1. //tagName[@value='visibleText']

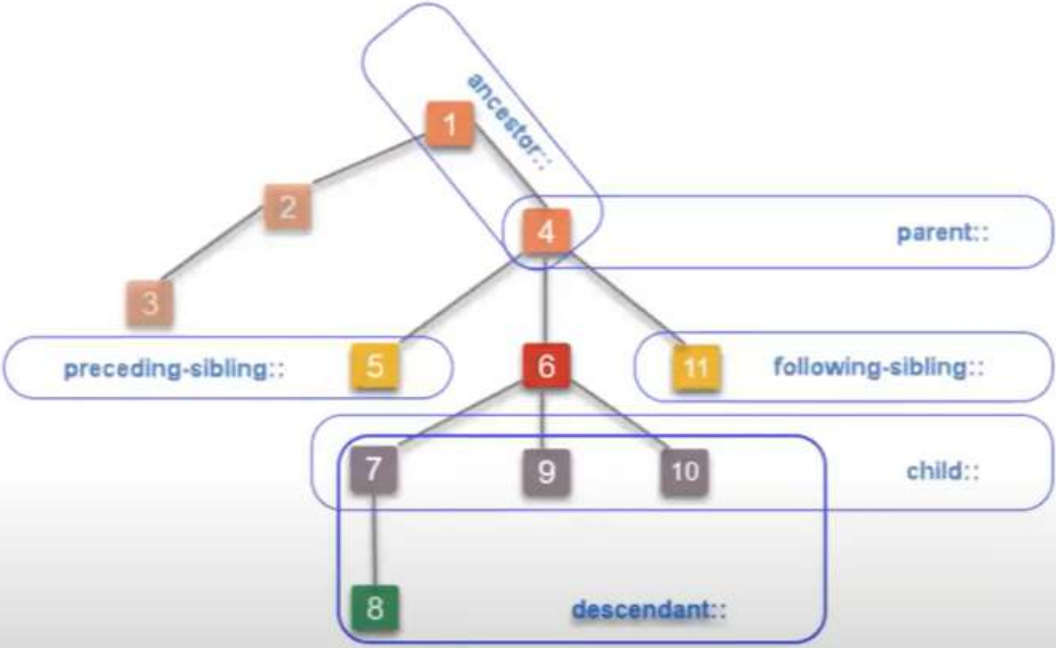
```
<!DOCTYPE html>
<html>
<body>

<form action="/action_page.php">
First <input type="text" name="FirstName" value="Janesh">
<br>
Last name: <input type="text" name="LastName" value="Kodikara">
```

First name: Janesh
Last name: Kodikara
Submit

Click the "Submit" button and

Examples :

1. //input[@value='Janesh']


ancestor::
1
2
3
4
parent::
preceding-sibling::
5
6
11
following-sibling::
7
9
10
child::
8
descendant::

Locating ancestors of a known element

The ancestor axis contains the ancestors of the known element; ancestor axis consists of the parent of a known element and the parent's parent so on.

Syntax :

//<xpathOfContextElement>/ancestor::<elementName> or //<xpathOfConte

parent of a known element and the parent's parent so on.

Syntax :

//<xpathOfContextElement>/ancestor::<elementName> or //<xpathOfCont
xtElement>/ancestor::*

```
A  <form id="frmLogin" method="post" action="/index.php/auth/validateCredentials">
     <div id="loginPanelHeading">LOGIN Panel</div>
   ▼ <div id="divUsername" class="textInputContainer">
       <input name="txtUsername" id="txtUsername" type="text">
        <span class="form-hint" style="display: none;">Username</span>
     </div>
```

Examples :

1. //input[@id='txtUsername']/ancestor::form : will select the form element
2. //input[@id='txtUsername']/ancestor::* : div element will be selected from the available candidates (div, form etc) as it comes first in the path if you use findElement method.

Locating a child element

The child axis contains the children of the context node

Syntax :

//<xpathOfContextElement>/child::<elementName> or
//<xpathOfContextElement>/child::*
//<xpathOfContextElement>/<elementName>

```
▼<div id="divUsername" class="textInputContainer">
  <input name="txtUsername" id="txtUsername" type="text">
  <span class="form-hint" style="display: none;">Username</span>
</div>
```

Examples :

In following examples context element's XPath is div[@id='divUsername']

1. //div[@id='divUsername']/child::input
2. //div[@id='divUsername']/input

In practice **/** is used instead of **child::** from the known XPath.

