Locators (Part 5)

This is the next tutorial in the selenium-java series. Please go through the previous tutorial before you start this one. In the last tutorial, we continued looking at more locators. In this tutorial we will continue our exercises with locators!

What you will Learn:

1. Traverse between siblings using following-sibling
2. Traversing from child to parent using parent::<tagname>
3. Identify element based on text
4. How to treat classnames that have spaces in them?

Traverse between siblings using following-sibling:

Let us inspect radio button on the facebook home page

Figure 1

Notice that all the 3 radio buttons are represented by 'span' tag that are parallel to each other. So we can say that these 3 'span' tags are **siblings** to each other having same parent tag 'span'

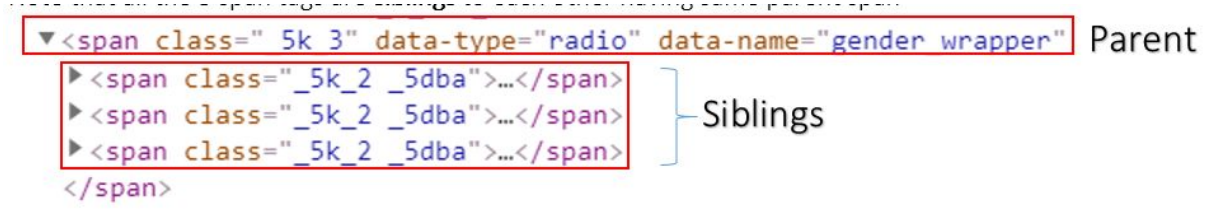

Figure 2

We have seen in the previous tutorial that below code will select first radio button

```
12          driver = new ChromeDriver();
13          driver.get("https://facebook.com");
14          driver.findElement(By.xpath("(//span[@class='_5k_3']/span)")).click();
```

Figure 3

Thus when we run the above script, 'Female' radio button gets selected

Figure 4

Keep in mind that, ‘Male’ radio button has 2 siblings viz ‘Female’ and ‘Custom’ radio buttons. Similarly, ‘Custom’ radio button has 2 siblings viz ‘Female’ and ‘Male’ radio buttons. Finally, ‘Female’ radio button has 2 siblings viz ‘Male’ and ‘Custom’ radio buttons. Now see below, let us inspect the xpath **//span[@class='_5k_2 _5dba']**, you will notice that 3 elements (3 radio buttons) have this xpath.

