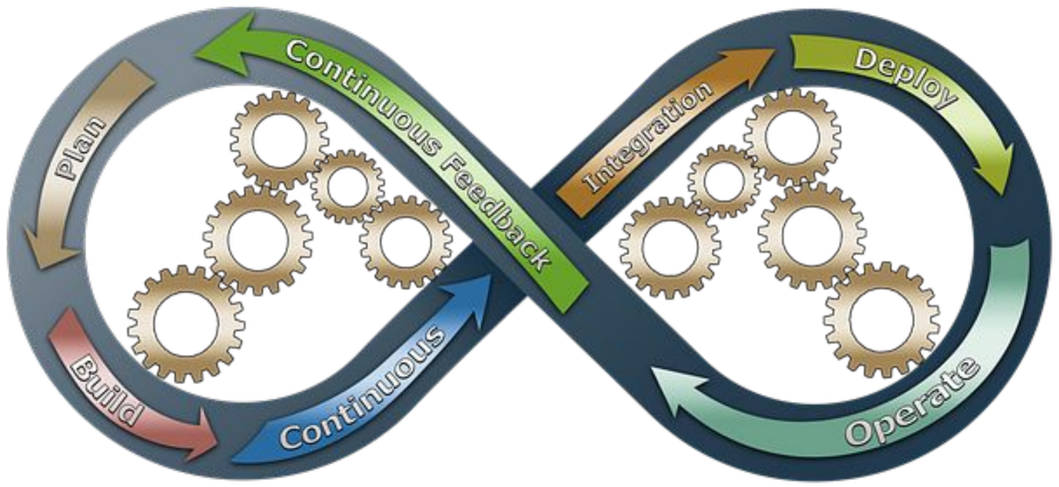


Session 4.4

Locators

AN INITIATIVE BY

UNICAL ACADEMY

UNICAL ACADEMY

Introduction

Let’s go!!!

Locators

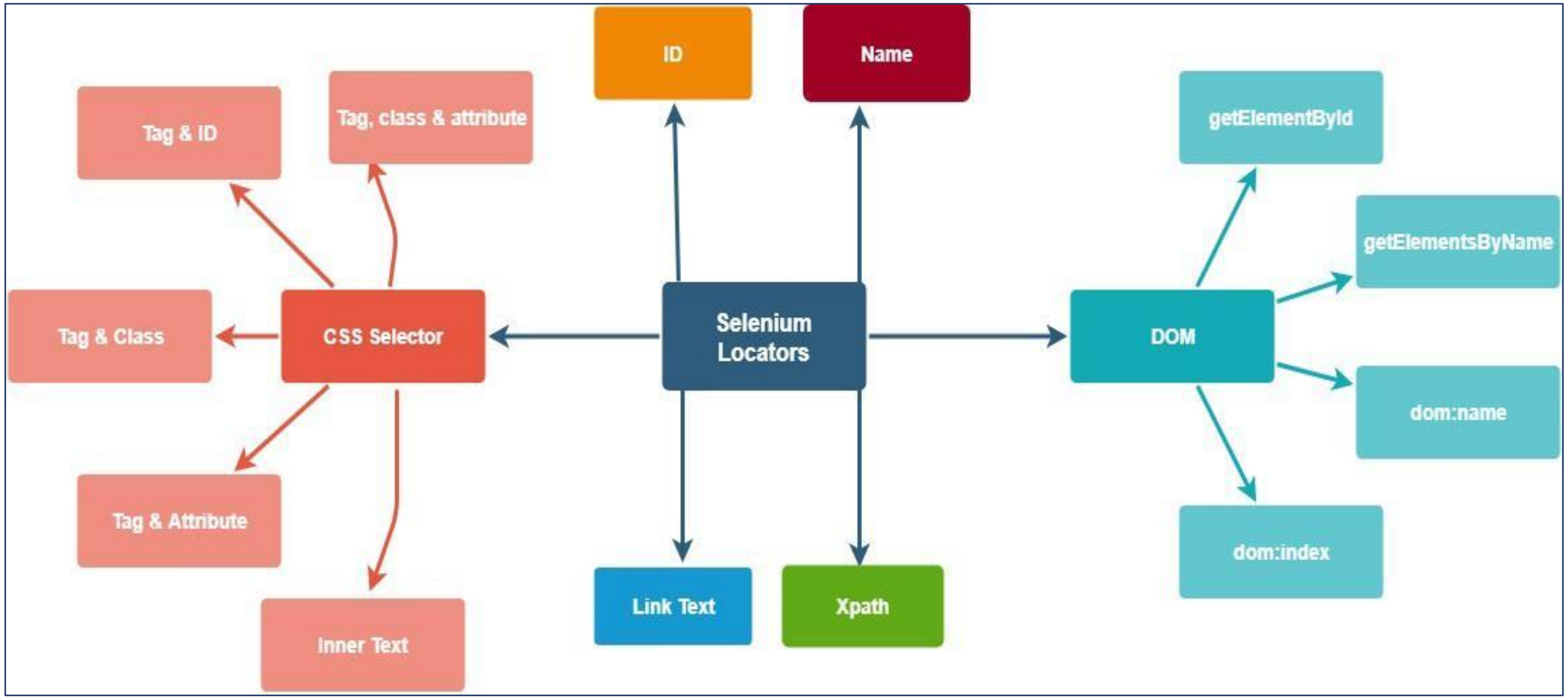
- The locator can be termed as an address that identifies a web element uniquely within the webpage. Locators are the HTML properties of a web element which tells the Selenium about the web element it needs to perform the action on.

- **Web elements:**
  - Text box
  - Button
  - Drop Down
  - Hyperlink
  - Check Box
  - Radio Button

- **Locator Types:**
  - ID
  - ClassName
  - Name
  - Link Text
  - Xpath
  - CSS Selector

UNICAL ACADEMY

DOM & Locators in Selenium



Object Identification

- Object Identification helps to handle a dynamic list of object id changes, based on the number of rows. It also provides better validation options for the list of objects in one screen.
- It finds elements where the relative XPath is different across devices.
- It helps you get dynamic data from the screen.