Following syntax could be used when a part of the attribute's values are NOT changed. We can use the non changing value for locating the element.

Syntax :

//elementName[contains(@attributeName,'substring of the value')] or
//*[contains(@attributeName,'substring of the value')]
//elementName[starts-with(@attributeName,'fixed prefix of the value')]

```
▼<div id="ptl-link">
  <a target="_blank" href="http://www.pragmatictestlabs.com">Pragmatic Test Labs</a>
</div>
```

Examples :

1. //a[contains(@href,'pragmatic')]
2. //*[contains(@href,'testlabs')]
3. //a[starts-with(@href,'pragmatic')]

The ends-with() function is part of XPath 2.0. Most of the browsers do not support Xpath 2.0 at the time of the writing.

Locating Elements with Multiple Attributes

Sometimes it may not be possible to locate an element with a single attribute uniquely as there could be more than one candidate elements with given attribute. In the real world, we have a similar scenarios. We cannot locate a person by just their first name or last name alone. We will have to use a combination of first name and last name to locate a person uniquely without making any confusion.

Similar technique is used in Selenium for locating elements when there are more than one elements with a given attribute. We will use two or more attributes together to locate an element uniquely.

Syntax :

//*[attribute1='value1'][attribute2='value2']...[attributeN='valueN'] or
//tagName[attribute1='value1'][attribute2='value2']...[attributeN='valueN'] or
//*[attribute1='value1' and attribute2='value2]
//tagName[attribute1='value1' and attribute2='value2]

```
▼<div id="divLoginButton">
  <input type="submit" name="Submit" class="button" id="btnLogin"
  value="LOGIN">
</div>
```

Examples :

1. //*[@type='submit'][@value='LOGIN']
2. //input[@class='button'][@type='submit'][@value='LOGIN'][@name='Submit'

Locating a parent element

The parent axis contains the parent of the context node. Every context element has only one parent element except root element (html).

Syntax :

//<knownXpath>/parent::* or

//<knownXpath>/parent::elementName

//<knownXpath>/..

Let's see how to locate the form element with respect to the username field. We need to select an element with unchanging XPath. In this case we will take the username field.

XPath of the known element : //input[@id='txtUsername']

```
T  ▼<form id="frmLogin" method="post" action="/index.php/auth/validateCredentials">
     <div id="logInPanelHeading">LOGIN Panel</div>
   ▼<div id="divUsername" class="textInputContainer">
       <input name="txtUsername" id="txtUsername" type="text">
       <span class="form-hint" style="display: none;">Username</span>
     </div>
```

Examples :

1. //input[@id='txtUsername']/parent::form
2. //input[@id='txtUsername']/parent::*
3. //input[@id='txtUsername']/..

Locating Elements when part of the visible text is static (partial match)

Syntax:

//tagName[contains(text(),'substring')]
//tagName[contains(.,'substring')]
//*[contains(text(),'substring')]

```
▼<div id="ptl-link">
  <a target="_blank" href="http://www.pragmatictestlabs.com">Pragmatic Test Labs</a>
</div>
```

Examples :

1. //a[contains(text(),'Pragmatic')]
2. //a[contains(., 'Test Labs')]
3. //*[contains(text(), 'Test Labs')]

Validate the XPath syntax before running the test scripts. Validating the XPath is discussed in a separate section.

//<xpathOfContextElement>/preceding::<elementName> or
//<xpathOfContextElement>/preceding::*

```
▼<div id="divUsername" class="textInputContainer" style>
  <input name="txtUsername" id="txtUsername" type="text" style>
  <span class="form-hint" style="display: none;">Username</span>
 </div>
▼<div id="divPassword" class="textInputContainer">
  <input name="txtPassword" id="txtPassword" type="password" style>
  <span class="form-hint" style="display: none;">Password</span>
 </div>
 <div id="divLoginHelpLink"></div>
```

Examples :

1. //span[text()='Password']/preceding::input

There will be two candidate elements (username and password elements). Selenium will select the password input element when findElement method is used. Elements are ordered from the context element (**span**).

Locating descendants of a known element

The descendant axis contains the descendants of a known element; descendant axis consists of the children of a context element and their children and so on.

