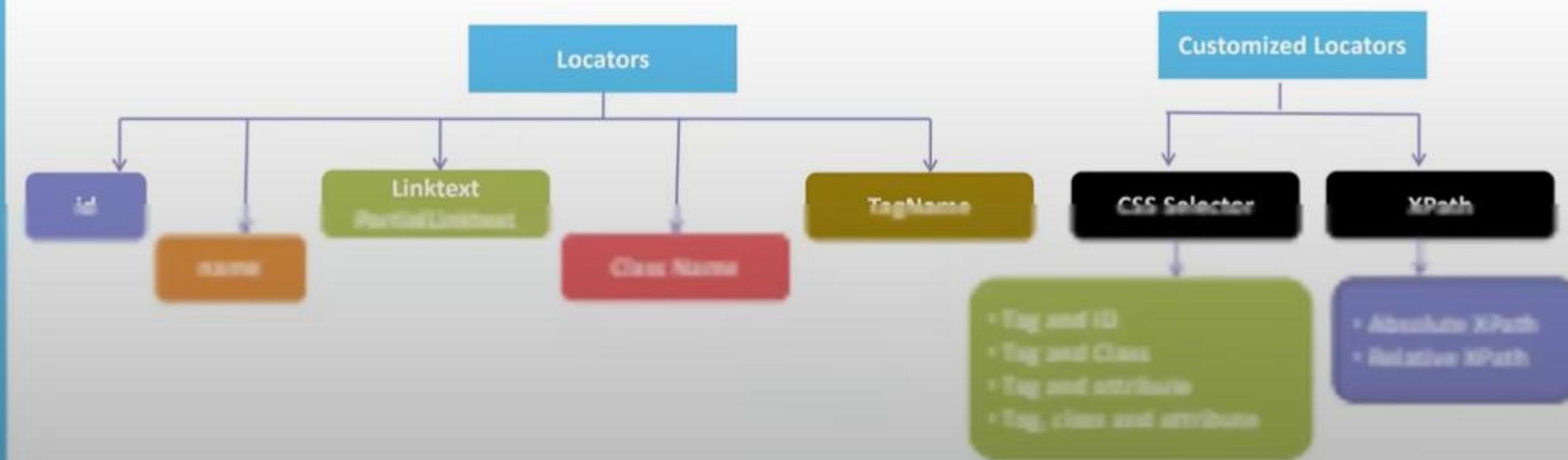


Selenium Locators

www.PavanOnlineTrainings.com

Types of Locators

- We can identify various elements on the web using **Locators**.
- Locators are addresses that identify a web element uniquely within the page.



Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

Locators.pdf

- We can identify various elements on the web using **Locators**.
- Locators are addresses that identify a web element uniquely within the page.

```
graph TD; Locators[Locators] --> id[id]; Locators --> name[name]; Locators --> LinktextPartialLinktext[Linktext  
PartialLinktext]; Locators --> ClassName[Class Name]; Locators --> TagName[TagName]; Locators --> CustomizedLocators[Customized Locators]; Locators --> CSSSelector[CSS Selector]; CustomizedLocators --> XPath[XPath]; CustomizedLocators --> CSSSelector; XPath --> AbsoluteXPath[Absolute XPath]; XPath --> RelativeXPath[Relative XPath];
```

www.PAYANONLINETRAININGS.COM

Locators

- id
- name
- linkText
- Partial LinkText
- class
- TagName

1:45:32 / 1:49:19

CC 4K 08:14 21-03-2024

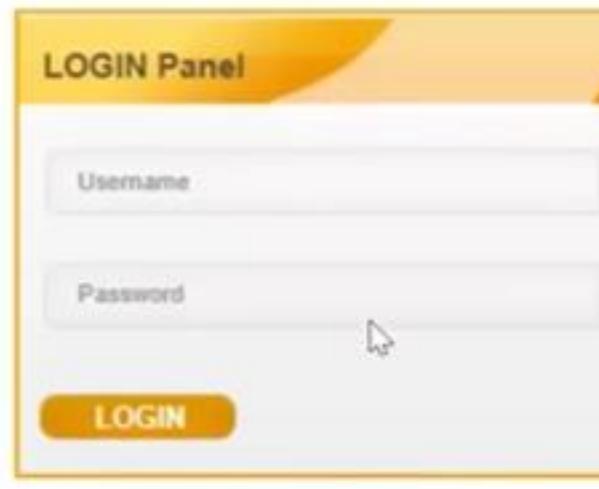
Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

www.PAYANONLINETRAININGS.COM

HTML Structure

LOGIN Panel



Element
Attribute
<input name="txtUsername" id="txtUsername" type="text">
Value

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head></head>
<body>
  <div id="wrapper">
    <div id="content">
      <style type="text/css">...</style>
      <div id="divLogin">
        <div id="divLogo">...</div>
        <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">Username</span>
          </div>
          <div id="divPassword" class="textInputContainer">
            <input name="txtpassword" id="txtpassword" type="password">
            <span class="form-hint">Password</span>
          </div>
          <div id="divLoginHelpLink">...</div>
          <div id="divLoginButton">
            <input type="submit" name="Submit" class="button" id="btnLogin" value="LOGIN">
          </div>
        </form>
      </div>
    </div>
  </body>
</html>
```

www.PAYANONLINETRAININGS.COM

ID

<http://automationpractice.com/index.php>

T-shirt

1:45:35 / 1:49:19

Search

CC 4K 08: A 21-03-2024

www.PAYANONLINETRAININGS.COM

ID

http://automationpractice.com/index.php

T-shirt 

```
<input class="search_query form-control ac_input" type="text" id="search_query_top" name="search_query" placeholder="Search" value="autocomplete="off"> == $0
```

```
driver.findElement(By.id("search_query_top")).sendKeys("T-shirt");
```

www.PAYANONLINETRAININGS.COM

Name

http://automationpractice.com/index.php

T-shirt 

1:45:38 / 1:49:19

CC 4K 08:21 A 21-03-2024

Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

- + 6 of 27

Name

http://automationpractice.com/index.php

T-shirt 

```
> <button type="submit" name="submit_search" class="btn btn-default button-search">...</button> == $0
```

```
driver.findElement(By.name("submit_search")).click();
```

Link Text / Partial Link Text

TOP SELLERS

Printed chiffon knee length dress with tank

1:45:40 / 1:49:19

CC 4K 08:21 A 21-03-2024

Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

www.PAYANONLINETRAININGS.com

Link Text / Partial Link Text

TOP SELLERS

-  Printed Chiffon Dress
Printed chiffon knee length dress with tank straps. Deep v-neckline.
\$16.40
-  Faded Short Sleeve T-shirts
Faded short sleeve t-shirt with high neckline. Soft and stretchy...
\$16.51

```
<a class="product-name" href="http://automationpractice.com/index.php?id_product=7&controller=product" title> Printed Chiffon Dress </a> == $0
```

```
driver.findElement(By.linkText("Printed Chiffon Dress")).click();
driver.findElement(By.partialLinkText("Chiffon Dress")).click();
```

www.PAYANONLINETRAININGS.com

Class Name

http://automationpractice.com/index.php

1:45:42 / 1:49:19

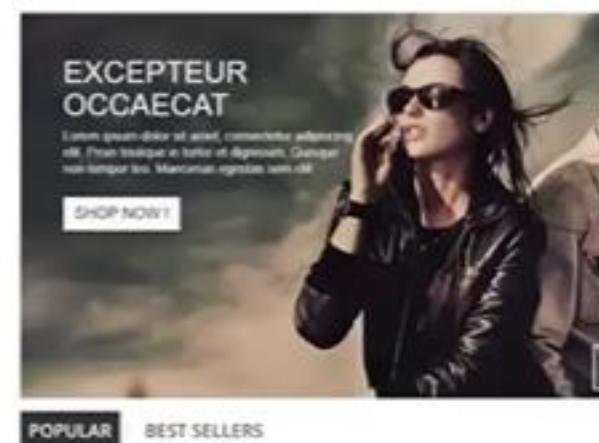
CC 4K 08: A 21-03-2024

```
driver.findElement(By.linkText("Printed Chiffon Dress")).click();
driver.findElement(By.partialLinkText("Chiffon Dress")).click();
```

www.PAVANONLINETRAININGS.COM

Class Name

http://automationpractice.com/index.php



```
<ul id="homeslider" style="max-height: 480px; width: 515%; position: relative; left: 960px;"> == $0
  ><li class="homeslider-container bx-clone" style="float: left; list-style: none; position: relative; width: 480px;">_</li>
  ><li class="homeslider-container" style="float: left; list-style: none; position: relative; width: 480px;">_</li>
  ><li class="homeslider-container" style="float: left; list-style: none; position: relative; width: 480px;">_</li>
  ><li class="homeslider-container" style="float: left; list-style: none; position: relative; width: 480px;">_</li>
  ><li class="homeslider-container bx-clone" style="float: left; list-style: none; position: relative; width: 480px;">_</li>
</ul>
```