UNICAL ACADEMY

Find Element Vs Find Elements

Find Element

- Returns the first most web element if there are multiple web elements found with the same locator
- Throws exception no Such Element Exception if there are no elements matching the locator strategy
- Find element by XPath will only find one web element
- Not Applicable

Find Elements

- Returns a list of web elements
- Returns an empty list if there are no web elements matching the locator strategy
- It will find a collection of elements whose match the locator strategy.
- Each Web element is indexed with a number starting from 0 just like an array

Syntax :

```
WebElement elementName = driver.findElement
            (By.LocatorStrategy("LocatorValue"));
```

Syntax to Declare Multidimensional Array:

```
List<WebElement> elementName =
driver.findElements(By.LocatorStrategy("LocatorValue"));
```

UNICAL ACADEMY

XPath

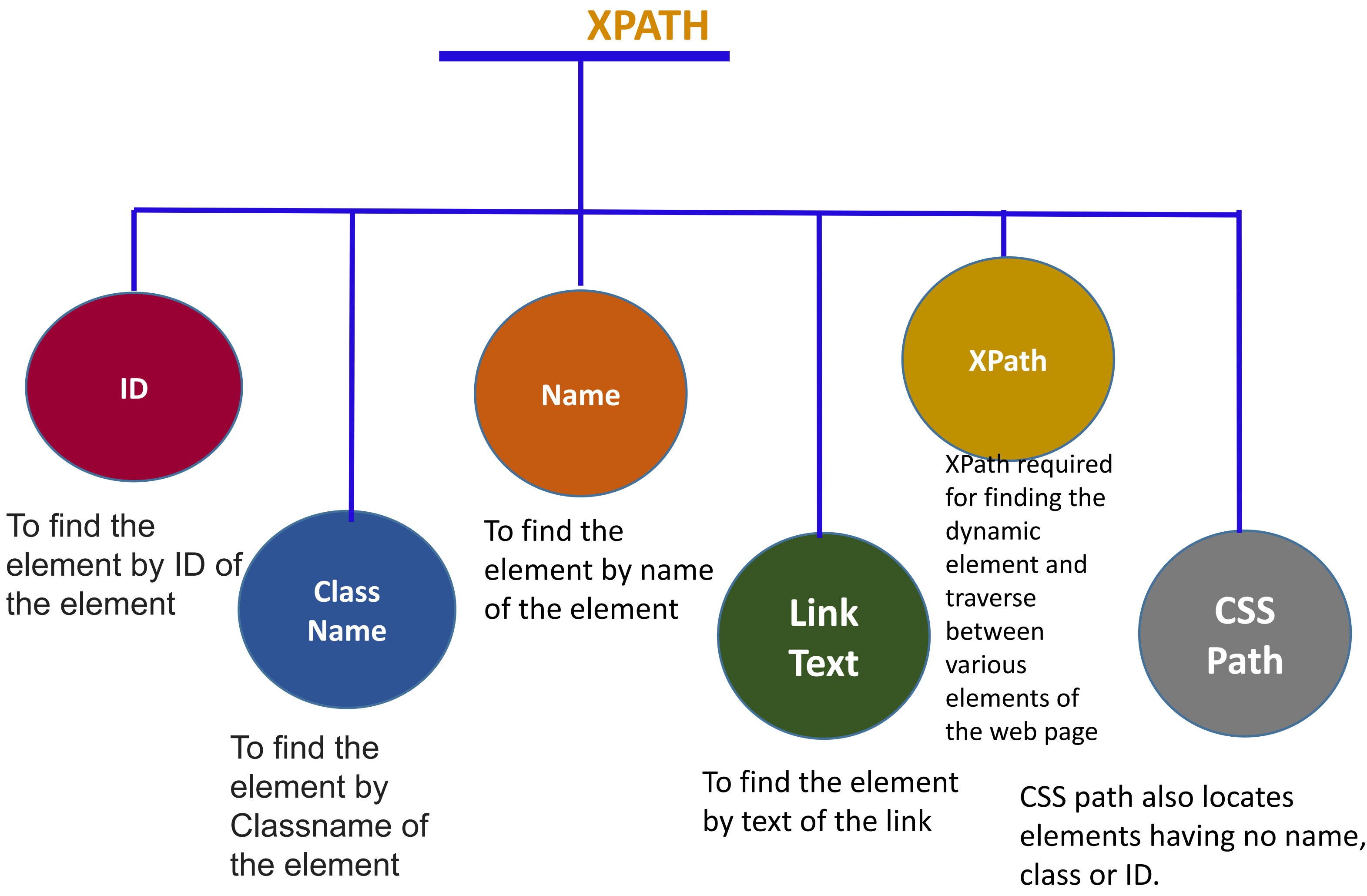
XPath

XPath in Selenium is an XML path used for navigation through the HTML structure of the page. It is a syntax or language for finding any element on a web page using XML path expression.

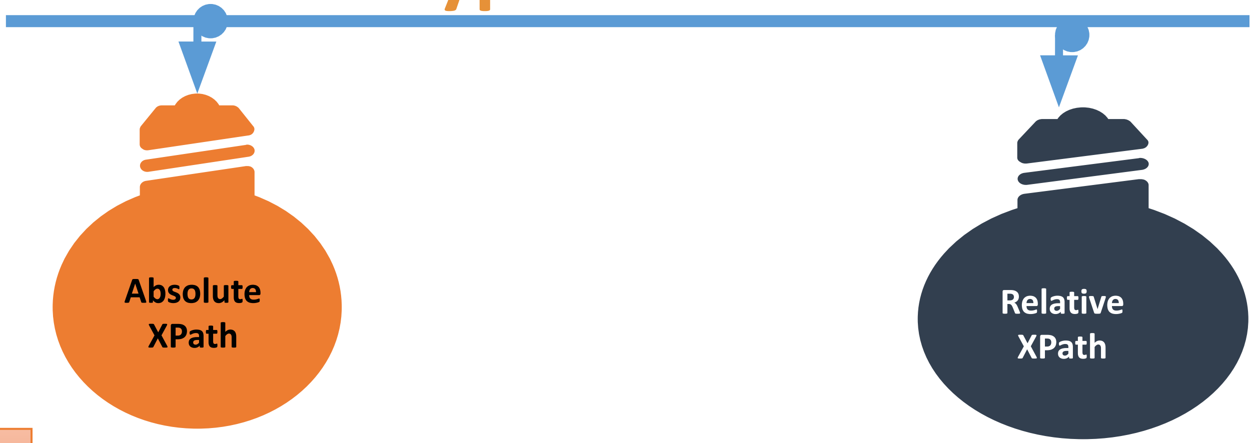
Syntax :

```
Xpath=//tagname[@attribute='value']
```

- **//** : Select current node.
- **Tagname:** Tagname of the particular node.
- **@:** Select attribute.
- **Attribute:** Attribute name of the node.
- **Value:** Value of the attribute


XPATH
ID
To find the element by ID of the element
Class Name
To find the element by Classname of the element
Name
To find the element by name of the element
Link Text
To find the element by text of the link
XPath
XPath required for finding the dynamic element and traverse between various elements of the web page
CSS Path
CSS path also locates elements having no name, class or ID.