Syntax :

//<xpathOfContextElement>/descendant::<elementName> or //<xpathOfContextElement>/descendant::*

```
A  <form id="frmLogin" method="post" action="/index.php/auth/validateCredentials">
     <div id="loginPanelHeading">LOGIN Panel</div>
   ▼<div id="divUsername" class="textInputContainer">
      <input name="txtUsername" id="txtUsername" type="text">
      <span class="form-hint" style="display: none;">Username</span>
    </div>
```

Examples :

1. //form[@id='frmLogin']/descendant::input
2. //form[@id='frmLogin']//input

You can use // instead of descendant:: keyword to locate descendants.

Keyword following:: can be used for locating element(s) anywhere below the tree with respect to a known element (context element).

Syntax :

//<xpathOfContextElement>/following::<elementName> or
//<xpathOfConextElement>/following::*

```
▼<form id="frmLogin" method="post" action="/index.php/auth/validateCredentials" sty
  <input type="hidden" name="actionID">
  <input type="hidden" name="hdnUserTimeZoneOffset" id="hdnUserTimeZoneOffset" va
  "5.5">
  <input type="hidden" name="_csrf_token" value="71389df12f99d00431a4c83a83fc2349
  id="csrf_token"> == $0
  <div id="logInPanelHeading">LOGIN Panel</div>
▼<div id="divUsername" class="textInputContainer">
  <input name="txtUsername" id="txtUsername" type="text">
  <span class="form-hint" style="display: none;">Username</span>
  </div>
▼<div id="divPassword" class="textInputContainer">
  <input name="txtPassword" id="txtPassword" type="password" style>
  <span class="form-hint" style="display: none;">Password</span>
  </div>
  <div id="divLoginHelpLink"></div>
▼<div id="divLoginButton">
  <input type="submit" name="Submit" class="button" id="btnLogin" value="LOGIN"
  style>
  </div>
</form>
```

E

1

2

Examples :

1. //input[@id='txtUsername']/following::input
2. //input[@id='txtUsername']/following::*

```
▼<div id="divUsername" class="textInputContainer">
  <input name="txtUsername" id="txtUsername" type="text">
  <span class="form-hint" style="display: none;">Username</span>
</div>
▼<div id="divPassword" class="textInputContainer">
  <input name="txtPassword" id="txtPassword" type="password" style>
  <span class="form-hint" style="display: none;">Password</span>
</div>
<div id="divLoginHelpLink"></div>
▼<div id="divLoginButton">
  <input type="submit" name="Submit" class="button" id="btnLogin" value="LOGIN"
  style>
</div>
</form>
```

Examples :

1. //input[@id='txtUsername']/following::input
2. //input[@id='txtUsername']/following::*

There are two candidate elements. Any descendant elements after the first candidate in path are excluded by Selenium when you use findElement method.

To select the login button input element with respect to the username field.

1. //input[@id='txtUsername']/following::input[last()]
2. //input[@id='txtUsername']/following::input[2]

first()

last()

Locating Elements with static Visible Text (exact match)

Following syntax is used for locating elements containing exact text within opening tag and closing tag (**inner text**).

Syntax:

//tagName[text()='exact text'] or
/*[text()='exact text']

Let's consider locating following hyperlink

```
▼<div id="ptl-link">
  <a target="_blank" href="http://www.pragmatictestlabs.com">Pragmatic</a>
</div>
```

Examples :

1. //a[text()='Pragmatic']

Locating following sibling

Keyword **following-sibling::** is used to locate the element(**s**) comes after a context element within same HTML hierarchy. Following siblings are the elements (children) of the context node's parent that occur after the context element in document order

Syntax :

//<xpathOfContextElement>/following-sibling::<elementName> or
//<xpathOfContextElement>/following-sibling::*

