XPath cheat sheet

Types of XPath

1 Absolute XPath

```
/html/body/div/div/section/section
/div/div/div/input
```

2 Relative XPath

```
//*[@id='row1']/input
```

/ vs //

```
/  - short for child node
// - short for descendant or self node
```

at the beginning of xpath

```
/  - selects a root element
// - selects element anywhere on the page
```

in the middle of xpath

```
/  - selects child of the element
// - selects descendant of the element
```

// vs .//

dot introduces a relative location path, starting at the context node

Examples:

```
WebElement parentElement =
driver.findElement(By.id("someId"));
```

1

```
By childLocator1 = By.xpath("//input");
        parentElement.findElement(childLocator1);
```

This will ignore parentElement and will search for input element anywhere on the page

2

```
By childLocator2 = By.xpath(".//input");
        parentElement.findElement(childLocator2);
```

This will search input element that is descendant of the parentElement

Index

```
//tag[index]
    //h5[2]
```

- get second element with tag h5

```
//tag1[index1]/tag2[index2]
    (//div[@class='row'])[3]/h5[2]
```

- find third div element that has class row, and then get second h5 direct child

```
(//tag1[@attribute='value']/tag2)[index]
    (//div[@class='row']/input[@class='text'])[2]
```

- get all input elements with class text that are children of any div elements with class row, and then get second element from that list

Position functions

position()=2 works same way as index [2]

```
//h5[position()=2] same as //h5[2]
```

Operators we can use with position

position()=2	Equal
position()!=2	Not equal
position()>2	Greater than
position()>=2	Greater than or equal to
position()<2	Less than
position()<=2	Less than or equal to

last() *- get last element from the list*

```
//h5[last()]
```

We can also use subtraction with the last function

```
//h5[last()-1]
```

XPath formula

```
//tag[@attribute='value']
```

Example:

```
//div[@class='round-button']
```

Text Function

<div>Full element text</div>

```
//div[text()='Full element text']
```

Contains Function

Work with attribute values

<div id='username123'>

```
//button[contains(@id,'username')]
```

And with text

<div>Lets learn how to automate tests</div>

```
//div[contains(text(),'how to automate')]
```

Starts-With Function

Work with attribute values

<input class="input-field">

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

And with text

<p>This page is created to be able to reproduce the most common Selenium Exceptions.</p>

```
//p[starts-with(text(),'This page is created')]")
```

not Function

```
//div[not(@id='login')]
//a[not(text()='Click here')]
//input[not(contains(@class,'input'))]
//p[not(starts-with(text(),'Selenium'))]
```

Finding elements relative to other elements

```
//div[./input]
```

Find div element that has input child

```
//input[parent::div[@id='row2']]
```

The same as //div[@id='row2']/input

Selecting Several Paths

Use the vertical bar to combine two or more XPath expressions into one

```
//div[@id='row1']/button |
//div[@id='row1']/input
//h2 | //h5 | //p
```

SVG elements

To get to SVG element, use wildcard in place of tag name, and use name function for the SVG element tag

```
//*[name()='svg']//*[name()='rect' and @transform]
//*[name()='rect' and contains(@transform,'rotate
(45.0')]
```

SYNTAX

EXPLAINATION:

/	**Absolute XPATH** - Starts at the top of the DOM, or a direct decendant (child)
//	**Relative XPATH** - Looks anywhere on the page. Starts at any element on the page with this tag, or an indirect decendant
div	*Example of an element tag*
[ ]	**Predicates** - Used to find a specific node or a node with a specific value
@	*Attribute*
=''	Specific attribute value to search for
.	Uses the node that is in context
..	Selects the parent of the current node

XPath Operators

Using **'OR'**

```
//button[@name='Add' or @name='Remove']
```

Using **'AND'**

```
//button[@id and @class='btn'and @style and
@name='Add']
//button[@id][@class='btn'][@style][@name='Add']
```

XPath wildcards

`//*[@class]` - *Element with any tag that has 'class' attribute*

`//button[@*='btn']` - *Any button element where any attribute has value 'btn'*

`//div[@*]` - *Div element that has any attribute*

XPath axes

Formula:

```
axisname::nodetag[predicate]
```

XPath axes:

ancestor::   ancestor
Selects all ancestors of the current nodes

descendant::   descendant
Selects all children, grand-children etc... of the current node

parent:: parent *Only the parent of the current node*

following-sibling:: *Siblings after the current node*

preceding-sibling:: *Siblings before the current node*

Examples:

```
//button[@id='btn']/parent::div
```

Find div parent of button element with id "btn"

```
//button[@id='btn']/following-sibling::label
```

Find label sibling that is located after button element with id "btn"

```
//button[@id='btn']/preceding-sibling::label
```

Find label sibling that is located before button element with id "btn"

```
//button[@id='btn']/parent::div/following-sibling::div/div
```

combination of few axes in the same expression

To learn more about XPath or test automation with Selenium visit

https://practicetestautomation.com/

Practice Test Automation