Using Contains, Sibling, Ancestor to Find Element in Selenium

If a simple XPath is not able to find a complicated web element for our test script, we need to use the functions from XPath 1.0 library. With the combination of these functions, we can create more specific XPath. Let's discuss a 3 such functions –

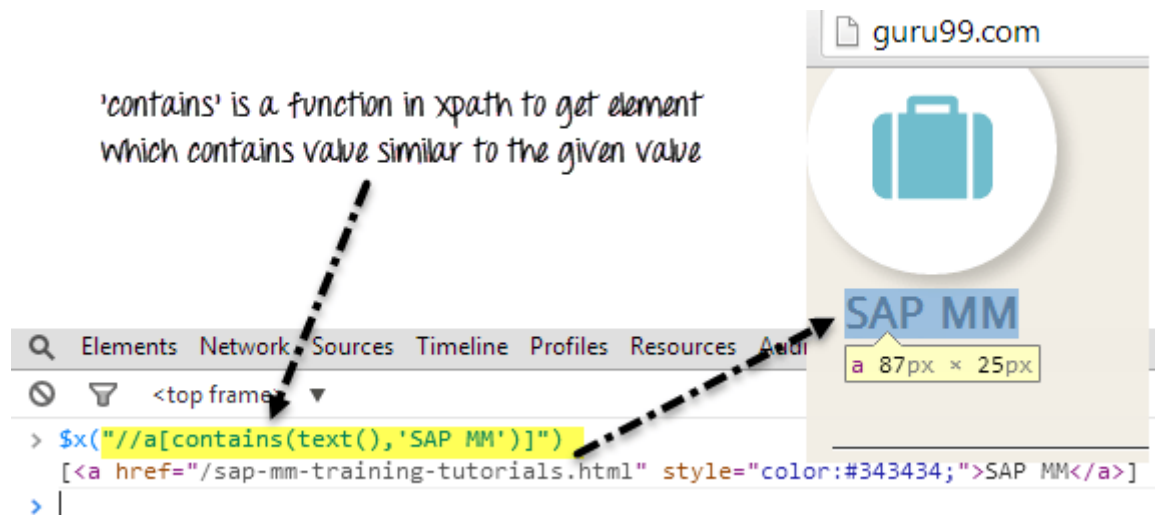1. Contains
2. Sibling
3. Ancestor

Let's study them in detail -

Contains: By using 'contains' function in XPath, we can extract all the elements which matches a particular text value.

Ex. Here we are searching an anchor .contains text as 'SAP M'.

```
"//h4/a[contains(text(),SAP M)]"
```

'contains' is a function in xpath to get element which contains value similar to the given value

guru99.com

SAP MM
a 87px × 25px

Elements Network Sources Timeline Profiles Resources Au...

<top frame> ▼

> $x("//a[contains(text(),'SAP MM')]")
  [SAP MM]
> |

Sibling: Using sibling keyword, we can fetch a web element on the which is related to some other element.

Example: Here on the basis of sibling element of 'a' we are finding 'h4'

```
"//div[@class='canvas- graph']//a[@href='/accounting.html'][i[@class='icon-usd']]/following-sibling::h4"
```


margin-top: -95px;">
  ▶ ...
  ▼ <h4>

When we need to find element on the basis of some other control we can use following-sibling keyword of xPath.
Here in this example we are trying to get 'h4' on the basis of 'a'

$
ACCOUNTING
h4 125px × 20px

<top frame> ▼

> $x("//div[@class='canvas-graph']//a[@href='/accounting.html'][i[@class='icon-usd']]/following-sibling::h4")
  [▶ <h4>...</h4>]
> |
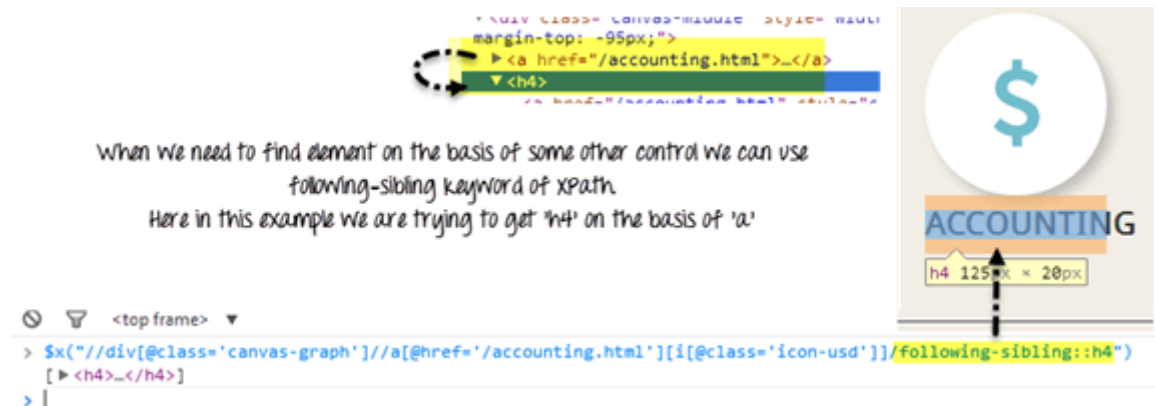
Ancestor: To find an element on the basis of the parent element we can use ancestor attribute of XPath.