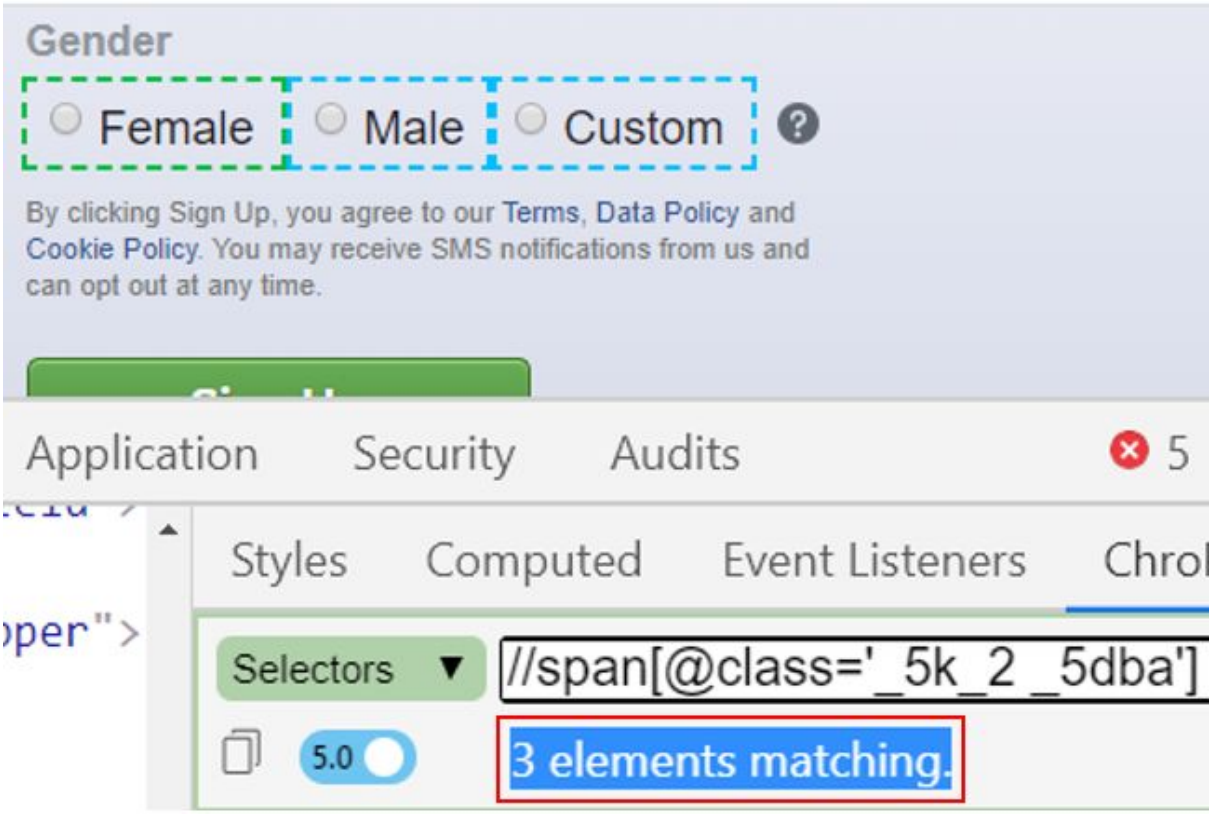

Figure 5

Now let us suppose that ‘Female’ radio button is selected by default (having above xpath). So, during runtime, how will you select ‘Custom’ radio button having the same xpath? So how can we jump between these siblings? How to traverse from one sibling to another sibling?

