@@@@@@@@@@@@@@@@@  **Practice URLs** @@@@@@@@@@@@@@@@@@@@@@@@@@

Jquery DropDown - https://www.jqueryscript.net/demo/Drop-Down-Combo-Tree/

AutoCompletion Dropdown - <https://www.twoplugs.com>

Broken Link validation - - <http://www.deadlinkcity.com>

DatePicker - <https://www.redbus.in/>

Mouse Action - <https://swisnl.github.io/jQuery-contextMenu/demo.html>

Keyboard Actions - <https://the-internet.herokuapp.com/key_presses>

Multiple Keyboard Action - <https://text-compare.com/>

Common Website - <https://demo.nopcommerce.com/>

BankMini Project - <https://demo.nopcommerce.com>

Missing Certificate - [cacert.com](http://cacert.com)

Practice Website - https://www.hyrtutorials.com/p/basic-…

Broken Image - https://www.theworldsworstwebsiteever.com/

@@@@@@@@@@@@@@@@@ **Shortcuts** @@@@@@@@@@@@@@@@@@@@@@@@@@@@

Console tab in Developer tool while searching for element:

Control + L = to clear the console

$$(“CSS Selector syntax”) = to search webElement using CSS Selector

@@@@@@@@@@@@@@@@@ **Selenium Intro & History** @@@@@@@@@@@@@@@@@@@@@@

**Selenium Introduction and History Summary**

* Its a collection of Interfaces which integrates several third party services, project management framework and other systems like apache poi, Log4j and others and has become one of most powerful test suite.
* Developed in 2004 by Jason Huggins while working in Company called ThoughtWorks for one of his internal project
* Selenium was initially started with Java Script
* Selenium now supports multiple languages and almost every browser available in current market.
* Selenium components are - Selenium IDE, RC, GRID and WebDriver known as Selenium Suite
* After Selenium 2 Selenium WebDriver got introduced. Selenium RC got deprecated after Selenium 3 onwards.
* Selenium RC was supported till 2.44 Release release after that it got depreciated.
* Selenium 3 removed the native support of Firefox browser also the driver communication using Jason wire protocol started happening with all browsers drivers which got provide by browser vendor like chrome by google, edge by Microsoft and so on.
* Selenium 3 release were from 3.0.0 to 3.141.59.
* Selenium 4 got introduced on 13th Oct, 2021 with W3C protocol

**What’s new in Selenium 4?**

* Better window Management - to open any link in new window or new tab
* Better documentation
* Deprecation of Desired Capabilities (Now we work through Browser Capabilities)
* Introduction of w3c protocol

**Advantages and Disadvantages of Selenium?**

* Advantages:

a. Supports multiple language

b. Works on most of the OS

c. Supports almost every browser

d. Fast and open source

* Dis-advantages:

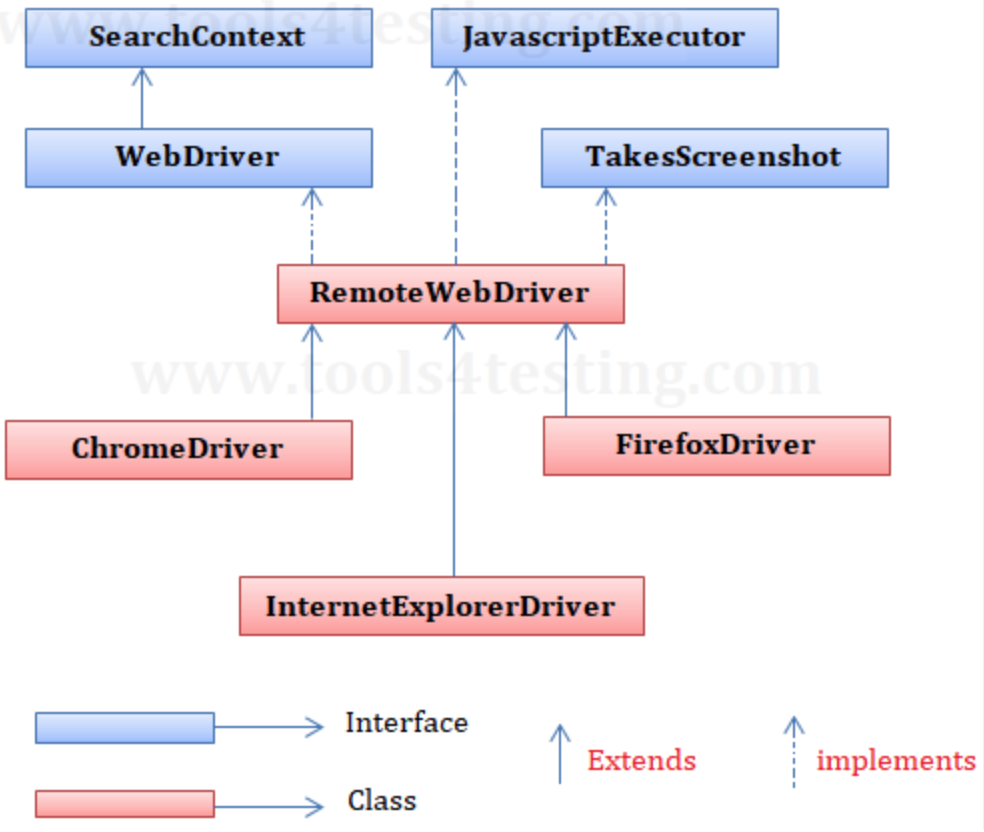
a. Doesn’t support windows apps

b. Doesn’t have dedicated Report Generation

c. Doesn’t support Excel file

c. Doesn’t support image validation

**Architecture of Selenium & Comparison between Selenium 3 and 4 version?**



* Selenium 3 Architecture

Client Library

Encod/Decod

Driver

Browser

Chrome, Safari

Jason Wire Prot

Chrome, Safari et

Java, Python, C#

* Selenium 4 Architecture

W3C

Client Library

Encod/Decod

Browser

Chrome, Safari et

Driver

Java, Python, C#

Chrome, Safari

@@@@@@@@@@@@@@@@@ **WebDriver & WebElements** @@@@@@@@@@@@@@@@@@

**What is Selenium WebDriver?**

* Selenium WebDriver officially came into picture in 2011 along with Selenium 2 version which was developed by a lead Developer Simon Steward. He along with Jason Huggins and couple of Google chrome developers, started working on in around 2006 or 2007 but finally came in 2011.
* When WebDriver got introduced it had native support for Firefox browser and for other browsers Selenium communicated via Jason Wire Protocol for encoding and decoding.
* Selenium Webdriver is a java interface which contains various abstract method and variables. This WebDriver interface is implemented by multiple classes like ChromeDriver class, Firefox driver class etc.
* WebDriver can also be considered as an Api (Application programming Interface). It contains various methods and classes by which communication happens between client and server.