```
int sliders=driver.findElements(By.className("homeslider-container")).size();
System.out.println(sliders);
```

www.PAVANONLINETRAININGS.COM

TagName

http://automationpractice.com/index.php

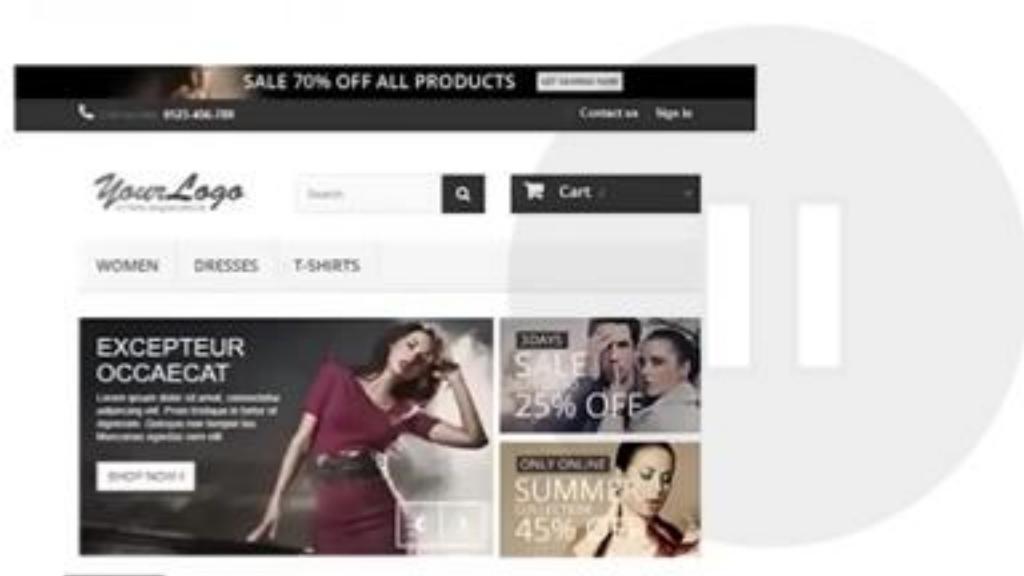
Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

- + 9 of 27

TagName

<http://automationpractice.com/index.php>



```
int links=driver.findElements(By.tagName("a")).size();
System.out.println(links);
```

www.PAYANONLINETRAININGS.COM

1:45:46 / 1:49:19

Search

4K

21-03-2024

Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

- + 11 of 27

CSS Selectors

www.PavanONLINETRAININGS.com

CSS Selector - Cascading Style Sheets

- Tag & ID (OR) #id
- Tag & class (OR) .class
- Tag & attribute (OR) [attribute=value]
- Tag , class & attribute

1:45:47 / 1:49:19

CC 4K 08: A 21-03-2024



www.PavanOnlineTrainings.com

CSS Selector – Tag and ID

<https://www.facebook.com/>

A screenshot of the Facebook login page. It shows an email input field containing "abc@gmail.com", a password input field, a blue "Log In" button, a "Forgotten account?" link, and a "Create New Account" button.

<input type="text" class="inputtext _55r1 _6luy" name="email" id="email" data-testid="royal_email" placeholder="Email address or phone number" autofocus="1" aria-label="Email address or phone number"> == \$0

driver.findElement(By.cssSelector("#email")).sendKeys("abc@gmail.com");

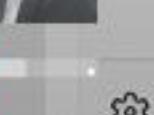
(or)

driver.findElement(By.cssSelector("input#email")).sendKeys("abc@gmail.com");

www.PavanOnlineTrainings.com

CSS Selector – Tag and Class

<https://www.facebook.com/>



Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

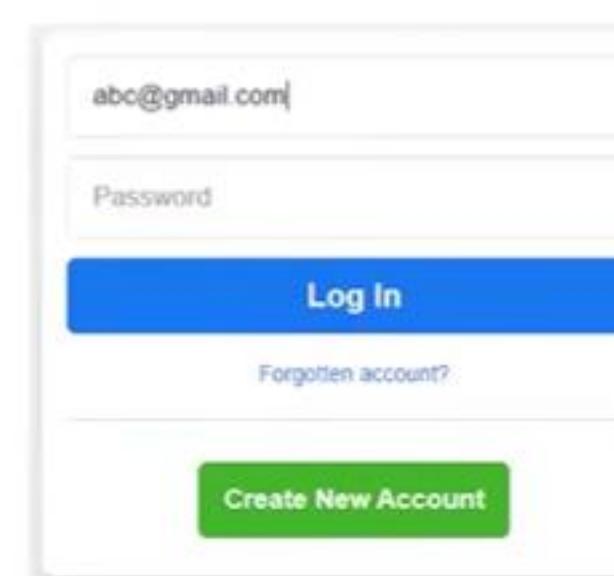
- + 13 of 27

```
driver.findElement(By.cssSelector("input#email")).sendKeys("abc@gmail.com");
```

www.PAYANONLINETRAININGS.COM

CSS Selector – Tag and Class

<https://www.facebook.com/>



The screenshot shows the Facebook login interface. A red box highlights the email input field, which contains the placeholder "abc@gmail.com". Below it is a password input field labeled "Password". A blue "Log In" button is centered below the fields. To its right, a link for "Forgotten account?" is visible. At the bottom, there's a green "Create New Account" button.

```
<input type="text" class="inputtext _55r1_6luy" name="email" id="email" data-testid="royal_email" placeholder="Email address or phone number" autofocus="1" aria-label="Email address or phone number">> == $0
```

```
driver.findElement(By.cssSelector(".inputtext")).sendKeys("abc@gmail.com");
```

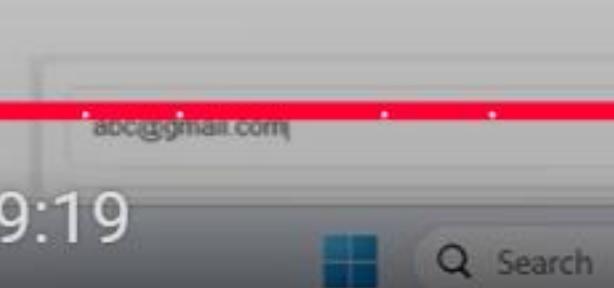
(or)

```
driver.findElement(By.cssSelector("input.inputtext")).sendKeys("abc@gmail.com");
```

www.PAYANONLINETRAININGS.COM

CSS Selector – Tag and Attribute

<https://www.facebook.com/>



The screenshot shows the Facebook login interface, identical to the previous one, with the email input field highlighted in red and containing "abc@gmail.com".

1:45:53 / 1:49:19

Search

CC 4K 08:24A 21-03-2024

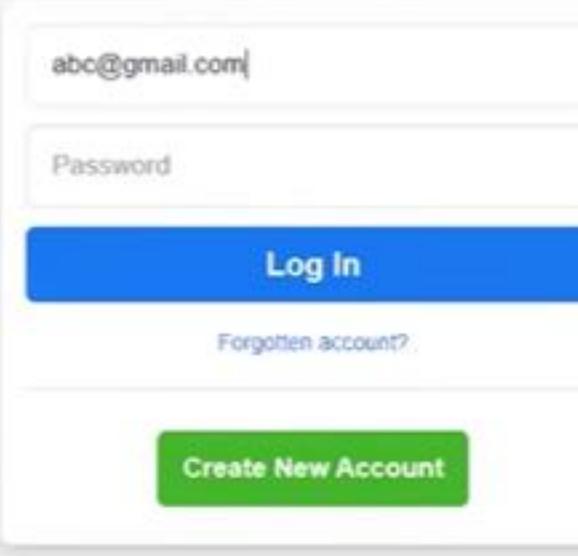
Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

- + 14 of 27

CSS Selector – Tag and Attribute

<https://www.facebook.com/>



The screenshot shows a Facebook login form. The email input field has the value "abc@gmail.com". The HTML code for this field is shown as a tooltip:

```
<input type="text" class="inputtext _55r1 _6luy" name="email" id="email" data-testid="royal_email" placeholder="Email address or phone number" autofocus="1" aria-label="Email address or phone number"> == $0
```

driver.findElement(By.cssSelector("input[name=email]")).sendKeys("abc@gmail.com");

(or)

driver.findElement(By.cssSelector("input[name=email]")).sendKeys("abc@gmail.com");

www.PAVANONLINETRAININGS.COM

CSS Selector - Tag, class and attribute

<https://www.facebook.com/>

input type="text" class="inputtext _55r1 _6luy" name="email" id="email" data-testid="royal_email" placeholder="Email address or phone number" autofocus="1" aria-label="Email address or phone number" style="width: 100%; height: 30px; border: 1px solid #ccc; border-radius: 5px; padding: 5px; font-size: 14px; margin-bottom: 10px;">

1:45:54 / 1:49:19

CC 4K 08:21 A 21-03-2024

Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

(or)

