Selenium Interview Questions

1. What is Selenium? Why do you prefer Selenium Automation Tool? What are the common test automation tools? What are the testing types that can be supported by Selenium? Challenges, issues and limitations of Selenium tool?

Selenium is a suite of software tools to automate Web Browsers.

Prefer to Use Selenium Tool

- It is an Open source suite of tools mainly used **for Functional and Regression Test Automation.**
- Selenium supports various Operating environments:
  - MS Windows,Linux,Macintosh etc…
- Selenium supports various Browsers:
  - Mozilla Firefox,IE,Google Chrome,Safari,Opera etc…
- Selenium supports various programming environments to write programs (Test scripts):
  - Java,C#,Python,Perl,Ruby,PHP, JS

Selenium and QTP/UFT are most commonly used tools

Scenarios we cannot automate using Selenium WebDriver:

- Automating Captcha is not possible
- We cannot read barcode using Selenium WebDriver
- Windows based pop ups
- Image validation and comparison
- PDF validation and comparison

2. What is Selenium IDE? What is Selenese? What is Selenium RC? What is Selenium Grid? When do you use Selenium Grid? What are the advantages of Selenium Grid?

Selenium IDE Selenium IDE (Integrated Development Environment) is a Firefox plugin.
Selenium RC AKA Selenium Remote control / Selenium 1.
Selenium WebDriver AKA Selenium 2 is a browser automation framework that accepts commands and sends them to a browser. It is implemented through a browser-specific driver. It controls the browser by directly communicating with it.
Selenium Grid is a tool used to distribute your test execution on multiple platforms and environments concurrently.

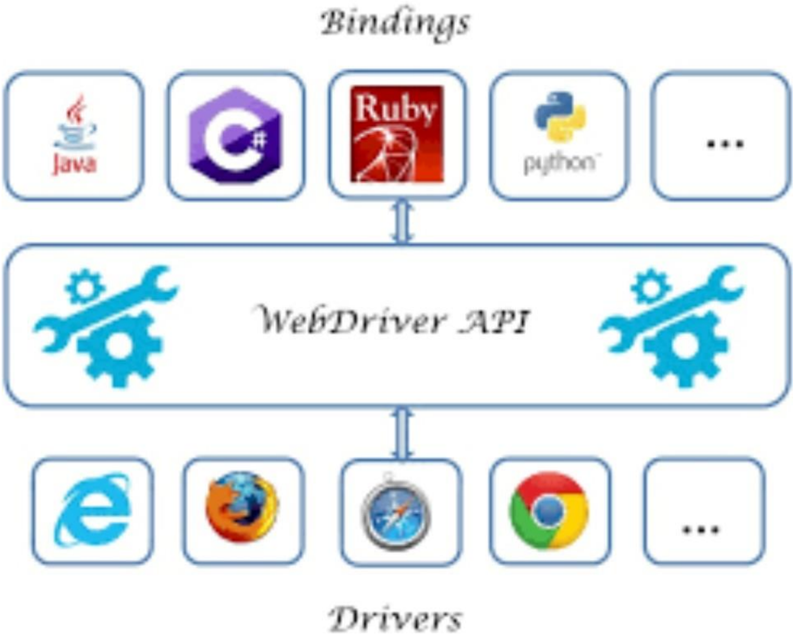
Usage or Advantage of Selenium Grid:

- It allows running test cases in parallel thereby saving test execution time.
- It allows multi-browser testing
- It allows us to execute test cases on multi-platform

3. What is Selenium WebDriver? What is the difference between selenium 3 and selenium 2? Which one do you incorporate in your framework? How does Selenium interact with the Web browser?

Selenium webDriver is one tool in selenium
Selenium 1
(Selenium IDE + Selenium RC + Selenium Grid)
Selenium 2 and 3
(Selenium IDE + Selenium RC + Selenium WebDriver + Selenium Grid)

We are using the Selenium 3.0 because of compatibility issues which can support most of the latest browsers any way we are creating maven project any dependency we can easily update in pom.xml file



4. What are the types of WebDriver APIs available in Selenium? Which WebDriver implementation claims to be the fastest? What is the super interface of WebDriver? What is a headless browser?

Different webdriver API:

- Gecko Driver
- InternetExplorer Driver
- Chrome Driver
- Safari Driver
- HTMLUnit Driver

HTMLUnitDriver, because it is headless browser which have less User Interface

Super interface of WebDriver is SearchContext Interface.

