Object Identification

->In All Selenium tools, Object will be identified using locators, locators will be written looking at the html source of the Webelement

->we have 8 locators in selenium-webdriver, but most of the time we go for “xpath” to identify the object/element

→ in real time application,most of the elements will never find id & name unique attribute , in such cases we have to go for Xapth to identify the object

Xpath :

It’s one of the locator in webdrivertool,which is used to navigate to entire html document & identify the object based on web element attribute& visible text[inner text]

Xpath Symbols

//→ go to entire html document or else Go to descendant Elements

/ -> traverse from parentHTMLTag to childHTMLTag

/.. -> Traverse from childHTMLTag to parentHTMLTag

[] -> provide tag backend attribute

@ -> attribute symbol

*->macth any htmlTag [regular expression]

Xpath keywords

And -> match both the attribute

Or -> match any one of the attribute

Xpathaxes [used for xpathoptmization]

==========

/following-sibling:: → traverse from currentHTMLtag to next sibling htmlTag

/preceding-sibling:: -> traverse from currentHTMLtag to previous sibling htmlTag

/descendant::From the current node skip all sub sub…. child nodes & navigate required descendant node

/ancestor::skip all parents nodes & navigate required super parent node

/child::traverse from parentHTMLTag to childHTMLTag

/parent::Traverse from childHTMLTag to parentHTMLTag

==========Xpath Functions==========

Text()

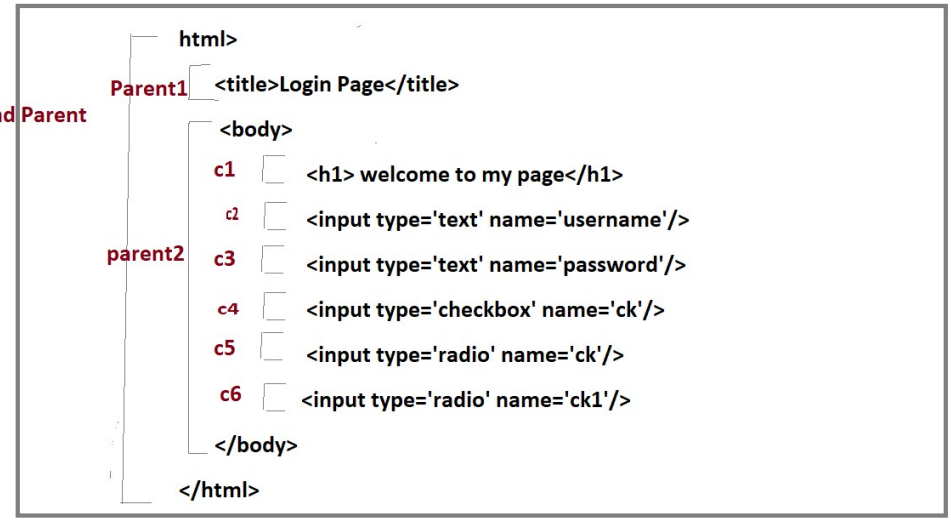
Contains()

Normalize-space()

Last()

There are two types of Xpath

1. **Absolute xpath**
2. **Relative Xpath**



1 Absolute xpath

- Whenever xpath is written from root-Htmltag to child-Htmltag , followed by / is called absolute xpath

1. whenever we copy the xpath from inspect window, its automatically generate Absolute xpath.

- Absolute xpath should not be used in real time selenium test , because it fails to identify the same object , whenever object location or requirement is getting changed

EG:

```
//html/body/input[1]
//html/body/input[2]
```

```
    WebDriver driver = newFirefoxDriver();
driver.get("file:///C:/Users/Raveesh/Desktop/myPage.html");

driver.findElement(By.xpath("//html/body/input[1]")).sendKeys("admin");
driver.findElement(By.xpath("//html/body/input[2]")).sendKeys("manager");
driver.findElement(By.xpath("//html/body/input[3]")).click();
```

UI Changes

```
<html>
  <title>My Page</title>
  <body>
   <div>
        <input type='text' name='username'/>
   </div>
   <div>
        <input type='text' name='password'/>
   </div>
   <div>
         <input type='checkbox' name='ck'/>
         <input type='radio' name='ck'/>
         <input type='radio' name='ck1'/>
  </div>
 </body>
```

2 Relative Xpath

. whenever xpath written directly to the webelement using webelement attribute or visible text is called relative xpath

➔Relative will never get failed, even though object location / requirement is getting changed

⇨ Because of Agile Methodology better to go for relative Xpath

Syntax

//htmlTag[@attribute='value']

//input[@name='username']

