Development (BDD) methodology in real world?**

****

1. **What is the limit for the maximum number of scenarios that can be included in the feature file?**

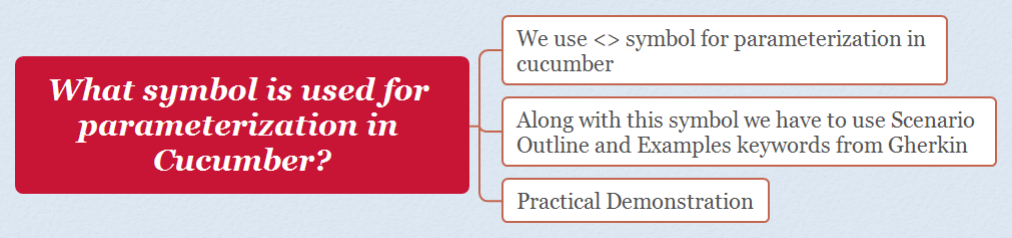
****

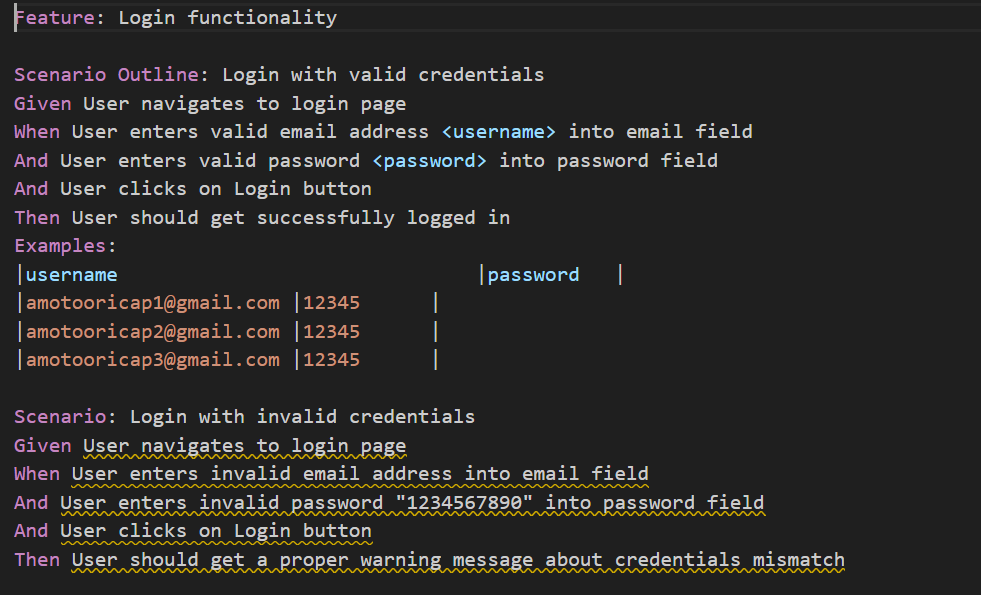
1. **What is the use of Background keyword in Cucumber?**

****

If first step is common in all scenarios, instead of writing it in all scenarios it can be put in Background.

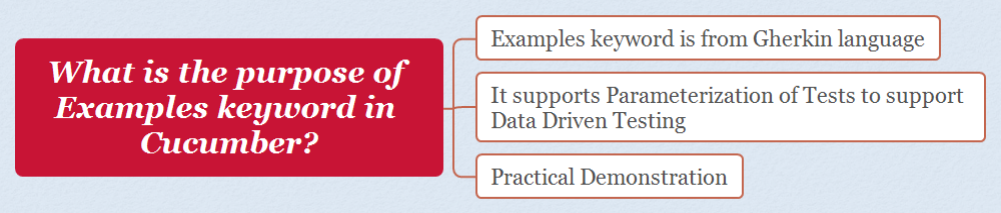
1. **What symbol is used for parameterization in Cucumber?**

****

****

Add Tidy Gherkin plugin in chrome to convert feature file into step definition.

