**What Are Browser Elements?**

Elements are the different components (fields) that are present on web pages. The most common elements we notice while browsing are:

* Text boxes
* Buttons
* Images
* Hyperlinks
* Radio buttons
* Check boxes
* Text area
* Drop down box/ List box/ Combo box
* Web Table/ HTML Table
* Frames

**What is a WebElement?**

WebElement represents an ***HTML element***. HTML documents are made up by *HTML elements*. HTML elements are written with a ***start*** tag, with an ***end*** tag, with the **content** in between: ***<tagname> content </tagname>***

The HTML **element** is everything from the start tag to the end tag: ***<p> My first HTML paragraph. </p>***

HTML elements can be nested (elements can contain elements). All HTML documents consist of nested HTML elements.

***findElement*** command of ***WebDriver*** returns ***WebElement***.

***WebElement element = driver.findElement(By.id(“UserName“));***

Only after the elements are located on the web page, we can perform operations (actions) and start testing them.

Finding correct GUI elements is a prerequisite for creating an automation script but, accurate identification of GUI elements is much more difficult than it sounds.

Sometimes, you might even end up working with incorrect GUI elements or no elements at all! Hence, using the right locator ensures that the tests are faster; stability of the automation script will increase or has lower maintenance over releases.

Any of the below 8 attributes can be used to locate elements uniquely.

Since the elements are located using these attributes, we refer to them as **“Locators”**.

The locators are:

1. By.id
2. By.name
3. By.className
4. By.tagName
5. By.linkText
6. By.partialLinkText
7. By.cssSelector
8. By.xpath
9. **By Id** - Locates element using id attribute of the web element.

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

1. **By className** - Locates the web element using className attribute.

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

1. **By tagName** - Locates the web element using its html tag like div, a, input etc.

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

1. **By name** - Locates the web element using name attribute.

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

1. **By linkText** - Locates the web element of link type using their text.

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

1. **By partialLinkText** - Locates the web element of link type with partial matching of text.

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

1. **By cssSelector** - Locates the web element using css its CSS Selector patterns.

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

1. **By xpath** - Locates the web element using its XPaths.

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

WebDriver has ***Find Element and Find Elements***methods to locate element on the web page.

Every method of the **WebDriver** either returns any value, object, element or returns void.

The same way ***findElement*** method of ***WebDriver*** returns a ***WebElement***.

findElement() method accepts ***By Object*** as a Parameter/Argument.

***findElement()***

* *On One Match: returns WebElement*
* *On Zero Match: throws NoSuchElementException*
* *On One+ Match: returns the first appearance of element in DOM*

***findElements()***

* *On One Match: returns list of one WebElement only*
* *On Zero Match: return an empty list*
* *On One+ Match: returns list with all matching WebElements*

----------------------------------------------------------------------------------------------------------------

Form.html:

<html>

<body>

<form id="login">

<input name="username" type="text"/>

<input name="password" type="password"/>

<input name="submit" type="submit" value="Login"/ class="loginsubmit">

</form>

</body>

</html>

----------------------------------------------------------------------------------------------------------------

Difference between getText() and getAttribute("attribute-name"):

getAttribute() -> It fetchs the text that containing one of any attribute in the HTML tag . Suppose there is HTML tag like

<input name="Name Locator" value="selenium">Hello</input>

Now getAttribute() fetchs the data of the attribute “value” which is "Selenium"

**Method Returns:** The attribute's current value or null if the value is not set.

driver.findElement(By.name("Name Locator")).getAttribute("value")

// The field value is retrieved by the getAttribute("value") Selenium WebDriver predefined method and assigned to the String object.

getText() -> returns the innerText of a WebElement. Get the visible (i.e. not hidden by CSS) innerText of this element, including sub-elements, without any leading or trailing whitespace.

Returns: The innerText of this element.

driver.findElement(By.name("Name Locator")).getText();

“Hello” will appear

----------------------------------------------------------------------------------------------------------------

**Difference between Click() and Submit() in Selenium**

-> Both click() and submit() both are used to click Button in Web page.

-> Selenium Webdriver has one special method to submit any form and that method is submit(). submit() method works same as clicking on submit button.

**When to use .click() method :**

-> You can use .click() method to click on any button. There is no restriction for click buttons.

-> That means element's type = "button" or type = "submit", .click() method will works for both.

-> If button is inside <form> tag or button is outside <form> tag, the click() method will work.

**When to use .submit() method :**

-> we can use .submit() method for only submit form after click on button.

-> That means element's type = "submit" and button should be inside <form> tag, then only submit() will work.

-> if element's type = "button" means submit() will not work.

-> If button outside of the <form> tag means submit() will not work

For Example, Submit() will work if submit button should be inside <form> tag and element type="submit" as below

<form>

<input id="submitbutton" name="submitbutton" type="submit" value="Next step" class="g-button g-button-submit">

</form>

But click() method will work for all buttons in webpage without any restrictions.

**Selenium 4 relative locators:**

<https://www.browserstack.com/guide/relative-locators-in-selenium#:~:text=Selenium%203%20has%208%20locators,was%20previously%20called%20friendly%20locators>.

**Link text and partial link text**

<https://www.guru99.com/locate-by-link-text-partial-link-text.html>

**Practice pages:**

<https://practicetestautomation.com/practice-test-login/>