Types of XPath



Absolute XPath

It is the direct way to find the element, but 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(/)

Syntax :

```
/html/body/div[2]/div[1]/div/h4[1]/b/html[1]/body[1]/div[2]/div[1]/div[1]/h4[1]/b[1]
```

Relative XPath

Relative Xpath starts from the middle of HTML DOM structure. It starts with double forward slash (//). It can search elements anywhere on the webpage, means no need to write a long xpath and you can start from the middle of HTML DOM structure.

Syntax to Declare Multidimensional Array:

```
Relative XPath: //div[@class='featured-box cloumnsize1']//h4[1]//b[1]
```

Cascading Style Sheet (CSS) Selectors

CSS Selectors

- **CSS Selectors** are one of the locator strategies offered by Selenium to identify the web elements.
- The **CSS Selectors** mainly use the character sequence pattern, which identifies the web elements based on their HTML structure.

Syntax :

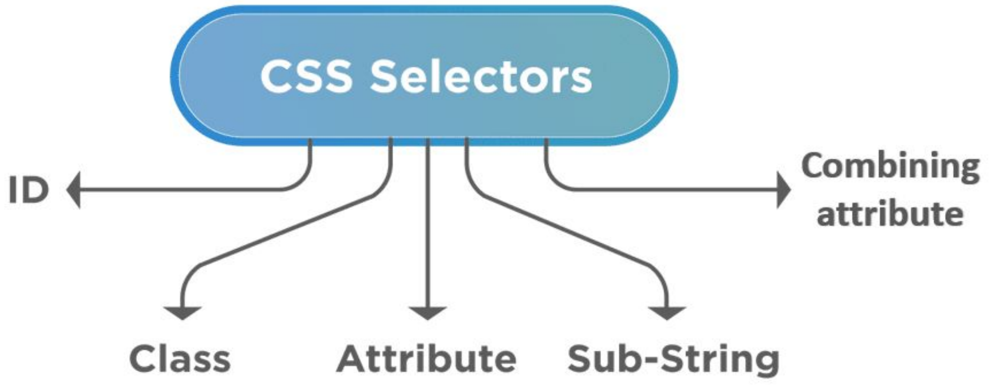
```
node[attribute_name = ‘attribute_value’]
```

- node is the tag name of the HTML element, which needs to locate.
- attribute_name is the name of the attribute which can locate the element.
- attribute_value is the value of the attribute, which can locate the element.

Cascading Style Sheet (CSS) Selectors

UNICAL ACADEMY

Different ways to Create CSS Selectors



ID

ID in CSS Selector to identify and locate a web element.

Syntax :

```
css=<HTML tag><#><Value of ID attribute>
```

- **HTML tag**: used to denote the web element to be accessed
- **#**: used to symbolize the ID attribute. Note that the hash is mandatory in cases where ID attribute is used to create a CSS Selector.
- **Value of ID attribute**: the value of the ID attribute being accessed. This value is always preceded by the hash.

UNICAL ACADEMY

Cascading Style Sheet (CSS) Selectors

Class

The ***class*** attribute of the *HTML* tags can also identify the elements on a Web Page.

Syntax :

```
css=<HTML tag><.><Value of Class attribute>
```

- **.** : The dot is used to symbolize the **Class attribute**. Note that the dot is mandatory in cases where a Class attribute is used to create a CSS Selector. The value of the Class is always preceded by a dot.

Attribute

Apart from the ***id*** and ***class*** attributes, all other attributes present within the *HTML* tag of the element can also be used to locate web elements using the *CSS Selectors*

Syntax :

```
css=<HTML tag><[attribute=Value of attribute]>
```

- **Attribute**: Used to create the CSS Selector. It can be a value, type, name, etc. It is best to select an attribute with a value that uniquely identifies the web element being accessed.
- **Value of attribute**: the value of the attribute that is being accessed.

UNICAL ACADEMY

Cascading Style Sheet (CSS) Selectors

SubString

In Selenium, CSS allows the matching of a partial string which, offers a way to create CSS selectors utilizing sub-strings.

This can be done in three ways.

1) Matching a prefix
2) Matching a suffix
3) Matching a Substring

Prefix

The purpose of this is to correspond to the string by using a matching prefix.

Syntax :

```
css=<HTML tag><[attribute^=prefix of the
string]>
```

- **^** : the symbol used to match a string using a prefix
- **Prefix**: the string on the basis of which the match operation is performed.

Cascading Style Sheet (CSS) Selectors

Suffix

The purpose of this is to correspond to the string by using a matching Suffix.

Syntax :

```
css=<HTML tag><[attribute$=suffix of the string]>
```

- **#**: the symbol used to match a string using a suffix.
- **Suffix:** the string on the basis of which the match operation is performed.

Substring

The purpose of this is to correspond to the string by using a matching substring.

Syntax :

```
css=<HTML tag><[attribute*=sub string]>
```

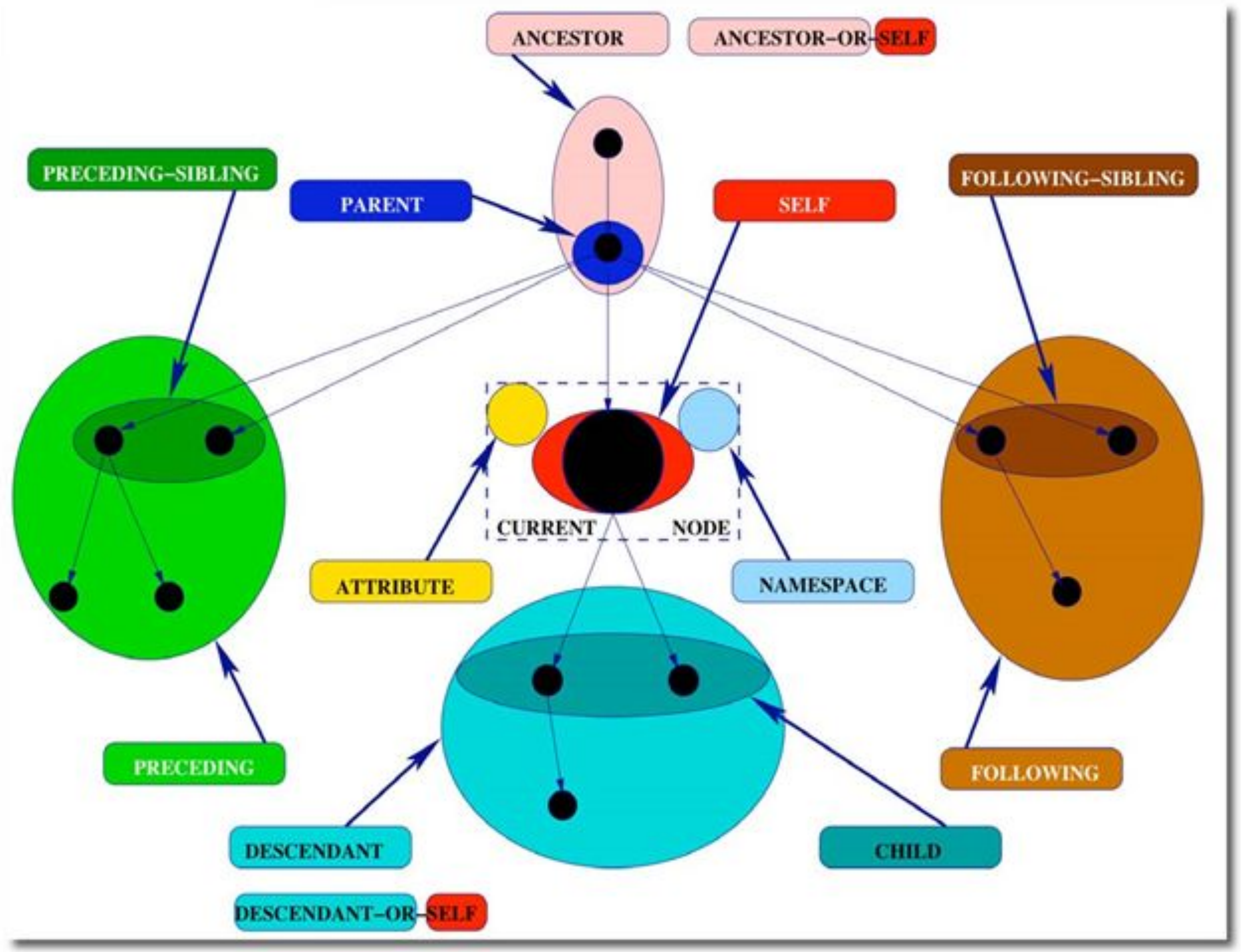
- *****: the symbol to match a string using sub-string
- **Sub string**: the string on the basis of which the match operation is performed.

