**Selenium Class**

# Chapter#5 – Locators in Selenium WebDriver

1. **Sample Code for Locators in Selenium WebDriver**
2. **Using Developer Tool**
3. **Locators in Selenium**

## Sample Code for Locators in Selenium WebDriver

//Launching Firefox browser

WebDriver driver = new FirefoxDriver();

//Opening google.com

driver.get("http://www.google.com");

//Initializing webelement searchBox

WebElement searchBox = driver.findElement(By.name("q"));

//Writing a text "Selenium" in the search box

searchBox.sendKeys("Selenium");

So, here we see that in order to perform an operation on the web element – searchBox, we first need to locate it. Here, By.name(“q”) is a locator which when passed to the findElement method returns a searchBox web element.

Before going further with the different types of locators available in Selenium, let’s first see, how to get the different attributes of an element that are used in the locators.

## Using Developer Tool

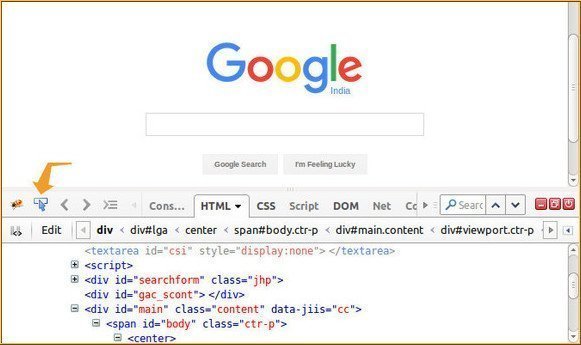
Locating web elements requires knowledge of their HTML attributes. For the HTML source code of specific elements, we can use the inbuilt developer tool (launched by pressing f12 in a browser).

Steps for finding the element’s from HTML attributes-

Launch the website to be automated e.g. – https://www.google.com

Press F12 to launch the developer tool.

Click on the inspect-element icon as displayed in the image below.



## Locators in Selenium

There are a total of 8 locators in Selenium WebDriver-

1. **By Id** – Locates element using id attribute of the web element.

WebElement element = driver.findElement(By.id("elementId"));

2. **By className** – Locates the web element using className attribute.

WebElement element = driver.findElement(By.className("objectClass"));

3. **By tagName** – Locates the web element using its HTML tag like div, a, input, etc.

WebElement element = driver.findElement(By.tagName("a"));

4. **By name** – Locates the web element using the name attribute.

WebElement element = driver.findElement(By.name("male"));

5. **By linkText** – Locates the web element of the link type using their text.

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

6. **By partialLinkText** – Locates the web element of link type with the partial matching of text.

WebElement element = driver.findElement(By.partialLinkText("Click"));

7. **By cssSelector** – Locates the web element using CSS

WebElement element = driver.findElement(By.cssSelector("div#id"));

8. **By XPath** – Locates the web element using its XPaths

WebElement element = driver.findElement(By.xpath("//div[@id='id']"));

Now you have successfully learned how to locate elements in Selenium. As you can see locating elements by id, className, tagName, name, linkText, and partialLinkText is simple.

We just have to select the right locator based on the uniqueness of the element e.g. we prefer using id because the id of elements is generally unique. But there can be scenarios where we might not have id attributes of web elements, also other locators like name, and className might not fetch the unique required web element. In those scenarios, we should use CSS selectors and XPath locators.

These locators are very powerful and help in creating robust locators for complex web elements. We have dedicated separate tutorials for these **two locators in our coming posts – CSS Locators and XPath Locators**.