**.WebElement :**

Any element [html] present in the webPage is called a webelement

**Locators :**

* Using locators we can find the webelement.
* locators are present in By (abstract class).

|  |
| --- |
| id |
| name |
| className |
| x-path |
| CSS Locator |
| tagName |
| Link Text |
| Partial Link text |

**Document Object Model ( DOM )**

In DOM

<html>

<tagName attributeName = " attributeValue"/>

<tagName Create An Account />

</html>

* tagName must be in **pink color**.
* attributeName must be in **orange color**.
* attributeValue must be in **blue color**.
* Text must be in **black color**.

Example:

WebElement uemail = driver.findElement(By.*id*("email"));

uemail.sendKeys("abc@gmail.com");

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**WebElement Methods :**

* **sendKeys()** ---> we pass the input values .
* **click()** ----> to click on the webelement
* **getText()** ----> to get the text of the webelement is present in application.
* **getAttribute()** :

refName.getAttribute ("value") ----> input value that we passed into a textBox.

refName.getAttribute ("name") ----> return attributeValue

* **isDisplayed()** ---> to check whether an element is visible (or) available in page.
* **isEnabled()** ----> to check whether an element is editable (or) clickable
* **isSelected()** ---> To Check whether an element is selected (or) not.
* **clear()** ---> it will clear the passed data.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**X-Path :**

Absolute x -path

Relative x-path

**Relative x-path :**

it starts with double forward slash (//).

it will search the middle of the DOM Page

**Syntax :**

//tagName [@attributeName = 'attributeValue']

//tagName [contains(@attributeName,'attributeValue')]

(//tagName[@attributeName='attributeValue'])[index]

//tagName[contains(text(),'half of the text')]

//tagName[text()='full text']

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Screenshot :**

**Points To Remember:**

* Covert driver object to **TakeScreenshot**
* Call **getScreenshotAs** – method to create image file
* Move image file to new destination path
* Use FileUtils to Copy file at destination

TakesScreenshot (Interface)

TakesScreenshot ts = (TakesScreenshot) driver; //----> narrowing

File source = ts.getScreenshotAs(OutputType.FILE);

File destination = new File("path//name.format");

FileUtils.copyFile(sourcefile,destinetion);

type casting :

short s = (short) long; // narrowing type casting -> convert a higher datatype into lower datatype

TAKING SCREENSHOT BELOW JARS ARE REQUIRED