UNICAL ACADEMY

Dynamic XPath

Dynamic XPath

- Dynamic XPath is also called as custom XPath and it is one way to locate element uniquely.
- Dynamic XPath is used to locate exact attribute or decrease the number of matching nodes/result from a webpage and following XPath expressions can be used for the same:



Dynamic XPath

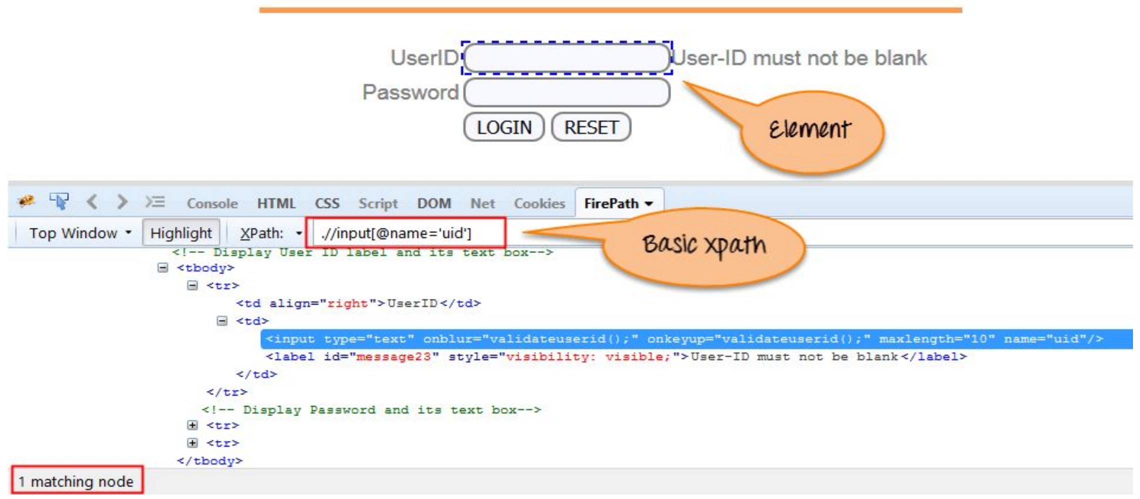
UNICAL ACADEMY

Basic XPath

XPath expression select nodes or list of nodes on the basis of attributes like **ID, Name, Classname**, etc. from the XML document as illustrated below.

Syntax :

```
Xpath=//input[@name='uid']
```



```
.//input[@name='uid']
```

```
<!-- Display User ID label and its text box-->
⊟ <tbody>
  ⊟ <tr>
      <td align="right">UserID</td>
    ⊟ <td>
        <input type="text" onblur="validateuserid();" onkeyup="validateuserid();" maxlength="10" name="uid"/>
        <label id="message23" style="visibility: visible;">User-ID must not be blank</label>
      </td>
    </tr>
   <!-- Display Password and its text box-->
  ⊞ <tr>
  ⊞ <tr>
 </tbody>
```

Contains()

Contains() is a method used in XPath expression. Contains is used to locate the web element who matches the specific text from multiple blocks.

Example :

```
.//*[@class='product']//h4[contains(text(),'Text')]//ancestor::div[@class='table-good']
.//*[@class='product']//h4[contains(.,'Text')]//ancestor::div[@class='table-good']
```

UNICAL ACADEMY

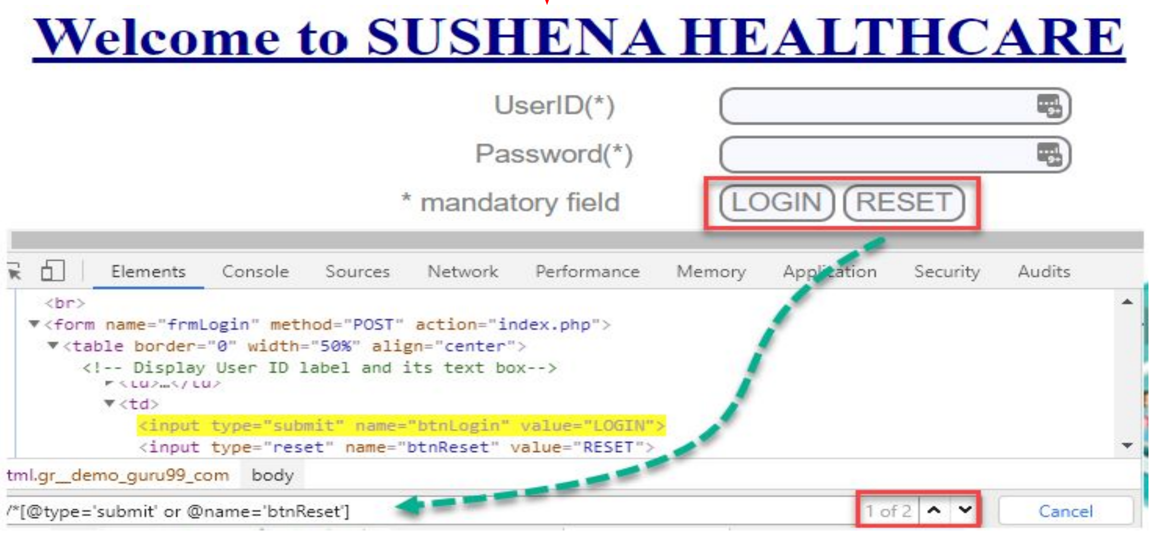
Dynamic XPath

Using OR & AND

In OR expression, two conditions are used, whether 1st condition OR 2nd condition should be true. It is also applicable if any one condition is true or maybe both. Means any one condition should be true to find the element.

