**\*) Drawbacks of css** **:-**

1. backward traversing
2. duplicates
3. only text (we can’t use css)

**\*) Navigations: -**

1. . -> current project
2. / -> immediate child
3. // -> any child
4. [] -> index value starts from 1

**Que) When we use Xpath?**

* 1. when targeted element not have id name link.
  2. when targeted element only have text.
  3. when targeted element have duplicates.

**\*) Xpath has two types :-**

1. **Absolute Xpath :-**

we are going to write a xpath from root element

its lengthy

/ stands for absolute xpath

1. **Relative Xpath :-**
   * + - 1. we are going to write a xpath from any child element
         2. less complex or lengthy
         3. in real time projects we generally use relative xpath
         4. // stands for relative xpath

**\*) drawbacks of absolute Xpath :-**

* 1. index value (any element is added or removed from the structure index value gets changed)
  2. lengthy
  3. time consuming

**Eg :-**

html

body

divA[1]

inputA[1]

inputB[2]

divB[2]

inputC[1]

inputD[2]

./html/body/div[1]/input[1] ---> A

./html/body/div[1]/input[2] ---> B

./html/body/div[2]/input[1] ---> C

./html/body/div[2]/input[2] ---> D

./html/body/div[1]/input ---> AB

./html/body/div[2]/input ---> CD

./html/body/div/input[1] ---> AC

./html/body/div/input[2] ---> BD

./html/body/div/input ---> ABCD

html/body/div[1]/input[1]|html/body/div[2]/input[2] ---> AD

html/body/div[1]/input[2]|html/body/div[2]/input[1] ---> BC

**\*) Relative Xpath :-**

1. **Co-relative Xpath :-**

**Eg :-**

html

body

divA[1]

inputA[1]

inputB[2]

divB[2]

inputC[1]

inputD[2]

//div[1]/input[1] ---> A

//div[1]/input[2] ---> B

//div[2]/input[1] ---> C

//div[2]/input[2] ---> D

//div[1]/input[1]|//div[2]/input[2] ---> AD

**Drawbacks :-**

1) if any element is added or removed from the structure we need to write Xpath again. (maintenance efforts)

1. **Xpath by Attribute:-**

**Syntax: -** //tagname[@AttributeName='AttributeValue']

**Eg:-** //img[@alt='Grocery']

**Drawbacks :-**

* + 1. when targeted element source code have only text.
    2. when duplicates are there.
    3. you have to pass complete attribute value.

Instead of xpath by group index we can use and , or ,not.

1. **And**

//input[@type=’text’ and @value=’A’ ] ---> A

1. **Or**

//input[@value=’B’ or @type=’submit’]

1. **Not**

//input[@value=’A’ and not(@type=’submit’)]

1. **Xpath by text function :-**

**Syntax: -** //tagname[text()='attribute text']

**Eg:-** //div[text()='Grocery']

1. **Xpath by contains :-**

**1)Xpath by contains attribute :-**

**Syntax:-** //tagname[contains(@attaribute\_name,'partial attribute value')]

**Eg:-** //td[contains(text(),'Please identify ')]

**2)Xpath by contains text():-**

**Syntax:-** //tagname[contains(text(),'partial text value')]

**Eg:** //input[contains(@placeholder,'Enter Email ID / Username')]

1. **independent and dependent:-**

**steps:-** 1) Identify which is fixed and which is dynamic.

2) Write Xpath for fixed element.

3) Update the Xpath by backward traversing

4) Update Xpath by writing xpath for dynamic element.

**Eg :- 1)** //td[text()=’Buddhist train’] - Xpath for fixed element

//td[text()=’Buddhist train ’]/.. – Backward traversing 🡪 /..

//td[text()=’Buddhist train ’]/..//td[@data-title=’helpline’]

**Eg :- 2)** //td[text()='Tourism']/..//td[@data-title='mail']

1. **Xpath by group index:-**

**Eg:-** (//div[text()='realme 9 (Stargaze White, 128 GB)'])[1]/../..//div[@class='\_30jeq3 \_1\_WHN1']

1. **Xpath by axes :-** 
   1. Child
   2. Parent
   3. Following - axes
   4. Preceding - axes
   5. Ancestors
   6. Descendant
   7. Following - Sibling
   8. Preceding - Sibling