1. **What is the purpose of Examples keyword in Cucumber?**

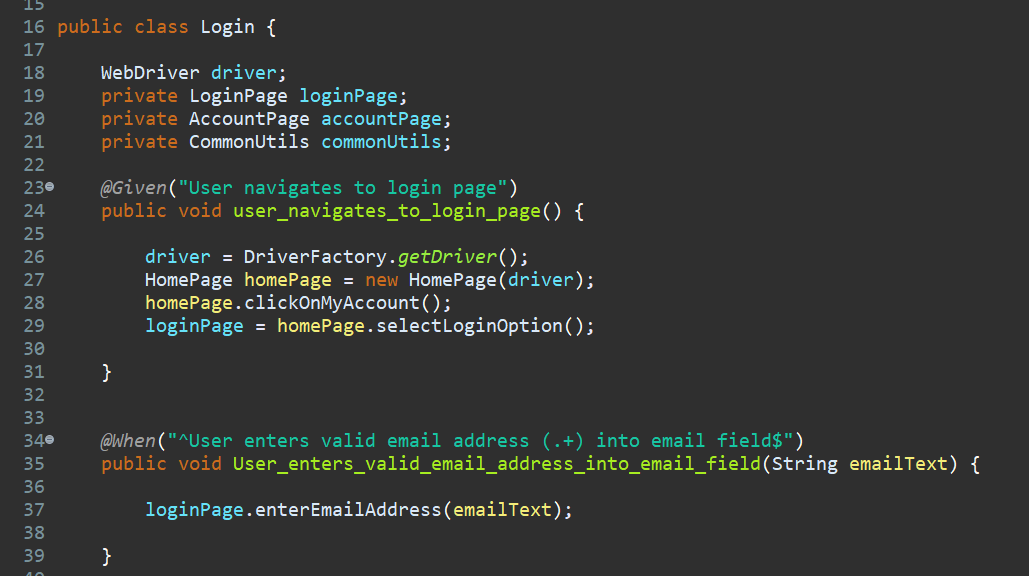
****

1. **What is the File extension of a Feature File?**

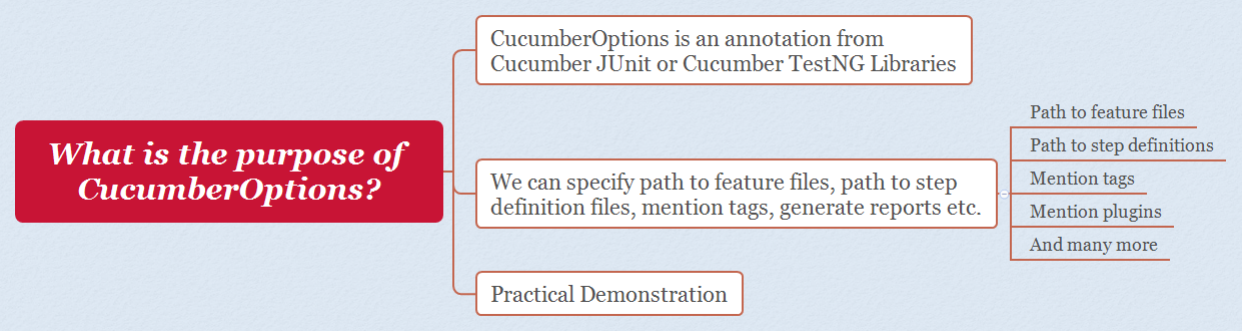
****

1. **Provide an example of step definition file in Cucumber**

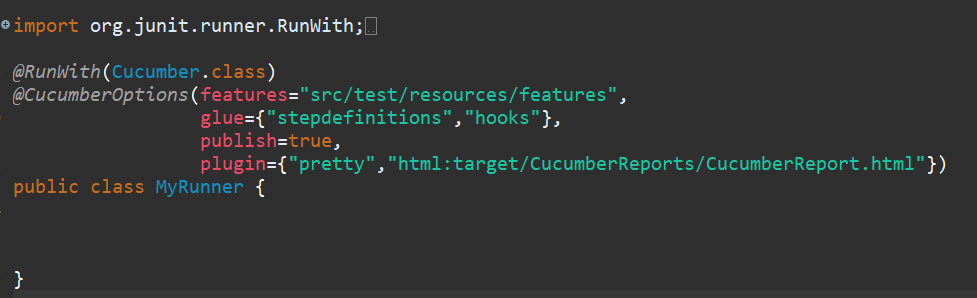
****

****

1. **What is the purpose of CucumberOptions ?**

****

Runner file to link feature file and step definition file.