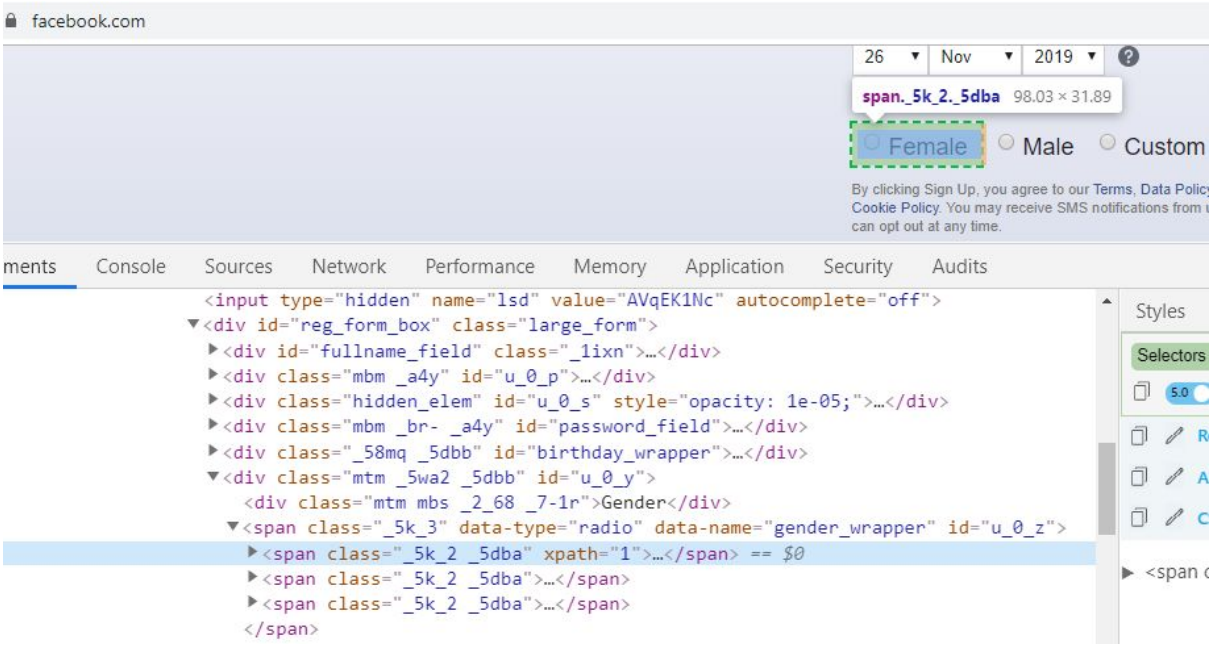

Figure 6

The syntax for identifying siblings is **/following-sibling::<tagname>**. In our case, the tagname of radio button is 'span'. So the first sibling can be represented as **/following-sibling::span[1]** and the second sibling as **/following-sibling::span[2]**.
So, to select first sibling viz 'Male' radio button, the complete xpath would be combination of first radio button's xpath & first sibling's xpath viz
//span[@class='_5k_2 _5dba']/following-sibling::span[1]

So we have

```
12          driver = new ChromeDriver();
13          driver.get("https://facebook.com");
14          driver.findElement(By.xpath("//span[@class='_5k_2 _5dba']/following-sibling::span[1]")).click();
```

Figure 7

Note that, none of the 3 radio buttons are selected when you launch facebook page, see below

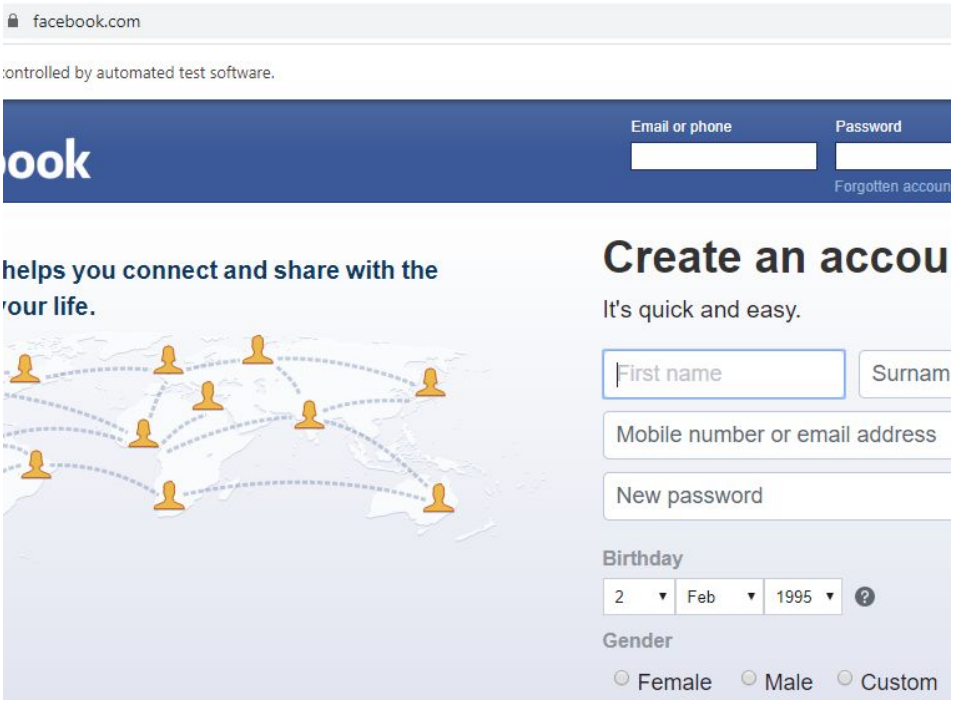

Figure 8

Now the first sibling of 'Female' radio button is 'Male' radio button. When you run the script, the first sibling 'Male' gets selected, see below