**Question 42: List of Interface in Selenium?**

* Here is the list of Interface used in Selenium:
  + WebDriver
  + TakesScreenshot
  + JavascriptExecutor
  + Action
  + Alert
  + ExpectedConditions
  + SearchContext
  + Navigation
  + OutputType
  + WebElement
  + TargetLocator
  + Options
  + Timeouts

**Question 44: Methods of WebDriver interface:**

1. close()

2. get()

3. getTitle()

4. getPageSource()

5. getCurrentUrl()

6. getWindowHandle()

7. getWindowHandles()

8. manage()

9. navigate()

10. quit()

11. switchTo() Methods of

**Question 44: Methods of WebElement interface:**

* Clear()
* click()
* sendkeys()
* getAttribute()
* getCSSValue()
* getLocation()
* getsize()
* getTagName()
* isDisplayed()
* isEnabled()
* isSelected()
* Submit()
* getText

**Interface used in Selenium and their methods:**

- Here is the commonly used interface and their methods

* + **TakesScreenshot** - getscreenshotAs(arg)
  + **JavascriptExecutor** - executeScript(“window.scrollBy(0,1000)”)
  + **Action** - context click, double click, click
  + **Alert** - accept(), dismiss()
  + **ExpectedConditions** - elementtobeclickable(), elementtobedisplayed() etc.
  + **Navigation** - Need to find out
  + **OutputType** - This is used under TakesScreenShot Interface as OutputType.FILE
  + **TargetLocator** - Used to locate given frame or window
  + **Options** - Used to manage stuff that we do in Browser, getCookies(), addCookie()
  + **Timeouts** - Used for implicit or explicit wait

**Desired Capabilities : What is desired Capabilities in Selenium?**

* When we launch any browser, Selenium opens a dummy browser which won’t have your manual saved plugins or favourites (when you launch url manually). To perform these kinds of manual changes, WebDriver provides WebDriverCapabilities class which have inbuilt functions to handle this.
* With newer Selenium Selenium versions DesiredCapabilities have been deprecated. We have dedicated driver level like ChromeOption, GeekoOption class which has all DesiredCapabilities features as well couple of new features.
* Syntax -

ChromeOptions option = new ChromeOptions();

option.addAgruments(“- - start - - maximized “”); - to open the page in maximize mode

option.setAcceptInsecureCerts(true); - to open any website with expired certificate

WebDriver driver = new ChromeDriver(option); - initialise ChromeDriver now

**XPath : How it works and different ways to locate the object**

* Xpath works on DOM (Document Object Model). DOM gets created once we launch any website and based on different tags and attributes, we use xpath to identify the correct tag name and attributes and attribute’s value to identify the correct webElement. We have different xpath ways to find the these values: Here are couple of Relative xpath syntax:
* //input[@id = “rewrewrew”]
* //img[@id = “rewrewrew” and @name = “ewer”]
* //a[text() = “login”]
* contains - //input[contains(@class, “test2323”)] - it will search with partial value or complete val.
* Starts-with - //button[starts-with(@href, “https://www.”) - it will require starting attribute value
  + Normalize-space() - //input[normalize-space(text(), “First Name”)] - This is used when text has spaces in the beginning and at the end. Normalise-space removes those spaces and finds the element.

**Xpath Axes : What is Xpath Axes, Ancestor, parent, following sibling, child etc.**

* Axes means relationship from the current element to the target element. This is used when we try to find out any webElement but there is no unique attribute or value. In that case we look for relationship of this webElement with its axes to find out the element. There are different types or methods we have for Axes but we can categories it based on below relationships:
* Ancestor - Parent and Grand Parents
* Parent - Parent
* Sibling - Sibling tag under same parent
* Child - Child tag
* Descendant - Child and Grand Children
* Following - This will return all the following elements/tags without any relations
* Preceding - Preceding - This will return all the preceding (downwards tags) without any relations
* Ancestor
  + ancestor - //ul[@id = 'nav1']/ancestor::div
  + ancestor-or-self - //ul[@id = 'nav1']/ancestor-or-self::div
* Descendant
  + descendant - //ul[@id = 'nav1']/descendant::li
  + descendant-or-self - //ul[@id = ‘nav1’]/descendant-or-self::li
* Sibling
  + Preceding sibling - //div[@class = ‘test2’]/proceding-sibling::td/child::input
  + Following sibling - //ul[@id = 'nav1']/following-sibling::select/child::option
* Parent - //ul[@id = ‘nav1']/parent::div\
* Child- //ul[@id = ‘nav1’]/child::div\
* Preceding - //ul[@id = ‘nav1']/following::div\
* Following - //ul[@id = ‘nav1']/preceding::div\