//input[@name='password']

Case 1: how to identify the object using One attributes

Syntax

//htmlTag[@attribute='value']

⇨ Xpath is written directly to the element using unique attribute

EG:

//input[@name='username']

//input[@name='password']

Case 2 : how to identify the object using multiple attributes

//htmlTag[@attribute1='value' and @attribute2='value']

//input[@name='ck' and @type='checkbox']

⇨ when object not able to find the element usingoneattribute , we can go for multiple attribute using and keyword

---

Case 3 : how to identiy the object using visible text

<htmlTag> java</htmlTaag>

|

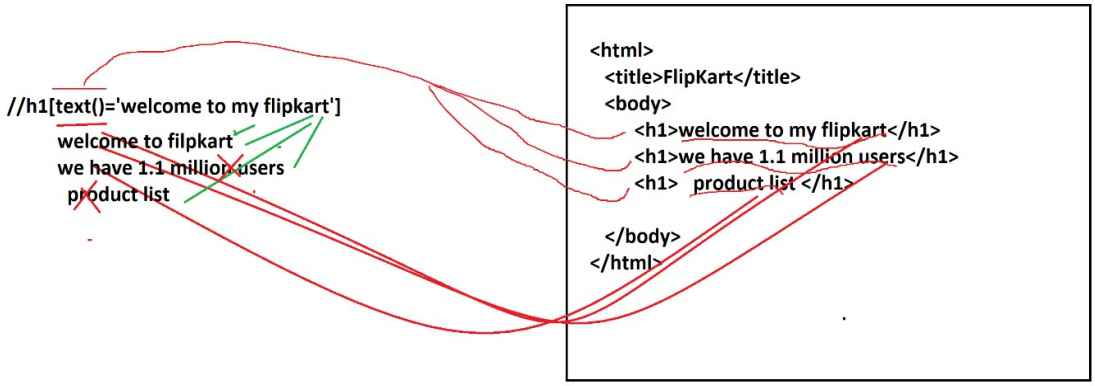
Visible text

If text is written between start &endTag is called Visible Text

Syntax

Text()=’expected Value’

//htmlTag[text()=’Expected value’]



EG :

//h1[text()='welcome to amazon'] :correct

//h1[text()='we have 1.1 million users'] : Wrong

//h1[text()='we hav'] : wrong

//h1[text()='product list'] : wrong

1. text() is used to identify the object using visible text of the element
2. text() cannot identify dynamic text(the text which is changing dynamically)
3. text() cannot identify the object using part of the string , because its complete string matching function
4. it cannot ignore the space before & after the string
5. text() function navigate to html document & try to identify expected visible text in GUI, & return true if expected completely matches with UI text

---

case 4 : how to identify dynamic object

```
<html>
  <title>FlipKart</title>
  <body>
      <h1>welcome to my flipkart</h1>
      <h1>we have 1.2 million users</h1>
      <h1>  product list  </h1>
       <div>
           <img src='image1.png' height='200'/>
           <input type='button' value='add to cart'/>
           <input type='button' value='cancel'/>
       </div>
       <div>
           <img src='image1.png' height='200'/>
           <input type='button' value='add to cart'/>
           <input type='button' value='cancel'/>
       </div>
  </body>
</html>
```

welcome to my flipkart

we have 1.2 million users

product list

contains(text()/@attribute, “expected value”)

//htmlTag[contains(text()/@attribute, “expected value”)]
EG :

//h1[contains(text(),’we hava’)] : correct
//h1[contains(text(),'users')]: correct

//input[contains(@value,'RS')] : correct
//h1[normalize-space(text().’product list’)] : correct

1. contains function is used to identify the dynamic object
2. contains can automatically ignore the spaces before & after the string
3. contains function can identify the object using part of the string or complete string
4. using normalize-space function we can ignore the space before & after the String or attribute

---

Case 5 :how to identify elements , when attribute or visible text contains spaces ,

```
<html>
  <title>FlipKart</title>
  <body>
      <h1>welcome to my flipkart</h1>
      <h1>we have product 1.1 million</h1>
      <h1>  we have product  </h1>

  </body>
</html>
```

Syntax :

//htmlTag[normalize-spave(text()/@attribute) = 'expected value']

//h1[normaliz-space(text())='we have product']

