**1.What are locators, different types of locators that can be used in Selenium and their priorities?**

**Answer:** Locators provide a way to access the HTML elements from a web page. In Selenium, we can use locators to perform actions on the text boxes, links, checkboxes and other web elements. They are the basic building blocks of a web page.

**Locator Types:**

Selenium doesn’t have any inbuilt capability or mechanism to locate any UI elements on the Web Pages. In order to find UI elements on the Web Page, Selenium has to take the help of Locators.We have 8 types of locators in Selenium:

* id
* name
* className
* linkText
* partialLinkText
* tagName
* cssSelector
* xapth

Locators can be classified into two categories:

Structure-based locators: locators that rely on the structure of the page to find elements.

      --XPath

      --CSS

Attributes-based locators: locators that relies on the attributes of the elements to locate them

      --Id

      --Name

      --Link

      --CSS

Please refer below link to get more details about locators  
<https://www.automationtestinginsider.com/2019/09/selenium-locators-and-their-priorities.html>

**2. What is the difference between ‘/’ and ‘//’ in XPath?**

**Answer:**

Single Slash “/” – Single slash is used to create Xpath with absolute path i.e. the xpath would be created to start selection from the document node/start node.

Double Slash “//” – Double slash is used to create Xpath with relative path i.e. the xpath would be created to start selection from anywhere within the document.

**3. What is an XPath?**

**Answer:**In Selenium automation, if the elements are not found by the general locators like id, class, name, etc. then XPath is used to find an element on the web page.

**XPath** (XML Path Language) is a query language for selecting nodes from an XML document. In addition, XPath may be used to compute values (e.g., strings, numbers, or Boolean values) from the content of an XML document.

**Some important points about xpath:**

It is slowest among all locators. But it provides you reliable ways to locate web elements.

**Syntax**- Xpath=//tagname[@attribute='value']

**Example**- //input[@type='text']

**Pros**- Allows very precise locators

**Cons**- XPath engines are different in each browser, hence make them inconsistent across browsers. That means if you write XPath for your application in Chrome browser, it may not work on IE.

**4. What is the difference between Absolute and Relative XPath? Give examples?**

**Answer:**

**Absolute XPath** :- It starts with the root node or a forward slash (/). The advantage of using absolute is, it identifies the element very fast.

Disadvantage is, if any thing goes wrong like some other tag added or removed in between, then this path will no longer works.

**Example:**

If the Path we defined as

1. html/head/body/table/tbody/tr/th

If there is a tag that has added between body and table as below

2. html/head/body/form/table/tbody/tr/th

The first path will not work as ‘form’ tag added in between

**Relative Xpath:**- A relative xpath is one where the path starts from the node of your choise – it doesn’t need to start from the root node.

It starts with Double forward slash(//)

**Syntax:**

//table/tbody/tr/th

Advantage of using relative xpath is, you don’t need to mention the long xpath, you can start from the middle or in between.

Disadvantage here is, it will take more time in identifying the element as we specify the partial path not (exact path).

If there are multiple elements for the same path, it will select the first element that is identified.

**5. What is the disadvantage of Absolute XPath and why is Relative XPath recommended over it?**

**Answer:** Disadvantage is, if any thing goes wrong like some other tag added or removed in between, then this path will no longer works.

**Example:**

If the Path we defined as

1. html/head/body/table/tbody/tr/th

If there is a tag that has added between body and table as below

2. html/head/body/form/table/tbody/tr/th

The first path will not work as ‘form’ tag added in between

A relative xpath is one where the path starts from the node of your choise – it doesn’t need to start from the root node.

It starts with Double forward slash(//)

**6. What is the concept that makes XPath Expressions powerful out of all the locators?**

**Answer:**XPath AXES is the concept which makes the XPath Expressions powerful out of all the locators. i.e. By using XPath AXES, we can traverse both forward and backward in the HTML code of the web pages.

**7. Why CSS Selectors have higher priority over XPath Expressions?**

**Answer:**  Below are the reasons why CSS Selectors need to be considered over XPath Expressions:

 When compared to XPath Expressions, CSS Selectors locate the UI elements faster.

 Selenium may not be able to locate few UI elements using XPath Expressions while executing the Automation scripts on Internet Explorer Browser.

CSS is faster. It also improves the performance. It is very compatible across browsers.CSS is best for IE as XPath does not work in IE always.

**8. What are the names of add-ons which can auto-generate the XPath Expressions and CSS Selectors?**

**Answer:**Below are the add-ons

Firebug – deprecated

Firepath - deprecated

MRI – for IE browser

ChroPath – Chrome, Mozilla, edge etc

**9. How to retrieve CSS Properties of an element?**

**Answer:**The values of the css properties can be retrieved using a get() method:

**Syntax:**

driver.findElement(By.id(“id“)).getCssValue(“name of css attribute”);

driver.findElement(By.id(“id“)).getCssValue(“font-size”);

**10. What is the major differences between XPath Expressions and CSS Selectors?**

**Answer:** One of the important differences between XPath and CSS is, with XPath we can search elements backward or forward in the DOM hierarchy while CSS works only in a forward direction. This means that with XPath we can locate a parent element using a child element.