Lets understand these 3 functions using an example –

Test Steps

1. Go to http://guru99.com
2. In the section 'A few of our most popular courses', search all Web Elements which are sibling of a WebElement whose text is 'SELENIUM'
3. We will find element using contains , ancestor and sibling function

USING Contains and Sibling

```
import java.util.List;

import java.util.concurrent.TimeUnit;
```

```java
import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

import org.testng.annotations.Test;

public class SiblingAndParentInXpath {

    @Test

    public void testSiblingAndParentInXpath(){

        WebDriver driver = new FirefoxDriver();          driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

        driver.get("http://guru99.com/");

        //Search element inside 'Popular course' which are sibling of control 'SELENIUM' ,Here first we will find a
h2 whose text is ''A few of our most popular courses' ,then we move to its parent element which is a 'div' , inside
this div we will find a link whose text is 'SELENIUM' then at last we will find all of the sibling elements of this
link('SELENIUM')

        List<WebElement> dateBox = driver.findElements(By.xpath("//h2[contains(text(),'A few of our most popular co
urses')]/parent::div//div[//a[text()='SELENIUM']]/following-sibling::div[@class='rt-grid-2 rt-omega']"));

        //Print all the which are sibling of the the element named as 'SELENIUM' in 'Popular course'

        for (WebElement webElement : dateBox) {
```

```
            System.out.println(webElement.getText());


        }



        driver.quit();



    }


}
```
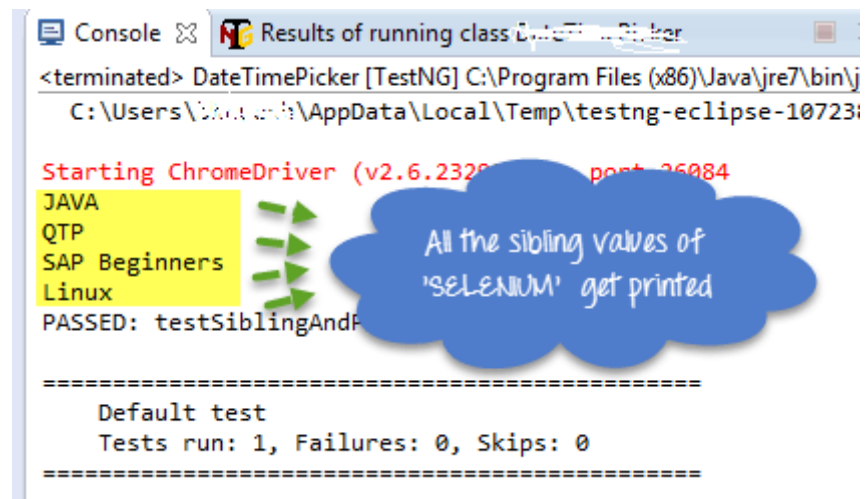
Output will be like:

Using ancestor function

We can achieve the same functionality with the help of a function 'ancestor' as well.

Now suppose we need to Search All elements in 'Popular course' section with the help of ancestor of the anchor whose text is 'SELENIUM'

Here our xpath query will be like

"//div[.//a[text()='SELENIUM']]/ancestor::div[@class='rt-grid-2 rt-omega']/following-sibling::div"

Complete Code

```java
import java.util.List;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.Test;

public class AncestorInXpath{

@Test

    public void testAncestorInXpath(){

            WebDriver driver = new FirefoxDriver();                 driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

        driver.get("http://guru99.com/");

        //Search All elements in 'Popular course' section with the help of ancestor of the anchor whose text is 'SELENIUM'
```

```
        List<WebElement> dateBox = driver.findElements(By.xpath("//div[.//a[text()='SELENIUM']]/ancestor::div[@clas
s='rt-grid-2 rt-omega']/following-sibling::div"));

        //Print all the which are sibling of the element named as 'SELENIUM' in 'Popular course'

        for (WebElement webElement : dateBox) {

            System.out.println(webElement.getText());

        }



        driver.quit();

    }

}
```

Output will look like-

Summary:

- There are some situation when regular xpath cannot be used to find element. In these situation we need different functions from xpath query.
- There some important xpath functions like contains, parent, ancestors, following-sibling etc.
- With the help of these functions we can create complex xpath expressions.
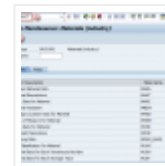
RELATED ARTICLES

TEST MANAGEMENT

Project Team: Develop, Building & Manage: A Step by Step Process

SOFTWARE TESTING

What is Test Maturity Model (TMM)?
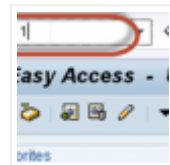
SAP - MM

MM17: Mass Maintenance of Material Master in SAP

PHP

PHP Date Function

SAP - BASIS

How to Limit Logon Attempts in SAP

SELENIUM

Using Apache Ant with Selenium