Example :

Xpath=//*[@type='submit' or @name='btnReset']



Welcome to SUSHENA HEALTHCARE

UserID(*)
Password(*)
* mandatory field
LOGIN RESET

Elements Console Sources Network Performance Memory Application Security Audits

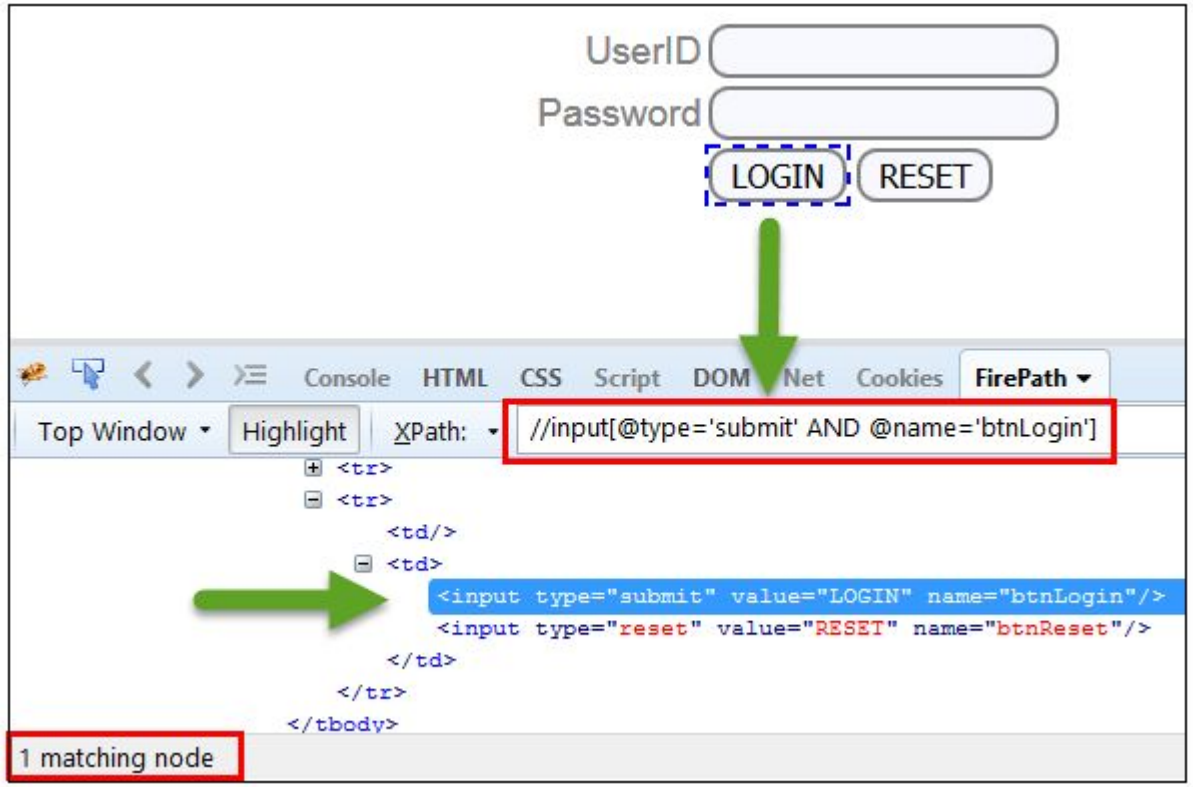
```
<br>
▼<form name="frmLogin" method="POST" action="index.php">
  ▼<table border="0" width="50%" align="center">
    <!-- Display User ID label and its text box-->
    ▸<td>…</td>
    ▼<td>
      <input type="submit" name="btnLogin" value="LOGIN">
      <input type="reset" name="btnReset" value="RESET">
```

tml.gr__demo_guru99_com body

/*[@type='submit' or @name='btnReset'] 1 of 2 Cancel

Example :

Xpath=//input[@type='submit' and @name='btnLogin']



UserID
Password
LOGIN RESET

Console HTML CSS Script DOM Net Cookies FirePath

Top Window Highlight XPath: //input[@type='submit' AND @name='btnLogin']

```
⊞ <tr>
⊟ <tr>
    <td/>
  ⊟ <td>
      <input type="submit" value="LOGIN" name="btnLogin"/>
      <input type="reset" value="RESET" name="btnReset"/>
    </td>
  </tr>
</tbody>
```

1 matching node

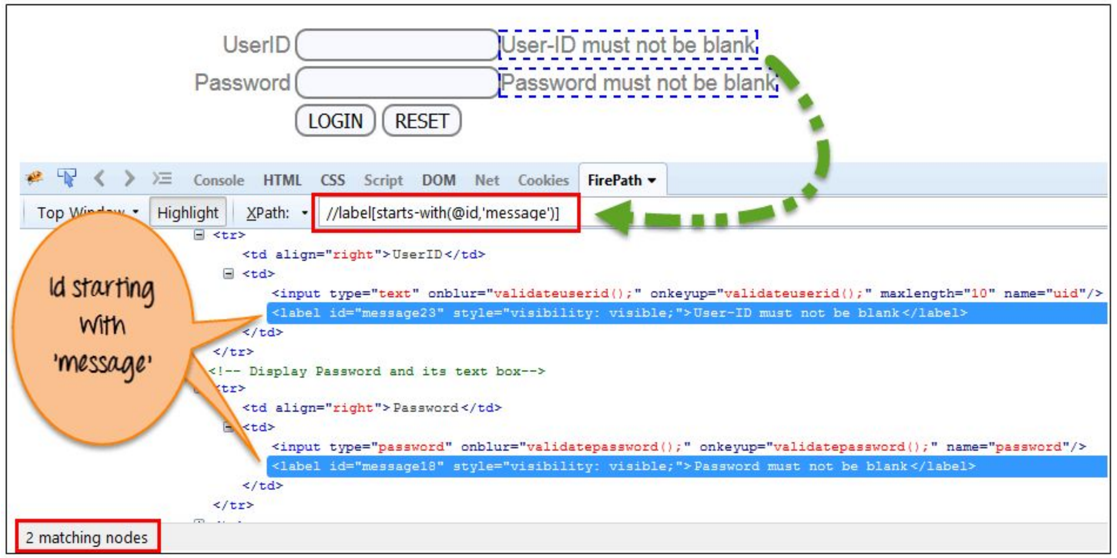
Dynamic XPath

Xpath Starts-with

XPath starts-with() is a function used for finding the web element whose attribute value gets changed on refresh or by other dynamic operations on the webpage. In this method, the starting text of the attribute is matched to find the element whose attribute value changes dynamically.