5. What are the Open-source Frameworks supported by Selenium WebDriver?

- JUnit
- TestNG
- Cucumber
- JBehave

6. What are the Locators available in Selenium? Which locator do you prefer?

There are eight locators in selenium to identify the webelements on the webpage : ID, ClassName, Name, TagName, LinkText, PartialLinkText, XPath, CSS Selector.

You can mention that preference of the locator depends on the project/application

7. What is a XPath? What is the difference between Absolute Path and Relative Path? What is the difference between "/" and "//". In which situations are you going to use Xpath?

XPath is used to locate the elements. Using XPath, we could navigate through elements and attributes in an XML document to locate web elements.

- **Single Slash "/"** – Single slash is used to create XPath with absolute path i.e. the XPath would be created to start selection from the document node/start node and it is called Absolute XPath.

Example : /html/body/td/tr/div[1]/div[2]

- **Double Slash "//"** – Double slash is used to create XPath with relative path i.e. the XPath would be created to start selection from anywhere within the document. And it is called Relative XPath.

Example: //input[@id='username']

8. Write a complex xpath or css expression? Which one is better? What is the difference?

xpath:
"//label[@for='personal_txtLicExpDate']/following-sibling::img"

Both **CSS** and **XPATH** are better in the Selenium Automation

When we want to automate the test Case on Internet Explorer then CssSelector is best and for the rest (Browser i.e. Firefox, Chrome and Safari) xpath is better feasible

Difference between both Xpath and CSS:

Generally, CSS is easy to use and readable over XPATH.

CSS is native to browsers and XPATH is not

Speed wise CSS and XPATH can be equal, or either one would be a bit speedier than other. Therefore, speed comparison can be ignored. However, off the record, I have encountered that **CSS is much faster than XPATH in IE Browser.**

9. What is the difference between Assert and Verify in Selenium? What is the difference between using Assertions vs using If condition? What are Soft Assert and Hard Assert in Selenium? What are the verification points available in Selenium?

Assert: If the assert condition is true then the program control will execute the next test step but if the condition is false, the execution will stop and further test step will not be executed.
Verify: Whether verified condition is true or false test script execution will continue

Hard assertion breaks a test immediately with an Assert Exception after the assert method reports failure. If one of the many test cases fails, then the execution continues with the remaining tests.

Soft assertion doesn't throw an exception on failure instead it collects all the errors until it finishes. And that's how we could manage to run all the steps of a test @Test

In Selenium WebDriver, there is no built-in features for verification points. It totally depends on our coding style.
Some of the Verification points are:

- To check for page title
- To check for certain text
- To check for certain element (text box, button, drop down, etc.)

10. How to launch a browser using Selenium WebDriver? Explain the line of code

```
// set the property for chrome driver, specify its location via the webdriver.chrome.driver
System.setProperty("webdriver.chrome.driver", driverPath+"chromedriver.exe");

// instantiate an instance of ChromeDriver, which will be driving our browser:
public static WebDriver driver = new ChromeDriver();

// navigate to the specific url
driver.get("http://google.com");
```

11. Is the FirefoxDriver a Class or an Interface?

FirefoxDriver is class that has extended RemoteWebDriver. And Remote RemoteWebDriver class implements WebDriver.

12. What are the different exceptions you have faced in Selenium WebDriver? What is StaleElementReferenceException? Have you encountered it ever and how you handled it?

- ElementNotVisibleException
- NoSuchElementException
- NoSuchFrameException
- NoAlertPresentException
- NoSuchWindowException
- StaleElementReferenceException - One of the worst Exception for an automation engineer. WebDriver throw this exception when element is in the DOM and even visible on the screen but you can't access the element as DOM reference change. This can happen if a DOM operation happening on the page is temporarily causing the element to be inaccessible.