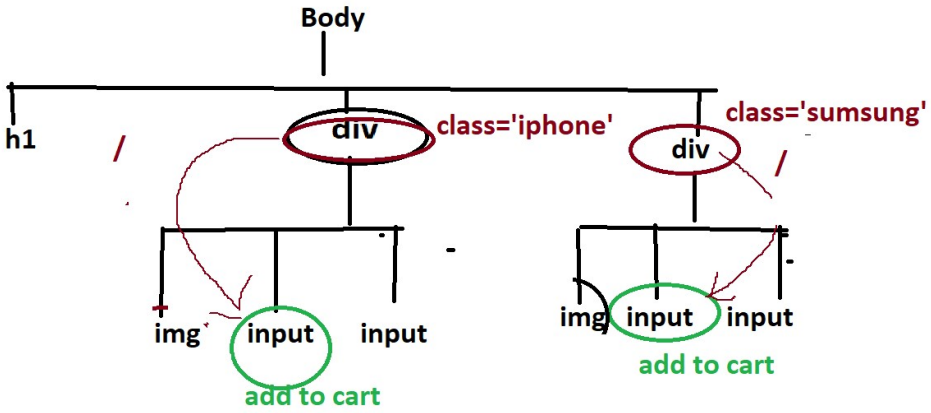
➔Normalize-space is xpath function which used to ignore the spaces before & after the String

Case 6 : how to identify the when similar/duplicate/same object present multiple times in GUI

welcome to my flipkart

Add to cart Cancle

Add to cart Cancle



```
<html>
  <title>FlipKart</title>
  <body>
      <h1>welcome to my flipkart</h1>
      <div class='iphone'>
        <img src='image3.png' height='100'/>
        <input type='button' value='Add to cart'/>
        <input type='button' value='Cancle'/>
      </div>
      <div class='samsung'>
        <img src='image4.png' height='100'/>
        <input type='button' value='Add to cart'/>
        <input type='button' value='Cancle'/>
      </div>
  </body>
</html>
```

Syntax

//parentHTmlTag[@att=’value’]/childHtmlTag[@att=’value’]

//div[@class='samsung']/input[@value='Add to cart']
//div[@class=iphone]/input[@value='Add to cart']

Solution :

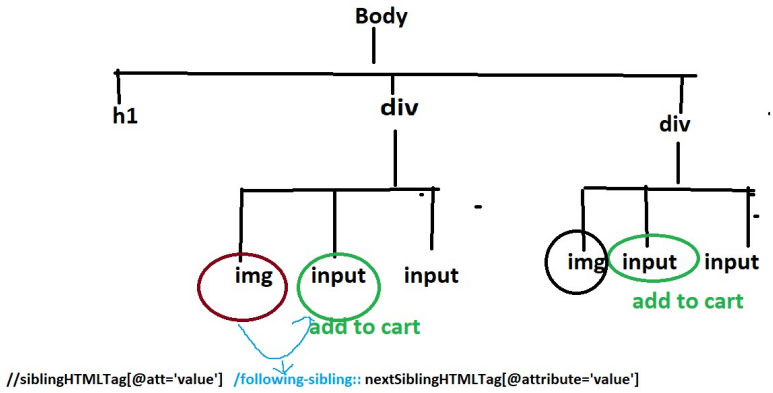
When object not able to identify using multiple attribute , in such cases will take reference of parent / grand parenthtmlTag to identify the object uniquely

Xpath – axes -> used to optimise the xpath

Case 7: how to identify the object using following sibling as a reference

```
<html>
  <title>FlipKart</title>
  <body>
      <h1>welcome to my flipkart</h1>
      <div>
        <img src='image3.png' height='100'/>
        <input type='button' value='Add to cart'/>
        <input type='button' value='Cancle'/>
      </div>
      <div>
        <img src='image4.png' height='100'/>
        <input type='button' value='Add to cart'/>
        <input type='button' value='Cancle'/>
      </div>
  </body>
</html>
```

welcome to my flipkart



Syntax

//siblingHTMLTag[@att='value'] /following-sibling:: nextSiblingHTMLTag[@attribute='value']

//img[@src='image3.png']/following-sibling::input[@value='Add to cart']

Solution:

Whenever element not able to identify the object using parent/grandparent, in such case will take a help of following sibling axes to identify the element using sibling as a reference