```
driver.findElement(By.cssSelector("input[name=email]")).sendKeys("abc@gmail.com");
```

www.PAYANONLINETRAININGS.COM

CSS Selector - Tag, class and attribute

<https://www.facebook.com/>

```
<div class="_6lux">
  <input type="text" class="inputtext _55r1 _6luy" name="email" id="email" data-testid="royal_email" placeholder="Email address or phone number" autofocus="1" aria-label="Email address or phone number" style=">
</div>
<div class="_6lux">
  <input type="password" class="inputtext _55r1 _6luy" name="pass" id="pass" data-testid="royal_pass" placeholder="Password" aria-label="Password" style=">
</div>
```

```
driver.findElement(By.cssSelector("input.inputtext[data-testid=royal_email]")).sendKeys("abc@gmail.com"); //Email
driver.findElement(By.cssSelector("input.inputtext[data-testid=royal_pass]")).sendKeys("abc"); //Password
```

www.PAYANONLINETRAININGS.COM

1:45:55 / 1:49:19

CC 4K 08: A 21-03-2024

XPath

WWW.PAYANONLINETRAININGS.COM

XPath

1. What Is XPath?

2. Types Of XPaths

- Absolute
- Relative

3. How to capture XPath?

4. Writing Dynamic XPath by different ways:

- Using 'OR' & 'AND'
- Using Contains()
- Using Starts-With()
- Using Text()
- Chained XPath

WWW.PAYANONLINETRAININGS.COM



Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

- + 18 of 27

Chained XPath

www.PAYANONLINETRAININGS.COM

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.
- XPath is used to find the location of any element on a webpage using HTML DOM structure.
- XPath can be used to navigate through elements and attributes in DOM.

www.PAYANONLINETRAININGS.COM

DOM – Document Object Model

* DOM is an API interface provided by browser.

1:46:01 / 1:49:19

Search

CC 4K 08:21 A 21-03-2024

www.PAYANONLINETRAININGS.COM

DOM – Document Object Model

- DOM is an API Interface provided by browser.
- When a web page is loaded, the browser creates a Document Object Model of the page.

HTML	DOM View	Rendered View
<pre><!DOCTYPE html> <html> <head> </head> <body> <button id="myBtn">Click Me</button> <input type="text" /> <p id="demo1"> This is static text message </p> <p id="demo2"> Hello!</p> </body> </html></pre>	<pre>DOCTYPE: html HTML HEAD #text: #text: BODY #text: BUTTON id="myBtn" #text: Click Me #text: INPUT type="text" #text: P id="demo1" #text: This is static text message #text: P id="demo2" #text: Hello! #text:</pre>	<p>Click Me</p> <p>This is static text message</p> <p>Hello!</p> <p>XPath works here</p>

Absolute XPath

1:46:02 / 1:49:19

Search

4K

08: A

21-03-2024

Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

- + 20 of 27

www.PAVANONLINETRAININGS.COM

Absolute XPath

- It is the direct way to find the element.
- The disadvantage of the absolute XPath is that if there are any changes made in the path of the element then that XPath gets failed.
- It begins with the single forward slash(/), which means you can select the element from the root node.
- Below is the example of an absolute XPath expression of the element
- Ex:

Absolute Xpath : /html[1]/body[1]/div[1]/div[1]/header[1]/div[3]/div[1]/div[1]/div[1]/a[1]/img[1]

Relative XPath

- 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.

You can start from the middle of the HTML DOM structure and no need to write long XPath

1:46:11 / 1:49:19

CC 4K 08:24 A 21-03-2024

Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

- + 21 of 27

Relative XPath

- 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.
- You can start from the middle of the HTML DOM structure and no need to write long XPath.

Ex:

Relative Xpath : //img[@class='logo img-responsive']

Syntax for Relative XPath

- XPath contains the path of the element situated at the web page. Standard syntax for creating XPath is.
- // : Select current node.

1:46:13 / 1:49:19

Search

CC 4K 08:24 A 21-03-2024

Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

Relative Xpath : //img[@class='logo img-responsive']

www.PAYANONLINETRAININGS.COM

Syntax for Relative XPath

- XPath contains the path of the element situated at the web page. Standard syntax for creating XPath is.
- // : Select current node.
- Tagname: Tagname of the particular node.
- @: Select attribute.
- Attribute: Attribute name of the node.
- Value: Value of the attribute.
- Xpath=//tagname[@attribute='value']

www.PAYANONLINETRAININGS.COM

1:46:15 / 1:49:19

CC 4K 08: A 21-03-2024

www.PAVANONLINETRAININGS.COM

XPath with OR

Signup for Free

<https://accounts.lambdatest.com/register>

```
<div class="form-group">
  <input type="text" placeholder="Company Name" name="organization_name" value class="form-control" style="xpath="1"> == $0
</div>
```

`//input[@name='organization_name' or @placeholder='Organization/Company Name']`

www.PAVANONLINETRAININGS.COM

XPath with AND



Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

- + 24 of 27

XPath with OR

```
//input[@name='organization_name' or @placeholder='Organization/Company Name']
```

www.PAYANONLINETRAININGS.com

XPath with AND

Signup for Free

https://accounts.lambdatest.com/register

The screenshot shows a sign-up form with fields for Full Name, Email, Desired Password, Company Name, and Phone. A blue arrow points from the 'Full Name' input field to its corresponding XPath expression in the code view.

```
<div class="form-group">
<input type="text" placeholder="Full Name*" name="name" value required="required" class="form-control" xpath="1"> == $0
</div>
```

```
//input[@name='name' and @placeholder='Full Name*']
```

www.PAYANONLINETRAININGS.com

XPath with contains()

1:46:19 / 1:49:19

CC 4K 08:14 21-03-2024

Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

- + 23 of 27

XPath with OR

Signup for Free

<https://accounts.lambdatest.com/register>

```
<div class="form-group">
  <input type="text" placeholder="Company Name" name="organization_name" value class="form-control" style="xpath="1"=> == $0
</div>
```

`//input[@name='organization_name' or @placeholder='Organization/Company Name']`

XPath with AND

<https://accounts.lambdatest.com/register>

www.PAVANONLINETRAININGS.COM

www.PAVANONLINETRAININGS.COM

1:46:17 / 1:49:19

CC 4K 08:24 A 21-03-2024

Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

- + 24 of 27

XPath with AND

Signup for Free https://accounts.lambdatest.com/register

Sign up with Google

- or Signup via email -

Full Name*

Email*

Desired Password* Show

Company Name

Phone (+1 555 555 5555)*

I agree to LambdaTest's Privacy Policy & Terms of Service

FREE SIGN UP

//input[@name='name' **and** @placeholder='Full Name*']

<div class="form-group">
 <input type="text" placeholder="Full Name*" name="name" value required="required" class="form-control" xpath="1" /> == \$0
</div>

XPath with contains()

www.PAYANONLINETRAININGS.COM

www.PAYANONLINETRAININGS.COM

1:46:19 / 1:49:19

CC 4K 08: A 21-03-2024

Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

- + 25 of 27

XPath with contains()

https://www.lambdatest.com/

Live Automation Pricing Resources Support Log in Start Free Testing

> ...

//a[contains(text(), 'Testing')]

//a[contains(@id, 'value')]

XPath with starts-with()

https://www.lambdatest.com/

Live Automation Pricing Resources Support Log in Start Free Testing

1:46:20 / 1:49:19

Search

CC 4K

08: A 21-03-2024

XPath with starts-with()

<https://www.lambdatest.com>

Live Automation Pricing Resources Support

```
▶ <a class="nav-link" href="https://accounts.lambdatest.com/register" onclick="onStartTesting()" xpathtest="1" style xpath="1">...</a>
```

```
//a[starts-with(text(), 'Start')
```

```
//a[starts-with(@name, 'value')]
```

Locators.pdf

Session 24 - Selenium with Java | Locators - XPath | XPath Functions | XPath Types

- + 27 of 27