**11. How can we locate an element by only partially matching its attributes value in XPath?**

**Answer:**XPath supports the contains() method. It allows partial matching of attribute’s value and thus helps when the attributes use dynamic values while having some fixed part. See the below example to understand better:

xPath :  //\*[contains(@section, 'Mobiles')]

**12. How can we locate elements using their text in XPath?**

**Answer:** You can try creating xpath using text() function like below:

//span[contains(text(),'Users')]

//span[contains(text(),'Contents')]

**13. How can we move to parent of an element using XPath?**

**Answer:**Use “//ancestor::” to move to parent node like this

Find the element of Search and the xpath is “//span[text()='Search']”

the we want to move the parent node of this node

and the xpath is “//span[text()='Search']//ancestor::ul”

**14. How can we move to nth child element using XPath?**

**Answer:** We can use below xpath like

First Child

//div[@id='rcnt']/child::div[1]

Fifth Child

//div[@id='rcnt']/child::div[5]

Where index starts from 1

**15. What is the syntax of finding elements by class using CSS Selectors?**

**Answer:**Locate by className (using symbol . dot)

      Syntax: tag.className

      Example: input.input

**16. What is the syntax of finding elements by id using CSS Selectors?**

**Answer:** Locate by id (using symbol # hash)

    Syntax: tag#id or #id

    Example: input#user\_login or #user\_login

**17. How can we select elements by their attribute value using CSS Selector?**

**Answer:** Locate by Name or Attribute or using multiple attributes:

**Syntax:**

      tagName[attributeName='value']

      [attributeName='value']

      tagName[attribute1='value'][attribute2='value']

      tagName[attribute1='value'],[attribute2='value']

**Example:** few examples given below

input[name=‘username']

input[name='log'],[id='input'],[type='text']

**18. How can we move to nth child element using CSS Selector?**

**Answer:**In CSS Selector, we have a very useful structural pseudo-class selector i.e. ‘nth-child’ selector. ‘nth-child’ can be used to select ordered elements by giving the expression (an+b) or single element with positive integer as an argument. Consider if you want to get odd rows from a HTML table

nth-child (2n+1) expanded below.

(2\*0) + 1=1=1st Row

(2\*1) + 1=3=3rd Row

(2\*2) + 1=5=5th Row

Selenium WebDriver-Java code below counts no. of odd rows in a table.

int inCount=driver.findElements(By.cssSelector("table[id='tbl1'] tr:nth-child(2n+1)")).size();

The below code gets first row from a table.

WebElement firstRow=driver.findElement(By.cssSelector("table[id='tbl1'] tr:nth-child(1)"));

**19. What is XPath Axes and what are the different Axes available?**

**Answer:** XPath Axes are used to find dynamic elements

XPath AXES is the concept which makes the XPath Expressions powerful out of all the locators. i.e. By using XPath AXES, we can traverse both forward and backward in the HTML code of the web pages.

Axes methods are used to find those elements, which dynamically change on refresh or any other operations. There are few axes methods commonly used in Selenium Webdriver like child, parent, ancestor, sibling, preceding etc.

Following are the axes techniques:

* Ancestor:
* Child:
* Descendant
* Following:
* Following-sibling:
* Parent
* Preceding:
* Preceding-sibling:

Ancestor:  ancestor selects any Parent and Grandparent of the current node.

Example: //input[@id=‘user123']/ancestor::\*

Child: Selects all children elements of the current node.

Example: //div[@id='user123']/child::\*

Descendant: Descendant lets you select Children and Grandchildren of the current node.

Following: Following returns all in the document after the closing tag of the current node.

Following-sibling: Following-sibling returns all the sibling after the closing tag of the current node.

Parent: Parent returns the parent of the current node.

Preceding: Preceding returns all in the document before the current node.

Preceding-sibling: returns all the sibling before the current node.

Self: returns the current node.

**20. Write an xpath to find all the hyperlinks on a web page?**

**Answer:**  Please refer below program.

**21. What if we don’t have locators information? Is there any method to automate?**

**Answer:**  Yes, It is possible by using robot class

For example you want to click a button but you don’t have its elements information, you can simply use below code and you can click it. You can change mouse movement values (robot.mouseMove) according to your requirement.