**CSS Selection - How it works and different ways to locate the object**

* CSS selector is one of effective way to find the webElement where we don’t have commons tags or unique identification available.
* There are 4 types of CSS Selection we use:
  + CSS Simple Selector - Used to find elements based on easy attributes like id, class
  + CSS Pseudo - Class Selector - Used to find elements based on index, location or position
  + CSS Combinators Selector - Used for defining the relationship between element and the HTML webpage
  + CSS Attribute Selector - Used to find elements based on attribute and attribute’s value
* CSS Simple Selector
  + Css Element Selector - By.cssSelector(“button”);
  + Css ID Selector - By.cssSelector(“#test12”); - # for ID
  + Css Class Selector - By.cssSelector(“.test123”); - . For Class
  + Css Universal Selector - By.cssSelector(“\*”); - Use Findelements
* CSS Attribute Selector
  + AttributeName - input[placeholder] - placeholder is a an attribute name
  + AttributeValue - input[placeholder = ‘test123’]
  + AttributeValue - Partial text - whole word - input[placeholder~=‘question’]
  + AttributeValue - Partial text - text - input[placeholder~=‘ques’]
  + AttributeValue - Starts with - whole word - input[class|=‘my’]
  + AttributeValue - Starts with - text - input[class^=‘my’]
  + AttributeValue - Ends with - text - input[class$=‘test’]
* CSS Combinators Selector -
* There are 4 types of relationships or types under CSS Combinator Selector:
  + Descendant Selector - Child and Grand Child - .container input[id = ‘fname'] - space is used
  + Child Selector - .container >div>input[id = ‘fname’] - ‘>’ is used for child
  + Adjacent Sibling Selector - Just immediate tag after 1st one - .container + input[id = ‘fname’] - Use + symbol for adjacent sibling
  + General Sibling Selector - After the 1st tag, all tags are sibling under same parent - .container ~ input[id = ‘fname’]

**Xpath vs CSS : Difference between xpath vs CSS Selector and which is better?**

* xPath can travel backward as well as forward using ancestor and descendant whereas CSS Selector only travel forward. We can differentiate both on below criteria :
  + Performance - xPath is approx. 10-20% faster than CSS Selector
  + User-friendliness - CSS Selector is little more user friendly than xpath
  + Ease of webElement Find - xPath is more effective to find any element as xpath can travel back and forward but css selector only travel forward.

**JavaScript : How to use JavaScript in WebDriver?**

* JavaScript is used for couple of websites where elements are not getting handled by WebDriver.
* To use JavaScript we need to use JavascriptExecutor Interface and assign driver by typecasting it.
* We can inspect any webElement and go to Console in dev mode/ Inspect mode - DOM page and write the script in console and execute the same using js.executeScript(“java script”);
* Here is the complete syntax:

JavascriptExeuctor js = (JavascriptExeuctor)driver;

js.executeScript(“argument[0].document.getElementByID(“test”).click”); - Finding element to

**JS Executor : How to use JavaScript executor to find any webElement and perform Click or send value?**

* We can find webElement using JavaScript as well
* To do so we need to follow JavaScript syntax to perform. Everything in Javascript is document so we need to follow syntax for object/webElement Identification:
  + document.getElementByID(“test”)
  + document.getElementByTagName(“test”)

Syntax:

Javascript code :

1. To find element using id, class or name:

document.getElementByID(“test”)

2. To find element via xPath, we use evaluate

document.evaluate(“input[@id=‘firstName'],document,null,XPathResult.FIRST\_ORDERED\_NODE\_TYPE,null)

JavascriptExeuctor js = (JavascriptExeuctor)driver;

js.executeScript(“argument[0].document.getElementByID(“test”).click”); - In this code, agrument[0] is used when we use WebDriver to find any element and perform and any action. Else we directly provide the complete syntax of JS.

**PageLoad Time : What is PageLoad and why we use it?**

* Selenium by default waits for 300 seconds till it gives any exception for page is not loaded correctly.
* We can change it using PageLoad method to desired time using similar syntax like implicit wait:
  + driver.manage().timeout().pageLoad(DurationofSeconds(10);
* This works similar to implicit wait like you have to define only once during webDriver initialisation.

@@@@@@@@@@@@@@@@@ **WebDriver Methods** @@@@@@@@@@@@@@@@@@@@@@@

**Launching Browser : How to Launch url, website, capture title of the page, url and page source of the page?**

* Using following web driver commands:

a. Launch url: 1. System.setProperty(“webdriver.chrome.driver”, “location”) 2. driver= new ChromeDriver();

b. WebDriverManager is also used to load the browser

c. Launch website: driver.get(“url”);

d. Get Title = driver.gettitle();

e. Get Page Source = driver.getpagesource();

**Element Validation : How to check is element is displayed, enabled or selected?**

* By using below commands:

a. isDisplayed();

b. isEnabled();

c. isSelected();

**Navigate.To : What is the difference between get and** [navigate.to](http://navigate.to) **methods?**

* Both are same as navigate.To() method internally calls get() method it-self. So we can say that both are synonyms except the writing
* Both these methods are used to open a browser
* There is no difference between both of these, as you see the library file it actually mentioned that [navigate.to](http://navigate.to)() actually call get() commands only
* We can say both are synonyms
* So in short we can say that only difference is parameters

**Find Element vs Elements : What is the difference between between findElement & findElements?**

* findElement:
  + Return only single element
  + Can be assigned in WebElement variable
  + In case no such element found throws exception
* findElements:”
  + Return more than 1 element
  + Can be assigned in List<WebElement>
  + No error in case no element is found

**GetAttributes vs GetText : Difference between getattribute() vs gettext()**

* geAttribute(“value” or “id” etc.) will capture the attribute inside the tag.
* getText() - for some webelement, you may see some text inside the html tag. This will return the same text. If no text is present, no value will be returned.

**DropDown Filed : How to handle dropdown values**

* By using Select Class - Syntax:

Select sl = new Select(dropdown webelement);

sl.selectbyvisibletext(“value”);

or sl.selectbyIndex(3);

or sl.selectbyvalue(“4”);

**Bootstrap Dropdown : How to handle Bootstrap Dropdown value?**

* BootStrap is a CSS framework which is used to develop fields. These bootstrap fields looks different but internal tag is different. Field will appear as button but tag may be link or something else.
* To handle this we need to locate the field and capture the webelement using xpath
* If it’s a dropdown we need to capture all dropdown values (using the correct tag and capture all list or dropdown values.
* Capture under List<WebElement> variable name and using for loop read all the values and select the correct one.

