XPATH

What is XPATH?

- XPATH is defined as XML path, it is a syntax or language for finding any element on the webpage using XML path expression.
- It is a type of locator and it is a path of an element which is present in the HTML tree.

Note:

- XPATH is also called as the XML path language.
- XPATH is a query language for selecting nodes from an XML document.
- XPATH is used to find the location of any element on a web page using HTML DOM structure.
- XPATH contains the path of the element situated at the web page.

Why XPATH?

Dynamic elements

- The element which is changing dynamically during runtime is called as dynamic element.
  Here, changing refers to change in the position or change in property values.
  In order to handle the dynamic elements we use XPTH.

Backward traversing in not allowed in CSS.

- If you want to identify parent of any element, we need to traverse in a backward direction. CSS doesn't support to traverse in a backward direction.
- XPATH is the solution to identify the parent by traversing in a backward direction.

Let's take an another example

```
<html>
    <body>
        UN<input type="text"/>
        PW<input type="text"/>
    </body>
</html>
```

- In the above sample page, we can't use the CSS selector for identifying the element. Coz it identifies the duplicate value coz both the elements have the same name field.

Another Example:

```
<html>
    <body>
        <div>Login</div>
        <div>forgot Password</div>
    </body>
</html>
```

- In the above example, we have two elements, where both the elements doesn't have any attribute value except the text.
- So in order to identify the element using the text of an element. Using CSS selector is not possible to identify the element. We use XPATH for identifying the element using the text of an element.

XPATH is categorized into two types

1. **Absolute XPATH**
2. **Relative XPATH**

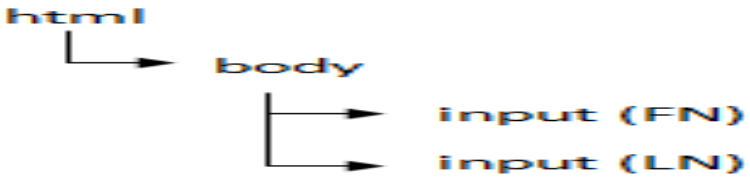
Absolute XPATH

- Specifying complete path of the element form the root till the required element is called as absolute **XPATH**.

Let's take an example of a below HTML.

```
<html>
    <body>
        UN<input type="text"/>
        PW<input type="text"/>
    </body>
</html>
```

HTML tree for the above will look like:



- We write the XPATH expression using '/ '(forward slash). The first forward slash represents beginning of the tree (root).
- After every forward slash we should specify tag of immediate child element. We can also use index which starts from 1.

HTML Code:

```
<html>
    <head>
        <title>Qspiders</title>
    </head>
    <body>
        <center><hr>
            Firstname: <input type="text">
            <br><br><hr>
            Lastname: <input type="text"><hr>
        </center>
    </body>
</html>
```

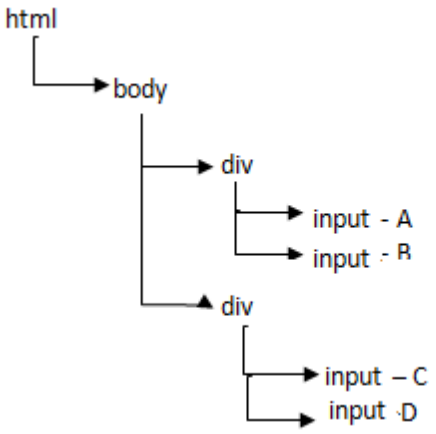
```
from selenium import webdriver

driver = webdriver.Firefox()
driver.maximize_window()
driver.get('file:///C:/Users/PriyaPramod/Desktop/IMP/HTML%20Pages/XPath/Demo1.html')
driver.implicitly_wait(30)

driver.find_element_by_xpath('/html/body/input[1]').send_keys('admin')
driver.find_element_by_xpath('/html/body/input[2]').send_keys('manager')

driver.close()
```

Let's consider the following html tree to derive XPATH expression

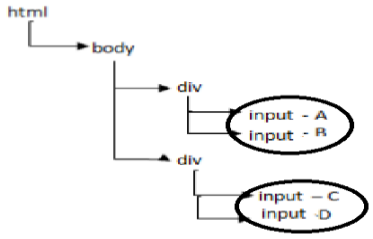


Below is the Absolute XPATH for the above HTML.