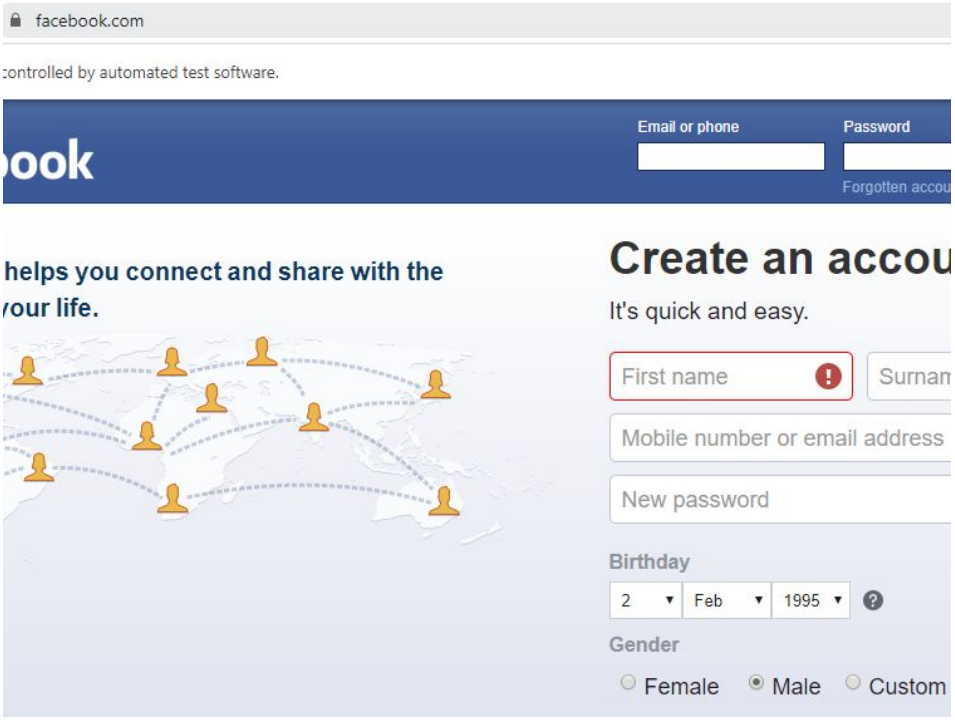

Figure 9

Comment line#14. Similarly below will select 2nd sibling viz 'Custom' radio button

```
5  public class MultipleElements {
6
7      public static void main(String[] args) {
8          System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TRAINING\\Software\\chrom
9
10         WebDriver driver = null;
11
12         driver = new ChromeDriver();
13         driver.get("https://facebook.com");
14         //driver.findElement(By.xpath("//span[@class='_5k_2 _5dba']/following-sibling::span[1]")).click();
15         driver.findElement(By.xpath("//span[@class='_5k_2 _5dba']/following-sibling::span[2]")).click();
```