**JQuery : How to handle JQuery dropdown**

* Same approach as Bootstrap Dropdown values

**Auto Suggestion : How to handle Auto Suggestion in dropdown**

* Need to find out the common tag which contains all auto suggestions. Like use xpath taking class name under ul/li tag etc. Then capture all inside List<WebElement> allsug = xpath
* Use for loop to get each of this value and get text and use if condition to check if value match and get it clicked.

Sample syntax:

driver.get(“<https://www.google.in>");

driver.findElement(By.Name(“q”).sendkeys(“Selenium”);

List<WebElement> allautosug = driver.findElements(By.Xpath(“//div/ul/li[@class=“abc”]”);

for(WebElement s: allautosug)

{

s.getText(); - you can get it printed also if want to see all auto suggestions

if(s.getText().equals(“Selenium download”)

{

s.click;

}

}

**WebTable : How to handle WebTable?**

* The most important point while handling webtable is to find out xpath of that element which highlights all element of that table. Capture all the webelements into List<WebElement> all field and then extract it using for loop. We can use nesting loop(for row and column) to capture all the row and column values. Using .getText() method to get the value.

**Mouse Actions : How to use Mouse operation like Right, left, double click etc?**

* To use Mouse actions we need to use Actions Object
* Actions act = new Actions(driver)
* We need to find the element where we need to perform the action and pass it to action method as below:
  + act.contextClick(xpath).perform();
* For drag and drop, we need to find xpath for source and target element and pass it under act.dragAndDrop(src, tar).perform();
* we have plenty of actions which we can perform in the same way

**Keyboard Actions : How to handle keyboard Actions.**

* We follow the same approach and Actions Object and its classes
  + We need to use below script

- Actions act = new Actions(driver);

- act.sendKeys(Keys.Enter).perform();

**Frames : How to handle frames and its webelement interaction?**

* We can use switchTo().frame(provide xpath of iframe/frame)
* Then we can perform normally
* 1 thing to remember in case of more than 1 frame, after completing transaction in 1 frame, we need to come back to default frame the move to 2nd frame.

**Mouse Hover : How to handle mouse hover?**

* Mouse Hover also belongs to Mouse Actions methods.
* Use Actions act = new Actions(driver) object and its methods.
* For Mouse hover, find xpath of menu as well as sub menu(which shows as hover) and store these into 2 Webelement variables.
* Then Use act.moveToElement(menu).moveToElement(submenu).click or perform();

**Slide Bars : How to handle Slider or sliding bar (range)**

* First we need to find the xpath of slider left point
* Get/print out its location and size using getLocation() and getSize methods
* Sliding action comes under Mouse action so use Actions act = new Actions(driver)
* Use method dragAndDropBy(slider point, changed location - how much you want, endpoint) - Refer location size printout and use accordingly.

**ToolTip : How to handle Tooltip and capture the capture**

* Tooltip is nothing but any input box and when we keep our mouse over that, we get a message. - To find the tooltip box, simply find using xpath and use driver.getAttribute(“title”) method to capture tooltip message. Generally tooltip message is inside the same tooptip box tag.

**Link - New Tab : How to open link in another tab or multiple link in multiple windows?**

* To open link in new window, use ‘Keys.chrod() method and inside that send keyboard commands like control + tab and control + Enter. Using below syntax:
  + String val = Keys.chord(Keys.Control/Command, Keys.Enter/Return);
  + Capture the link element using xpath which you want to open in new window
  + And instead of sending click(), use sendkeys(val); send chord value.

**Link - New Window : How to open any url in new window or new tab?**

* In Selenium 4 onwards, we have a method as
* driver.switchTo().newWindow(WindowType.Tab) - To open a new tab
* driver.switchTo().newWindow(WindowType.Window) - To open a new window
* After that simply use driver.get(“url”) command to open that website

**Actions vs Actions : Difference Between Actions and Action**

* Actions - Its a Selenium Webdriver Class and has plenty of methods to perform for Keyboard and Mouse actions
* Action - Its an Interface from Selenium WebDriver and if we want to store any action into a variable we can use Action Interface to do so. For example:

Action buttonclick = act.sendKeys(Keys.Enter); - We haven’t use perform method here we are only storing action into Action variable i.e. buttonclick here.

**Cookies : How to handle Cookies?**

Set<Cookie> cookies = driver.manage().getCookies(); //to get cookies details

for(Cookie co: cookies)

{

co.getCookies();

}

Cookie cookie = new Cookies(“test123”, “24234234”);

driver.manage().addCookie(cookie);

cookies = driver.manage().getCookies(); \* need to get cookies once again after adding or deleting

driver.manage().deleteCookie(cookie);

cookies = driver.manage().getCookies(); \* need to get cookies once again after adding or deleting

@@@@@@@@@@@@@@@@@ **Complex Methods** @@@@@@@@@@@@@@@@@@

**WebTable : How to handle WebTable with Pagination?**

* To handle webtable with pagination, we need to capture text at the bottom of the page where we generally gets the message like page 1 of 231 etc. and use get indexof() method to capture 231 value
* Use same approach of Question 14 (how to capture webTable to extract the value of each page row and column value.
* 1 important fact is generally when we click on any page no (initially it is link) but after clicking on the page, it turns to span tag and becomes active. So we need to handle this logic.
* To handle this logic refer SDET video part21.

**Date Picker : How to handle Date Picker in Calendar?**

* We need to have target and current date details
* We can define our desire date into 3 String variable for Day, Month and Year;
* Capture the month and year element on the calendar and get its text
* Use While(true) and inside it capture the element of month and year and spilt both and use if else condition to match the condition of your desired month and year.
* After that capture all the dates of the calendar, use for loop to extract all the values and then use if condition to match your desired date.

**Question 60: How to deal with Data/Calander WebElement on page?**

* Few important points while dealing with Calander:
* First we need to verify the user input means the date and format is correct or not
* To verify user input, we need to use SimpleDateFormat and convert/parse the user input date
* Need to find out the correct xpath using Ancestor or child or decendant xpath to get the correct data and reach to right date, month and year.