****

1. **How cucumber can be integrated with Selenium WebDriver?**

****

Go to mvnrepository.com and search cucumber

Copy the dependency of below mentioned and paste it in pom.xml:

**1.** [**Cucumber JVM: Java**](https://mvnrepository.com/artifact/io.cucumber/cucumber-java)

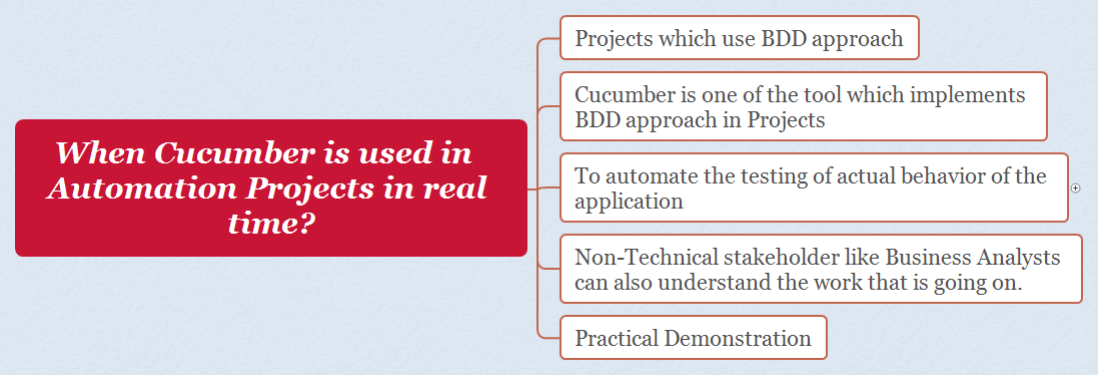
**2.** [**Cucumber JVM: JUnit 4**](https://mvnrepository.com/artifact/io.cucumber/cucumber-junit)

**3.** [**Cucumber JVM: Core**](https://mvnrepository.com/artifact/io.cucumber/cucumber-core)

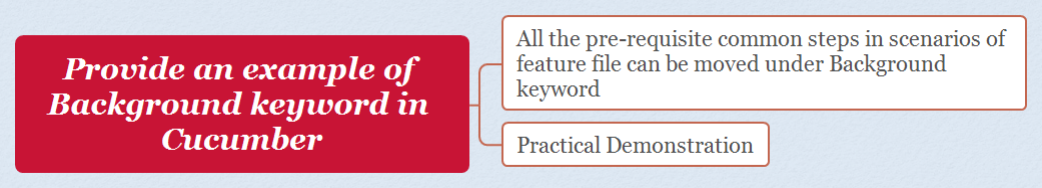
**(version should be same)**

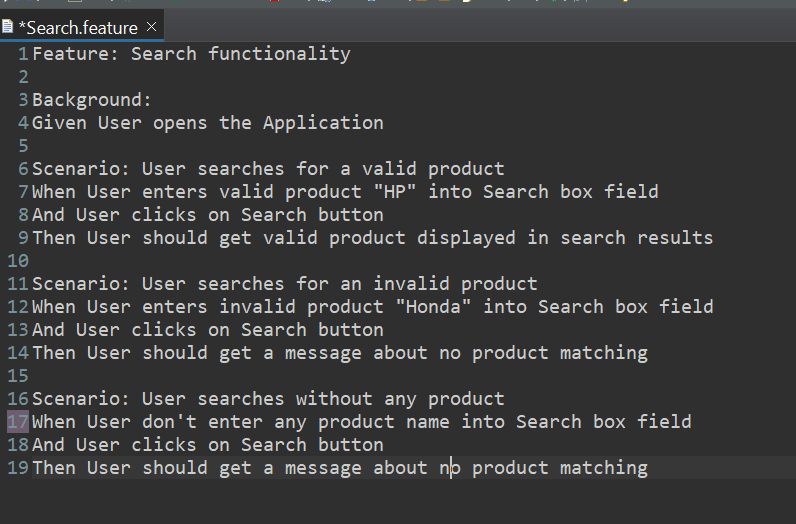
Run the feature file without the step definition and it will give you the step definition in console.