```
><a class="nav-link" href="https://accounts.lambdatest.com/register" onclick="onStartTesting()" xpathtest="1" style xpath="1">...

```
//a[starts-with(text(), 'Start')]
```



```
//a[starts-with(@name, 'value')]
```



www.PavanONLINETRAININGS.COM



## XPath with Text()



https://www.lambdatest.com/



```
<li class="nav-item">
 ... == $0

```



```
//a[text()='Pricing']
```



www.PavanONLINETRAININGS.COM



1:46:24 / 1:49:19

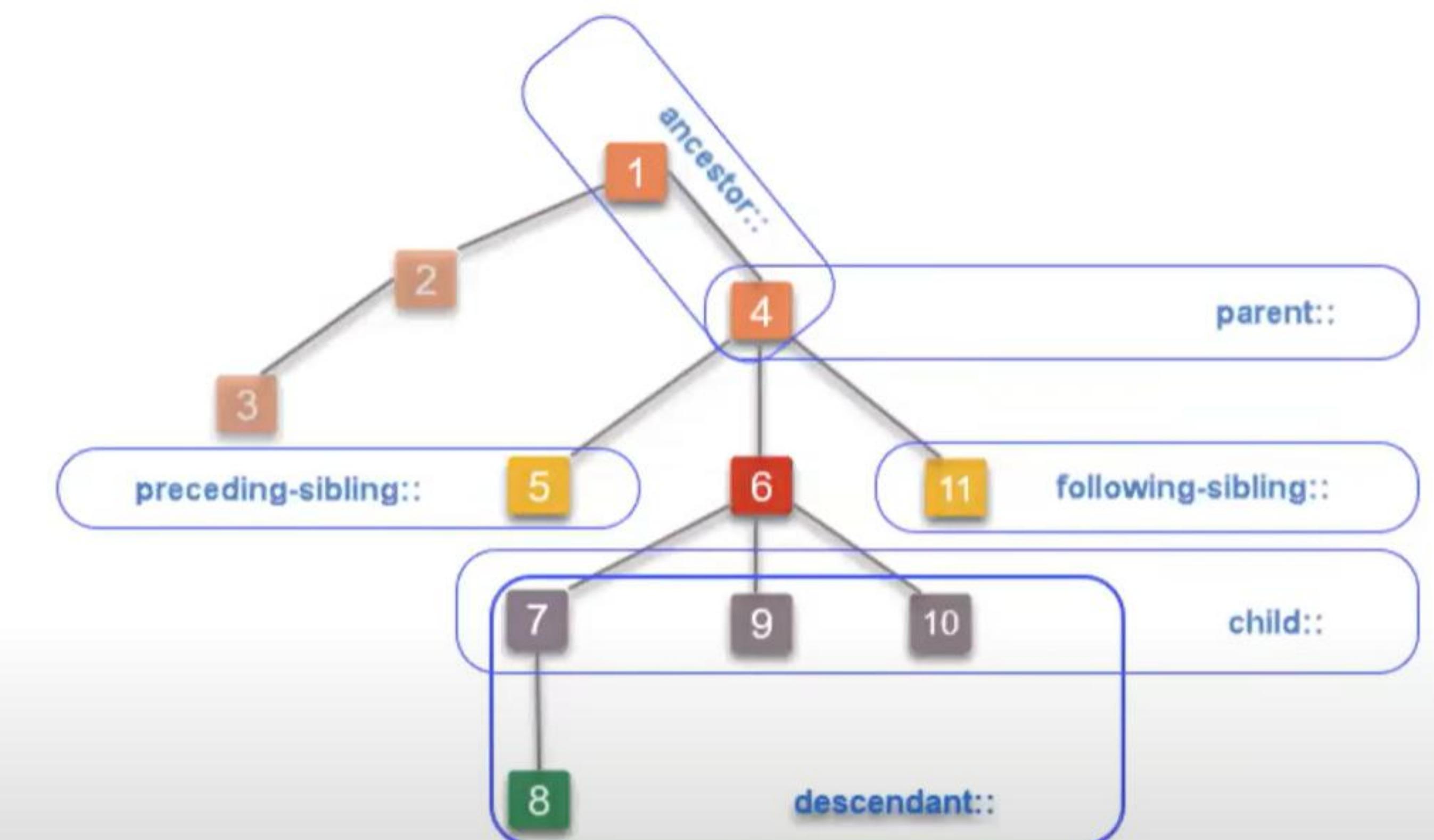


Search



CC 4K 08:24 21-03-2024


```



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).

Activate Windows
Go to Settings to activate Windows.

Syntax :

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']

```
<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" value="User" />  
    <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']/..

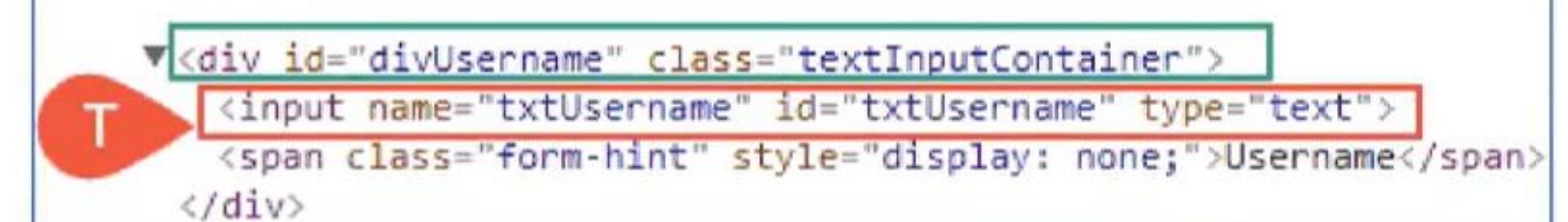
Activate Windows
Go to Settings to activate Windows.

Locating a child element

The child axis contains the children of the context node

Syntax :

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



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.

Locating grand children

Syntax :

```
//<xpathOfContextElement>/*/<elementName>
```

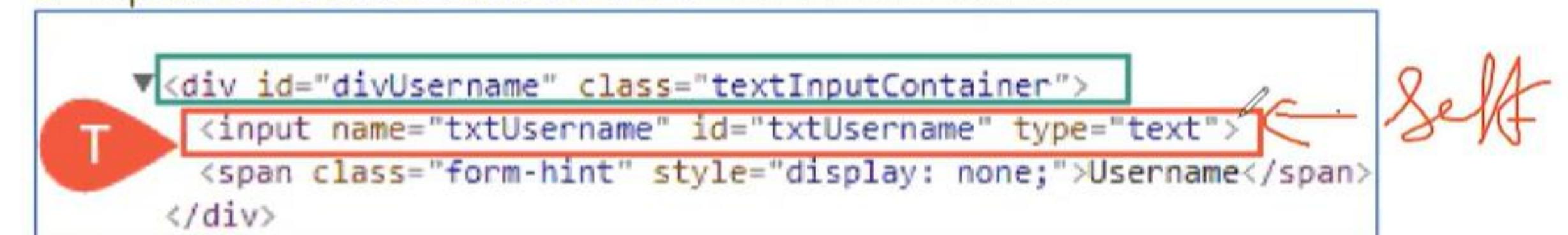
Activate Windows
Go to Settings to activate Windows.

Locating a child element

The child axis contains the children of the context node

Syntax :

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



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.

Locating grand children

Syntax :

```
//<xpathOfContextElement>/*/<elementName>
```

Activate Windows
Go to Settings to activate Window...

Session 25 - Selenium with Java | Locators - XPath Axes | SelectorsHub

```
* <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

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

Activate Windows
Go to Settings to activate Windows.

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

Syntax :

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

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 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.

available candidates (div, form etc) as it comes first in the path if you use findElement method.

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 //<xpathOfCon  
textElement>/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>
```

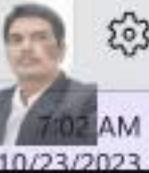
Examples :

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

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

Locating following elements

Activate Windows
Go to Settings to activate Windows.



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 the 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]

Locating preceding element

Keyword **preceding::** is used for locating an element before a known (XPath) element.

The **preceding** axis contains all nodes that are descendants of the root of the tree in which the context node is found, are not ancestors of the context node, and occur before the context node in document order

Syntax :

//<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**).

2. //span[text()='Password']/preceding::input[2]

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`).

2. `//span[text()='Password']/preceding::input[2]`

Above can be used for selecting the username field.

