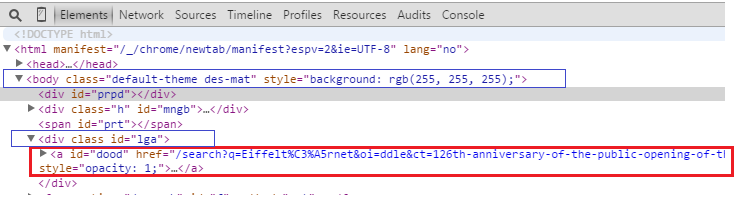
---------------------------------------------------------------------------------------------

## **How to write Xpath**

Syntax: //tagName[@attributeName=’attributeValue’ ]/childTag

## Example:



1. In the above html, if we need to identify tag “a”, xpath will be,

Xpath: **//a[@id=’dood’]**

1. In the above html, if we need to identify tag “a”, from parent tag <div class id=”lga”>, xpath will be

Xpath: **//div[@id=’lga’]/a**

1. In the above html, if we need to identify tag “a”, from parent tag <body>, xpath will be

Xpath: **//body/div[3]/a** (notice that we have not used attribute with body since its unique)

OR

Skip all intermediate nodes between body and first “a” tag.

Xpath: **//body/descendant::a[1]**

We can also use attributes of “a” as below for precise identification.

Xpath: **//body/descendant::a[@id=’dood’]**

--------------------------------------------------------------------------------------------

EX 02:

Html…

|  |
| --- |
| <div id=”divId” class=”divClassName” type=”divType”> someText </div>  <div id=”divId2”> </div>  <a href=”www.somewebpage.co.in” class=”aClass” > someLinkText <a>  </div> |

Xpath (can use any attribute – id, class…)

//div[@id=’divId’]

//div[@class=’divClassName’]

//div[@type=’divType’]

//div[text()=’someText’ ]

//\*[@id=’divId’]

Identify child element using parent tag and attribute

//div[@id=’divId’]/div[1]

//div[@id=’divId’]/div/a[1]

Refer child tag

Syntax: //parentTagName[@attributeName=’attributeValue’]/descendant::childTagName[@childAttributeName=’childAttributeValue’]

//div[@id=’divId’]/descendant::div[@id=’divId2’]

//div[@id=’divId’]/descendant::a[@class=’aClass’]

//div[@id=’divId’]/descendant::a[text()=’ someLinkText’]

Multiple child nodes with same tag name: Use Indexing, Start-Index: [1]

//div[@id=’divId’]/descendant::a[1]

Can use x: instead of descendant::

//\*[@id='lga']/x:div

//label[text()='Seaside & Country Homes']/preceding-sibling::input[@type='checkbox']

//table/descendant::span/input[@type='checkbox']

To identify partial value of attribute (used in dynamic Xpath)

//\*[starts-with(@id,'div')]

//\*[contains(@id, 'div')]

One or more nodes with text Model

//ul[@class='featureList' and ./li[contains(.,'Model')]]

Using Two Attribute

Syntax: //tagName[@attributeName1=’attributeValue1’ and @attributeName2=’attributeValue2’ ]

//input[@id='radOnewayOrReturnTrip' and @value='O']

//\*[@id='radOnewayOrReturnTrip' and @value='O']

Usage

*driver*.findElement(By.xpath("//div[@id='ires']/descendant::a[1]"))

--------------------------------------------------------------------------------------------

## **How to write cssSelectors:**

EX:

Html…

<div id=”divId” class=”divClassName” type=”divType”> someText </div>

<div id=”divId2”> </div>

<a href=”www.somewebpage.co.in” class=”aClass” > someLinkText </div>

To identify element using class name:

Syntax: tagName.ClassName OR .ClassName,

.divClassName

div.divClassName

To identify element using id:-

Syntax: tagName#idValue OR #idValue,

#divClass

div#divClass

To identify element using attribute:-

Syntax: tagName[attributeName=’attributeValue’] OR [attributeName=’attributeValue’]

[type=’divType’]

div[type=’divType’]

To identify element using Partial attribute value:-

For attribute value starts-With:

Syntax: tagName[attributeName^=’attributeValue’]

div[type^=’div’]

For attribute value ends-With:

Syntax: tagName[attributeName$=’attributeValue’]

div[type$=’Type’]

For attribute value has-Sub-String:

Syntax: tagName[attributeName\*=’attributeValue’]

div[type\*=’Ty’]

Usage:

*driver*.findElement(By.cssSelector(".divClass"));

*driver*.findElement(By.cssSelector("div#divId"));

*driver*.findElement(By.cssSelector("#divId"));

*driver*.findElement(By.cssSelector("div[type=’divType’]"));

*driver*.findElement(By.cssSelector("div[type^=’div’]"));

---------------------------------------------------------------------------------------------