Solution: We could refresh the page and try again for the same element. or Wait for the element till it gets available
wait.until(ExpectedConditions.presenceOfElementLocated(By.id("table")))

13. How To Login Into Any Site If It Is Showing Any Authentication Pop-Up For Username And Password?

To do this we pass username and password with the URL
http://username:password@url
e.g. http://myUserName:myPassword@gmail.com

Testing URL: http://abcdatabase.com/basicauth
Code: driver.get("http://test:test@abcdatabase.com/basicauth ");

By using Alerts in Selenium, we could handle authentication pop up.

```
//Selenium-WebDriver Java Code for entering Username & Password as below:
driver.findElement(By.id("userID")).sendKeys("userName");
driver.findElement(By.id("password")).sendKeys("myPassword");
driver.switchTo().alert().accept();
driver.switchTo().defaultContent();
```

14. How to handle hidden elements in Selenium WebDriver? Is there a way to click hidden LINK in WebDriver?

First store that element in object, let's say element and then write following code to click on that hidden element

```
WebElement element= “property of element”;
JavascriptExecutor js = (JavascriptExecutor)driver;
js.executeScript("arguments[0].click();", element);
```

15. What are the types of waits available in Selenium WebDriver? What is Implicit/ Explicit/ Fluent Wait In Selenium WebDriver?What is WebDriverWait In Selenium WebDriver?

Implicit Wait

The implicit wait will tell the web driver to wait for certain amount of time before it throws a "No Such Element Exception". The default setting is 0. Once we set the time, web driver will wait for that time before throwing an exception.

```
driver.manage().timeouts().implicitlyWait(TimeOut, TimeUnit.SECONDS);
```

Explicit Wait

The explicit wait is used to tell the Web Driver to wait for certain conditions (Expected Conditions) or the maximum time exceeded before throwing an "ElementNotVisibleException" exception.
The explicit wait is an intelligent kind of wait, but it can be applied only for specified elements.

```
WebDriverWait wait=new WebDriverWait(drv,30);
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("ckdmc"));
```

Fluent Wait

The fluent wait is used to tell the webdriver to wait for a condition, as well as the frequency with which we want to check the condition before throwing an "ElementNotVisibleException" exception. Frequency: Setting up a repeat cycle with the time frame to verify/check the condition at the regular interval of time.

Consider a scenario where an element is loaded at different intervals of time. The element might load within 10 seconds, 20 seconds or even more than that if we declare an explicit wait of 20 seconds. It will wait till the specified time before throwing an exception. In such scenarios, the fluentwait is the ideal wait to use as this will try to find the element at different frequency until it finds it or the final timer runs out.

```
Wait wait = new FluentWait(WebDriver reference)
.withTimeout(timeout, SECONDS)
.pollingEvery(timeout, SECONDS)
```

```
.ignoring(Exception.class);
```

16. How to input text in the text box using Selenium WebDriver? How to input text in the text box without calling the sendKeys()?

we can input text using sendkeys

```
WebElement email = driver.findElement(By.id("email"));
email.sendKeys("your@email.here");
```

as well as

```
JavascriptExecutor executor = (JavascriptExecutor)driver;
executor.executeScript("document.getElementById("email").value='your@email.here';);
```

17. How to get a text of a web element? How to get an attribute value using Selenium WebDriver?

- String buttonText = driver.findElement(By.cssSelector("div.success > button")).getText();
- String innerText = driver.findElement(By.cssSelector("div.success")).getAttribute("innerHTML");

18. How to press ENTER key on text box In Selenium WebDriver? How to press TAB key in Selenium?

driver.findElement(By.id("Value")).sendKeys(Keys.RETURN); **or**
driver.findElement(By.id("Value")).sendKeys(Keys.ENTER); **or**
driver.findElement(By.id("Value")).sendKeys(Keys.TAB);

19. How to pause a test execution for 5 seconds at a specific point?

```
Thread.sleep(5000);
```

20. What is the alternative to driver.get() method to open an URL using Selenium WebDriver? What is the difference?

```
driver.navigate().to("http://www.example.com");
```

The navigate interface also exposes the ability to move backwards and forwards in our browser’s history:

```
driver.navigate().forward();
driver.navigate().back();
```

21. What is the difference between driver.close() and driver.quit() methods? close()

- It is used to close the browser or page currently which is having the focus.

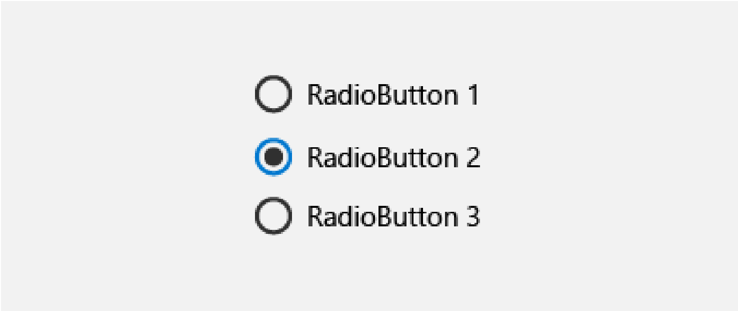
quit() - It is used to shut down the webdriver instance or destroy the web driver instance(Close all the windows).