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::*
```

```
<div id="divUsername" class="textInputContainer">  
  <input name="txtUsername" id="txtUsername" type="text" style="width: 100%; height: 100%; border: none; border-bottom: 1px solid #ccc; padding: 0; margin: 0;">  
  <span class="form-hint" style="display: none;">Username</span>  
</div>
```

Examples :

Activate Windows
Go to Settings to activate Windows.

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 Several XPaths

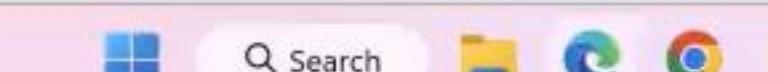
You can locate multiple elements by separating two or more XPath expressions with | character.

Syntax :

XPath1 | Xpath2.... | XPathN

If the first XPath is available in the first element is selected by Selenium for further actions. If both are available first one is Selected when you use findElement method. Both will be selected

Activate Windows
Go to Settings to activate Windows..



7:24 AM
10/23/2023

Locating preceding sibling

Keyword **preceding-sibling::** is used to selects the sibling(s) that comes before the context node with a known XPath, those elements (children) of the context node's parent that occur before the context element in document order.

Syntax :

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

Username

Examples

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

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

Session 25 - Selenium with Java | Locators - XPath Axes | SelectorsHub

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 Several XPaths

You can locate multiple elements by separating two or more XPath expressions with | character.

Syntax :

XPath1 | XPath2.... | XPathN

If the first XPath is available in the first element is selected by Selenium for further actions. If both are available first one is Selected when you use findElement method. Both will be selected

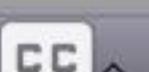
Activate Windows
Go to Settings to activate Windows.



49:06 / 1:13:34

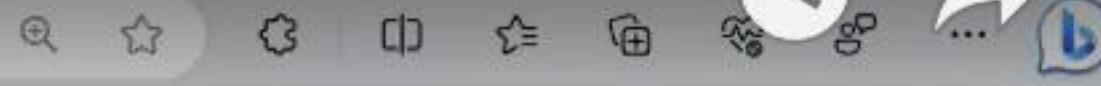


Search



7:26 AM
23/3/2023

Session 25 - Selenium with Java | Locators - XPath Axes | SelectorsHub



Syntax :

XPath1 | Xpath2.... | XPathN

If the first XPath is available in the first element is selected by Selenium for further actions. If both are available first one is Selected when you use **findElement** method. Both will be selected if you use **findElements** method. If only second XPath (Xpath2) is available then second element will be selected. If both of them are NOT available Selenium gives an error, **NoSuchElementException** when you use **findElement** method.

Example :

1. //input[@id='txtUsername'] //input[@name='txtPassword']

In this example available elements can be located.

This is useful when you know one of them would exist when the page is loaded.

2. //*[@id='txtUsername'] //*[@name='txtPassword'] //*[@name='btnLogin']



Working with Operators

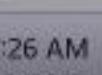
Activate Windows
Go to Settings to activate Windows.



49:07 / 1:13:34



Search



7:26 AM
23/23/2023

You can use various operators to compare numeric value in attributes and inner text. We have addition (+), subtraction (-), multiplication (*), division (div), equal (=), not equal (!=), less than (<), less than or equal (<=), greater than (>), greater than or equal (>=), **and**, **or** , mod operators.

Lets see an example



DEAL OF THE DAY

124.00

List: \$169.00 (27% off)

Ends in 01:03

Audio-Technica ATH-M50xGM

...

124.00 == \$0

::after

> <div class="a-row a-spacing-top-mini unitLineHeight">...</div>

::after

> <div class="a-row a-spacing-mini">...</div>

> <div class="a-row a-spacing-mini">...</div>

> <div class="a-row a-spacing-mini">...</div>

> <div class="a-row a-spacing-mini">...</div>

...

div div div div div span.a-size-medium.inlineBlock.unitLineHeight.dealPriceText

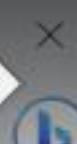
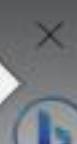
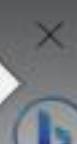
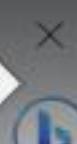
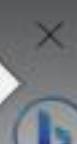
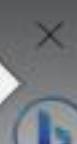
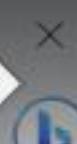
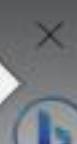
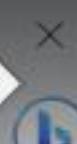
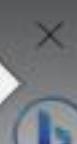
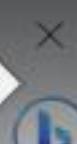
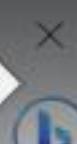
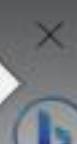
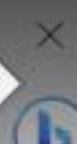
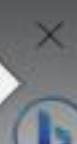
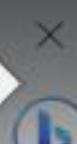
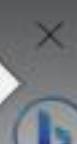
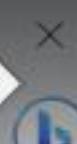
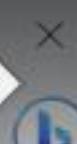
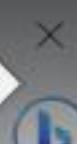
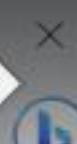
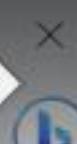
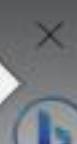
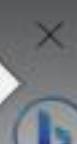
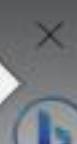
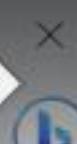
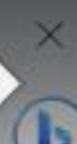
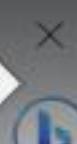
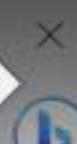
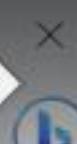
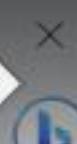
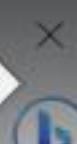
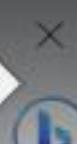
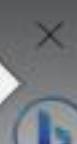
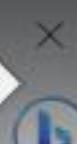
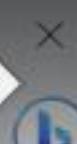
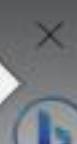
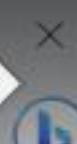
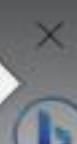
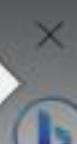
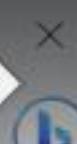
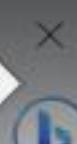
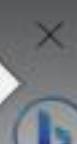
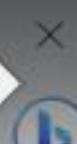
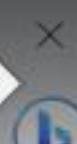
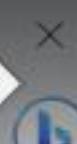
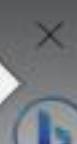
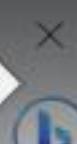
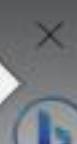
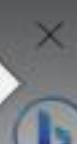
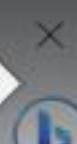
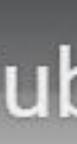
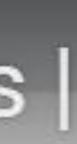
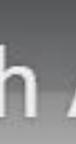
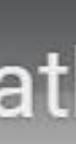
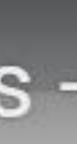
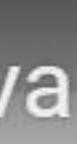
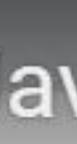
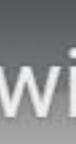
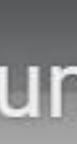
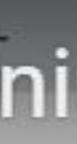
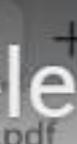
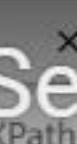
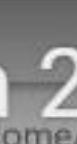
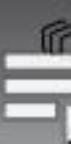
//span[text()>100]

1 of 30 ▲ ▼ Cancel

Say you need to locate element(s) with deal price greater than 100

