Locators

HTML Tags:

<html>-defines root of the web page

<a>-defines the hyperlink

<button> defines the button

<div> defines the section in a document

<span> defines the section in a document

<iframe> defines a inline frame

<frame> defines a frame

<img> defines a image

<input> defines a input control

<select> defines a drop down

<option> defines a option in drop down

<table> defines a table

What is Locator?

**Locator** is a command that tells **Selenium** IDE which GUI (**Graphical User Interface)** elements ( say Text Box, Buttons, Check Boxes etc) its needs to operate on

**Why do you need Find Element/s command?**

Interaction with a web page requires a user to locate the web element.

Find Element command is used to uniquely identify a (one) web element within the web page. Whereas, Find Elements command is used to uniquely identify the list of web elements within the web page. There are multiple ways to uniquely identify a web element within the web page such as ID, Name, Class Name, Link Text, Partial Link Text, Tag Name and XPATH.

Type of locators

1. id- Most preferred
2. name-if Id does not exit
3. className- when class is unique with out white spaces
4. linkText-if it is a link
5. partialLinkText- when portion of the link changes dynamically
6. tagName-for collection of objects
7. xpath-none of the above works
8. cssSelector-last option

When cannot use specific Locators

1. id- Numbers
2. name-Duplicate
3. className- white spaces, duplicate
4. linkText-duplicate
5. partialLinkText- duplicate
6. tagName-duplicate
7. xpath-X path can change
8. cssSelector-CSS can change

What is Locator:

Locator is an command it tells selenium IDE(eclipse) which GUI elements like text box or check box dropdown or button or link needs to operate.

Locators are used to inspect the elements in web page through DOM—document object model.

They are many types of locators we have.

In that the most efficient and most preferred locator is by ID.

Since ID are supposed to be unique for each web element.

If ID value is not unique means the next preferred locator is by Name.

Locating element by name are very similar to locating by ID.

Instead of using Id value will use name value here.

If name is unique means we can use. If you have multiple or duplicate names we should not use locator as name.

After name the next preferred locator is by className

We can use--when class is unique with out white spaces

We can not--Duplicate and white spaces

After claaName the next preferred locator is by linkText

This type of locator will applies only to the hyperlink texts.

If linkText value is having duplicates means again we should not prefer this locator.

The next preferred locator to inspect hyperlinks by partialLinktext

If partialLinkText has duplicate values again will not prefer this locator

Locating by tagName

This locator will use for collection of elements/objects.

When tagname is duplicate will not use this.

## Locating by CSS Selector

cssSelector:

cssSelector is a advance locator technic .

we can find color font-size, background-color and line-height of the particular element we will use cssSelector.

get.cssvalue(By.ID/By.Name/By.linktext/ By.partialLinktext/By.classname/By.tagName/By.path)

Example:

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

public class GetCssValue {

public static void main(String[] args) {

System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");

WebDriver driver=new ChromeDriver();

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

driver.manage().timeouts().implicitlyWait(30,TimeUnit.SECONDS);

//ger url

driver.get("https://login.salesforce.com/");

//login button details

//get font size

String fontSize\_login = driver.findElement(By.id("Login")).getCssValue("font-size");

System.out.println("font size of login button is "+fontSize\_login);

//get color

String color\_login = driver.findElement(By.id("Login")).getCssValue("color");

System.out.println("color of login button is "+color\_login);

//get backgrong color

String backgroundColor\_login = driver.findElement(By.id("Login")).getCssValue("background-color");

System.out.println("background color of login button is "+backgroundColor\_login);

//get line-height

String lineheight\_login = driver.findElement(By.id("Login")).getCssValue("line-height");

System.out.println("line height of login button is "+lineheight\_login);

}

}

**public interface WebElement extends SerachContext, TakeScreenShot{**

**public interface SerachContext{**

**public abstract class By {**