22. What is the difference between driver.findElement() and driver.findElements() commands? What is the return type of findElements?

findElement() will return only **single WebElement** and if that element is not located or we use some wrong selector then it will throw NoSuchElementexception.

findElements() will return **List of WebElements** – It returns empty list, when element is not found on current page as per the given element locator mechanism. It doesn't throw NoSuchElementException

23. How do you work with radio buttons which do not have id attribute?



We can use click() ,if “id” is not present we can use xpath(i.e you can use relative or absolute) ,css or other locator type

24. How do you handle an element in different windows? Getwindowhandle vs Getwindowhandles and the return types. Write a method to switch to window based on title?

We can handle multiple windows in selenium webdriver using **Switch To** method which will allow us to switch control from one window to another window.

getWindowHandle() will get the handle of the page the webDriver is currently controlling. This handle is a unique identifier for the web page. This is different every time you open a page even if it is the same URL.

```
String handle= driver.getWindowHandle();
```

getWindowHandles() method or commands returns all handles from all opened browsers by Selenium WebDriver during execution

```
Set<String> handle= driver.getWindowHandles();//Return a set of window handle
```

You can use SwitchTo().Window("handle") to switch to the window you desire.
You can use SwitchTo().Window("mywindowID"), if you know the window ID.

SwitchTo().Window("") will always go back to the base/main window.

Method to switch to window based on title

```
public switchToWindow(String titleName) {
                driver.switchTo().window("titleName"); //titleName =windowName
}
```

25. How would you handle element that is in different frame? How to identify the frame which does not have Id as well as name?

Using switch commands we can switch to different frame to handle elements:

We can switch to the frame 3 different ways:

1. **Switch to Frame by Name or ID**

   ```
   driver.switchTo().frame("iframe1")
   ```

2. **Switch to Frame by WebElement**

   ```
   WebElement iframeElement = driver.findElement(By.id("IF1"));

   //now use the switch command

   driver.switchTo().frame(iframeElement);
   ```

3. **Switch to Frames by Index**

   ```
   driver.switchTo().frame(0)
   ```

26. How do you handle the calendar elements? WebTables?

Step 1- Click on calendar
Step 2- Get all td of tables using findElements method
Step 3- using for loop get text of all elements
Step 4- using if else condition we will check specific date
Step 5- If date is matched then click and break the loop.

```
public class CalendarHandling {
    public static void main(String[] args) {

        System.setProperty("webdriver.firefox.marionette","G:\\Selenium\\Firefox
                                        driver\\geckodriver.exe");
                WebDriver driver=new FirefoxDriver();
                driver.get(“URL)
                driver.findElement(By.id("datepicker")).click();
                List<WebElement>
allDates=driver.findElements(By.xpath("//table[@class='ui-datepicker-
calendar']//td"));

        for(WebElement ele:allDates) {
                        String date=ele.getText();
                        if (date.equalsIgnoreCase("28")) {
                                ele.click();
                                break;
                        }
                }
}
```

WebTable

```
List<WebElement> rows =
        driver.findElements(By.xpath("//table[@id='resultTable']/tbody/tr")); for (int i = 1; i
        <=rows.size(); i++) {
                String rowData = rows.get(i).getText();
                   if (rowData.contains(ID)) {
                        driver.findElement(By.xpath("//table[@id='resultTable']/tbody/tr["+
                                                            i
                                                +"]/td[1]/input")).click();
                        break;
                    }
         }
```

27. How to select a value in a dropdown? How to check the multiple selected value in a dropdown?

We can use Select class

```
Select dropdown = new Select(driver.findElement(By.id("designation")));
```

- dropdown.selectByVisibleText("Value");
- dropdown.selectByIndex(1);
- dropdown.selectByValue("value");

isMultiple() Method

isMultiple() method is useful to verify if targeted select box is multiple select box or not means we can select multiple options from that select box or not. It will return boolean (true or false) value. We can use it with if condition before working with select box

```
Select listbox = new
Select(driver.findElement(By.xpath("//select[@id='FromLB']")));

            if (blistbox.isMultiple()){
                  listbox.selectByVisibleText("Value");
                  listbox.selectByVisibleText("Value");
            }
```