```
//span[contains(@class,'dealPriceText') and text()>100]
```

You can combine the techniques learned to build complex XPaths and locate any element in DOM.



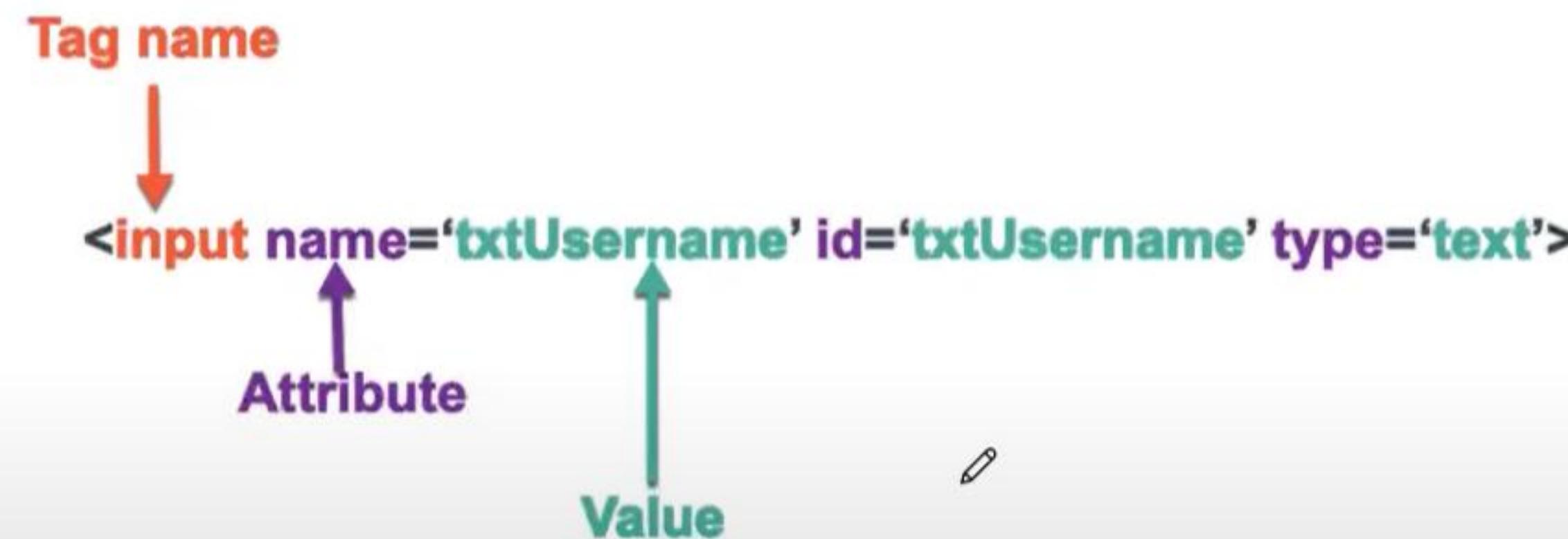
Session 25 - Selenium with Java | Locators - XPath Axes | SelectorsHub

dynamic elements. This leads to failure of the automated test scripts (brittle) when web pages with dynamic contents are automated. Most of the testers rely on extracting the XPaths from browser plugins. These tools have limitations and do not provide the best XPath for dynamic elements.

We will discuss XPath in detail with examples and explore a few tools to generate XPaths easily.

Terms used in XPath

Let's get familiar with the basic terms used in XPath syntax.



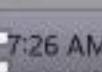
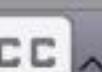
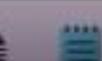
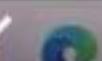
Activate Windows
Go to Settings to activate Windows.




49:34 / 1:13:34

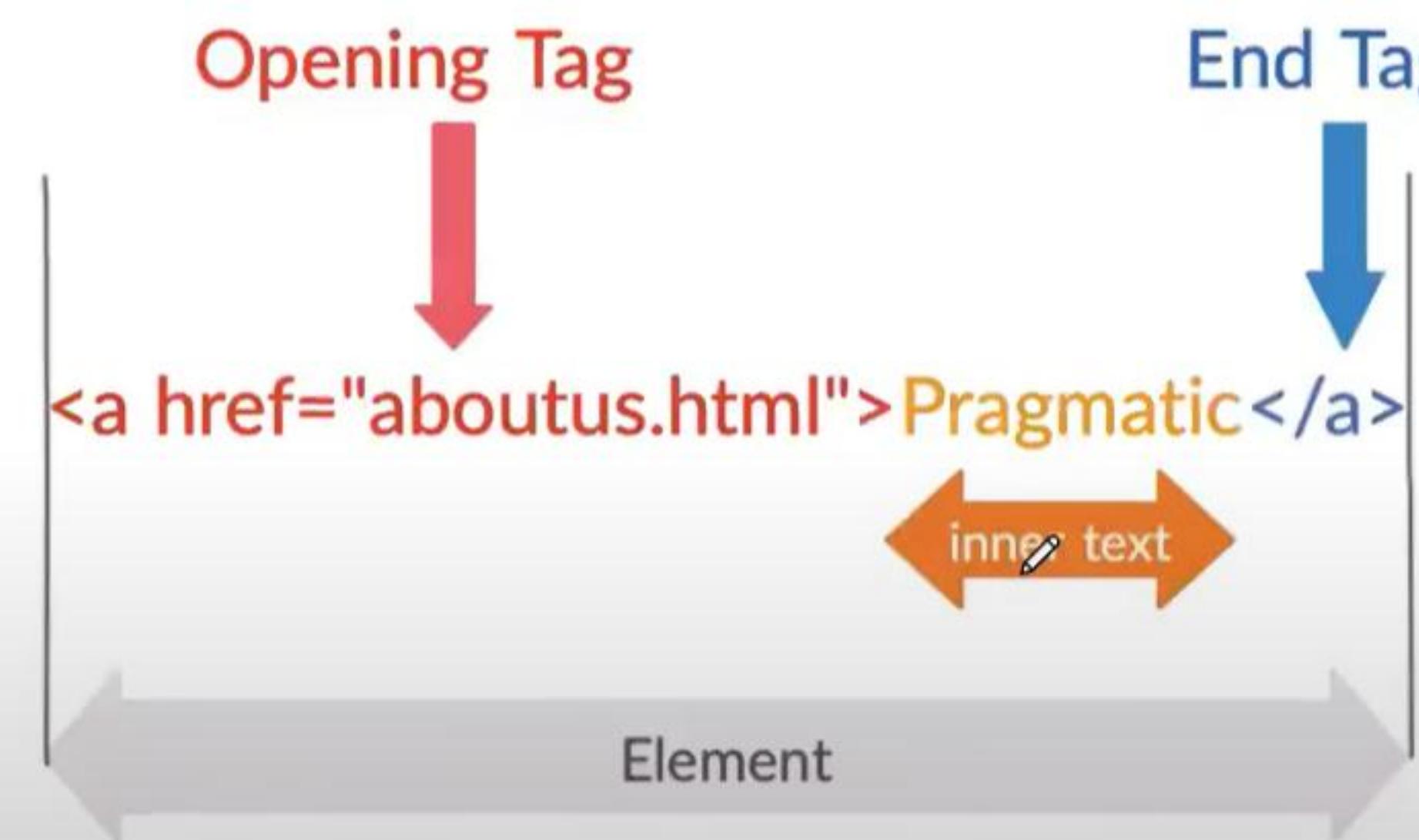


Search



7:26 AM

7/23/2023



Using XPath for locating elements in HTML

We have our own way of introducing (explaining) XPath to the trainees in our public training programs.

Using XPath for locating elements in HTML

We have our own way of introducing (explaining) XPath to the trainees in our public training programs.

We write following two equations. A=B and B=C. We ask the students what can be derived from these two expressions. Immediately students reply with the answer A=C, even before the question is asked :-).

Then we give following two statements

- - XPath can be used for locating the elements in XML
 - XML and HTML have similar syntax (HTML is a subset of XML)

Hence we can derive **XPath** can be used **for locating elements in HTML** pages (web pages) too . Selenium uses XPath to locate elements in web pages.

Why do we need to master many XPath syntaxes?

We may not be able to locate some elements using their ID or Name as some elements do not have unique attributes (id or name). Some attributes are dynamically changed. There could be elements without any attribute too. Hence we may have to locate them differently than the static elements.

Activate Windows
Go to settings to activate Windows.



XPath can be used for

- - Locating elements with respect to a known element
 - Locating elements with partially static attribute values
 - Locating elements without attributes or without unique attributes

XPath can do **bidirectional** navigation (Going forward and backward)

XPath is one of the most **flexible** and strongest location strategies.

Types of XPath

We have grouped XPaths into 2 types:

Absolute XPath

Types of XPath

We have grouped XPaths into 2 type

Absolute XPath

Absolute XPaths starts with the root of the **HTML** pages

Absolute XPaths are not advisable for most of the time due to following reasons

1. Absolute XPaths are lengthier and hence they are not readable
 2. Absolute XPaths are not resilient. They tend to break when minor structural changes are introduced to the web pages

Absolute XPaths shall be used only when a relative XPath cannot be constructed. (highly unlikely). **It is not recommended** to use absolute XPath in Selenium.

Syntax: Absolute XPaths start with /htm

Example: /html/body/div[1]/div/div[2]/form/div[2]/input

Relative XPath

Relative XPaths is used for locating elements with respect to a element with known (solid) XPath. The element of your choice is referred relative to a known element.

Session 25 - Selenium with Java | Locators - XPath Axes | SelectorsHub

(solid) XPath. The element of your choice is referred relative to a known element.

Syntax: Relative XPaths are started with two forward slashes '//'.

Examples :

1. `//div[@id='divUsername']/input`
2. `//form/div[@id='divUsername']/input`

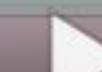
3. `//form/*/input`

There could be zero or more elements between the **context element** (starting element with a known Xpath) and the target element

NOTES:

3. Allowance of zero or more elements between context element and target element

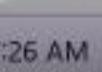
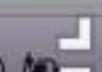
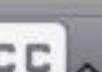
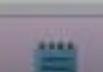
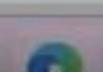
Activate Windows
Go to settings to activate Windows.



49:45 / 1:13:34



Search



7:26 AM
7/23/2023

There could be zero or more elements between the **context element** (starting element with a known Xpath) and the target element

NOTES

1. Absolute XPaths are faster than the relative XPath
 2. Use shortest possible relative XPaths

What should be considered when choosing an XPath?

It is important to consider following while choosing an XPath from available options.

A good locator is

- Unique
 - Descriptive
 - Resilient
 - Shorter in length

When you want to locate a single element your XPath should have only one candidate element (**unique**). It will be easier to identify the element if it is descriptive and short(for **Maintainability**). XPaths generated from tools may not be **user-friendly**. It should be possible to locate an element with given XPath when the test is run again in subsequent releases too. XPath should be selected in such a way it is valid even after changes in DOM (**resilient**). You will have multiple XPath options. A shorter XPath shall be selected to make it more readable in your test scripts.

be possible to locate an element with given XPath when the test is run again in subsequent releases too. XPath should be selected in such a way it is valid even after changes in DOM (**resilient**). You will have multiple XPath options. A shorter XPath shall be selected to make it more readable in your test scripts.

Sample HTML code

Session 25 - Selenium with Java | Locators - XPath Axes | SelectorsHub

File | mac/Home/Desktop/XPath.pdf

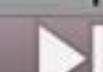
Search | Star | Settings | Home | Favorites | Help | More

FOLLOWING HTML (DOM) will be used for the explaining most of the XPath syntax in this post.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  ><head>...</head>
  ><body>
    ><div id="wrapper">
      ><div id="content">
        ><style type="text/css">...</style>
        ><div id="divLogin">
          ><div id="divLogo">...</div>
          ><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">Username</span>
            ></div>
            ><div id="divPassword" class="textInputContainer">
              ><input name="txtPassword" id="txtPassword" type="password">
              ><span class="form-hint">Password</span>
            ></div>
            ><div id="divLoginHelpLink">...</div>
            ><div id="divLoginButton">
              ><input type="submit" name="Submit" class="button" id="btnLogin" value="LOGIN">
            ></div>
          ></form>
        ></div>
      ></div>
    ></body>
</html>
```



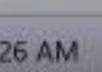
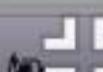
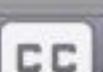
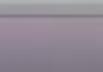
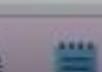
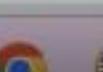
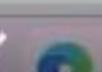
Activate Windows
Go to Settings to activate Windows.