Writing the XPATH to identify independent element.

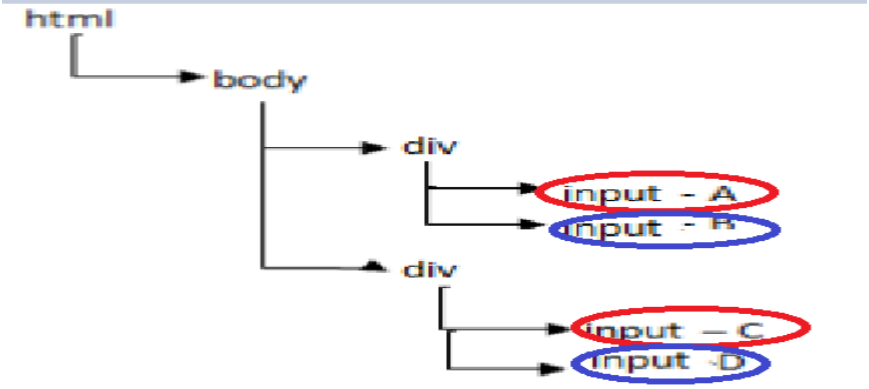
Xpath	Matching Element
/html/body/div[1]/input[1]	A
/html/body/div[1]/input[2]	B
/html/body/div[2]/input[1]	C
/html/body/div[2]/input[2]	D

Xpath to identify A, B and C, D



/html/body/div[1]/input	AB
/html/body/div[2]/input	CD

Xpath to identify A, C and B, D

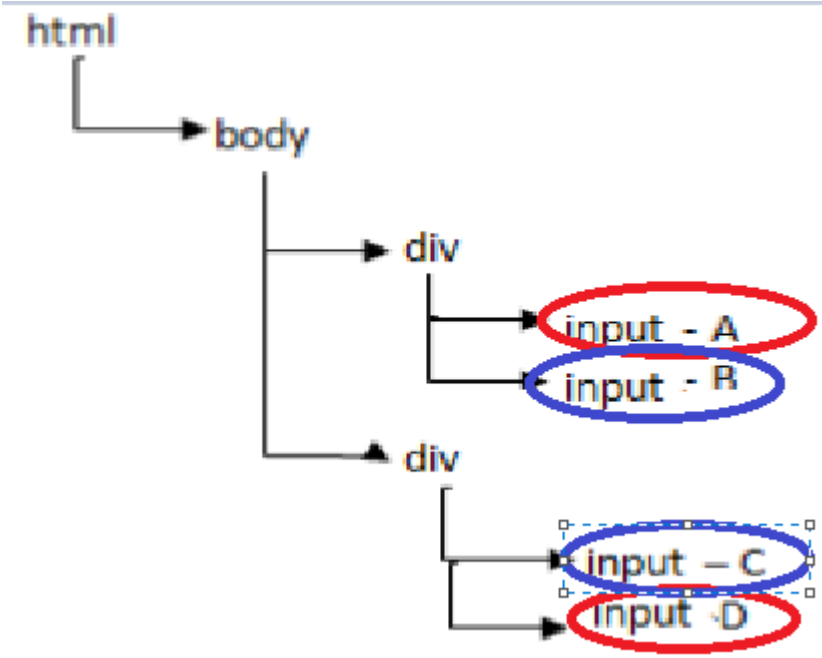


/html/body/div/input[1]	AC
/html/body/div/input[2]	BD

Xpath to identify A, B, C & D

/html/body/div/input	ABCD

Xpath to identify A, D & B, C



- In the above example, consider the scenario where we want to identify A, D & B, C. But both the elements are not in a same position in their respective parent.
- In this case, writing the one XPATH expression to identify both the elements is not possible. Because both the elements are present in the different position.
- So in this case, we derive two XPATH and using the union operator "|", we join two XPATH to make one expression to identify the elements.

/html/body/div[1]/input[1]| /html/body/div[2]/input[2]	AD
/html/body/div[1]/input[2]| /html/body/div[2]/input[1]	BC

- We can join two or more XPATH to make a one expression.