Robot robot = new Robot ();

robot.delay(3000);

robot.mouseMove(10,200);

robot.mousePress(InputEvent.BUTTON1\_MASK);

robot.mouseRelease(InputEvent.BUTTON1\_MASK)

**22. What is the use of following-sibling ?**

**Answer:** following-sibling Selects all siblings after the current node

[Graphical user interface, text, application, email

Description automatically generated](https://1.bp.blogspot.com/-RqHQZZgVUBs/XndeYIhLYvI/AAAAAAAAAdc/Fj6YLcoGchAIWoaNYLHVWZmzhonVNsopACLcBGAsYHQ/s1600/following-sibling.webp)

In above image, ul has 8 children(8-li) on total.

And, each li has 7 more siblings(i.e brothers/sisters). Now, suppose you want to identify the 4th li where only first li can be identified uniquely. To identify the the 4th li we can proceed as follows:

 Identifying the 1st li element, via the xpath

 //li[@class='classroomlink hidden-xs hide']

 if we want the identify the 4th li then if we proceed by giving slash after it i.e

 //li[@class='classroomlink hidden-xs hide']/

 Then it will look for the elements inside the li. And so it is not possible to navigate to the 4rd li.

 Here we can make use of the following-sibling to identify our element by

 //li[@class='classroomlink hidden-xs hide']/following-sibling::li[3]

**23. We have two similar hidden elements with same attribute, how can you write xpath?**

**Answer:** I will show you how we can use some of these above functions in xpath to identify the objects.

<html>

<body>

<input type='checkbox' name='chk'>first

<br><input type='checkbox' name='chk'>second

<br><input type='checkbox' name='chk'>third

<br><input type='checkbox' name='chk'>forth

<br><input type='checkbox' name='chk'>fifth

<br><input type='checkbox' name='chk'>sixth

</body>

</html>

Select last element

xpath=(//input[@type='checkbox'][last()]

Second position

xpath=//input[@type='checkbox'][position()=2]

**24. What is the difference between Xpather and Xpath Checker?**

**Answer:** Answer will be provided soon

**25. What is the best way to locate a web element if there is no unique XPath?**

**Answer:** In case of no unique XPath available, the first alternative to be used is CSS Selectors. CSS (Cascading Style Sheets) has the following advantages over XPath :

Easier to use

Stable, i.e. does not change dynamically as XPaths do

Typically faster than XPath

Consistent support across browsers

Syntax to use CSS Selectors in Selenium Webdriver : By.cssSelector(".classXYZ")

<https://www.automationtestinginsider.com/2019/09/how-to-locate-web-element-using-css.html>

**26. In XPath, I wants to do partial match on attribute value from beginning. Tell me two functions using which I can do It?**

**Answer:** Contains() and starts-with() function in XPath is used when we are familiar with pattern of dynamically changing attribute’s value of an element on HTML pages. This not only works with dynamic values of any of the html attributes but it also works when we want to write XPath on the basis of partial pattern of any of attribute.

**27. How we can retrieve the dynamically changing Ids?**

**Answer:**  You can go for the following methods:

1.Absolute XpathPath method

This is the easiest way to solve the issue. Absolute XPath is the path starting from the root. It will be something like :

/html[1]/body[1]/nav[1]/div[1]/form[1]/div[1]/div[1]/div[1]/input[1]

But the risk with this method is, if something changes in the structure of your web page, your code will break. So this is not a recommended method.

2.Use Relative XPath using contains or starts with text

This is the preferred method for handling dynamic web elements if you observe a pattern in the attribute values like ID or Class of the web element.

For eg consider the HTML snippet:

<input type="submit" id=" submit\_234678" value="Subscribe">

This Subscribe button on the page has an ID with a dynamically changing number in it (‘234678’). This keeps on changing every time you refresh the page. But it always starts with submit. So you can use a relative XPath as given below to identify the web element:

XPath - //input[starts-with(@id, ‘submit\_’)]

Now, consider another example:

<input type="submit" id=" 1224-subscribe" value="Subscribe">

In this case you can write XPath as:

XPath - //input[contains(@id, ‘subscribe’)]

3.Identify by index

Sometimes, you will have multiple elements with same locator value. For example there may be two submit buttons with id starting with ‘Submit’. In this case you can use findElements method and locate the element using the index.

driver.findElements(By.xpath(//\*[contains(@id, ‘submit’).get(0).click();

Here get(0) is used to get the first web element with matching XPath.

4.Use Multiple attributes to locate an element

To identify a particular element you can use multiple attributes if a single attribute is not enough to identify your web element uniquely.