49:48 / 1:13:34



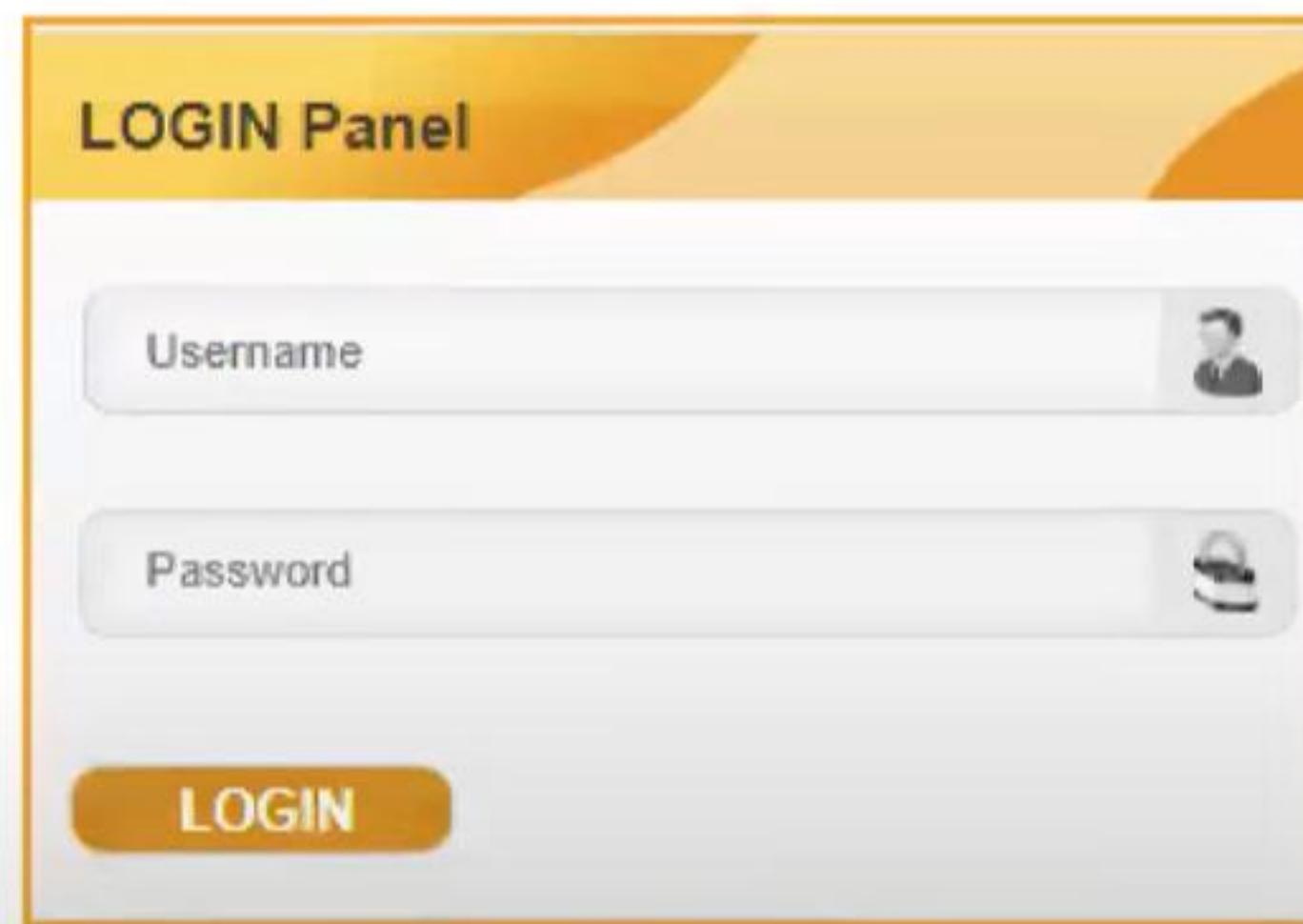
Search



7:26 AM

7/23/2023

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



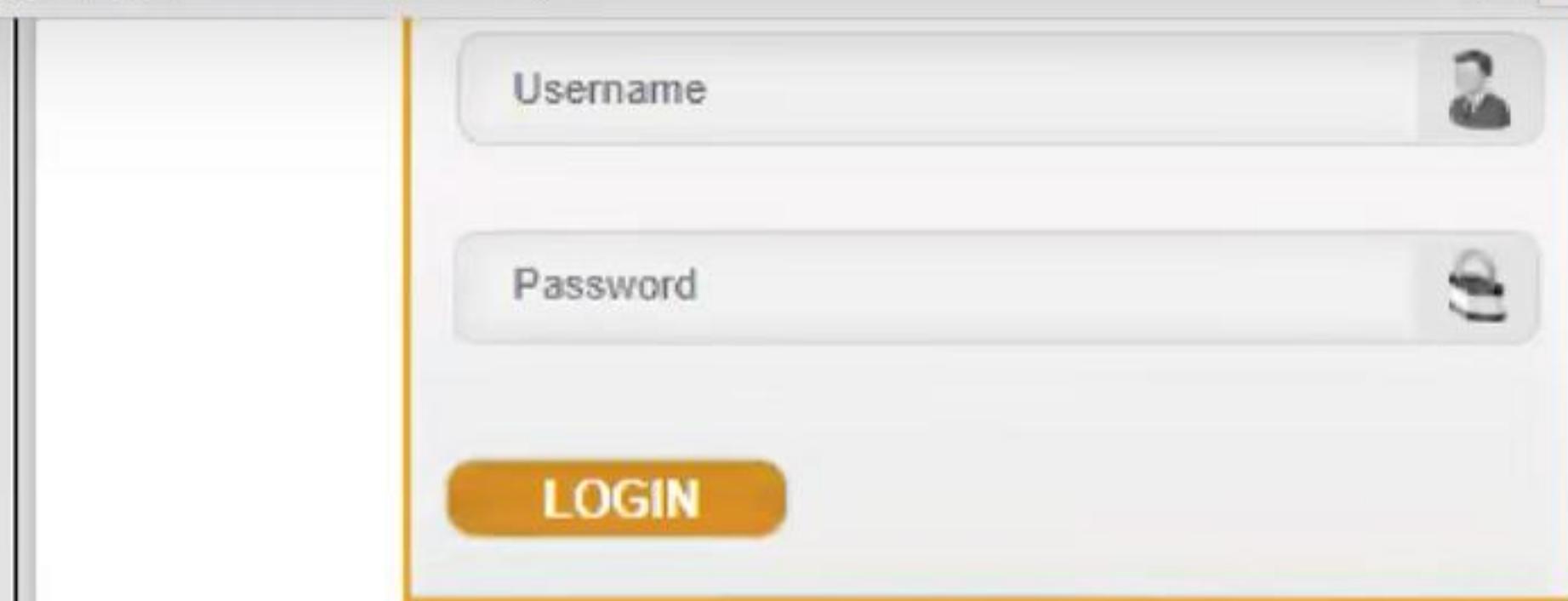
Locating an Elements with a Known Attribute

The following syntax can be used for locating elements when at least one of the attribute's value is unique and static.

Syntax:

```
//^|(@attributeName='value')
```

Session 25 - Selenium with Java | Locators - XPath Axes | SelectorsHub



Locating an Elements with a Known Attribute

The following syntax can be used for locating elements when at least one of the attribute's value is unique and static.

Syntax:

```
//*[@attributeName='value']
```

Let's locate the **username** field in following.

