**SELENIUM**

* Selenium is an open-source framework used for automating web browsers. It provides a set of tools and libraries that enable testers to automate browser-based interactions, perform web testing, and automate repetitive tasks on web applications.
* Configuration of Selenium:

1. Download selenium.
2. Download Java
3. Create Maven Project in IntelliJ.
4. Download all the binaries from Maven repository (Web Driver Manager, Selenium).
5. Save the Pom.xml file.
6. Create a package in test and create a class called java class.

* Driver.getTitle( ); is the method used to get the title of that application.
* Driver.quit( ): is used to quit current browser window + all the sessions related to same sessions.
* Driver.close( ): is used to close the current browser window.
* Selenium doesn’t interact with the UI but it interacts with the HTML path i.e. DOM component of element.
* Find element(): looks for the first element. The first web element is nothing but the tag <input> and type, class etc will comes under attributes.
* **Locators**: locators are a way to identify and locate web elements on a web page when automating web testing or interacting with a web application using tools like Selenium WebDriver. Locators provide a means to uniquely identify elements such as buttons, input fields, links, checkboxes, and other elements on a web page.

1. Id
2. Name
3. cssSelector
4. xpath
5. tagname
6. classname
7. linkText
8. partialLinkText

* These are the major locators we use to follow from selenium 2.0. But from selenium 4.0 there are some relative locators called:

1. Above
2. Below
3. Near
4. toRightof
5. toLeftof

* Xpath: We can give for find elements By.id, By. Xpath, By.name, By. Class etc in order to identify the locator easily.
* Xpath by visible text: //a[text()='Go To Registration ']
* Xpath by id: //input[@id='firstname']
* Xpath by name: //input[@name='firstname']
* Xapth by tagname: //<a> or href
* Xpath for text: //div[text()='Messages'] (tagname with text)
* Xpath by Placeholder: //label[@aria-label="Remember me Checkbox"] , //input[@aria-placeholder="Password"]
* Xpath by index: (//input[@id='firstname'])[1]
* **Radio Buttons:**  Radio buttons can be selected by using ID of that respective radio button or any other appropriate locator.
* To check whether the radio button is selected or not “isSelected( )” method is used.
* How to handle dropdowns for select tag with the dropdowns. Models a SELECT tag, providing helper methods to select and deselect options.
* First we need to give public Select​([WebElement](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/WebElement.html" \o "interface in org.openqa.selenium) element)Constructor. A check is made that the given element is, indeed, a SELECT tag. If it is not, then an sUnexpectedTagNameException is thrown.

**Parameters:**

element - SELECT element to wrap

**Throws:**

[UnexpectedTagNameException](https://www.selenium.dev/selenium/docs/api/java/org/openqa/selenium/support/ui/UnexpectedTagNameException.html) - when element is not a SELECT

**Iframe:** An Iframe is a content coming from different source and we are presenting that data in that page. If the elements are there in Iframe we cannot directly interact with them. For that we need to switch to Iframe and perform the action. We can write the Iframe in 3 different types: driver. SwitchTo( ).frame( ) and we can pass the arguments into the following syntax directly or we can directly pass ID of the frame into the syntax even.

By Web element: driver. SwitchTo( ). Frame( ) >>this is the syntax for passing the web elements.

By Index: driver. SwitchTo( ). Frame( )>> this is the syntax for passing the index of the frame.

**Extent report :** Extent report is used to generate the detailed interactive test reports.

Extent Report offers features like detailed test execution summaries, step-by-step logs, screenshots, charts, and graphs, which make it easier to analyze test results and identify issues. It allows testers and stakeholders to gain insights into the test execution process and make informed decisions based on the generated reports.

**GIT: It is an local repository.**

**Repo = is a folder or a package.**

**The below mentioned are the 4 elements of the GIT:**

**Github = one example of remote repository hosting service.**

**Local repository is the 2nd element of the GIT**

**Staging area**

**Working directory:**

1. **Git init**
2. **Git status we have given**
3. **We have done working directory**
4. **The untracked one is known as working directory.**