**WebElement:**

* WebElement is an interface. Anything present on the web page such as text field, button, link…etc is called as WebElement.
* Before performing any operation/action such as typing text, clicking, selecting…etc, selenium has to identify the element uniquely.
* In-order to identify the element uniquely, selenium uses **locators** which is a property value(attributes) or expression of the element.
* The property values are given by the application developer using HTML code.
* Hence in-order to specify the locators in selenium, we should get the property values from the HTML source code of the web page , for this basic knowledge of html is required.
* See **mypage.html** file for html code

**WebElement interface has following methods:**

1. clear()
2. click()
3. findElement()
4. findElements()
5. getAttribute()
6. getCssValue()
7. getLocation()
8. getRect()
9. getScreenshotAs()
10. getSize()
11. getTagName()
12. getText()
13. isDisplayed()
14. isEnabled()
15. isSelected()
16. sendKeys()
17. submit()

* In Selenium to search the element we use ‘**findElement()**’ & ‘**findElements()**’ methods, which takes ‘**locator**’ as argument.
* ‘**Locators**’ are the static methods of ‘**By’** class. (‘By’ is abstract class)

Selenium identifies the element using locators. In Selenium we have **8-types of locators**. All locators take one argument of type string

1. By.tagName(“value”)
2. By.id(“value”)
3. By.name(“value”)
4. By.className(“value”)
5. By.linkText(“Logout”)
6. By.partialLinkText(“Logo”)
7. By.cssSelector(“value”)
8. By.xpath(“value”)

**Examples for locators:**

1. Using ‘**tagName’** locator in Selenium:

**Ex**:- driver.findElement(By.tagName(“input”));

1. Using ‘**id’** locator in Selenium:

**Ex**:- driver.findElement(By.id(“email”));

1. Using ‘**name’** locator in Selenium:

**Ex**:- driver.findElement(By.name(“field1”));

1. Using ‘**className’** locator in Selenium:

**Ex**:- driver.findElement(By.className(“input text”));

1. Using “**linkText**” locator in Selenium:

**Ex**:- driver.findElement(By.linkText(“ForgetPassword”));

1. Using “**partialLinkText**” locator in Selenium:

**🡪**If the text of the link is partially changing then we use partialLinkText.

**Ex**:- driver.findElement(By.partialLinkText(“Forget”));

1. Using ‘**cssSelector**” locator in Selenium:

🡪 “css” stands for Cascading Style Sheets.It is an expression in selenium

🡪 ‘cssSelector’ is a locator which has following **syntax**.

|  |
| --- |
| ***tagName[Attribute\_name=’Attribute\_value’]*** |

**Ex:-** driver.findElement(By.cssSelector(“input[type=’text’]”));

1. Xpath

**Note: 1**

<html>

<body>

UN: <input type=”text”><br>

Pw: <input type=”password”>

</body>

</html>

For the above input fields, there is no id, name, since it is not a link, we can’t use linkText or partialLinkText but we can use tagName. tagName for both the fields are same hence if we use it, it always performs the action on the 1st field only. In these scenarios we can use cssSelector as shown below

driver.findElement(By.*cssSelector*("input[type='text']")).sendKeys("abc");

driver.findElement(By.*cssSelector*("input[type='password']")).sendKeys("xyz");

Note: 2

<html>

<body>

Fname: <input type=”text”><br>

Lname: <input type=”text”>

</body>

</html>

In the above code both lines are same so we don’t use cssSelector.