22/02/2017

CSS Selector using Firebug & Firepath add on:

To check css expression is correct or not, we use firebug & firepath add on of firefox browser.
To Install it, go to Tools → addon's → extensions → search firebug → click install button of firebug → search firepath → click install button of firepath → Restart the browser.

Open the required page in firefox browser, press F12 (firebug), click on firepath, select css, type the css expression 'a.c1'. It will highlight the matching element.

In Google Chrome, press F12, opens the developer tool bar, then press Ctrl + F5, then type the css expression. It will highlight the source code of matching element.

Using the xpath locator.

It is the path of the element in HTML tree.

Ex: Sample HTML Tree :-

document
↳ Html
↳ Head
↳ Body
↳ a

"document" is the root element of the HTML ~~too~~ tree. while specifying the path we use single forward slash (/) which represents immediate child element.

→ Example of xpath present in HTML Tree.
"/Html/Body/a."

using xpath in Selenium:

driver.findElement(By.xpath("/Html/body/a")).click();

Ex: 1 Html
1↳ Body
↳1 div
1↳ Input (A)
2↳ Input (B)
↳2 div
1↳ Input (C)
2↳ Input (D)

} Refer Demo-Html.

In xpath, we can use Index. It starts from 1 (one).
If we do not specify the index, it represents all the elements which are having specified tag.

Eg: /html/body/div - It matches with all the divisions present on the page i.e. 2 (two)

/html/body/div[2] - It matches with second division on the page.

In the HTML tree, if the element is only one, then specifying the index as 1 (one) or without specifying the index represents the same element.

Ex: /html = /html[1]

Absolute xpath: Specifying complete path of the element is called as absolute xpath.

	Xpath	element
1)	/html/body/div/input	A B C D
2)	/html/body/div[1]/input	A B
3)	/html/body/div[1]/input[1]	A
4)	/html/body/div/input[1]	A C
5)	/html/body/div/input[2]	B D
6)	/html/body/div[1]/input[1] | /html/body/div[2]/input[2]	A D

NOTE: To combine 2 xpath expression, we use pipe (|).

__/__/____

Relative xpath: Writing absolute xpath on real time application will be very difficult, because path will be very lengthy. In order to reduce length of expression, we use relative xpath. To write relative xpath we use one of the xpath axes called 'descendant' which represents any child element. xpath axes will have following syntax:

/axes :: tag.

Ex: /descendant :: input. which matches with all the inputs, which are descending from root element, nothing but all the 'inputs' present anywhere in the web page. and this is similar to following absolute xpath - /html/body/div/input.

'descendant' is very frequently used axes and it has a shortcut '//' (double slash)

	Relative xpath	Element
1)	//input	A B C D
2)	//div[1]/input	A B
3)	//div[1]/input[1]	A
4)	//input[1]	A C
5)	//input[2]	B D
6)	//div[1]/input[1] | //div[2]/input[2]	A D

NOTE: xpath is broadly categorized into 2 types:

1) Absolute xpath
2) Relative xpath.

I·Q: Derive an xpath which matches with all the images and all the links present on the webpage.

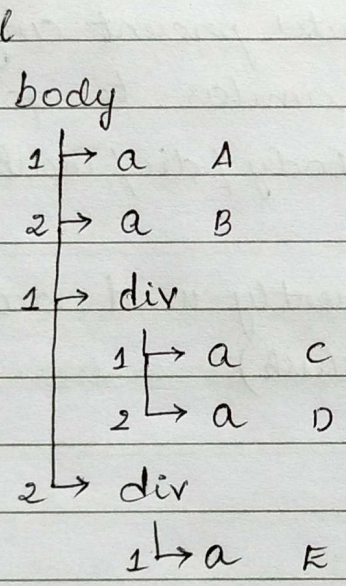
//img | //a

I·Q: What is the difference b/w '//a' and '//table//a'

//a → represents/matches with all the links present anywhere in the webpage.

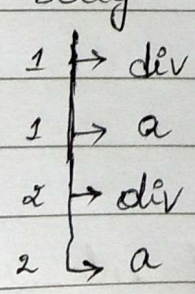
whereas //table//a → matches with all the links present in all the tables

Ex 1:



* //a → 5 → A B C D E
* //div//a → 3 → C D E
* //a[1] → 3 → A C E
* //div//a[1] → 2 → C E
* //div//a[2] → 1 → D

Ex 2:



Xpath by attributes while writing xpath expression, we can include attributes of the element and the syntax is

tag[@Attribute name = 'Attribute Value']

Ex:1 for /Html/body/div/input[@value = 'A'] → Absolute xpath

//input[@value='A'] → Relative xpath.

Ex2: //input[@id = 'Email']

//input[@name = 'SignIn' and @value = 'Next']

In xpath, we can include more than one attribute using 'and' & 'or'.