**\*) to find any web element on web page (eg - links)**

* 1. //a ---> to find all link
  2. (//a)[1] ---> to find first link
  3. (//a)[last()] ---> to find last link
  4. (//a)[position() mod 2 = 0] ---> to find all even links
  5. (//a)[position() mod 2 = 1] ---> to find all odd links

**\*)Methods of Webelements:-**

* 1. click() - to click on any targeted element.
  2. clear() - it will remove the text from targeted textbox.
  3. sendKyes() - it can perform clicking operation using enumeration. (Keys.ENTER)
  4. getAttribute() - if you pass the attribute name it will return its respective value.
  5. getText() - it will return text visible by it attribute.
  6. getSize() - it will return size of targeted element.(width and height).
  7. getLocation() - it will return a location (x,y axes) of targeted element.
  8. getRect() - it will return rectangular location of targeted ele (width, height x,y axis)
  9. getcssvalue() - if you pass css attribute you will get css attribute value.
  10. isEnabled() -

1. if you able to perform any action on targeted ele it returns true else false.
2. Compulsory it will have disabled word in source code.
3. Tagname must have input or button else It will always return true.
4. Default value is true
   1. isSelected() –
      * 1. if targeted element is selected (eg. Checkbox) it returns true else false.
        2. Targeted ele must be checkbox or radio button else It will always return false.
   2. isDisplayed() - if targeted element is displayed on webpage it returns true else false.
   3. submit() –
      * 1. tagname strictly should have button and type is submit .
        2. parent or ancestors should strictly have tagname called form .
        3. it will perform clicking operations same as click() but it have conditions.
   4. getTagName() - it will return tag name of respective targeted element.

**Synchronisation:-**

It is a process of matching application loading speed as well as automation scripting speed is known as synchronisation.

Automation speed is more as compare to application loading speed.

We can’t increase the speed of application loading.

We can decrease the speed of automation script using.

When you are working on multiple pages then only you required synchronisation.

1. **Implicate wait -**
   * + - 1. no need to write every time.
         2. For every 500mili seconds it searches for targeted element.
         3. We write interval in the form of seconds
         4. Applicable only for find elements and find element
         5. NoSuchElementException
         6. Time interval units are miliseconds,nanosec.

Findelement ()

Findelements()

Element is present no is the time over no wait for 500ms

In html.

polling period/ Wait freq/

Yes Yes timeInterval

Web element NoSuchElementException

Or

List<web element>

1. **Explicate wait -**
   * + - 1. if condition is satisfied returns true and other lines are executed.
         2. If condition is not satisfied returns TimeOutException and other lines are not executed.
         3. Applicable for other then find elements and find element.
         4. TimeOutException.
         5. Time interval unit is seconds.

Conditions is is wait for 500

Or condition condition milliseconds

Predicates satisfied satisfied

The control will be released TimeOutException

To next line of code. with appropriate msg.

1. **Thread.sleep -** also called as blind wait.
2. **Fluent wait -** concept of adv selenium
3. **Pageload timeout-**

**\*) TakeScreenShoot:-**

Commansio.jar is required to create a folder automatically.

**\*) For Each loop :-**

**Syntax-** for (datatype reference variable : collection)

{

Statements;

}

**\*) Select Class:-**

Must have tag-name as select.

1. Boolean isMultiple() :-

when targeted element is a multi-list dropdown then it returns true else returns false.

1. List<WebElement> getOptions() :-

It is used to fetch all the options present in dropdown.

1. selectByVisibleText() :-

It will select the option from dropdown on the basis of visible text.

1. selectByIndex():-

It will select the option from dropdown on the basis of index value.

Index value starts from 0.

1. selectByValue():-

It will select an option from dropdown on the basis of Value of value attribute.

1. getFirstSelectedOption():-

When you select multiple options from multi list dropdown getFirstSelectedOption will return firstly selected option.

If we use this method with single list dropdown it will return default value selected in dropdown.

This method is used in multi list dropdown.

**\*) Steps to create Maven Project :-**

File->new->project-> select a wizard (Maven folder)->select maven project->click on next->click on checkbox (create a simple project)->click on next-> enter group id and artifact id -> click on finish.