Known element

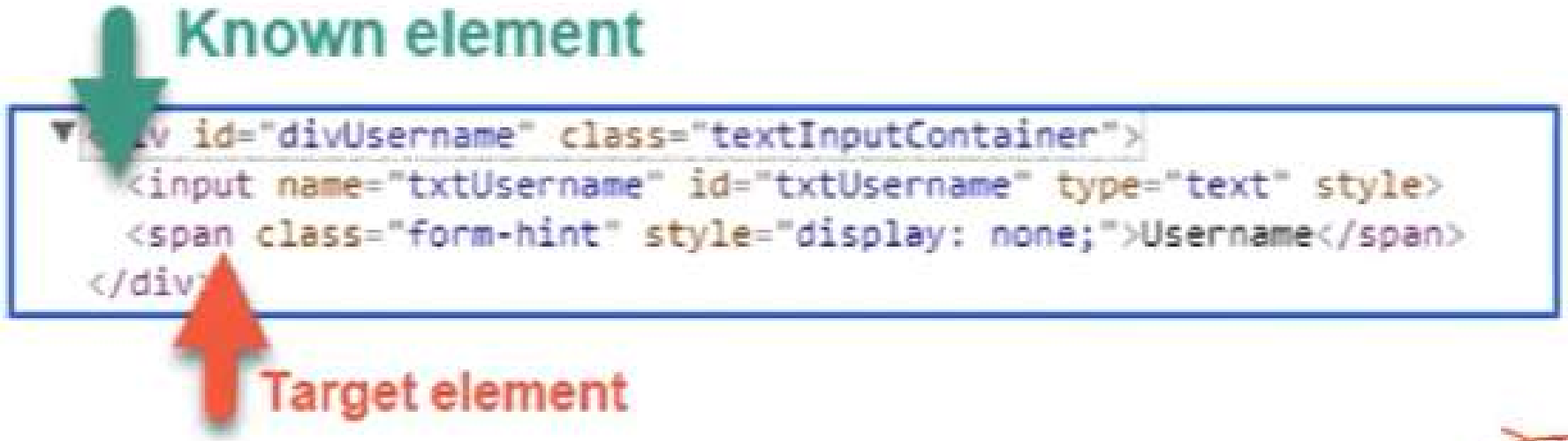
```
▼<div id="divUsername" class="textInputContainer">
  <input name="txtUsername" id="txtUsername" type="text" style>
  <span class="form-hint" style="display: none;">Username</span>
</div>
```

Target element

node's parent that occur after the context element in document order

Syntax :

//<xpathOfContextElement>/following-sibling::<elementName> or
//<xpathOfContextElement>/following-sibling::*



Examples :

1. //*[@id='txtUsername']/following-sibling::span
2. //*[@id='txtUsername']/following-sibling::*

Locating grand children

Syntax :

//<xpathOfContextElement>/*/<elementName>

//<xpathOfContextElement>/child/<elementName>

A

```
▼<form id="frmLogin" method="post" action="/index.php/auth/validateCredentials">
  <div id="loginPanelHeading">LOGIN Panel</div>
  ▼<div id="divUsername" class="textInputContainer">
    <input name="txtUsername" id="txtUsername" type="text">
    <span class="form-hint" style="display: none;">Username</span>
  </div>
```

Examples :

1. //form/*/input
2. //form/div/input

before the context element in document order.

Syntax :

//<xpathOfKnownElement>/preceding-sibling::<elementName> or
//<xpathOfKnownElement>/preceding-sibling::*

```
▼<div id="divUsername" class="textInputContainer">
  <input name="txtUsername" id="txtUsername" type="text">
  <span class="form-hint" style="display: none;">Username</span>
</div>
```

Examples :

1. //span[contains(text(),'Username')]/preceding-sibling::input
2. //span[contains(text(),'Username')]/preceding-sibling::*

With this we complete discussion of XPath with axes. Please note that we have not discussed attribute, ancestor-or-self, descendant-or-self, namespace and self **axes** in this article as they do not have practical usage in the context of Selenium.

Locating Elements when prefix of the inner text is static

You can locate the elements when part of the starting text of the inner text are static.

Syntax :

//tagName[starts-with(text(),'Prefix of Inner Text')]

//*[starts-with(text(),'Prefix of Inner Text")]

```
▼<div id="ptl-link">
  <a target="_blank" href="http://www.pragmatictestlabs.com">Pragmatic Test Labs</a>
</div>
```

Examples :

1. //a[starts-with(text(),'Pragmatic')]
2. //*[starts-with(text(), 'Prag')]

Terms used in XPath

Lets get familiar with the basic terms used in XPath syntax.

Tag name

<input name='txtUsername' id='txtUsername' type='text'>

Attribute

Value