1. **When Cucumber is used in Automation Projects in real time?**

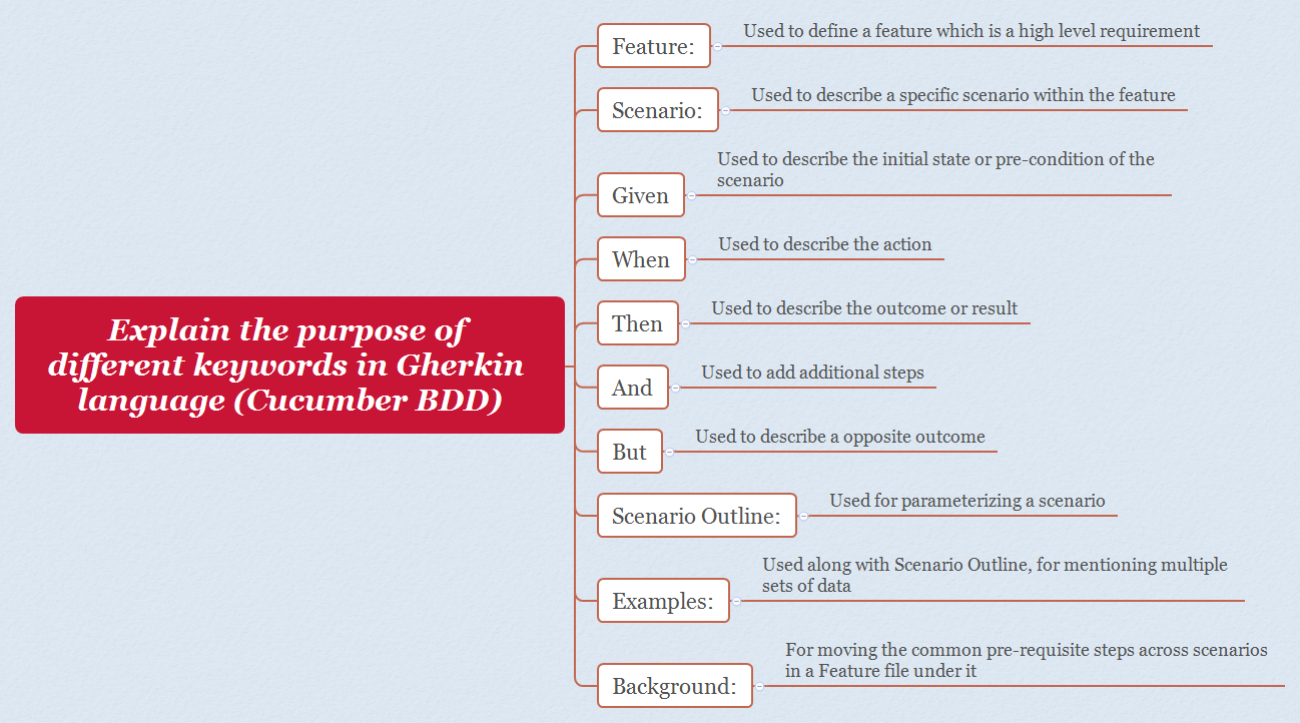
****

1. **Provide an example of Background keyword in Cucumber**

****

****

1. **Explain the purpose of different keywords in Gherkin language - Cucumber BDD**

****