1) CHILD: 24-02-2017

In xpath, Navigating from one element to another element is called as traversing. To perform this we use xpath axes. In order to move from a parent element to its immediate child element, we use an xpath axes called child.

Ex: /child::html/child::body.

Above xpath is navigating from root element to its child i.e. html and then to its immediate child which is body.

Single forward slash (/) is shortcut for child axes.

⇒ /html/body

2) DESCENDANT: Descendant represents any child of given parent element.

Ex; /descendant::input.

The shortcut for descendant is double forward slash //
⇒ //input - matches with all the inputs

3) PARENT: To navigate from child element to its immediate parent element, we use parent axes. Shortcut is '/..'

⇒ Refer to DemoF.Html.

Ex: //option[text()='Idly']/parent::select

⇒ //option[text()='Idly']/..

4) ANCESTOR: To navigate from child element to any of its parent element, we use an axes called ancestor.

Ex: //option[text()='Idly']/../../..

⇒ //option[text()='Idly']/ancestor::html

5) FOLLOWING-SIBLING: following-sibling represents all the elements present under the same parent but after the current element

Ex: //option[text()='Idly']/following-sibling::option

Idly & Puliyogare & All

#op

//option[text()='Idly']/following-sibling::option[1]

Idly & Puliyogare ONLY

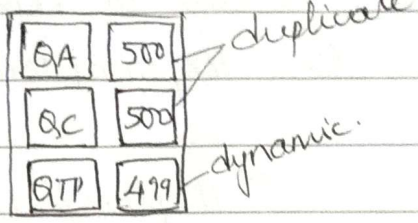
6) PRECEDING-SIBLING: represents all the elements present under the same parent, which are present before the current element.

Ex: //option[text()='Dosa']/preceding-sibling::option ⇒ Idly & Puliyogare (Dosa)

//option[text()='Dosa']/preceding-sibling::option[1] ⇒ Puliyogare (Dosa)

Independent & dependent XPATH ::

```
<table id="t1" border="1">
  <tr>
    <td> QA </td>
    <td> 500 </td>
  </tr>
  <tr>
    <td> QC </td>
    <td> 500 </td>
  </tr>
  <tr>
    <td> QTP </td>
    <td> 499 </td>
  </tr>
</table>
```



I.Q. The above table represent subject & its costs. Some of the costs are duplicate & some of them ~~of~~ are completely dynamic. Identify the cost of QC.

//td[.='500'] - It matches with both cost of QC & QA, where find elements returns first one ie. QA.

In order to identify the cost of QC., first we start with QC itself and then we navigate to respective ~~column~~ row where cost is present.

Ex: //td[text()='QC']/following-sibling::td

This can be also written as //td[.='QC']/../td[2]

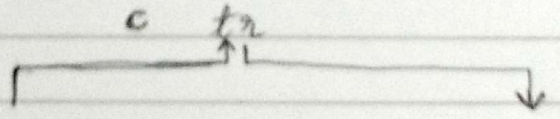
__/__/____

The cost of QTP is not duplicate. but it is completely dynami
To identify such completely dynamic element, we can use same technique ie. navigating from unique element to duplicate or dynamic element. This is also called as Independent - Dependent XPATH.

Steps to write Independent - dependent XPATH:

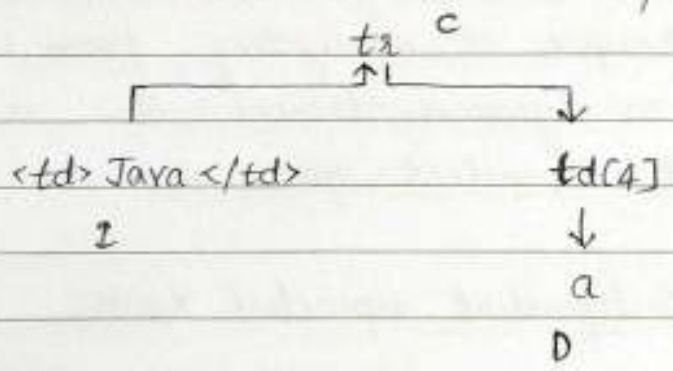
Step 1: Inspect Independent element and notedown its source code.
Step 2: Place the mouse pointer on the source code of independent element. Then move the mouse pointer in up-ward direction till it highlights both Independent & dependent elements. It is called as common parent. Write its path above the source code of Independent element.
Step 3 Use the arrow keys to navigate from common parent to dependent ~~par~~ element, add its path under common parent
Step 4 Derive the xpath from independent element common paren and then to dependent element.

Ex: Identifying the version of python language present in the selenium download page.



→ //td[.='Python']/../td[2]

IQ Derive an xpath to identify 'download' link of java present in selenium download page.



⇒ //td[. = 'Java']/../td[4]/a

⇒ //td[. = 'Java']/..//a[. = 'Download'] → If column is changing

Assgn: Derive an xpath to identify price of SAMSUNG GALAXY ON7 (Gold, 8GB) Mobile phone present in flipkart.com

Q: Derive an xpath to identify checkbox of a task present in task table of actiTIME appln