Figure 10

When you run the script, the second sibling 'Custom' gets selected, see below

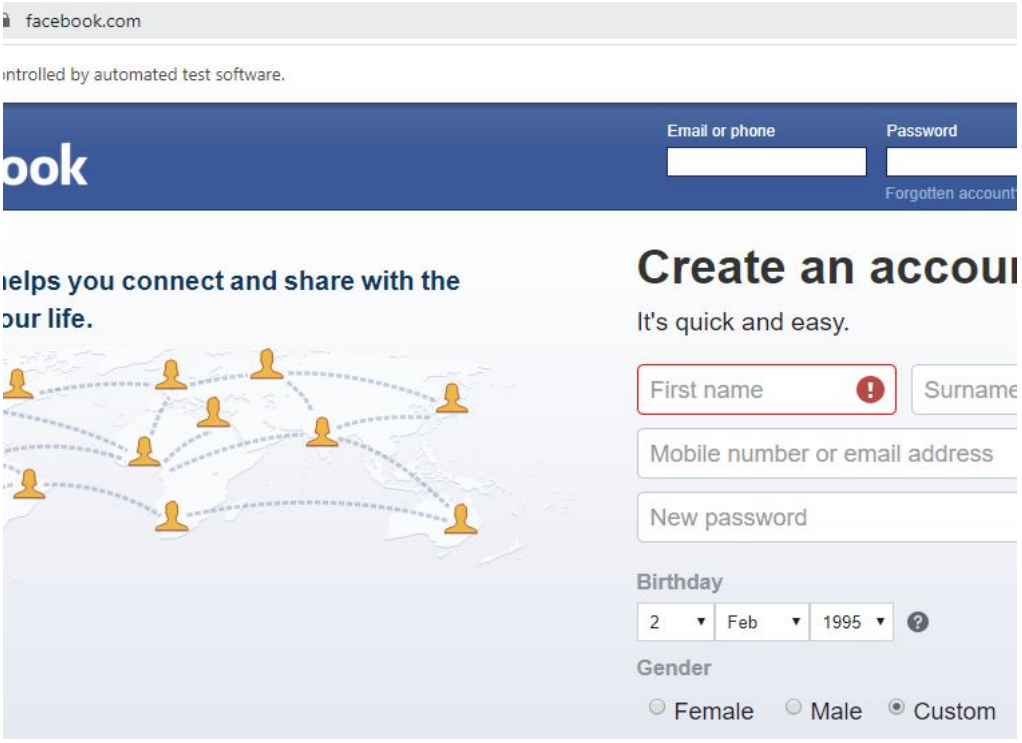

Figure 11

Let us uncomment line#14. Now when you run the script, you will notice that 'Male' radio button gets selected followed by 'Custom' radio.

```
5  public class MultipleElements {
6
7      public static void main(String[] args) {
8          System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TRAINING\\Software\\chro
9
10         WebDriver driver = null;
11
12         driver = new ChromeDriver();
13         driver.get("https://facebook.com");
14         driver.findElement(By.xpath("//span[@class='_5k_2 _5dba']/following-sibling::span[1]")).click();
15         driver.findElement(By.xpath("//span[@class='_5k_2 _5dba']/following-sibling::span[2]")).click();
```

Figure 12

We have seen in figure#5 that the xpath //span[@class='_5k_2 _5dba'] matches 3 elements. The selenium script will select first radio button (since selenium scans the page from left to right). So let us add line#14. Now when you run the script, you will notice that 'Female' radio button gets selected followed by 'Male' radio button and finally 'Custom' radio.

```
12          driver = new ChromeDriver();
13          driver.get("https://facebook.com");
14          driver.findElement(By.xpath("//span[@class='_5k_2 _5dba']")).click();
15          driver.findElement(By.xpath("//span[@class='_5k_2 _5dba']/following-sibling::span[1]")).click();
16          driver.findElement(By.xpath("//span[@class='_5k_2 _5dba']/following-sibling::span[2]")).click();
```

Figure 13

You can also write line#14 as below. When you run the script, you will notice that 'Female' radio button gets selected followed by 'Male' radio button and finally 'Custom' radio.

```
12          driver = new ChromeDriver();
13          driver.get("https://facebook.com");
14          driver.findElement(By.xpath("(//span[@class='_5k_3']/span)")).click();
15          driver.findElement(By.xpath("//span[@class='_5k_2 _5dba']/following-sibling::span[1]")).click();
16          driver.findElement(By.xpath("//span[@class='_5k_2 _5dba']/following-sibling::span[2]")).click();
```

Figure 14

So this way you can traverse from one sibling to another.

Traversing from child to parent using parent::<tagname>

We have seen in our previous tutorial that using absolute xpath, we can traverse from parent to child. Let us try to do reverse now, let us see how to traverse from child to parent (this situation will arise if a parent does not have any static attribute value and the child has static attribute). The child xpath **//label[@class='_58mt']** matches 3 elements (only the TEXT of 3 radio buttons, but not the actual radio buttons)

