XPATH by Axes functions

- Xpath axes search different nodes in XML document from current node.
- Xpath axes are the methods used to find dynamic elements.
- Axes methods are used to find those elements, which dynamically change on refresh or any other operations.
- Following are the few axes methods commonly used in selenium.

➔ Following
➔ Preceding
➔ Ancestor
➔ Parent
➔ Child
➔ Descendant
➔ Following-sibling
➔ Preceding-sibling

HTML Code:

```
<html><head>
   <title>Qspiders</title>
</head>
  <body style="font-family:arial">
    <ul id="parent">
      <li id="One"><a href="" class="listLink"><span class="position">1</span>One</a></li>
      <li id="Two"><a href="" class="listLink"><span class="position">2</span>Two</a></li>
      <li id="Three"><a href="" class="listLink"><span class="position">3</span>Three</a>
        <ul>
            <li id="A"><a href="" class="listLink"><span class="position">1</span>A</a></li>
            <li id="B"><a href="" class="listLink"><span class="position">2</span>B</a></li>
            <li id="C"><a href="" class="listLink"><span class="position">3</span>C</a></li>
            <li id="D"><a href="" class="listLink"><span class="position">4</span>D</a></li>
            <li id="E"><a href="" class="listLink"><span class="position">5</span>E</a></li>
            <li id="F"><a href="" class="listLink"><span class="position">6</span>F</a></li>
        </ul>
    </li>
    <li id="Four"><a href="" class="listLink"><span class="position">4</span>Four</a></li>
    <li id="Five"><a href="" class="listLink"><span class="position">5</span>Five</a></li>
    <li id="Six"><a href="" class="listLink"><span class="position">6</span>Six</a></li>
    </ul>
    </body>
</html>
```

Following

- This function selects all the following elements in the document of the current node (element).

Syntax:

```
//Current-element-expression//following::TagName
```

Example:

```
//a[contains(text(),'One')]//following::a
```

Preceding

- Selects all the elements that come before the current element.

Syntax:

```
//Current_Element_Expression//preceding::Tag_Name
```

Example:

```
//a[contains(text(),'Six')]//preceding::a
```

Ancestor

- The ancestor function is used to identify all the ancestor elements (Parent, Grand Parent, and Great Grand Parent) of the current element.

Syntax:

```
//Current_Element_Expression//ancestor::Tag_Name
```

Example:

```
//a[contains(text(),'One')]//ancestor::ul
```

Parent

- Selects the parent of the current element.

Syntax:

```
//Current_Element_Expression//parent::Tag_Na
```

Example:

```
//a[contains(text(),'One')]//parent::li
```

Child

- Selects all the child (Direct) elements of the current element.

Syntax:

```
//Current_Element_Expression//child::Tag_Name
```

Example:

```
//li[@id='One']//child::a
```

Descendant

- Selects the all the child's (Descendants) of the current element.

Syntax:

```
//Current_Element_Expression//child::Tag_Name
```

Example:

```
//li[@id='Three']//descendant::a
```

XPATH by following-sibling

- The following-sibling axis function used to identify the following sibling of the context node.

Syntax:

```
//XpathExpression/following-sibling::SiblingTagNameame
```

Example:

```
//input[@id='username']/following-sibling::input
```

XPATH by preceding-sibling

- The preceding-sibling axis function used to identify the preceding sibling of the context node.

Syntax:

```
//XpathExpression/preceding-sibling::tagName
```

Example:

```
//input[@id='password']/preceding-sibling::input
```

XPATH by group index

- Sometimes we may have to handle the elements with XPATH index but index may give more than one match, which are under different parents, in these situations index might not help you. We may have to use Group index in these kind of scenarios.

Let's consider the below HTML code.

```
<html>
    <body>
        <div id="fruit"><br><br><br>
            <button type="button">Blueberry</button><br><br>
            <button type="button"  >Banana</button><br><br>
            <button type="button">Strawberry</button><br><br>
        </div>
        <div id="fruit">
            <button type="button">Apple</button><br><br>
            <button type="button" >Orange</button><br><br>
            <button type="button">Grape</button><br><br>
        </div>
    </body>
</html>
```

Now, write an XPATH for identifying the first element which is present in the first div using the Indexing logic.

Below is the XPATH.

```
//div[@id='fruit']/button[1]
```

It will find the two matching nodes.

It identifies the duplicate element. Coz, XPATH by indexing will find the elements which are under different parents. In this kind of scenarios, we go for using the XPATH by group indexing.

“Group index puts all matches into a list and gives indexes them. So here we will not have any duplicates matches”

```
Syntax : (//XPathExpression)[index]
```

We have to use parenthesis to make a XPATH into group XPATH after it index the XPATH.

Now, for the same above scenario, write the XPATH using the group indexing.

```
(//div[@id='fruit']/button)[1]
```

Above XPATH will identify only one XPATH.

Example Progrm:

```
public class XPath_by_groupIndex {
    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver", "D:/Softwares/Drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("file:///C:/Users/PriyaPramod/Desktop/HTML%20Pages/XPath/Xpath_GroupIndex.html");
        driver.manage().timeouts().implicitlyWait(100, TimeUnit.SECONDS);

        WebElement button = driver.findElement(By.cssSelector("(//div[@id='fruit']/button)[1]"));
        button.click();
        driver.close();
    }
}
```

Important:

1. What is the difference between //a, //a[1] and (//a) [1]?

- //a – Matches with all the links present in the entire web page
- //a[1]- Matches with all the first link
- (//a)[1] – Matches with only the first link

- //input[@type='checkbox'] – Matches with all the checkbox in the entire web page
- (//input[@type='checkbox'])[1]- Matches with the first checkbox
- (//input[@type='checkbox'])[5] - Matches with the fifth checkbox

In XPATH by group index, we use a function called "last ()"to select the last element.

Below is the syntax to select the last node in XPATH by group index.

```
Syntax : (//XPathExpression)[last()]
```

Example program.

```
public class Xpath_by_group_index_last_function {
    public static void main(String[] args) throws InterruptedException {
        System.setProperty("webdriver.chrome.driver", "D:/Softwares/Drivers/chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("file:///C:/Users/PriyaPramod/Desktop/HTML%20Pages/XPath/Xpath_GroupIndex.html");
        driver.manage().timeouts().implicitlyWait(100, TimeUnit.SECONDS);

        WebElement button = driver.findElement(By.cssSelector("(//div[@id='fruit']/button)[last()]"));
        button.click();
        driver.close();
    }
}
```

Derive the 'XPATH' which matches with first and last image?

```
(//img)[1]|(//img)[last()]
```