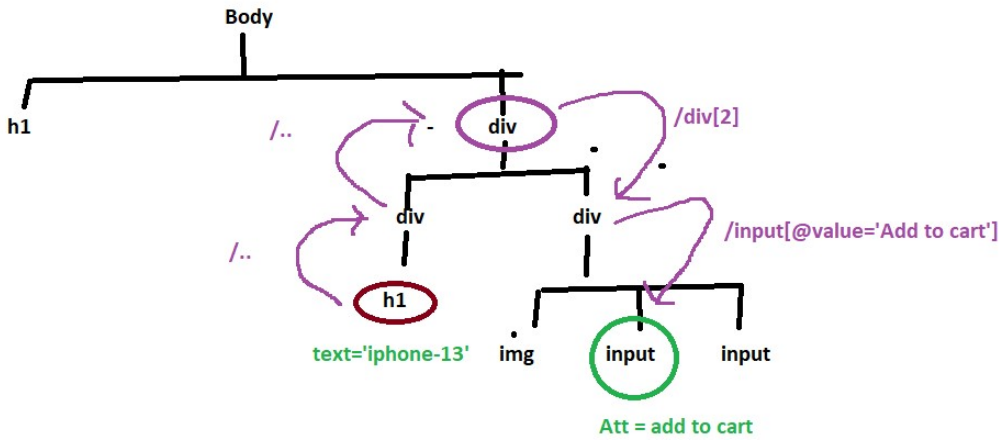
---

Case 8 : how to identify the object using preceding-sibling as a reference

//currnetHTMLTag[@att=’value’]/preceding-sibling::siblingHTMLTag[@att=’value’]

➔Whenever element not able to identify the object using parent/grandparent, in such case will take a help of preceding-sibling to identify the object

Case 9: how to Identify the expected Element using other element has reference



EG :

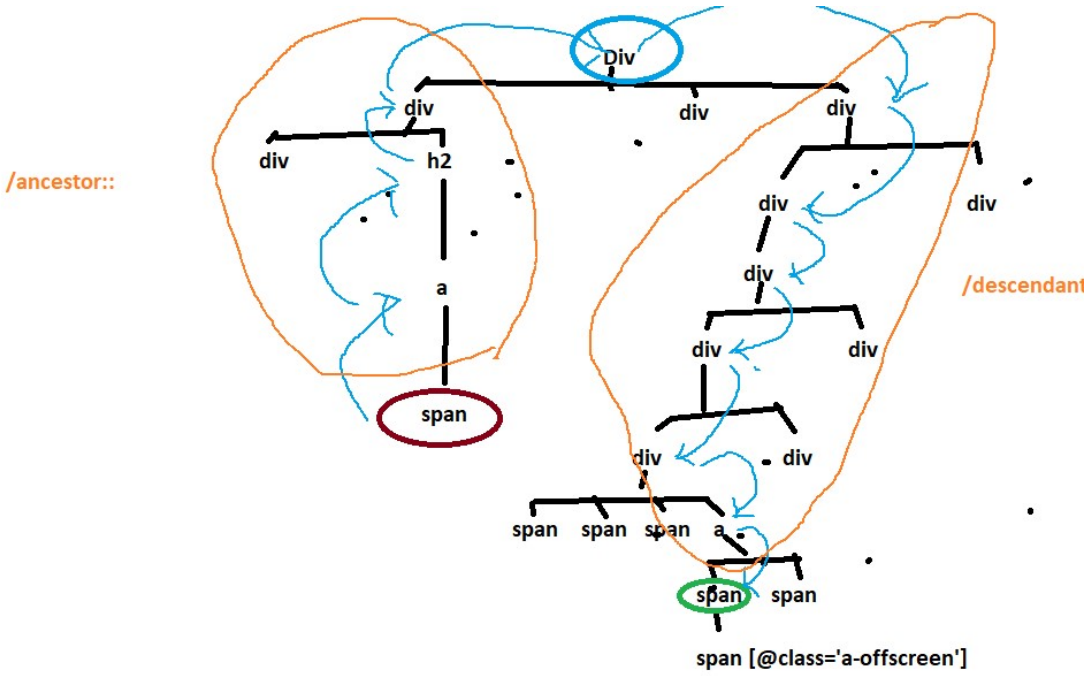
//h1[text()='iphone-13']/../..div[2]/input[@value='Add to cart']

//h1[text()='iphone-13']/../..div/input[@value='Add to cart']

//h1[text()='iphone-13']/../following-sibling::div/input[@value='Add to cart']

1. Identify the reference [unique] element
2. Identify the required Element
3. Identify the common Parent for both required & reference Element
4. Write html tree structure
5. By taking a help of reference element try to traverse to requires element via common parent

Case 10 : how to identify the object using /descendant as a reference



EG: with out descendant

//span[text()='Apple iPhone 13 (128GB) - Midnight']

/../../../..

/div[3]/div[1]/div/div[1]/div[1]/a/span[@class='a-price']/span[@class='a-offscreen']

EG : with descendant

//span[text()='Apple iPhone 13 (128GB) - Midnight']

/../../../..

/descendant::span[@class='a-price']/span[@class='a-offscreen']

//span[text()='Apple iPhone 13 (128GB) - Midnight']

/../../../..

