**Selenium**

It is an automation testing tool. To automate the web applications.

It is a open source testing tool which is compatible with java, python, c#, ruby etc. it is not applicable to automate desktop applications.

**Components in selenium**: -

1.selenium IDE

2.selenium RC🡪cross browser testing

3.selenium Grid🡪faster (multiple testcases running parallelly on different machines)

4.selenium web driver.

Here web driver is commonly used for automating the software. It is supported to CHROME, FIREFOX, OPERA, SAFARI, EDGE etc.

**Features**: -

It is a open source.

Easy to install and usage.

Parallel execution possible.

Takes less time to automate testcases.

Cloud based testing platform-recorded the test cases and those are re-usable.

**web elements**: -

Anything which is present in the web page is web element.

Ex: -text box, check box, radio button, buttons, select tags, list box, dropdowns etc. we can find the web elements with the help of locators.

For finding the single web element we use to write with

driver. find element (By. locator (‘value’)--🡪 every time it returns single web element.

to find no of elements we use

List<web element> l=driver. Find elements (By. locator(‘value’))--🡪every time it returns list of web elements.

**Locators:** -

Locators are used to find the unique web elements to check whether the element is present or not. There are 8 locators available in selenium.

**1.id**🡪 finding the element with id locator

Syntax -🡪Web element e=driver. Find element (By.id(‘value’))

**2.class name**-🡪 finding the element with classname locator

Syntax -🡪Web element e=driver. Find element (By. classname(‘value’))

**3.name**--🡪- finding the element with name locator

Syntax -🡪Web element e=driver. Find element (By.name(‘value’))

**4.linked text**🡪 finding the element with linkedtext locator

Syntax -Web element e=driver. Find element (By. linked text(‘value’))

**5.partial linked text**-🡪find with half of the text as well

Syntax - finding the element with partial linkedtext locator.

**6.Xpath(XML path)🡪**find the web elements with the help of all locators. There are 2 types of xpaths.

1.Relative Xpath –Starts with //--can takes from any node

2.Absolute Xpath—starts with /--Always takes from the beginning node

Syntax -driver. findelement(BY. xpath(//tagname[@attribut=’value’]).

We can write xpath with id, name, classname, text, contains etc.

**Traversing: -** it is used to finding the html code for dynamically changing web elements

//tagname[@id=’value’]//td//tr[2].

**Xpath by group index**: -

Finding the single element from the group of similar elements with the help of index.

Syntax :- //tagname[@attribute=’value’][2].

**7.CSS selector** 🡪cascaded sheet style.

Syntax -driver. findelemet (By. Css selector(tagname#’value’) --🡪for id

Syntax -driver. findelemet (By.Css selector(tagname.’value’) --🡪for classname

While writing the CSS selector with class name we need to replace spaces with dot(.).

**Differences b/w xpath and CSS selector**

|  |  |
| --- | --- |
| Xpath | Css selector |
| It is slower compared to CSS selector | It is faster |
| It is supported with text as well | It can’t be supported with text |
| Here both forward and reverse traversing possible for finding the web element. | Here only forward traversing is possible for finding the web element. |

**Waits in selenium:-**

To maintain the synchronization we use some waits in selenium. Those are

1.implicit wait

2.Explicit wait

3.Fluent wait

**1.implicit wait :-**

It is the Global wait which is applicable to all the web elements. Every time web driver waits for that time to finding the web element. The advantage of this wait is to wait sometime before shown **No such element exception**.

Syntax –

driver. Manage (). Timeouts (). implicitly Wait (DurationOfseconds,1000)

2.**Explicit wait :-**

This wait is applicable for the single web element. This wait used to tell the webdriver for certain conditions before showing the **element is not visible** **exception and Timeout exception.**

Syntax

Webdriver wait wait=new Webdriver wait (Duration.Ofseconds,1000)

Wait. Until (Expected conditions. visibility of(element)),

Wait. until (Expected conditions. element to be clickable)

**Fluent wait:-**

Fluent Wait in Selenium marks the maximum amount of time for Selenium WebDriver to wait for a certain condition (web element) becomes visible. It also defines how frequently WebDriver will check if the condition appears before throwing the “**ElementNotVisibleException**”.

Syntax

Wait<WebDriver> wait = new Fluent Wait <WebDriver>(driver)

. with Timeout (Duration. Of Seconds (30L))

. polling Every (Duration. Of Seconds(5L))

. ignoring (NoSuchElementException.class).

In this wait poling time will check for every 5 secs whether that element is present or not.

These are the waits frequently used in selenium to wait the web driver certain period of time.