**Selenium Locators**

Selenium gives user options to locate elements in 8 different ways.

* **Id**
* **Name**
* **Linktext**
* **Partial Linktext**
* **Tag Name**
* **Class Name**
* **CSS**(Cascaded Spread Sheets)
* **XPath**

**Selenium Locators: Locate element by Id**

The most preferred, the easiest and efficient way to locate an element on a web page is By ID. Id will the attribute on the page which will be unique like you bank account number or employee id. Ids are the safest and fastest locator option and should always be the first choice even when there are multiple choices.

Example 1: <input id="txtUserName" type="text">

Example 2: <input id="txtUserName" name="userName" type="text">

In first example its straight forward we only have Id, but in second we have Id as well as Name as an attribute. We can write the script as

WebElement Ele = driver.findElement(By.id("txtUserName "));

But in many cases we found that we have common Id or dynamic Ids *(like in case of google, Gmail or the application using GWT)*. In that case we need to use different locators.

**Selenium Locators: Locate element by Name**

This is a fall back option when Id for element is not present. But mostly the names are used again and again, so make sure that the name is unique on the page before using it.

Example:

<input id="txtUserName" name="userName" type="text">

WebElement ele= driver.findElement(By.name("userName "));

**Selenium Locators: Locate element by LinkText**

Finding an element with link text is very simple. This locator is used in case you want to locate any hyperlink only. But make sure, there is only one unique link on the web page. If there are multiple links with the same link, in such cases Selenium will perform action on the first matching element with link on page.

**Example:**

In above image we have three hyperlinks. If we want to locate the Forgot Password? Link. The locator will be

<a href="#">Forgot Password? </a>

WebElement hyperlink = driver.findElement(By.linkText("Forgot Password?"));

We have 2 links with text Forgot Email. If we try to locate 2nd link with locator  
**LinkText=\*Forgot Email.** Selenium will locate 1st link.

In this case if we want to locate the 2nd link, we will need to use exact keyword with colon **(exact:)**. The locator in that case will be

linkText= exact:\*Forgot Email

<a style="background-color: transparent;" href="#">\*Forgot Email</a>

WebElement hyperlink = driver.findElement(By.linkText("exact:\*Forgot Email"));

**Selenium Locators: Locate element by Partial LinkText**

Partial LinkText works same as LinkText, only difference is you can use a part of the text from link.

Example:

<a href="#">Forgot Password? </a>

WebElement hyperlink = driver.findElement(By. PartialLinkText ("Password"));

**Selenium Locators: Locate element by Tag Name**

Tag Name we can use for the elements like drop downs, check boxed, radio buttons. Following html is for drop down with 3 values. To select that drop down we will use tagName locator.

Example:

<select name="selCity" id="selCity">

<option value="none">--Select--</option>

<option value="PUNE">Pune</option>

<option value="ADI">Ahmedabad</option>

</select>

WebDriver command:

Select select = new Select(driver.findElement(By.tagName("select")));

select.selectByVisibleText("Pune");

or

select.selectByValue("PUNE");

**Selenium Locators: Locate element by Class Name**

This locator we can use as a fall back option for either name or Id. But the same condition applied here Class name should be unique or selenium will locate the first element present on the page with the class name we have used to locate element.

# ****Selenium Locators: Locate element by CSS selector using html tag attributes****

This a fall back when all options fail, you can use parent child relation in tags, in case you need to use complex strategy to locate elements. CSS selectors are string representation of HTML tags, attributes, Id, Class. It’s somewhat complex strategy compared to the previous we seen. But we can locate the elements which don’t have even Id or name or class using CSS selectors.