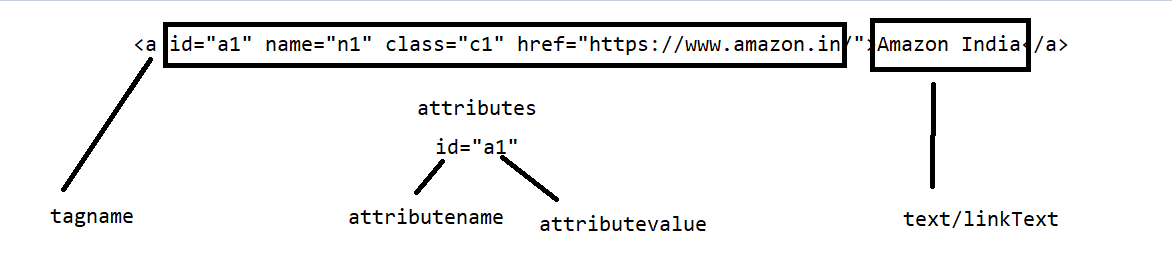
Locators in Selenium:

* Before performing any action on a web element its important that first we need to identify the element uniquely
* To identify the webelement we use locators from selenium
* There are 8 different types of locators they are

1. tag
2. id
3. name
4. class
5. linkText
6. partialLinkText
7. cssSelector
8. xpath



cssSelector:

cssSelector is one of the locators in selenium which can be used if any of the above locators are not available

|  |  |
| --- | --- |
| syntax | tagname[attributename='attributevalue'] |

xpath:



* it is a path of the element from the html tree
* there are 2 types of xpath 1) absolute xpath 2) relative xpath
* absolute xpath -> denoted by /.

it is a complete path from the root element till the element that we want to locate

/ meaning – immediate next element

ex: /html/body/table/tbody/tr/td/table/tbody/tr[1]/td/div/table/tbody/tr/td[1]/div[1]/div

* relative xpath 🡪 denoted by //.

// any element under this tag

different ways of writing xpath

1. using single attribute

|  |  |
| --- | --- |
| syntax | //tagname[@attributename='attributevalue'] |
| examples: | //div[@class='atLogoImg'] |
|  | //input[@placeholder='Username'] |
|  |  |

**Notes from 25/5/2023**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Different types of relative XPath

1) Using single attribute

syntax: //tagname[@attributename='attributevalue']

2) Using Text function

syntax: //tagname[text()='textvalue']

<div>Login </div>

//div[text()='Login ']

<a href="register.php" style="color: #0000ee;text-decoration: underline;">REGISTER</a>

//a[text()='REGISTER']

3) Using more than one attribute using AND / OR

syntax: using AND //tagname[@an='av' and @an='av']

syntax: using OR //tagname[@an='av' or @an='av']

<input type="text" name="username" value="" id="username" class="textField" placeholder="Username">

//input[@type='text' and @name='username']

//input[@placeholder='Username' or @name='username']

4) Contains function

How did you handle dynamic elements from your application

syntax: //tagname[contains(@an,'av')] --> attribute

syntax: //tagname[contains(text(),'tv')] --> text value

<b>First

Name: </b>

//b[contains(text(),'First')]

<a>inbox(10)</a>

//a[contains(text(),'inbox')]

5) Starts with

syntax: //tagname[starts-with(@an,'av')] --> attribute

syntax: //tagname[starts-with(text(),'tv')] --> text value

<b>First

Name: </b>

//b[starts-with(text(),'First')]

Example of dynamic elements possibly which you could come accross

<input id="caiosf4726572572Customer842937534cnsdnjd"></input>

<input id="caiosdsadfds6572572Customer842937534cnsdnjd"></input>

<input id="cafsd6572572Customer842fsdfsdnjd"></input>

//input[contains(@id,'Cust')]

<input id="icici84378953nfdjsnfdjsnf"></input>

<input id="icicicxhcudisfh9899797987"></input>

//input[starts-with(@id,'icici')]

other xpaths like following preceding ancestor --- so many ways to write xpath

6) Following and Grouping

sytnax: //tagname[@an='av']/following::tagname

//b[contains(text(),'First')]/following::input[1] --> only in xpath the index will start from 1 not 0

(//input[@type='text' or @name='username'])[2]