**Keyboard Actions - Multi Selection : How to handle multiple keyboard actions like Control C and Control V?**

* Use Action methods and perform set of actions step by step:
* Actions act = new Actions(driver);

For Selecting :

act.keyDown(Keys.Command);

act.sendKeys(“A”);

act.keyUp(Keys.Command);

act.perform();

To Copy:

act.keyDown(Keys.Command);

act.sendKeys(“C”);

act.keyUp(Keys.Command);

act.perform();

For next Tab:

act.sendKeys(Keys.Tab).perform();

To Paste:

act.keyDown(Keys.Command);

act.sendKeys(“V”);

act.keyUp(Keys.Command);

act.perform();

**Page Factory : What is this and how to use it?**

* Selenium WebDriver provides Page Factory Class to support Page Object Model approach. Using @FindBy Annotation, user can declare and define webElement using different locators. Also using initElements method, user can initialise webElements. The approach is create multiples classes and use Page Factory to locate all available web elements and declare it.

@@@@@@@@@@@@@@@@@ **Working with Different Documents** @@@@@@@@@@@@@@

**Apache POI : What is Apache POI? How to create an excel file ?**

* Apache poi is an API provided by Apache Foundation which is a collection of different java Libraries. This Library gives the facility to read, write and manipulate different Microsoft files like excel, word or ms-office
* We have 2 formats for ms office old version and new version so make sure to download Apache poi for old office (.xls, word) and for new format, use apache poi-poi or common one.
* Working with Excel:

- Create HSSFWorkbook or XSSFWorkbook component/class for excel, and follow below syntax to create rows:

HSSFWorkbook workbook = new HSSFWorkbook(); - to create an workbook

HSSFSheet sheet = workbook.createSheet(); - to create sheet

sheet.createRow(0); - it will create 1st row

sheet.getRow(0).createcell(0).setCellValue(“Santosh”);

sheet.getRow(0).createcell(0).setCellValue(“Ankita”);

sheet.createRow(1);

sheet.getRow(1).createcell(0).setCellValue(“Santosh”);

sheet.getRow(1).createcell(0).setCellValue(“Santosh”);

- Create a file location where we want to keep this excel file

File file = new File(“.//Screenshot/test.xls”); - For location

FileOutputStream fo = new FileOutputStream(file); - FileoutputStream to write the file

workbook.write(fo); - write

workbook.close(fo); - close

**Excel Read : How to read the value of cell and column value of an excel file:**

File file = new File(“.//Files/test.xls”). - To provide the location

FileInputStream fs = new FileInputStream(file); - Class to read the file

HSSFWorkbook workbook = new HSSFWorkbook(fs); - Class to work with excel

HSSFSheet sheet = workbook.getSheetAt(0); - To get 1st sheet

- - - - - To read only one cell value - - - - - - -

String row = sheet.getRow(0).getCell(0).getStringCellValue(); -

System.out.println(row);

- - - - - - - - To read entire excel sheet value including different data type - - - - -

int rowcount = sheet.getPhysicalNumberofRows(); - To get how many row available

for(int I = 0;i<=rowcount;i++)

{

HSSFRow row = sheet.getRow(i);

int cellcount = row.getPhysicaNumberofCells(); - To read how many cell available

for(int j=0;j<cellcount;j++)

{

HSSFCell cell = row.getCell(j);

String cellvalue = getcellValue(cell); - Call method and providing cell as parameter

system.out.print(“||” +cellvalue);

}system.out.println();

}

- - - create a method to handle different format of data - - - - -

Public static String getCellValue(HSSFCell cell)

{

switch(cell.getCellValue))

{

case NUMBERIC:

return String.valueOf(cell.getNumericCellValue());

case BOOLEAN:

return String.valueOf(cell.getBooleanCellValue());

case STRING:

return cell.getStringCellValue());

default:

return cell.getStringCellValue());

}

}

**MS Excel : How to work with Excel?**

* To work with excel, first we need to add Apache POI dependency in POM File
* Its actually Data Driven Framework Validation
* The approach to handle excel should be like this - workbooks - sheet - row - cell
* First we need to import FileInputStream

FileInputStream file = new FileInputStream(“Location of excel file”);

XSSFWorkbook workbook = new XSSWorkbook(file);

XSSFSheet sheet = workbook.getSheet(“Sheet1”);

int rowcount = sheet.getLastRowNum();

int columncount = sheet.getRow(0).getLastColumn();

***// If we want to capture and get it printed all the rows and column values;***

for(int I=0;i<rowcount;i++)

{

XSSFRow Current\_row = sheet.getRow(I);

for(j=0;j<coulumncount;j++)

{

String value = current\_row.getCell(j).toString();

System.out.print(“ “+ value+ “ “);

}

System.out.println();

}

***// If we want capture the values and send the same value to website fields.***

* To Capture all the rows and column value, use for loop and then save it to variables like:

for(int I=1;i<=rowcount;i++)

{

XSSFRow Current\_row = sheet.getRow(I);

String firstName = Current\_row.getCell(0).getStringCellValue();

String lastName = Current\_row.getCell(0).getStringCellValue();

String email = Current\_row.getCell(0).getStringCellValue();

String phone = Current\_row.getCell(0).getStringCellValue();

// Find the elements where we need to provide these details:

driver.findElement(By.xpath(“//dfsfsdf”).sendkeys(fristName);

driver.findElement(By.xpath(“//dfsfsdf”).sendkeys(lastName);

driver.findElement(By.xpath(“//dfsfsdf”).sendkeys(email);

driver.findElement(By.xpath(“//dfsfsdf”).sendkeys(phone);

**Text and CSV File : How to ready data from text and csv file?**

- In Selenium to work with csv file, we need a simple csv parser library called opencsv. Opencsv is open and can be found under Apache 2.0 dependency. OpenCSV contains methods to read and write csv files. To read text file, we use FileReader/FileWriter class and its methods like BufferedReader/BufferedWrite to work with it.

**PDF : How to read data from pdf file?**

* PDF Full form - portable document format
* PDFBox dependency need to added
* Syntax

File file = new File(“location of pdf file with pdf name”);

FileInputStream fi = new FileInputStream(file); - To read the raw data

PDDocument pdf = PDDocument.load(fi); - to load the file

System.out.print(pdf.getPages().getCount()); - this will print the no of pages of pdf file

PDFTextStripper stripper = new PDFTextStripper(); - To read the text

pdf.getStartPage(1);

pdg.getEndPage(2);

String pdftext = stripper.getText(pdf);

Syso(pdftext);

**Images : How to compare 2 images**

- We can capture both the image location and size in variable and compare it.