/html/body/div[1]/input[1]| /html/body/div[1]/input[2]| /html/body/div[2]/input[1]	ABC

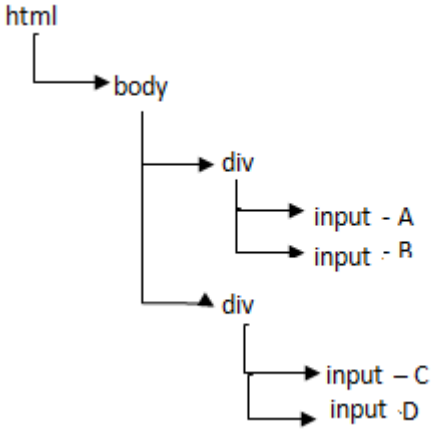
Drawback of absolute XPATH

1. Absolute 'XPATH' is very lengthy. In order to reduce the length of expression we can use relative 'XPATH'.

Relative XPATH

- Specifying the XPATH directly to the web element using web element back end attributes is called as relative XPATH.
- In relative XPATH, we use double forward slash ("//") which represents any child also called as descendent.
- "//" selects node in the document from the current node that match the selection.

Let's take the above HTML to derive the relative XPATH



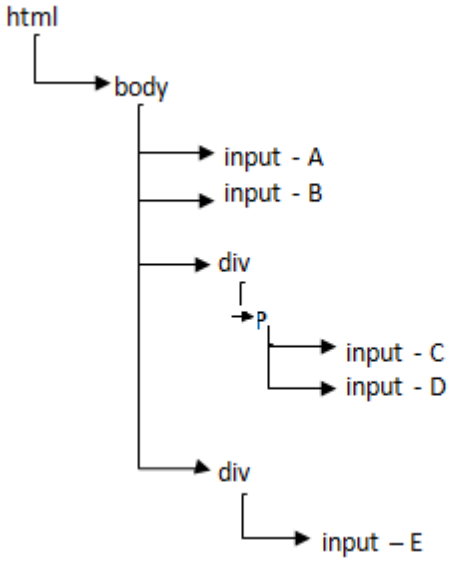
Xpath	Matching element
//div[1]/input[1]	A
//div[1]/input[2]	B
//div[2]/input[1]	C
//div[2]/input[2]	D
//div[1]/input	AB
//div[2]/input	CD
//input[1]	AC
//input[2]	BD
//input	ABCD
//div[1]/input[1]| //div[2]/input[2]	AD
//div[1]/input[2]| //div[2]/input[1]	BC
//div[1]/input[1]| //div[1]/input[2]| //div[2]/input[1]	ABC

Note:

1. **What is the different between single forward slash and double forward slash?**
- Single forward slash represent immediate child whereas double forward slash represents any child (descendent).

2. **What is the difference between '//a' and '//table//a'?**
- '//a' matches with all the links present which are in the entire page. Whereas '//table//a' matches with all the links which are present inside the table.
3. **Derive an 'XPATH' which matches with all the images present on the web page?**
- '//img'
- **Write an 'XPATH' which matches with all the links and all the images present on the web page?**
- '//a|//img'
4. **Important Note:** XPATH matches with hidden elements also.
5. **What is the difference between //input and //div//input?**
- //Input matches with all the inputs present in the entire web page.
- //div//input matches with all the inputs present inside the 'div'.
- //div/input matches with all the immediate child input of 'div'

Below is the HTML code for the above example



//div//input -> CDE
//div/input -> E
//input -> ABCDE
//p/input -> CD

Relative XPATH is categorized into following types

1. XPATH using attribute.
2. XPATH using text function.
3. XPATH using starts-with function.
4. XPATH using normalize-function.
5. XPATH using contains function.
6. XPATH using traversing.
7. XPATH using Independent and dependent.
8. XPATH by group index.
9. XPATH by Axes functions
    - Forward axes function.
        - ✓ Following
        - ✓ Child
        - ✓ Descendants
        - ✓ Following-sibling
    - Backword axes function.
        - ✓ Preceding
        - ✓ Ancestor
        - ✓ Parent
        - ✓ Preceding-Sibling