**What is XPath**

XPath is defined as **XML path**. **It is a syntax or language for finding any element on the web page using XML path expression**.

Why X path?

XPath is used to find the location of any element on a webpage using HTML DOM structure

Types of X path

1. Absolute X path
2. Relative X path

Absolute X path

Its begins with single forward slash (/).

select the element from the root node and ends with destination node

disadvantage of the absolute XPath is that if there are any changes made in the path of the element then that XPath gets failed

Example

html/body/div[1]/section/div[1]/div/div/div/div[1]/div/div/div/div/div[3]/div[1]/div/h4[1]/b

**Relative xpath:**

For Relative Xpath the path starts from the middle of the HTML DOM structure. It starts with the double forward slash (//), which means it can search the element anywhere at the webpage.

**// -denotes with current tag**

**Tagname/attaribute/attribute name/text**

**Attribute based Relative XPath:**

XPath expression select nodes or list of nodes on the basis of attributes like **ID , Name, Classname**, etc.

**//tag name[@attribute=’ attribute value’]**

**Partial Attribute based Relative XPath:**

**Contains**:

It is used when the value of any attribute changes dynamically

**//tag name[contains(@attribute,’ attribute value’)]**

**//tag name[contains(text(),’ text value’)]**

**Starts-with**

It is used when the starting value is static and ending value is changing dynamically

**//tag name[starts-with(@attribute,’ attribute value’)]**

**//tag name[starts with(text(),’ text value’)]**

**Ends-with**

It is used when the starting value is changing dynamically and ending value is static

**//tag name[ends-with(@attribute,’ attribute value’)]**

**//tag name[ends with(text(),’ text value’)]**

**Using OR & AND:**

**//tag name[@attribute1=’attribute1 value’ OR @attribute2=’attribute2 value]**

**//tag name[@attribute1=’attribute1 value’ AND @attribute2=’attribute2 value]**

**Text Based Relative X path**

find the element with exact text

**//tag name[text()=’ text value’]**

**Following:**

**Used My parent sibling or My parent sibling child**

**//tag name[text()=’ text value’]/following:: tag name**

**Tag name should be parent sibling or parent sibling child tag name downwords**

**Following-sibling:**

**Used for same sibling**

**//tag name[text()=’ text value’]/following - sibling:: tag name(same sibling downwords)**

**preceding:**

**Used My parent sibling or My parent sibling child**

**//tag name[text()=’ text value’]/preceding:: tag name**

**Tag name should be parent sibling or parent sibling child tag name upwords**

**preceding-sibling:**

**Used for same sibling**

**//tag name[text()=’ text value’]/ preceding -sibling:: tag name(same sibling up words)**

**Select from list of same locator**

**(//tag name[@attribute=’ attribute value’])[1]**

**(//tag name[@attribute=’ attribute value’])[2]**

**Goto parent**

**//tag name[@attribute=’ attribute value’]/parent::\***

****