



TECHNO
STUDY +

SELENIUM XPATH

Most powerful selector

PRIMARY DIFFERENCE

- XPath we can traverse both forward and backward whereas CSS selector only moves forward
- XPath can search by text, using contains functionality

GENERAL LOCATOR STRATEGY

1. ID first
2. CSS selector for everything else
3. When nothing else works, use XPath

WHAT IS XPATH?

- XPath is defined as **XML path**.
- XPath is used to find the location of any element on a webpage using HTML DOM structure.

TYPES OF X-PATH

There are two types of XPath:

1) Absolute XPath

2) Relative XPath

Abstract teal shapes are located in the bottom left corner of the slide, including a large light teal shape and a smaller dark teal shape.

ABSOLUTE XPATH:

/html/body/div[2]/div/section/
figure/figcaption/div[2]

RELATIVE XPATH:

//*[@id= 'contact_form']

```
<!DOCTYPE html>
<html lang="en">
  ><head>...</head>
  ><body>
    ><div class="wrapper row1">...</div>
    ><div class="wrapper row2">
      ><div id="container" class="clear">
        ><!-- Slider -->
        ><section id="slider" class="clear center">
          ><figure>
            ><figcaption>
              ><h2>Contact Us</h2>
              ><div class="small">...</div>
              ><div id="contact_form"> == $0
                "
                Username: "
                <input type="text" id="username">
                <br>
                "
                Password: "
                <input type="password" id="password">
                <br>
```

BASIC FORMAT OF RELATIVE XPATH

TECHNO
STUDY +

Select current
node

Selects
Attribute

Value of the
attribute

Xpath=//tagname[@Attribute='Value']

Tagname like Input,
Div, Img etc.

Attribute
Name

BASIC XPATH

Xpath=//input[@type='text']

Xpath=//label[@id='message23']

Xpath=//input[@value='RESET']

Xpath=//*[@class='barone']

Xpath=//a[@href='http://demo.guru99.com/']

Xpath= //img[@src='//cdn.guru99.com/images/home/java.png']

CONTAINS():

Xpath=//*[contains(@type,'sub')] type inde sub kelimesi geçen

Xpath=//*[contains(@name,'btn')]

Xpath=//*[contains(@id,'message')]

Xpath=//*[contains(@href,'guru99.com')]

Xpath=//h2[contains(text(),'Popular right now')]

STARTS-WITH FUNCTION

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

ENDS-WITH FUNCTION

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

The ends-with function is part of xpath 2.0 but browsers generally only support 1.0

USING OR & AND

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

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

Xpath =//span[@role='menuitem' or @class='text-gray-darker']

Or = || in java

And = && in java

Xpath =//a[@id='item_0_title_link' or @href='./inventory-item.html?id=0']

TEXT()

Xpath= `//*[text()='Accepted usernames are:']` ekranda gözüken yani tag lerinin arasında olan yazı

```

<div class="login_credentials_wrap">
  ▼ <div class="login_credentials_wrap-inner">
    ▼ <div id="login_credentials" class="login_credentials">
      <h4>Accepted usernames are:</h4> == $0
      "standard_user"
      <br>
  
```

FOLLOWING

following: This function will return the following elements of the particular component.

Xpath=//*[@type='text']//following::input

TECHNO
STUDY +

The screenshot displays the 'Guru99 Bank' login interface. It features two text input fields labeled 'UserID' and 'Password', and two buttons labeled 'LOGIN' and 'RESET'. A blue dashed box highlights the 'Password' field and the 'LOGIN' and 'RESET' buttons. An orange speech bubble points to this box with the text 'Showing 3 Nodes'. Below the browser window, the FirePath extension's XPath editor is visible, showing the XPath expression `//*[@type='text']//following::input` highlighted in a red box. Another orange speech bubble points to this box with the text 'Xpath using following'. The FirePath panel also shows a snippet of HTML code: `<tr>`, `<!-- Display Password and its text box-->`, and `<tr>`.

BY INDEX []

Xpath=//*[@type='text']//following::input[1]

TECHNO
STUDY +

The screenshot displays the Guru99 Bank login interface. It features a header with the text "Guru99 Bank" in an orange bar. Below the header, there is a login form with two input fields: "UserID" and "Password". The "Password" field is highlighted with a blue dashed border. Below the input fields are two buttons: "LOGIN" and "RESET". A speech bubble with the text "Showing particular Node" points to the "Password" field.

Below the login form, the FirePath extension interface is visible. The "XPath" field is set to `//*[@type='text']//following::input[1]`, which is highlighted with a red box. The "Highlight" button is active. The DOM tree shows the following structure:

```
<tr>
  <!-- Display Password and its text box-->
  <tr>
    <td align="right">Password</td>
    <td>
      <input type="password" onblur="validatepassword();" onkeyup="validatepassword();" />
      <label id="message18"/>
    </td>
  </tr>
```

CHILD

Xpath=//*[@id='java_technologies']//child::li

Xpath=//div[@class='cart_list']//a

Selects all children elements of the current node

TECHNO
STUDY +

FOLLOWING-SIBLING (SIBLINGS ON SAME LEVEL)

xpath=//*[@type='submit']//following-sibling::input

The screenshot shows a web form with fields for 'UserID' and 'Password', and two buttons: 'LOGIN' and 'RESET'. The 'RESET' button is highlighted with a blue dashed border. An orange speech bubble points to the 'RESET' button with the text 'xpath using following-sibling'. Another orange speech bubble points to the 'RESET' button with the text '1 Nodes matched'.

Below the form, the FirePath tool is shown. The 'XPath' field contains the expression `//*[@type='submit']//following-sibling::input`, which is highlighted with a red box. The tool shows the HTML structure of the form, with the `<input type="reset" value="RESET" name="btnReset"/>` line highlighted in blue. At the bottom, a red box indicates '1 matching node'.

PRECEDING-SIBLING

xpath=//*[@type='submit']
]
//preceding-sibling::input

.. GOING UP ONE LEVEL

`xpath=//*[@type='submit']/../span`

`Xpath=//*[@text()='ADD TO CART']/parent::div`

TECHNO
STUDY +