Xpath:

-->Xpath is a XML path which is used to find locators on the webpage using DOM Structure.
-->There are two types of Xpath: Absolute which is denoted by single slash and Relative xpath which is denoted by double slash.
-->In that in our project we are using Relative xpath because we can directly find the element anywhere at the webpage but in Absolute xpath we have to find from the head of the DOM structure.

Actions:

-->Actions is a predefined class which is used for mouse over actions in a webpage. There are few methods as
-->moveToElement() --> used to do mouse over actions
-->doubleClick() --> used for double click
-->contextClick() --> used for right click
-->dragAndDrop() --> to move a webelement from one place to another
-->keyUp() --> for key release
-->keyDown() --> for key press

Robot:

-->Robot is a predefined class present in java.awt package which is used for performing keyboard actions in a webpage. There are few methods as
-->keyPress() --> used to press a key
-->keyRelease() --> used to release a key

Select:

-->Select is a predefined class which is used to perform dropdown in a webpage. There are few methods as
-->isMultiple() --> to verify whether dropdown is multiselected will returns true in boolean value
-->selectByIndex() --> to select an option in dropdown by using index
-->selectByValue() --> to select an option in dropdown by using value
-->selectByVisibleText() --> to select an option in dropdown by using visible text
-->getAllSelectedOptions() --> to get the selected options in dropdown
-->getFirstSelectedOptions() --> to get the first selected option in dropdown
-->deSelectByValue() --> to deselect an option in dropdown by using index
-->deSelectByIndex() --> to deselect an option in dropdown by using index
-->deSelectByVisibleText() --> to deselect an option in dropdown by using index
-->deSelectAll() --> to deselect all the selected option in dropdown
-->getOptions() --> to get all options present in dropdown using list

JavaScript Executor:

-->If locator is found in DOM Structure but still it shows noSuchElement Exception due to hidden elements in the webpage,
if sendkeys not working, if click not working and for Scrollup and Scrolldown, we go for JavaScript Executor. It is an Interface.
There are few methods as
-->executeScript()
-->("arguements[0].setAttribute('value')",WebelementRef) --> sendkeys
-->("arguements[0].getAttribute()") --> get the particular value
-->("arguements[0].click()",WebelementRef) --> click
-->("arguements[0].scrollIntoView(true)",WebelementRef) --> scroll down
-->("arguements[0].scrollIntoView(false)",WebelementRef) --> scroll up

TakeScreenShot:

-->It is an Interface. It is used for capturing the webpage. This captured images will defaultly stored in temporary location.
We can store the captured image in our preferred location using CopyFile() Method present in FileUtils Class.
-->Here we have method called getScreenShotAs().

WebTable:

-->All the tables present in webpage is called webtable. First we need to fetch the rows using tagname "tr" and iterate the rows and
then fetch the data using tagname "td" and then iterate the datas and pick a particular data from the webtable using if conditions and
then fetch the data using tagname "th" and then iterate the headers.
-->There are two types -->Static - Data's are fixed.
-->Dynamic - Data's keeps changing

Windows Handling:

-->Consider a webpage has multiple windows example 10 windows, if any actions performed in 2nd window it will throw noSuchElement Exception
because the control will be still in the parent window (i.e)1st window. Now if you want to switch to the second window, windows Handling is used.
-->Method Used: driver.switchToWindow();
-->We can switch window by using --> String Id, String URL, String Title and There are few methods as
-->getWindowHandle() --> to get parent window ID
-->getWindowHandles() --> to get all windows ID which returns Set<WebElements> because Set does not allows duplicate.

Alert:

-->Alert is a kind of popup window or popup message. We can't find locators for Alert, so to handle an alert we use a method called
driver.switchToAlert(); If we are not handling alert, we can't do any operations.
-->Alert has 3 types-->
-->Simple ALert --> accept();
-->Confirm Alert --> accept(); dismiss();
-->Prompt Alert --> sendKeys(); accept(); dismiss();

Frames:

-->It is webpage embedded inside a webpage.
-->If any locators are placed inside a particular frame then we need to switch in to that particular frame and need to access the locators.
-->It is mainly for security purpose. We can switch to particular frame using ID, Name, Index and WebElement Reference.
--> Methods Used: driver.switchToFrame();
-->parentFrame(); --> switch to previous frame
-->defaultContent(); --> switch to parent window

Waits:

-->If locator is found but still noSuchElement Exception is thrown due to webpage loading and wait problems, we go for Waits concept.
-->Two types --> Static Wait --> Will wait for the maximum time given though the locator is found. Example: Thread.Sleep
--> Dynamic Wait --> Will not wait for the maximum time, if locator is found it will navigate to next step. There are two types here
-->Implicit wait --> Given common for all the locators
-->Explicit wait --> Given only for a particular locators. There are two types here
-->WebDriver wait --> Given only in terms of seconds
-->Fluent wait --> Given in all time formats and handle time out exception.