**\*) Actions Class :-**

Create object of actions class

Actions a= new Actions(driver)

**\*) Methods of Action Class:-**

1. moveToElement(Web Element)
2. contextClick(Web Element target) or right click
3. contextClick()
4. doubleClick(Web Element target)
5. doubleClick()
6. Click()
7. click(Web Element target)
8. click&hold()
9. click&hold(Web Element target)
10. release()
11. release(Web Element Target)
12. moveToElement(target,x-axis,y-axis)
13. Drag&drop()
14. moveByOffset(x-axis,y-axis)
15. SendKeys(char sequence)
16. SendKyes(target,char sequence)
17. Build()
18. Perform()
19. keyUp()
20. keyDown()
21. drag&dropBy()

**\*) JavaScript Executers:-**

**\*) popUps :-**

they are the gui window which appears on the top of the another window

there are some types of popups –

1. javascript popup
2. hidden division
3. file upload
4. control focus
5. control click
6. controlSetText
7. file download
8. notification
9. authentication
   1. JavaScript PopUp:-
      1. Alert popup – Have only ok button
      2. confirmation popup- have ok as well as cancel button
      3. prompt popup – have ok , cancel as well as text field.

**\*** **How to handle the popUps:-**

* + 1. Inspectable
    2. Selenium
    3. Programming Languages
    4. Tools
    5. Try to avoid popups

**\*) Find Elements:-**

**\*Maven repository is global repository**

**\*M2 repository is local repository**

**\*) TestNG (test next generation )**

1. Unit test tool
2. Junit and nunit combination
3. TDD framework- test driven development
4. Batch Execution – to reduce the time consumption
5. Parallel execution
6. Group execution

To run the methods by groups you need to add

Before classes tag in testNG.xml

<groups>

<run>

<include name="task">

</include>

</run>

</groups>

1. Cross browser execution

Add these tags after test tag in testNG.xml

<parameter name="Browser" value="Edge"></parameter>

<classes>

<class name="parallel.Test3"></class>

</classes>

Eg : for multiple Browsers :-

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">

<suite name="Suite">

<test thread-count="5" name="Test1">

<parameter name="Browser" value="Chrome"></parameter>

<classes>

<class name="parallel.Test3"></class>

</classes>

</test> <!-- Test -->

<test thread-count="5" name="Test2">

<parameter name="Browser" value="FireFox"></parameter>

<classes>

<class name="parallel.Test3"></class>

</classes>

</test> <!-- Test -->

<test thread-count="5" name="Test3">

<parameter name="Browser" value="Edge"></parameter>

<classes>

<class name="parallel.Test3"></class>

</classes>

</test> <!-- Test -->

</suite> <!-- Suite -->

1. HTML

**Methods Of TestNG :-**

1. @Test – main method
2. @parameter
3. @Before Method
4. @Before class
5. @Before suit (X)
6. @Before test (X)
7. @After class
8. @After test (X)
9. @After suit (X)
10. @After method
11. @data provide

**\*) Assertions :-**

1. **Hard Assert**
2. Condition mandatory
3. Static methods
4. If condition is not satisfied all the next lines are skipped
5. If one Method is failed next will pass
6. **Soft Asserts**
7. Condition is satisfied or not all the lines are get executed
8. Used in Non-Mandatory field
9. It includes only Non-static methods
10. If one method is failed then all the next methods will fail

**\*) POM – Page Object Model/ Object Repository/ Element Repository :-**

1. **Advantages:-**
2. It avoids StaleElementReferenceException
3. Reusability of code
4. Readability of code
5. Reduce maintenance efforts
6. **What is pom ?**

It is a design pattern which gives solutions for frequently occurring problems in automation testing process.

1. **Steps :-**

**Step1-** Create separate class for all pages and in suffix page login page , home page.\

**Step2-** Pass all the credentials in respective java class.

**Step3-** Make element as private and call getter method.

**Step4-** call @FindBy (@FindBy(id=”xyz”)) to find element to avoid stalereferenceException

**Step5-** Call Pagefactory is a class includes initelement() used to initialise @findBy()

**Step6-** Call the Base class

**Step7-** Create Action Method

**Step8-** It will take you to the next landing page.

**\*) Maven :-**

Maven is build management tool

Maven has two types plugins or software are eclipse plugins and command line plugins Eclips plugins are inbuilt plugins

Dependency is a jar file which downloads the data or tool from global repository (maven repository) to local repository (.m2 folder)

**Advantages of Maven :-**

* 1. It provides folder structure
  2. It supports advanced tools like Jenkins and testNG.
  3. It supports git and git hub
  4. Easily we can add the dependencies
  5. It continuously integrates with your frameworks.