- starts-with(@<attr>,'')

- ends-with(@<attr>,'')

- contains(text(), '')

- contains(@<attr>,'')

- last()

- hierarchy

/

//

/..

following-sibling::<tagname>

preceding-sibling::<tagname>

- what is absolute xpath and what is the disadvantage and advantage

- what is relative xpath

- what is partial xpath

- what exceptions we get if element is not found.

- xpath

--------

- refer all tags

- refer particular tag with specific attribute and value

- refer 2nd input tag

- refer particular tag with specific attribute

- refer last "li" tag of "ul" tag

- refer input tags whose attribute's value starts with "em"

- refer input tags whose attribute's value ends with "il"

- refer tags whose attribute's value contains "mi"

- refer tags, that contains text "ar"

- refer tags, that contains text "ar", but at last refer 2nd element

- refer input tag, whereever they are in div->form

- refer input tag, whereever they are in div->form

- refer input tag, whereever they are in div->form, after than extract siblings before and after like "button"

- refer parent tag of input tag

**XPATH**

xpath starts with ***//<tagname>[@<attribute>='value']***

The idea behind writing xpath is to get 1 matching node, means unique.

**//\*** = here \* means refering every tag in html and you will get count of number of tags presented on html.

***//input*** = It will select all input tags.

**//input[2]** = second input box from top to bottom html page

**//input[@id]** = element with "id" attribute

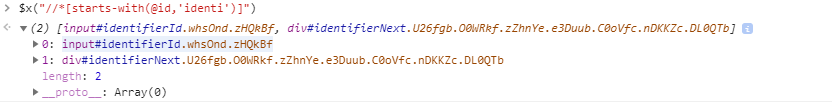
|  |
| --- |
|  |

***//input[@id='Email']*** = it will select that input box, whose id is email.

***//input[@id='Email'][@type='email']*** = (concatenating attributes)

***//ul[@id=’list’]/li[last()]*** = Last Child

**.//input[starts-with(@id,'Ema')]** for refering **email123uqi**

****

***//input[ends-with(@name,’name’)]*** = Attribute Ends With

**$x("//\*[@id='view\_container']/div/div/div[2]/div/div[1]/div/form/content") =** refer direct child elements like "content" of parent "form".

**$x("//\*[@id='view\_container']/div/div/div[2]/div/div[1]/div/form//content") =** refer direct and indirect child "content" elements of parent "form". "see // at the end".

**.//input[contains(@id,'Emai')]** for refering **13123email234yuY**

**.//input[@type='checkbox'][@value='soccer']** = all checkboxes having value soccer

**//\*[contains(text(),'rajan')]** = find those html elements, that contains "rajan" as text

**//\*([contains(text(),'rajan')])[2]** = find those html elements, that contains "rajan" as text and refer 2nd element only.

**//div[@class='search']/form/fieldset/input[2]** = refer div, which are applied with "search" class, that has form, field and inputs tag. Way for getting 2nd input tag

**//div[@class='search]/form/fieldset/input[2]/../..** = of above used xpath, how to locate to parent elements

**$x("//div[@class='search-container']/form/fieldset/input[2]/following-sibling::button")** = the hierarchy in which input's element exist, extract "button" elements from same hierarchy

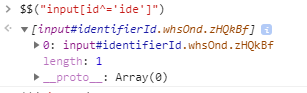
**$x("//div[@class='search-container']/form/fieldset/input[2]/preceding-sibling::input")** = extract all input elements coming before input[2] (in same hierarchy)

**CSS Path**

1. <tagname>#<id value> = Access particular web element by ID value.
2. #<Id value> = Searh element that has particular id.
3. <element name> = Search element
4. .<class name> = Search element that has particular class information
5. <element name>.<class name>[<attribute name>='<value'>] = Search web element label having class name and attribute value.
6. .<class name> > div = forward in hierachy and accessing div.
7. .<class name> > div:first-child = access first child of div in hierarchy
8. .<class name> > div:last-child = access first child of div in hierarchy
9. .<class name> > div:nth-child(2) = accessing second child of div in hierarchy

**How to make CSS selector of web element, that has dynamic ID**

1. input[id^='Ema'] = this means starts with Ema
2. input[id$='Ema'] = this means ends with Ema
3. input[id\*='Ema'] = this means contains Ema



**if class name contains spaces, this means for every space, there is new class, then what is the way to access web element having class name like that**

For eg. ABC XYZ QWE

CSS Path will be ABC.XYZ.QWE

**How to make unique XPATH (get 1 matching node), if we have two web elements which is same from all aspects**

In that case use herirachy concept, so in heiracrchy, there will be any parent element who holds unique id

*.//div[@id='login']/form[@id='load\_form']/div/div[2]/input*

**what is absolute xpath**

1. **if element has "id" then firebug by default generates relative path** otherwise absolute path.
2. absolute path will give incorrect information, if developer after bug fix, changes the location of web element.
3. aboslute can be amended but relative path does not.
4. if xpath starts with html till the location of element is absolute path, otherwise it is known as relative xpath
5. When we get absolute xpath, it is always unique
6. absolute xpath always starts with HTML like html/body and so on

**what is partial xpath**

1. **Breaking absolute xpath from the beginning is partial path**
2. Sometime we break until we get 1 matching node.

**Can we have xpath which is combination of partial xpath and relative xpath**

1. yes, for eg //div[1]/div/div/input[@id='Email']
2. Go on gmail
3. locate email textbox
4. make settings so that u can get absolute path
5. tweak it, to make the xpath smaller or until u get 1 matching node ( This will provide partial xpath)
6. At the end specify id information in input part
7. Now you will get xpath, which is combination of partial xpath and relative xpath.

**How to get xpath/CSS Selector in chrome**

right click on element whose xpath u want

click on inspect

a window is appeared and ur element's html is highlighed (elements tab)

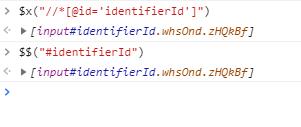
right click => copy => copy xpath / copy Selector

move to console tab ( to check wheather xapth is working or not )

for xpath = use $x("") = in double quotes paste the xpath = if xpath contains double quotes replace it with single quote.

for CSS Selector = use $$("") = in double quotes paste the selector = if selector contains double quotes replace it with single quote.

ctrl + L = to clean the console window if required



**XAPTH and CSS**

1. In css, we can move only forward in hierachy not backward, but in XPATH, we can move forward and backward.
2. as compared to CSS, XPATH are bit slow, when we are working in IE, but rest on other browsers, we will not find slowness.
3. In CSS, XPATH is not started with forward slashes and there is no use of @ symbol for accessing attribute of web element
4. What we do to generate XPATH, that we can do generate CSS Path but keep in mind 3rd point.
5. In chrome

1) XPATH = $x("<XPath>")

2) CSS = $$("<CSS PATH>")

**how will you get to know, wheather the element is dynamic or not**

inspect the element and use it's locator info in code.

then run the code, if you will get exception like "no such element exception" or "element not traceable". this means it's locator's info is generating differently at run time, so inspect the element at run time, u will see, that locator's info is completely changed.