Example :

```
Xpath=//label[starts-with(@id,'message')]
```



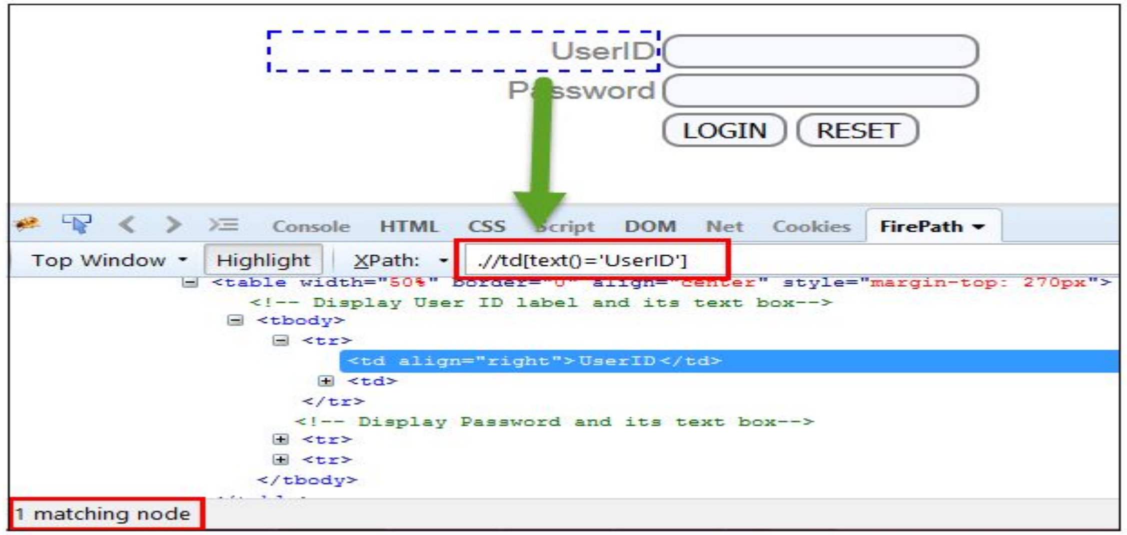
```
//label[starts-with(@id,'message')]
<tr>
  <td align="right">UserID</td>
  <td>
    <input type="text" onblur="validateuserid();" onkeyup="validateuserid();" maxlength="10" name="uid"/>
    <label id="message23" style="visibility: visible;">User-ID must not be blank</label>
  </td>
</tr>
<!-- Display Password and its text box-->
<tr>
  <td align="right">Password</td>
  <td>
    <input type="password" onblur="validatepassword();" onkeyup="validatepassword();" name="password"/>
    <label id="message18" style="visibility: visible;">Password must not be blank</label>
  </td>
</tr>
```

XPath Text()

The **XPath text() function** is a built-in function of selenium webdriver which is used to locate elements based on text of a web element. It helps to find the exact text elements and it locates the elements within the set of text nodes. The elements to be located should be in string form.

Example :

```
Xpath=//td[text()='UserID']
```



```
.//td[text()='UserID']
<table width="50%" border="0" align="center" style="margin-top: 270px">
<!-- Display User ID label and its text box-->
<tbody>
  <tr>
    <td align="right">UserID</td>
    <td>
  </tr>
  <!-- Display Password and its text box-->
  <tr>
  <tr>
</tbody>
```

Dynamic XPath

UNICAL ACADEMY

XPath Axes methods

S. No	Xpath Axes	Description	Example
1	**Following**	Selects all elements in the document of the current node() [UserID input box is the current node] as shown in the below screen.	Xpath=//*[@type='text']//following::input
2	**Ancestor**	The ancestor axis selects all ancestors element (grandparent, parent, etc.) of the current node as shown in the below screen.	Xpath=//*[text()='Enterprise Testing']//ancestor::div
3	**Child**	Selects all children elements of the current node (Java) as shown in the below screen.	Xpath=//*[@id='java_technologies']//child::li
4	**Preceding**	Select all nodes that come before the current node as shown in the below screen.	Xpath=//*[@type='submit']//preceding::input
5	**Following-sibling**	Select the following siblings of the context node. Siblings are at the same level of the current node as shown in the below screen. It will find the element after the current node.	xpath=//*[@type='submit']//following-sibling::input
6	**Parent**	Selects the parent of the current node as shown in the below screen.	Xpath=//*[@id='rt-feature']//parent::div
7	**Self**	Selects the current node or 'self' means it indicates the node itself as shown in the below screen.	Xpath =//*[@type='password']//self::input
8	**Descendant**	Selects the descendants of the current node as shown in the below screen.	Xpath=//*[@id='rt-feature']//descendant::a

Objects Identification in CSS Selectors

UNICAL ACADEMY

Pro's

- Often more robust than XPath – less likelihood of parent objects or structure causing problems for individual objects
- Often more concise than XPath
- Some evidence that they are faster than XPath
- More widespread browser support than XPath

