

XPATH in Selenium

-XPATH means Xml path.

-A query language to search web elements on web page.

2 types of XPATHs:

1. Absolute XPATH

- Absolute XPATH is a path from root of the DOM (HTML tag) to the element that we want to locate.

- E.g `/html/body/div[@class='container']/div[@id='main']/div[@class='LeftPanel']`

- Absolute XPATH always starts from single slash (/)

- Absolute XPATHs are fast in execution

- Absolute XPATHs are not recommended where your project is not consistent and changing frequently.

2. Relative XPATH:

- Relative XPATH is a path immediate to element that we want to search.

- E.g. `//a[@id='123']`

- Relative XPATH always starts from double slash (//)

- Relative XPATHs are slow in execution compared to Absolute XPATH.

- Relative XPATHs are recommended where your project is unstable.

Functions in XPATH:

1. contains():

This function can be used to locate an element by partial matching of text or attributes.

E.g for attribute:

`//tag-name[contains(@attrib,'value to search')]`

E.g for text:

`//div[contains(text(),'Apply Online')]`

2. starts-with():

This function can be used to locate an element by its starting value of attribute or text.

`//input[starts-with(@value,'Log')]`

`//input[starts-with(text(),'Login')]`

3. text():

This function can be used to locate an element with its text.

This function looks for exact match of the text. It will get failed even if cases are not matching.

e.g. `//tag-name[text()='Text of Web Element']`

4. last():

This function can be used to locate the last specified element of the DOM.

E.g. `//input[@attrib='value'][last()]`

5. position():

This function can be used to locate the element by its position.

E.g. `//input[@attrib='value'][position()=2]`

`//tag-name[index].`

E.g `//tr[1]/td[2]`

XPATH Axes:

When application under test has html elements where there tags names, list of attributes and value of attributes, everything is same. In such situations it becomes difficult to locate them by relative xpath.

We can find such elements by reading their relationships with other html elements. These relationships are called as XPATH Axes.

XPATH Axes mention relationship between html tags.

1. Parent
2. Child
3. following-sibling
4. preceding-sibling
5. ancestor
6. descendant
7. self
8. ancestor-or-self
9. descendant-or-self

Syntax:

`//<tagA>[@attrib='value']/<axes>::<tagB>[@attrib='valueB']`

`//<tagA>[@attrib='valueA']/parent::<tagB>[@attrib='valueB']`

-This line is telling that **tagB** is parent of **tagA**.

`//<tagA>[@attrib='value']/child::<tagB>[@attrib='valueB']`

tagB is child of tagA.