DAY 2 SELENIUM NOTES

- **What is automation?**

  If there is a repeating process, and we reduce or remove human interruption/interaction, we automate that process.

- **How does automating tests help us?**

  Precision
  Reduce mistakes
  More accurate
  Faster/Save time
  Save cost
  Auto-generated reports

What is Selenium?

- Basically, Selenium is a bunch of jar files that allows us to automate browsers.
- In simple terms, selenium is bunch of jar files (libraries)
- We use these jar files to automate browsers

Maven Project:

- Maven is a "**build**" automation tool
- "build" : repeating steps when we are creating a project
- Maven project is different than Java project
- Maven is NOT a tool for testers.
- It comes with certain type of folder structure and other files.

- **src**

  main : this is where developers write the source code of the webApp/webPage
  test : this is where developers write their unit tests
  pom.XML : this file is used to add and manage dependencies of our project

- **target** : this file is where the maven project is storing the compiled version of the project and some other information.
- it will only appear after you run your code.
- everything in this folder will be refreshed every time we run our code.

- **Is Maven for tester?**
- No. Maven is a tool created for developers.
- As testers (SDETs) we will take advantage of some of its functionality.

- **What version of Selenium we are using?**
- 3.141.59

- **What version of WebDriverManager we are using?**
- 5.1.1

- **If the developers are using both main and test folders, which folder the testers use?**
- Testers and developers are not working on the same project.
- As SDET you will not have access to the source code.

Selenium WebDriver vs Browser Driver

- **Selenium WebDriver:**
  Is actuallly the name of the tool
  Collection of libraries that allows us to automate browsers.

- **Browser Driver (is the translator)**
  Browser driver does not come in the selenium library
  we need to set it up seperately.
  If we are using WebDriverManager (BoniGarcia) easily sets up our driver
  If we are NOT using WebDriverManager, we have to download our browser drivers seperately.

.get(""); Method

- What does the method do?
  It gets given URL in an opened browser.
- Does it accept any argument?
  Yes it does.
  It accepts a String argument.
- Does it have a return type?
  void return type.

Basic Navigations

- driver.navigate() --> navigate methods allows us to do simple navigations
- driver.navigate().forward () --> will take page to forward page
- driver.navigate().back() --> will take page to previous page
- driver.navigate().refresh()--> will refresh the current page
- driver.navigate().to() --> exactly same as .get() method

.getTitle(); Method

- What does the method do?
  It gets the title of the page
- Does it accept any argument?
  No. It does not accept any argument.
- Does it have a return type?
  Yes. It returns a String.
  It gets the title of the current page, and returns it as a String.
- Does it throw any certain exception?
  No.

.getCurrentUrl(); Method

- What does the method do?
  Gets the URL of the current page.
- Does it accept any argument?
  No. It does not accept any argument.
- Does it have a return type?
  Yes. It returns a String.
  Gets the URL of the current page and return as String.
- Does it throw any certain exception?
  No.

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

- Maximizes the currently opened browser.
- This effect (maximizing) will happen on the line we call this method.

driver.manage().window().fullscreen();

- this one sometimes does not work for windows, so better not use it.

Why maximizing the page is important?

- Because in different sizes the content displayed will be different.
- This happens if the page is "responsive".

- **Responsive means** : if size gets smaller, it will display less content.
- Maybe it will store some web elements under some hamburger menu, or some other type of menu.

- **driver.close();**
- will close only the currently opened browser
- if we have more than one window open, it will only close the one that was focused on

- **driver.quit();**
- it kills the current session.
- it means if more than one window was opened, everything will be closed.
- after using .quit() method, we cannot execute any more line of codes.
- We will get "NoSuchSessionException”

- **What is a session?**
- Every time we run our Selenium code a session is created.
- That specific session will continue until we explicitly kill it or code execution comes to end.

Thread.sleep():

- It comes from java library
- It has nothing to do with SELENIUM
- Basically, stops the page for given duration in milli seconds

- 3000 milli seconds = 3 secs

Expected Value = will always come from documentation

Actual Value = will always come from the browser

Verification = we will compare '**expected**' vs '**actual**' values