CON'S

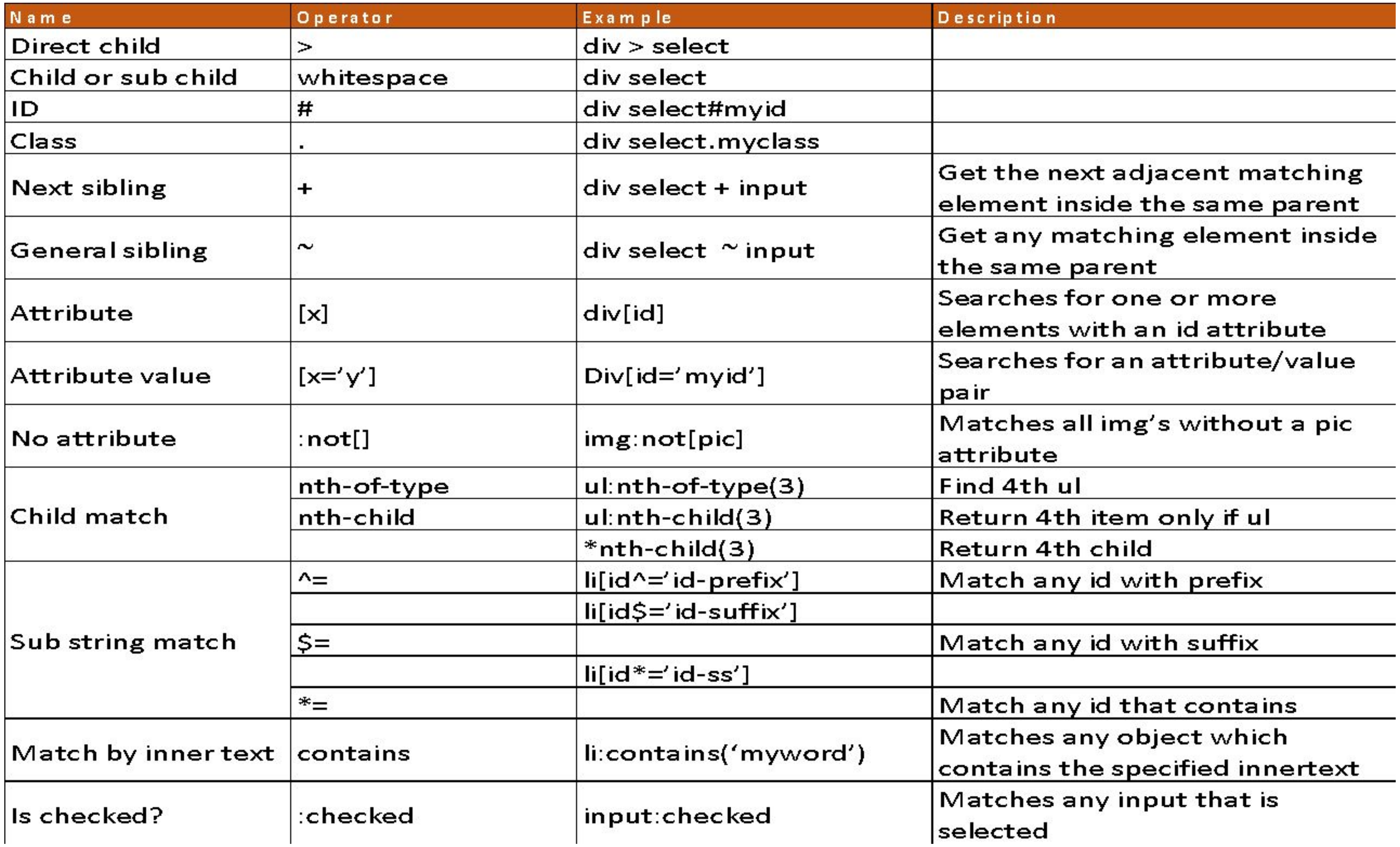
- Trickier to learn and master than XPath

Session 4.4

Objects Identification in CSS Selectors

UNICAL ACADEMY

Operators



Name	Operator	Example	Description
Direct child	>	div > select	
Child or sub child	whitespace	div select	
ID	#	div select#myid	
Class	.	div select.myclass	
Next sibling	+	div select + input	Get the next adjacent matching element inside the same parent
General sibling	~	div select ~ input	Get any matching element inside the same parent
Attribute	[x]	div[id]	Searches for one or more elements with an id attribute
Attribute value	[x='y']	Div[id='myid']	Searches for an attribute/value pair
No attribute	:not[]	img:not[pic]	Matches all img's without a pic attribute
Child match	nth-of-type	ul:nth-of-type(3)	Find 4th ul
	nth-child	ul:nth-child(3)	Return 4th item only if ul
		*nth-child(3)	Return 4th child
Sub string match	^=	li[id^='id-prefix']	Match any id with prefix
		li[id$='id-suffix']	
	$=		Match any id with suffix
		li[id*='id-ss']	
	*=		Match any id that contains
Match by inner text	contains	li:contains('myword')	Matches any object which contains the specified innertext
Is checked?	:checked	input:checked	Matches any input that is selected

Objects Identification in XPath

UNICAL ACADEMY

Pro's

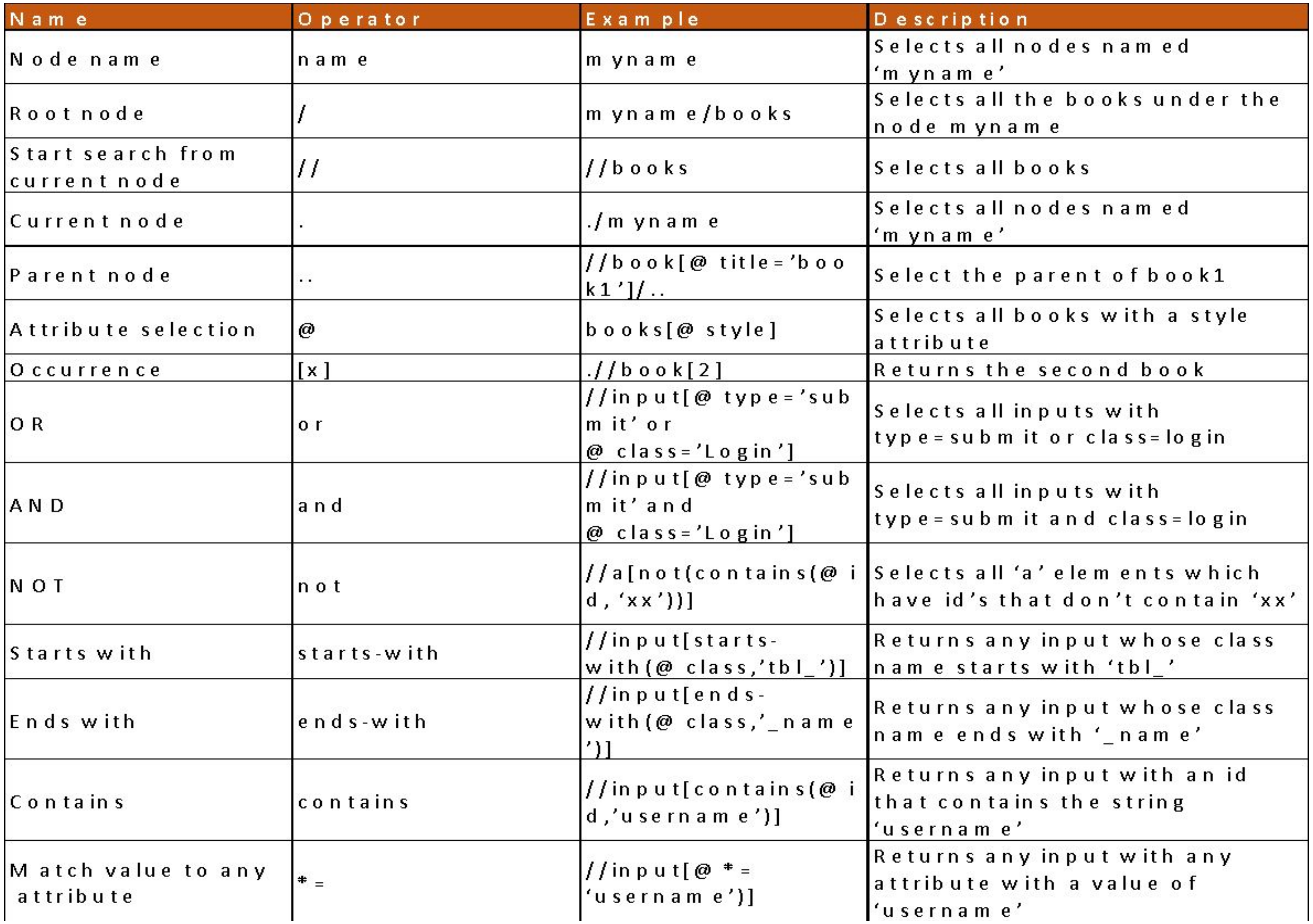
- Easy to understand – it's easy to follow the Xpath in the DOM
- Precision – Can uniquely identify any object
- More text recognition operators than CSS