//span[@class='a-price']/span[@class='a-offscreen']

Case 11 : how to identify the object using /ancestor as a reference

```
//span[text()='Apple iPhone 13 (128GB) - Midnight']
/ancestor::div[@class='a-section a-spacing-small a-spacing-top-small']
/descendant::span[@class='a-price']/span[@class='a-offscreen']
```

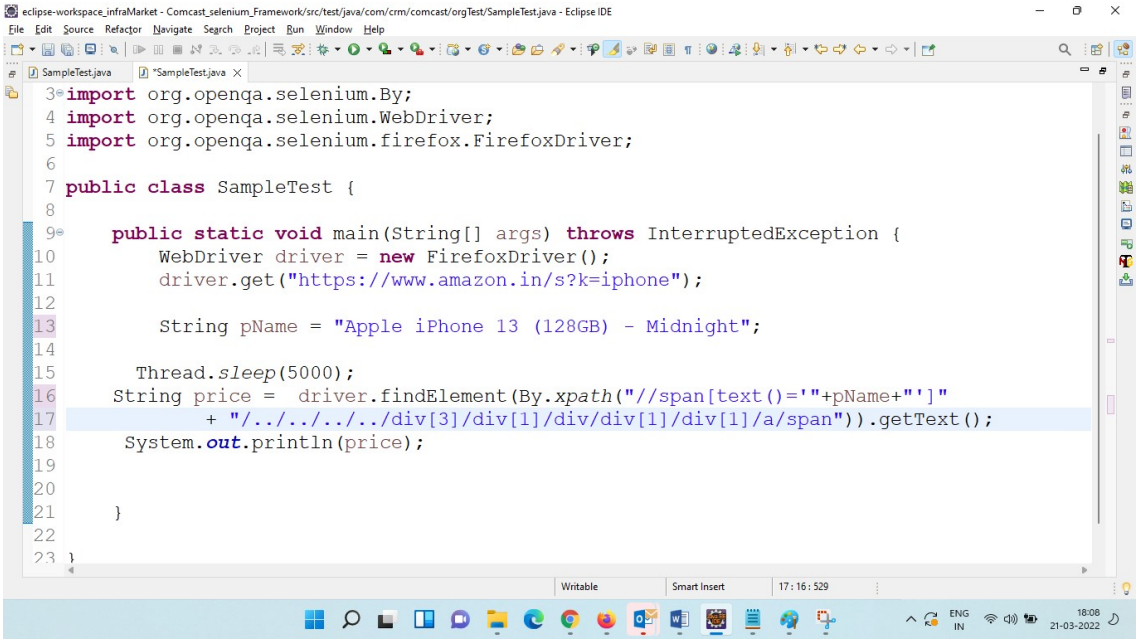
Case 12 : Dynamic Xpath

➔Xpath is getting created during runtime is called DynamicXpath

➔Dynamixxpath can be used to multiple elements by changing the variable value in runtime

➔Dynamic xpath reduces size of Elements in Object Repository

➔In dynamic xpath variable value always comes from External File , but in example variable value is hardcoded



Xpath vs CssSelector

Xpath	Css-selector
xpath is bidirectional	Css is unidirectional
xpath is slower compare to Css	Css is faster compare to Xpath
xpath navigate to Entire HTML DOM to identify the Element	Css is navigate to Entire Css-Document to identify the Elements using #id & .class attributes
We have xpath-axes & more function to Idebtify the Dynamic Elemenyts	Lesss feature to identify dynamic Element

---

Interview Questions

1. Difference between Xpath&Css-Selector
2. Why Xpath ? Why not other locator ?
3. Difference between Absolute & relative xpath ?Xpathtypes ?
4. Difference between / & //
5. What is dynamic xpath
6. Which locator is preferred to use in Selenium ?
7. Disadvantages of Xpath ?
8. What is Xpathaxces ?
9. Methods used in Xpath ?
10. How to identify Duplicate Object ?
11. How to identify Dynamic Object ?

12. How to identify multiple Elements using one locator ?
13. Have you used regular Expression in xpath ?
14. How to shadow elements ?
15. How to identify Hidden Elements ?
16. How to get All Hyper Links in GUI?
17. How to identify the Broken Links?
18. How to Identify the elements in Dynamic Webtable ?
19. How to get the row count of DynamicWebtable ?
20. How to get the row column count of DynamicWebtable ?
21. How to identify specific elements Dynamic Webtable ?
22. How to get complete Column elements in dynamic WebTable ?
23. How to get complete Row elements in dynamic WebTable ?