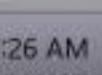
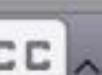
Activate Windows
Go to settings to activate Windows.



49:50 / 1:13:34



Search



7:26 AM
17/23/2023



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

We have **three** valid XPath

Examples

1. `//*[@id='txtUsername']`
 2. `//*[@name='txtUsername']`
 3. `//*[@type='text']`

NOTE: Third XPath **should not** be used even though it is a valid XPath. Because It will not be a unique XPath in most of the cases. There will be many elements with type='text'. Hence Selenium will not be able to locate the target element uniquely.

Single quotations should be used to enclose the values (in Java). You need to use escape character if you wish to enclose the values with double quotes.

Example : //*[@id='txtUsername']

In our real life we cannot locate a person with name **Mohammed** in a Muslim community.

Example : //*[@id='txtUsername']

In our real life we cannot locate a person with name **Mohammed** in a Muslim community.

How does Selenium works when there are multiple elements (candidates) for an XPath?

Selenium will pick the first element in the path if there are multiple candidates for a given XPath when webdriver.**findElement**(By.xpath("XPATH")) method is used. All the candidate elements can be assigned to a List when webdriver.findElements(By.xpath("XPATH")) method is used.

Examples

1. To select the third input element : //form/input[3]
 2. To select the last input element : //form/input[last()]

Locating elements by position is discussed further in a separate section.

findElement method throws **NoSuchElementException** error when there is no elements with given XPath. Use **findElements** method and check the size when working with non-present elements.

Locating Elements with a Tag-name and an Attribute

Syntax

Session 25 - Selenium with Java | Locators - XPath Axes | SelectorsHub

used.

Examples :

1. To select the third input element : //form/input[3]
2. To select the last input element : //form/input[last()]

Locating elements by position is discussed further in a separate section.

findElement method throws **NoSuchElementException** error when there is no elements with given XPath. Use **findElements** method and check the size when working with non-present elements.

Locating Elements with a Tag-name and an Attribute

Syntax:

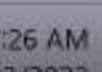
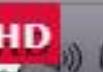
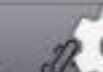
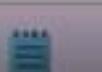
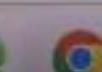
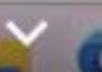
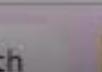
//tagName[@attributeName='value']



49:54 / 1:13:34



Search



7:26 AM
7/23/2023

Session 25 - Selenium with Java | Locators - XPath Axes | SelectorsHub

//tagName[@attributeName='value']

Let's consider locating the username input field again



```
<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']
2. //input[@name='txtUsername']

This is one of the most commonly used Xpath. Most of the plugs can generate above XPaths automatically

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:

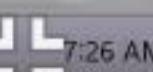
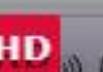
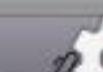
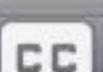
Activate Windows
Go to settings to activate Windows.



49:57 / 1:13:34



Q Search



7:26 AM

1. //input[@id='txtUsername']
2. //input[@name='txtUsername']

This is one of the most commonly used Xpath. Most of the plugs can generate above XPaths automatically

Locating Elements with static Visible Text (exact match)

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

Syntax

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

Let's consider locating following hyperlink



Examples

```
1. //a{text()}'Dremel'
```

Examples

1. //a[text()='Pragmatic
 2. //*[@text()='Pragmatic

NOTE: The inner text is case sensitive

Locating elements by the visible text is not advisable

1. If you are testing a multilingual application
 2. When same text is appearing in more than one location

Locating Elements when part of the visible text is static

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.

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.

Session 25 - Selenium with Java | Locators - XPath Axes | SelectorsHub

T

```
<div id="ptl-link">  
  <a target=""_blank"" href="http://www.practicallab.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.

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')]
```

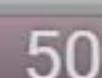
T

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

Examples :

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

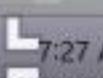
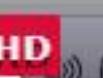
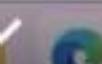
Activate Windows
Go to Settings to activate Windows.



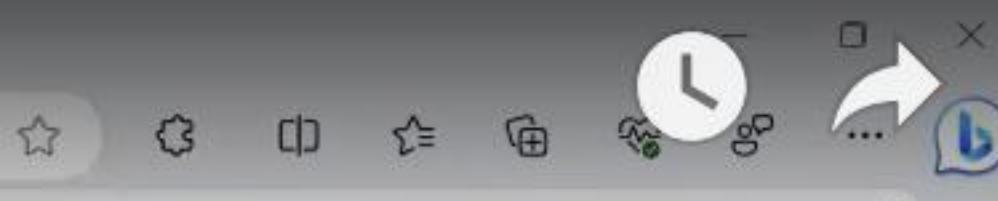
50:18 / 1:13:34



Search



7:27 AM
12/23/2023



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')]
```

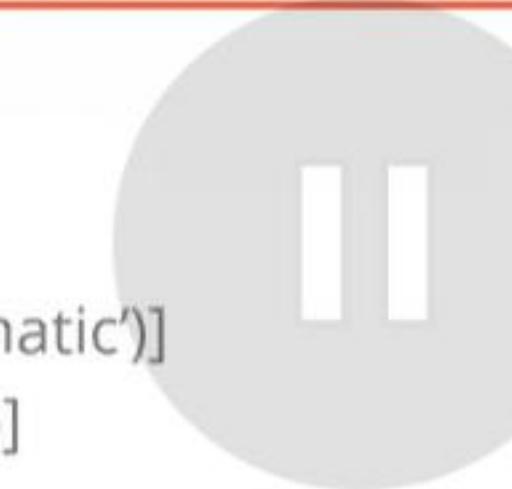
```
//*[starts-with(text(),'Prefix of Inner Text')]
```



```
<div id="pt1-link">  
  <a target="_blank" href="http://www.practicetestlabs.com">Pragmatic Test Labs</a>  
</div>
```

Examples :

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

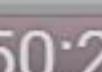
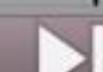


Locating elements with Visible text in input elements

Locating input elements with visible text should not be confused with the locating elements with inner text as in above sections. Attribute value should be used for locating the visible text in input type elements.

Syntax :

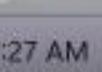
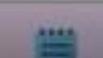
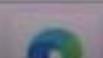
Activate Windows
Go to Settings to activate Windows.

50:20 / 1:13:34



Search



7:27 AM
17/23/2023

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

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

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

First name: Janesh
Last name: Kodikara

Click the "Submit" button and the following page will appear:

Examples

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

We have already discussed the process of locating elements by tag-name and an attribute.

Locating Elements with Multiple Attributes

Session 25 - Selenium with Java | Locators - XPath Axes | SelectorsHub

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

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

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

First name: Janesh

Last name:

Click the "Submit" button and the fo

Examples :

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

We have already discuss the process of locating elements by tag-name and an attribute.

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']`

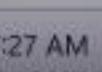
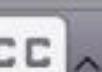
Activate Windows
Go to Settings to activate Windows.



50:37 / 1:13:34



Search



7:27 AM
17/23/2023



We have already discuss the process of locating elements by tag-name and an attribute.

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 :

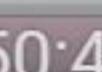
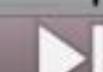
```
//*[attribute1='value1'][attribute2='value2']...[attributeN='valueN'] or  
//tagName[attribute1='value1'][attribute2='value2']...[attributeN='valueN'] or  
//*[attribute1='value1' and attribute2='value2']  
//tagName[attribute1='value1' and attribute2='value2']
```

```
▼<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']

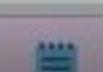
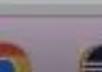
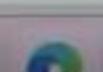
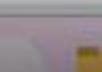
Activate Windows
Go to Settings to activate Windows.



50:40 / 1:13:34



Search

7:27 AM
12/23/2023

Examples

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

Locating Elements with Dynamic Attribute values

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 :



Locating Elements with Dynamic Attribute values

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="pt1-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')]

Activate Windows
Go to Settings to activate Windows.



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 relative to known element

If our **target** elements do not have unique attribute(s) or static innerHTML, we need to locate the elements with respective to an element who has an **unchanging XPath**.

We do this very well in our real life. We always give direction to an unknown location with respect to a well known location (a **landmark**). Giving direction to your home from a well-known shop, statue, railway station etc.

Session 25 - Selenium with Java | Locators - XPath Axes | SelectorsHub

If our **target** elements do not have unique attribute(s) or static innerHTML, we need to locate the elements with respect to an element who has an **unchanging XPath**.

We do this very well in our real life. We always give direction to an unknown location with respect to a well known location (a **landmark**). Giving direction to your home from a well-known shop, statue, railway station etc.

XPath has thirteen (**13**) different axes. We will look into a subset of useful axes in this blog post which can be used with Selenium.

Your target element should be closer to the known element (**context element**) to make XPath shorter, resilient and readable.

Activate Windows
Go to Settings to activate Windows.