28. How to capture Screenshot in Selenium WebDriver?

We use TakeScreenShot interface to capture screenshot

```
//Step 1. Convert web driver object to TakeScreenshot
  TakesScreenshot scrShot =((TakesScreenshot)webdriver);

//Step 2. Call getScreenshotAs method to create image file
  File SrcFile=scrShot.getScreenshotAs(OutputType.FILE);

 //Step 3. Move image file to new destination
  File DestFile=new File(fileWithPath);

//Step 4. Copy file at destination
 FileUtils.copyFile(SrcFile, DestFile);
```

29. What operations can you do using actions class?

We can perform following operation using action class:

- click (): Simply click on element
- doubleClick (): Double clicks on Element
- contextClick() : Performs a context-click (right click) on an element
- clickAndHold(): Clicks at the present mouse location (without releasing)
- dragAndDrop(source, target): Invokes click-and-hold at the source location and

moves to the location of the target element before releasing the mouse. source – element to grab, target – element to release

30. How To Perform Right Click (Context Click) In Selenium WebDriver? Double Click? Hover over on a web element? Drag And Drop?

Using action class we can perform all of this operations

```
Actions action = new Actions(driver);

WebElement element=driver.findElement(By.linkText("TEST"));

//Double click
action.doubleClick(element).perform();

//Mouse over
action.moveToElement(element).perform();

//Right Click
action.contextClick(element).perform();

//mouse over
WebElement element = driver.findElement(By.name("source"));
WebElement target = driver.findElement(By.name("target"));
action.dragAndDrop(element, target).perform();
```

31. How can we handle web based pop-up? How can you handle JavaScript Alerts? Can we inspect an alert? How many ways you can handle alert? How do you get text from alert? How do you send text to alert? How can we handle windows based pop up?

We can handle web based pop-ups using Alert Interface.
There are the four methods that we would be using along with the Alert interface.

```
Alert alert = driver.switchTo().alert(); or driver.switchTo().alert()
```

1) void dismiss() – The dismiss() method clicks on the “Cancel” button as soon as the pop up window appears.

```
alert.dismiss(); or driver.switchTo().alert().dismiss();
```

2) void accept() – The accept() method clicks on the “Ok” button as soon as the pop up window appears.

```
alert.accept(); or driver.switchTo().alert().accept();
```

3) String getText() – The getText() method returns the text displayed on the alert box.

alert.getText(); or driver.switchTo().alert().getText();

4) void sendKeys(String stringToSend) – The sendKeys() method enters the specified string pattern into the alert box.

alert.sendkeys(“text”); or driver.switchTo().alert().sendkeys(“text”);

Selenium WebDriver cannot handle window based pop up, for that we can use AutoIT.

32. How To Scroll Web Page Down Or UP Using Selenium WebDriver?

We use JavascriptExecutor

```
//to perform Scroll on application using Selenium
 JavascriptExecutor js = (JavascriptExecutor) driver;
 js.executeScript("window.scrollBy(0,250)", "");
```

```
//to perform Scroll on application using  Selenium
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript("window.scrollBy(0,document.body.scrollHeight)");
```

```
//to scroll an application to specified elements
JavascriptExecutor je = (JavascriptExecutor) driver;
je.executeScript("arguments[0].scrollIntoView(true);",element);
```

33. Have you done any cross browser testing within your Project?

Yes I have done cross browser testing in my framework using 3 browser initialization. Chrome , IE and Firefox testing using Webdriver.

34. How to Upload a file in Selenium WebDriver? How to Download a file in Selenium WebDriver?

sendKeys() method on the file-select input field to enter the path to the file to be uploaded. ...

```
WebElement uploadElement = driver.findElement(By.id("uploadfile_0"));
```

```
// enter the file path onto the file-selection input field
 uploadElement.sendKeys("C:\\newhtml.html");
```