Xpath- //button[starts-with(@id, 'save') and contains(@class,'publish')]

**28. How will you Identify the web-element that has same property values?**

**Answer:**  Thare are some function we can use in Xpath to identify the abject in above cases.

An XPath expression can return one of four basic XPath data types:

String

 Number

 Boolean

 Node-set

XPath Type : Functions

Node set : last(), position(), count(), id(), local-name(), namespace-uri(), name()

String : string(), concat(), starts-with(), contains(), substring-before(), substring-after(), substring(), string-length(), normalize-space(), translate()

Boolean : boolean(), not(), true(), false(), lang()

Number : number(), sum(), floor(), ceiling(), round()

Let’s Consider the below HTML

<html>

<body>

<input type='checkbox' name='chkbox'>first

<br><input type='checkbox' name='chkbox'>second

<br><input type='checkbox' name='chkbox'>third

<br><input type='checkbox' name='chkbox'>forth

<br><input type='checkbox' name='chkbox'>fifth

<br><input type='checkbox' name='chkbox'>sixth

</body>

</html>

In the above html file there are six checkboxes and all are having same attributes (same type and name)

How we can select the last checkbox based on the position. We can use last() function to indentify the last object among all similar objects.

Xpath: //input[@type='checkbox'])[last()]

Below Xpath will check or uncheck the second last checkbox and thrid last checkbox respectively.

xpath: //input[@type='checkbox'])[last()-1]";

xpath: //input[@type='checkbox'])[last()-2]";

You want to select second checkbox and forth checkbox then use below xpath

xpath: //input[@type='checkbox'])[position()=2]";

xpath: //input[@type='checkbox'])[position()=4]";

**29. A xpath is working in chrome but not on Firefox how would you handle without changing xpath? (css).**

**Answer:** If the problem is that a locator which works for one browser (say Firefox or Chrome), should work for another browser (say Internet Explorer) but does not then you probably have a timing problem. The classic timing problem in Selenium is that there is ALWAYS a timing issue locating some elements but some browsers are so fast that the timing is so fast you don't see the timing requirement but other browsers are so slow you do see the timing requirement. The solution is to handling the wait for event properly. This fixes the problem with slow browsers but has no effect on fast browsers.

**30. How to find all links present under body of the page ignore the links which are present under header and footer of the page.**

**Answer:**answer will be provided soon

**31. An element has an id “bng\_123” but its number is changing. How to handle it?**

**Answer:**Please refer answer of question#27

**32. There is a submit button in page it has id property. By using “id” we got “element not found exception”, how will you handle this situation? What might be the problem in this case?**

**Answer:** id changing dynamically . Please refer answer of question#27

**33. Suppose developer changed the existing image to new image with same xpath. Is test case pass or fail?**

**Answer:**case will Pass

**34. I want to find the location of "b" in the below code, how can I find out without using xpath, name,id, csslocator, index?**

<div>

 <Button>a</button>

 <Button>b</button>

 <Button>c</button>

 </div>

**Answer:**

//button[text()=’b’]

**35. How to write xpath if all digits changing every time - ABG123FG65**

**Answer:** Please refer answer of question#27

**36. xpath and css selector which one faster?**

**Answer:** CSS selectors perform far better than Xpath and it is well documented in Selenium community. Xpath engines are different in each browser, hence make them inconsistent. IE does not have a native xpath engine, therefore selenium injects its own xpath engine for compatibility of its API.

**37. How to find Xpath of 4 elements which has the same properties?**

**Answer:** Please refer answer of question#28

**38. List the disabled element from web page**

**Answer:**

**public** **class** **DisableEleTest** {

WebDriver driver;

**@Test**

**public** **void** **disableEleTest**() {

System.setProperty("webdriver.chrome.driver",

"C:\\Users\\Hitendra\\Downloads\\chromedriver\_win32\\chromedriver.exe");

driver = **new** ChromeDriver();

driver.manage().window().maximize();

driver.get("https://www.google.com/");

List<WebElement> allElements = driver.findElements(By.xpath("//\*"));

**int** totalWebElements = allElements.size();

System.out.println("Total WebElements: " + totalWebElements);

**int** enableCount = **0**;

**for** (WebElement ele : allElements) {

**if** (ele.isEnabled()) {

enableCount++;

}

}

System.out.println("Total Disable elements: " + (totalWebElements-enableCount));

driver.close();

}

}

**Output:**

Total **WebElements:** **206**

Total Disable **elements:** **1**

**39. How to verify no of web elements on a web page**

**Answer:**

List<WebElement> allElements = driver.findElements(By.xpath("//\*"));      //Identify all the elements on web page

int elementsCount = allElements.size();     //Count the total all element on web page

**40. can we use combination of relative and absolute xpath**

**Answer:**  Yes

**41. Dynamically changing web table, how to get the last row?**

**Answer:**  
In this case we will use last() method of xpath as shown below:

// How to print data from last row

System.out.println("Directly printing column values of last row of table: ");

List columnOfLastRow= driver.findElements(By.xpath("//table[@name='Table']/tbody/tr[last()]/td"));

**for**(WebElement **e:**columnOfLastRow)

{

System.out.println(e.getText());

}