1. //\*[@Attribute = ‘value’]
2. //tag [@Attribute = ‘value’]
3. //tag [@Attribute1 = ‘value1’ and @Attribute2 = ‘value2’]
4. //tag [text () = ‘value’]
5. //tag [text () = ‘value’ and @Attribute = ‘value’]
6. //tag [@ChildAttribute = ‘value’] /.. (To call a parent) (when a child is unique)
7. //tag [@ParentAttribute = ‘value’] /\* (To call a child) (when a parent is unique)
8. //tag [@ParentAttribute = ‘value’] / tag [condition] (To call a child) (when a parent is unique)
9. //tag [@ParentAttribute = ‘value’] // tag [condition] (To call a descendant child)
10. (//tag [@Attribute = ‘value’])/ tag [number] (To call a child)
11. //a[span[@id=’link’] – (Using child attribute to call a parent if having a single child)
12. //a [. //span[@id=’link’] – (Using child attribute to call a parent if having multiple child’s or descendants)
13. //tag [contains (text (), ‘value’)]
14. //tag [contains (Attribute, ‘value’)]
15. //tag [contains (Attribute, ‘value’) and text () =’value’ and @Attribute =’value’]
16. //tag [starts-with (Attribute, ‘value’)]
17. //tag [contains (text (), ‘value’)] // tr[2] --(To Select a row in a table, here we are selecting 2nd row]

Or we can use: //tag [contains (text (), ‘value’)] // tr[position() = 2]

1. //tag [contains (text (), ‘value’)] // tr[last()] – (To always select a last row in a table where no of rows are not fixed)
2. //tag [contains (text (), ‘value’)] // tr[last()-1] (To always select second last row in a table where no of rows are not fixed, we can do it likewise)
3. //tag [count(.//tr)] –(to count total no of rows)
4. //tag [count(.//tag(Attribute, ‘value’))] –[to count total no of respective element

**Function wise XPaths:**

**Normalize-space:** When you want to remove before and after space of any word you can use it

Ex: //p [normalize-space(text()) = ‘Tommy’]

//p [normalize-space(@name or any Attribute) = ‘aa’]

**transalate:** When you want to convert upper case to lower case characters and vice – versa

**//tagname[transalate(String,string1,string2)]**

Ex: //p [transalate(text(),’’ABCDEFGHIJKLMNOPQRSTUVWXYZ,’abcdefghijklmnopqrstuvwxyz’)]= ‘tommy’]

Note: What if we want to remove spaces and convert upper case to lower cases

//p [normalize-space(transalate(text(),’’ABCDEFGHIJKLMNOPQRSTUVWXYZ,’abcdefghijklmnopqrstuvwxyz’))]= ‘tommy’]

**String-length:** When you want to check how many strings for no of characters

Ex: //p [string-length(text()) >30] (here 30 are no of characters)

**round:** When you want to check a number with its round off digits

Ex: //p [round (text()) =54]

**floor:** When you want to check a number with its lower off digits

Ex: //p [floor (text()) =53]

**not:** When you want to exclude something

Ex: //input[@type=’Radio’ and not (@id = ‘gender-0’)]

**substring:** When you want to locate something before or after a certain character

Ex: //input[substring-after(text(), ‘:’) = ‘10am’]



**Parent :**

//div[@id=’Y2’]/ parent :: div

Or

//div[@id=’Y2’]/ parent :: \*

Or

//div[@id=’Y2’]//\* [Preferred]

**ancestor :**

//div[@id=’Y2’]/ ancestor :: div[@id =’A’]

**child :**

//div[@id=’B2’]/ child :: \*

Or

//div[@id=’Y2’]/\*

Or

//div[@id=’Y2’] /div[2]

**descendant :**

//div[@id=’B2’]/ descendant :: \*

Or

//div[@id=’Y2’]//\*

Or

//div[@id=’Y2’] //div[@id = ‘M2’] [Preferred]

Or

//div[@id=’B2’]/ descendant :: div[@id = ‘M2’]

**following :** Which ever element comes after the located one that will coming there

//div[@id=’B2’]/ following :: \*

**following-sibling:** Which ever sibling element comes after the located one that will coming there

//div[@id=’B2’]/ following -sibling :: \*

**preceding :** Which ever element comes before the located one that will coming there

//div[@id=’B2’]/ preceding:: \*

**preceding-sibling:** Which ever sibling element comes before the located one that will coming there

//div[@id=’B2’]/ preceding-sibling :: \*

Reference Links: <https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbURReU40TldGZmFjQTBEVnFETGNnR0JvTklsQXxBQ3Jtc0tsUVJQQlZGUU54SlViT0hHa1prekdGMkg3cjE4bWpMZERlYloyenNlSmtmSFhQdXVSN0dIcHZFLXdoY1d6LUk1dm5yOF9RVDJzdkJPWVU0RDdxdm1SR3d1WnlDSERDcmpyenBaeVFOOVZuejNaa3lESQ&q=https%3A%2F%2Ftheautomationzone.blogspot.com%2F2020%2F07%2Fsample-webtable-3.html&v=NhG__BL8zFo>