Download

```
System.setProperty("webdriver.gecko.driver", "src/Drivers/geckodriver");
FirefoxProfile profile = new FirefoxProfile();
```

```
  //Set the last directory used for saving a file
  profile.setPreference("browser.download.dir",
  "/Users/codeinhub/Desktop/Upload_Download/");

//Use for the default download directory
  profile.setPreference("browser.download.folderList", 2);

  // Set Preference to not show file download confirmation dialogue using MIME types
  Of different file extension types.

  profile.setPreference("browser.helperApps.neverAsk.saveToDisk",
          "application/msword.application/octet-stream.application/pdf");
profile.setPreference("pdfjs.disabled", true);

  //create FirefoxOptions and pass preference to it
  FirefoxOptions options = new FirefoxOptions();
  options.setProfile(profile);

  WebDriver driver = new FirefoxDriver(options);
  driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
  String url = "http://toolsqa.com/automation-practice-form/"; driver.get(url);

// click on link to download file
  driver.findElement(By.linkText("Test File to Download")).click();
```

Or using autoIT script we can upload / download

35. How can you find Broken Links/Images in a page using Selenium WebDriver?

Step to find broken links:

- Collect all the links from the webpage. All the links are associated with the Tag ‘a‘.
- Create a list of type WebElement to store all the Link elements in to it.
- Now Create a Connection using URL object(i.e ., link)
- Connect using Connect Method.
- Use getResponseCode () to get response code. eg 200

```
List<WebElement> links = driver.findElements(By.tagName("a")); for
          (int i=0; i<links.size(); i++) {
       String linkURL = links.get(i).getAttribute("href"); if
        (linkURL!= null) {
                 URL obj = new URL(linkURL); HttpURLConnection
                 conn = ((HttpURLConnection)
                 obj.openConnection());
```

```
                    int rCode = conn.getResponseCode();
                  if (rCode == 200) {
                       System.out.println(i+ " Link is valid------" + linkURL);
                  } else {
                       System.out.println(i+ " Link is broken------" + linkURL);
                  }
               }else {


                       System.out.println(links.get(i).getText()); System.out.println(i+ "
                       Link is broken ********" + linkURL);
            }
         }
```

36. Is it possible to automate the captcha using Selenium? List some scenarios which we cannot automate using Selenium WebDriver?

No we can't automate captcha.
Less Support for Image based Testing, No Other Tool integration for Test management

37. Suppose developer changed the existing image to new image with same xpath. Is test case pass or fail?

No test case will pass because we can not check image using selenium.

38. Scenario: you have 2 frames in the page and in 1 you need to enter some text in second you need to click a button. How can you do that? Can you switch from frame to frame directly?

I will first switch frame 1
driver.switchTo().frame(1); and enter text then come to default main

using **driver.switchTo().defaultContent();**

again I will switch to second frame
driver.switchTo().frame(2); and click on button

39. Scenario: there is a submit button in page it has id property. By using id we got element not found exception, how will you handle this situation? What might be the problem this case?

We can use xpath or check that button or check if button is inside frame or we can use submit() rather than click() or else we can use the javaScriptExecutor for that. Or maybe it takes some extra time for button to get loaded in DOM. So we can add Explicit wait until element becomes clickable.

40. Suppose I have 2 firefox installed in my system of different versions or same versions, now tell which firefox will selenium picks?

Which Gecko driver version match that firefox version and path you mention in system.setproperty will be pick by selenium

```
System.setProperty("webdriver.gecko.driver",
"D:\\Library\\drivers\\geckodriver.exe");
```

41. Write a dynamic XPath to locate a table's 2nd row 3rd column data.

```
//to get 2nd row 3rd column data
WebElement cellIneed =
tableRow.findElement(By.xpath("//table/tbody/tr[2]/td[3]"));
```

42. Write a program to scroll in selenium until an element is found?

```
WebElement element = driver.findElement(By.id("id_of_element"));
((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView(true);",
element);
```

43. Let's say I have a page and on that page I have html web table that have 4 columns and twenty rows. And I have to validate data in the last rows?

Use xpath index concept and take text of last row and do assertion

```
String url="https://www.toolsqa.com/automation-practice-table/";
driver.get(url);

//get numbers of rows
List<WebElement> rows =
driver.findElements(By.xpath("//table[@summary='Sample Table']/tbody/tr"));

int lastRow = rows.size();
```

```
WebElement lastRoww =driver.findElement(By.xpath("//table[@summary='Sample Table']/tbody/tr["+lastRow+"]"));

System.out.println(lastRoww.getText())
```

; OR:

```
List<WebElement> rows =
driver.findElements(By.xpath("//table[@summary='Sample Table']/tbody/tr"));

          for (int i=rows.size()-1; i>=0; i--) {

               String latRowValue=rows.get(i).getText();

               System.out.println(latRowValue);

               break;

          }
```