Screenshot logoimageScreenshot = new AShot().takesScreenshot(driver, logoImageElement);

ImageIO.write(logoImageScreenshot.getImage(), “png”, new File(“location”);

File file = new file(“location”);

**Question 32: What config change is required to show the screenshot in reports log?**

- We can use Listeners and try and catch block to get the screenshot in report log.

**Question 34: What is DOM?**

* DOM stand for Document Object Model. When we launch any webpage and url and we browse the page, browser creates a DOM in backend based on HTML code. XPATH works on the nodes which is maintained in this DOM structure.

**Question 35: Difference between Action and Actions**

* Actions is a class which is used to handle Keyboard and Mouse actions like context click, click etc.
* Action is an Interface -

**Question 36: How to handle dropdown filelds?**

* Dropdown fields are handled using Select Class. We can use getoptions() to capture all dropdown values and we can get the size.

**Questions 37: What is Assertion and tell the difference between Hard Assertion and Soft Assertion?**

* Assertion Class used to verify a condition using excepted and actual values. There are different assertion conditions available like:
  + Assert.assertequals()
  + Assert.assertnotequal()
  + Asserttrue()
  + Assertfalse()
  + Assertnull
  + Assertnotnull
* There are 2 types of Assertion - Hard Assertion and Soft Assertion
  + Hard Assertion - In this assertion if condition fails, the execution of next code will stop right there
    - Hard assertion uses - Assert class and its static class so we can directly use it like above assertion.
  + Soft Assertion - In this assertion even if condition fails, rest of the code still continues. We capture all the failure message and print it at the end.
    - Soft Assertion is not static so we need to define it like
      * SoftAssert softassert = new SoftAssert();
      * softassert.assertequals(actual result, expected result, “Message that we want to print”)
      * softassert.assertAll(); - At the end, we use to display all the results including failed one. If we don’t use this all Tcs will display as Pass even thought actually it failed during execution.

**Question 38: How to we execute only Failed TCs in the script?**

* In project folder we have 1 file named as TestOutput folder under which we will have failedTCs.xml file only

**Question 39: Define high level approach of Hybrid Framework**

* Once we have more than 1 framework/ approach used to automate the project, it is considered as Hybrid Framework.
* Data Driven - Storing all the data in a separate folder and using utilities classes to read that data. We create file utility and excel utility methods utilities package to read the data.
* Modular Driven - Module based creating TCs and storing it into separate package
* Method Driver - Creating and using several re-usable methods. Like creating Base class having common actions that we are performing on browser like launching the browser, closing it etc.
* TestNG configuration - TestingNG annotations are used to prioritise the Testcase in base class.
* Utility package - Under Utility package we will have webDriver Utility, JavaUtility, Excel Utility
  + WebDriver Utility - We have common WebDriver methods like Select, WebDriverWait, PagetoLoad, alertpopup accept
  + Java Utility - write methods like get random number, get date
  + FileUtility - write method to read the data from property file, read data from excel file,
* We also have POM Class - Page Object Model package where we are creating classes to capture init method we are using inside the constructor to initialise the web Element
* Resource Folder - Will have driver executable files
* Screenshot - Where we will have failed screenshot
* POM.xml file - to have all the required dependency and executing our framework using Jenkins
* Reports - HTML reports

**Question 40: How to run the TCs using Jenkin?**

* Need to configure our project and its location (GITHUB Location ) in Jenkings and run the tcs.
* When a build is deployed, Jenking will download the latest file from GITHUB and run the tcs.

**Question 41: What is GITHUB and how frequently Jenkings monitor the GITHUB ?**

- Github is a cloud based GIT repository hosting service.

* Generally we set Jenking configuration to monitor Github location on daily basis.

**Question 43: SearchContext Interface Methods**

* Here are couple of SearchContext Methods

1. findElement()

2. findElements()

@@@@@@@@@@@@@ **Framework** @@@@@@@@@@@@@@@@@@@@@@@@@@@@

**Framework High Level Intro and some highlights?**

* The objective of framework is to optimise code, re-usability code, folder structure, data management
* Data Driven, Hybrid, Page Object Models, Page Factory are not frameworks but these are approach or design pattern . We can Jnuit or TestNG as Framework as these have pre-built functionalities to design flow, design or suite design
* Packages under src/test/java : Base, Test Cases, Utilities, Ruffwork
* Folders under src/test/resources : Logs, Properties, TestData, Runners, Executables
* Files under Properties Folder -
  + RO.properties : to keep all locators
  + Config.properties : to keep all common details like browser name, url,

**Page Object Model : What is the approach for Page Object Model test automation framework?**

* In larger project where we have multiple pages and one page led to another one like
  + Login page
  + Home Page
  + Quote Page
  + Binder Page
  + Policy Page
  + Payment Page
  + Logout Page
* In Page Object Model means - we handle every page by capturing its webelement and applying methods.
* To do so, the approach is as below:
  + Create 1 Java Class for each page and capture each page all elements only, we don’t include any method or actions in this class
  + Create 1 common Java class and have all the actions and methods that we are going to apply on these elements.