CON'S

- Can get very long and unwieldy
- Some browsers have limited XPath support
- More brittle than CSS selectors in many circumstances

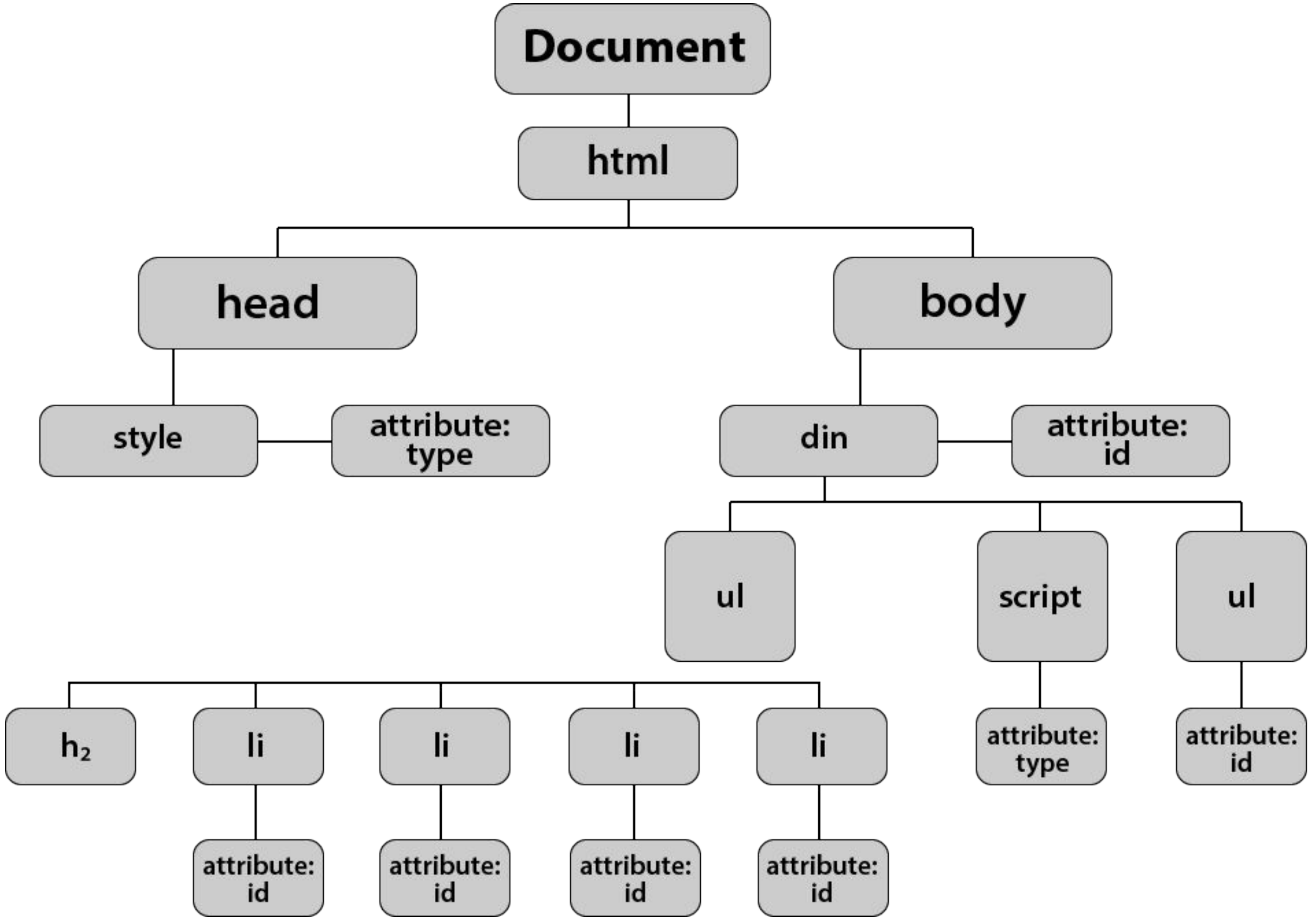
Objects Identification in XPath

UNICAL ACADEMY



Name	Operator	Example	Description
Node name	name	myname	Selects all nodes named 'myname'
Root node	/	myname/books	Selects all the books under the node myname
Start search from current node	//	//books	Selects all books
Current node	.	./myname	Selects all nodes named 'myname'
Parent node	..	//book[@ title='book1']/..	Select the parent of book1
Attribute selection	@	books[@ style]	Selects all books with a style attribute
Occurrence	[x]	.//book[2]	Returns the second book
OR	or	//input[@ type='submit' or @ class='Login']	Selects all inputs with type=submit or class=login
AND	and	//input[@ type='submit' and @ class='Login']	Selects all inputs with type=submit and class=login
NOT	not	//a[not(contains(@ id, 'xx'))]	Selects all 'a' elements which have id's that don't contain 'xx'
Starts with	starts-with	//input[starts-with(@ class,'tbl_')]	Returns any input whose class name starts with 'tbl_'
Ends with	ends-with	//input[ends-with(@ class,'_name')]	Returns any input whose class name ends with '_name'
Contains	contains	//input[contains(@ id,'username')]	Returns any input with an id that contains the string 'username'
Match value to any attribute	*=	//input[@ *= 'username')]	Returns any input with any attribute with a value of 'username'

DOM Structure



UNICAL ACADEMY

Session Recap