**Framework Introduction : Create a project and explain how you will design the Framework**

* Need to create couple of Packages:
  + PageObjects Package - Under this package, we will create 1 class for each page like Login, Home, Cart, Payment etc.
    - For Everypage class, we need to find elements (@FindBy) and their action methods
  + TestCase Package - Under this Package, we will create 1 BaseClass which will have TestNG annotation based common TCs which will be used across all the TCs like
    - Initial Browser setup (System.setProperty) - @BeforeClass as TestNG annotation Close/Quit as teardown methods using @AfterClass TestNG Annotation
    - All the TCs - We will create TC script as per Test Case that we want as per project

**Framework Definition : Explain your framework**

* Try to explain with graphical diagram to make the explanation easy.
  + Start with Language which language you use for scripting - like Java or Python
  + Type of the framework - Data Driven, Module Driven or Hybrid
  + Show the folder structure that you are going to design and maintain in your framework
  + Base Class - Every Framework has a base class which will have WebDriver, Implicit Wait, Logger, Reports or any common data that we are going to use across the project. Generally all TCs will extend this Base Class to use and read Base Class methods.
  + Element Locators - Explain how you maintain elements in Framework. If you follow Page Object Model, we have 1 java class for each page and as page level, we capture all the web elements and have their associated methods to perform its intended actions like click for button, send value for text fields, selection for radio and checkbox etc.
  + Utilities/Function Class - Every Project has some functions. These functions are common function which can be accessed across projects In most of the projects we divide these functions into Generic and Project specific. Calling data from an excel or xml data sheet, Explicit wait, actions, Capture Screenshot, sending email, these are considered part of utilities or function. We call this as Utilities functions and keeps under utilities package.
  + Config - Properties Files - Details like browser url, screenshot path, login credentials can be placed or stored in Properties file
  + TestNG Annotation - What kind of annotation that we are using in the project like BeforeClass, AfterClass, Test, Parameters etc. Test NG framework and its annotations are used to perform parallel execution, grouping and assertion testing.
  + Parametrisation - Explain how you parametrise your testing using excel file
  + Error Screenshot - How we capture error screenshot and also failed screen shots will be added to Extend Report
  + Test Data - All test will kept inside Excel sheet or XML sheet under Test Data Folder - Apache POI is used to read the data of excel
  + Reports - How to generate reports and send as an email
  + Version Control - GITHub or SVN
  + Continuous Integration Tool - Jenkings or Hudson
* Properties Files

@@@@@@@@@@@@@@@@@ **TestNG Framework** @@@@@@@@@@@@@@@@@@

**Question 61: What is TestNG Framework?**

* TestNG stands for Test with Next Generation which is Annotation based Test Automation framework inspired by JUNIT but have additional features and more powerful.
* TestNG uses your existing client library code and covert into different Tcs and set the priority, sequence and other setup instructions.
* On execution, TestNG publishes a reports and shows no of Tcs passed and failed etc..
* TestNG also provides an XML file which we use to set different execution instruction
* TestNG provides Parallel execution, XML support to run the test
* TestNG annotation works as per alphabetic orders
* We can create TestNG xml by manually or simple click on project and convert to TestNG
* TestNG XML follows a standard format hierarchy as below:
  + Suite-Test-Class-Method

**Question 62: TestNG - Annotations - How to setup and use?**

* Here are list of all different types of Annotation:
  + @Override
  + @Test
  + @BeforeSuite
  + @AfterSuite
  + @BeforeClass
  + @AfterClass
  + @BeforeMethod
  + @AfterMethod
  + @BeforeTest
  + @AfterTest
  + @BeforeGroup
  + @AfterGroup
  + @Parameters
  + @Factory
  + @DataProvider
  + @FindBy
  + @Listeners
  + @Ignore

**TestNG Top Features:**

* TestNG provides 2 important thing to work - Annotation and XML
* Running TCs from XML is called TestNG Suite
* Inside TestNG @Test Annotation we can use priority=1,2,3 to set the priority of the method to execute in the same order where priority 1 is on top priority
* We can also use dependsonMethod for exam @Test(priority =1, dependsonMethod =“Login”)
* We can also use groups to group your TCs (@Test(groups = “Regression”) or @Test(groups ={“Regression”, “Smoke”})
* In case we don’t want execute any TC, we can use exclude instead of include under <Suite><groups><run><exclude name = “method name”>
* We can also create Master Suite TestNG xml file which will call TestNG Suite class which have TC level instructions.

**TestNG Dependency - How to setup and use?**

* In practical situation, we have scenario where Tcs are interdependent. This scenario can be achieved using Dependency in TestNG.
* Syntax = @Test(priority=1, dependsOnMethods=“method name”) - Depends on 1 method
* Syntax = @Test(priority=1, dependsOnMethods={“method1”, “method2”) - Depends on more
* If 1 method is failed and 2nd method is dependent on that, then this method will be skipped
* If dependent method failed is failed, and we still want to execute, we can use soft dependency.
* Soft Dependency Syntax = @Test(priority=1, dependsOnMethods=“method name”, alwaysRun= true)

**TestNG Suite - How to setup and use?**

* TestSuite is nothing but collections of methods. In TestNG this TestSuite is available in form of an XML file.
* We can create this xml file manually or click right click on project and covert to TestNG format and it will create an TestNG.xml file.

**TestNG Master Suite - How to setup and use?**

* Its quite easy, copy any existing TestNG xml or create a new one
* Remove all contents after suite (Like class, method, parameters etc)
* Simply add <suite-files><suite-file path = “location of other TestNG Xmas”/suite></suite-files>
* You can add multiple TestNG xml files and run it - That’s all

**TestNG Groups - How to setup and use?**

* Groups are used to categories your Tcs into different groups like Reg, Core or Smoke etc.
* To create/setup group, its very easy, @Test(priority=1, dependsOnMethods = “Login”, groups =“Smoke”)
* We can add more than 1 groups for 1 single tc.

**TestNG Listeners - How to setup and use?**

* TestNG provides @Listeners annotation which listens to every event that occurs in Selenium code.
* TestNG is mainly used to capture screenshots whenever Tcs fails. Try and catch we can do but for a larger project, this is not an ideal way. So Listeners has several annotations which helps to achieve this.
* To use Listeners we can create a class and implement ITestListener interface which has server un-implemented methods like onTestPass, OnTestFail etc.
* For OnTestFail, inside this method , you can call take screenshot method inside that.

**TestNG TC Skip - How to Skip TC forcefully?**

* Sometimes in real projects we want to skip few TCs. To do that, we can use below syntax inside the method
* Inside the Test Case/ program method, we need to add
  + Throw new SkipException(“Messages that we want to print”); Throw new SkipException(“Message”);

**Question 33: How to do parallel testing?**

* To perform Parellel testing in Selenium. We can use TestNG xml file where we can set parallel attribute set on Suite, Method or Test level. We can also set the thread count like how many thread we want to use to execute the parallel execution.
* We need to add parallel = “tests” inside suite tag in testing.xml.

**How to read data(Parametrization) from an excel file like user name or password in TestNG?**

* We can create method to getData() and for this method use TestNG annotation as @DataProvider
* From the method where will be data used, we need to provide this data provider name like @Test(dataProvider = “getData”)
* While creating this getData Method, we need to use return type as object[][] as data will have row and column.

@@@@@@@@@@@@@@@@@ **Logs, Screenshots & Reports**@@@@@@@@@@@@@@@@@@

**How to take screenshot for failed TCs?**

- We can use try and catch block and inside try block we can have all our codes which we would like to execute and catch(Throwable t) and inside that we can use screenshot method to take the screenshot.

**What is TestNG Listeners?**

* TestNG Listeners are like triggers what it means is when ever something goes wrong in TestNG suite/tcs, Listerners will trigger and capture it. There are n number of TestNG Listeners available like BeforeTest, AfterTest, etc.
* While using Listeners in TestNG, we need to create a separate class and implement one of TestNG internal interface i.e. ITestListener
* When we will implement this ITestListerner, it will automatically ask us to implement unimplemented methods. Click on that and it will provide n number of unimplemented methods like:
  + onTestStart(ITestResult result) {}
  + onTestSuccess()
  + onTestFailure()
  + onTestSkipped()
  + Here is sample syntax:

public void onTestFailure(ITestResult result)

{

system.out.println(“Test Failed” + result.getTextName());

}

* Once methods are override here, we need to include this file in TestNG xml file as TestName under <Test> tag name in xml.
* Also we need to add <Listener> <Listener class name = “package.classname” /> After Suite Tag.

**How to get TestNG Report?**

* After TestSuite gets executed, once we refresh our project, we can see a new folder got created by TestNG i.e. Test-output. Under that we can find index.html file which will provide execution report.
* We can also refer emailable report which has execution details.
* We can also add ReportNG which is a TestNG supported reporting, we can add dependency in pom and implement along with Listeners methods.

**How to generate Extent Report?**

* Need to add Extent report dependency in pom file
* Create a method and import below classes
  + ExtentHtmlReport
  + ExtentReports
  + ExtentTests

**How to generate Log files and log information?**

* Need to add Log4J dependencies
* Need to create a .properties file and add to your project
* This properties file should have appenders and other format defined
* Need to use PropertyConfigurator.configure(path of your .property file)

**How to send email in Java & Selenium?**

* Need to add javamail dependency
* Need to create a class under utility package
* In that class, need to write all the required code for sending email. (Easily available on internet)
* Create another class under same utility package and defined all common values like email id, body , password etc. with static keyword.
* Now create another class and call email method and provide all the details as arguments inside the method.
* Before running this code, you need to go to Browser security and turn on less secure app access on.

@@@@@@@@@@@@@ **Version controlling & Continuous Integration** @@@@@@@@@@

**Question 63: What is GIT and GITHUB? What is Distributed and Centralised Repository?**

* GIT was developed by Linus Torvald who also created Linux corral.
* GIT is a Distributed Version Control System
* Repository - Where we store all our codes is knows as Repository
* Centralised Repository - When all backup codes (different version of your previous codes) are stored on single server and maintained.
* Distributed Repository - When every developer has a local copy of current codes and back up versions of their code. They can work even if there is no internet and once done and internet available they can commit the code and push their code to Remote Repository.
* GIT will help developer to use couple of commands and maintain the version controlling.

**Question 64: Explain GIT Architecture? What are GitHub common and popular commands**

* GIT has 4 Phases:
  + Working Directory -> Staging -> GIT Local Repository -> GITHUB Remote Repository

Untracked File ->(add) Tracked Files ->(commit)Committed Files.-> (push) Remote Files

* Git Commands:
* **Creating a Repository Commands:**
  + Git init - To initialise local git repository
  + Git clone existing-git-repository - to Clone an existing git repository

**- Basic Snapshoting**

* + Git status - To check the status
  + Git add - Add a file in staging area
  + Git add -A - Add all new and changed files to staging area
  + Git commit - Commit Changes
  + Git rm -r - Remove a file or folder
* **Branching and Merging**
  + Git branch - List branches
  + Git branch -a - List all branches (local and remote)
  + Git branch branchname - Create a new branch
  + Git push origin - - delete branch name - Delete a remote branch
  + Git checkout -b branchname - Create a new branch and switch to it
  + Git checkout -b branchname orginbranch - Cream a new remote branch and switch to it
  + Git branch -m [old branch name] [new branch name]
  + Git merge current branch - to marge current branch to active branch
  + Git stash - Stash changes in a dirty working directory
* Sharing and updating project
  + Git Push origin [branch name] - Push a branch into your remote repository
  + Git Push - Push changes to Remote Repository
  + Git Pull - Pull changes from Remote Repository
  + Git remote add origin path - Add a new repository
* **Inspection and Sharing**
  + Git Log - view changes
  + Git config - - global user.name <<“Santosh>> - It allows to set configuration options

**Question 65: What is Head in terms of GIT and how many GIT Heads can be re-presented?**

* A Head is nothing but reference to last commit object of a branch
* For every repository there will be a default Head referred as Master or Main but there is no restriction of head numbers.

**Question 66: What is Conflict in GitHub?**

* When 2 separate branches have changes in the same line in a file
* A file is deleted in one branch and trying to modify in another branch

**Question 67: What is Jenkins?**

* Jenkings is a freeware tool which supports Continuous Integration Process.

@@@@@@@@@@@@@ **Database Connection** @@@@@@@@@@@@@@

**Mention some important utilities classes we need to automate any project?**

* Read and write excel file
* Read and write csv, text and pdf files
* Send email
* Taking Screenshot with timestamped for failed Tcs
* DB Connection

**How to connect Database and perform db testing?**

* Need to add mysqlconnectorJava or SQL
* Need database url (ip and port), userid and password to connect any database