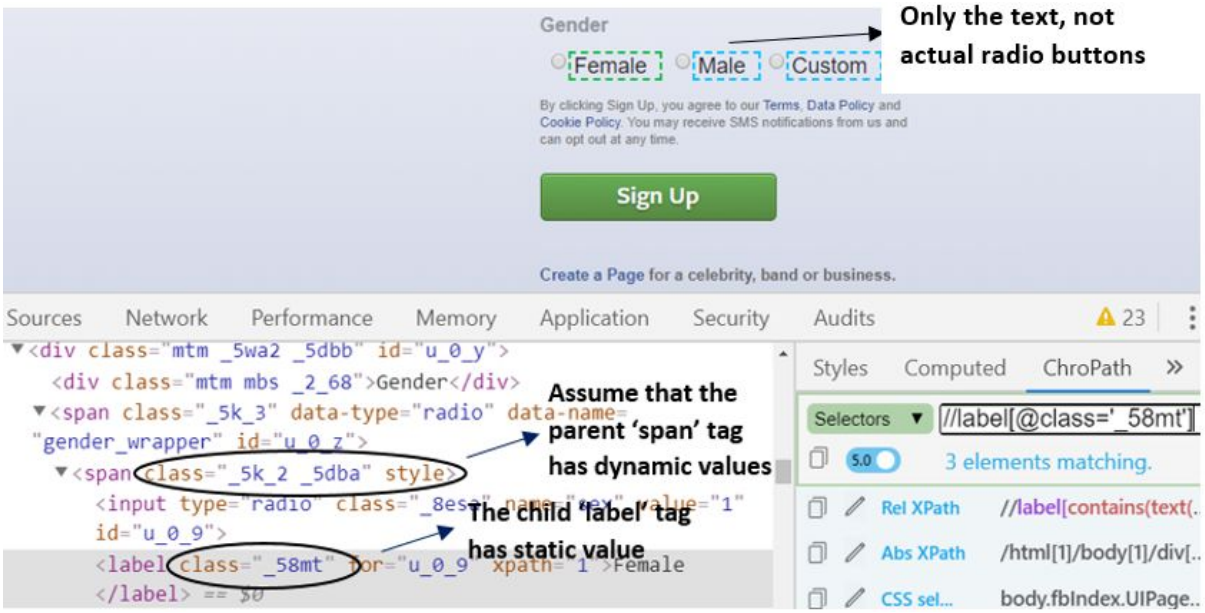

Figure 15

The parent of 'label' tag is 'span' tag. So the only solid information that we have about parent is that it is a 'span' tag. Now inspect the xpath **//label[@class='_58mt']/parent::span**

Notice a broader selection now. We now see that the selections include the actual radio buttons as well

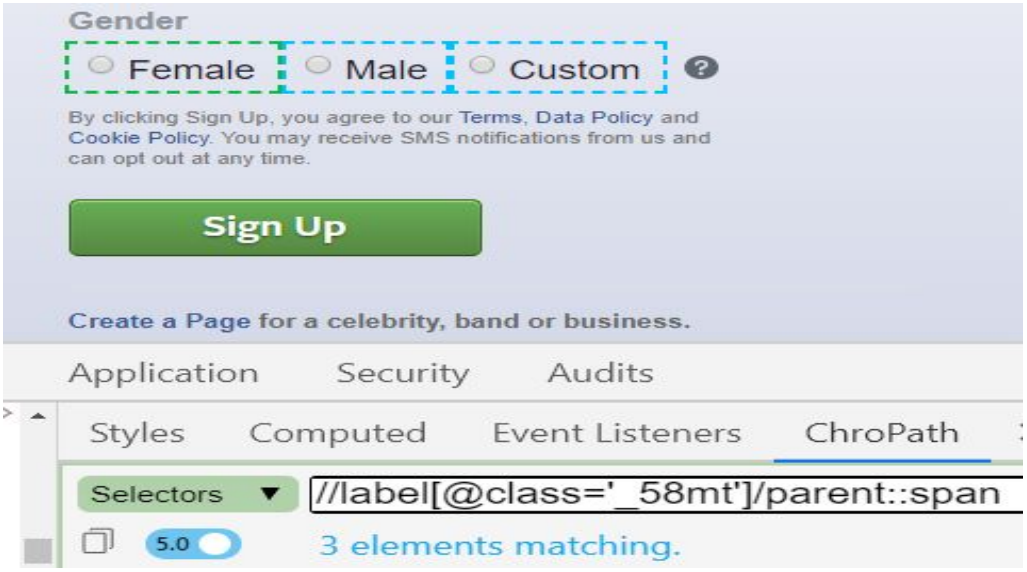

Figure 16

So lets use this xpath in line#14

```
12          driver = new ChromeDriver();
13          driver.get("https://facebook.com");
14          driver.findElement(By.xpath("//label[@class='_58mt']/parent::span")).click();
```

Figure 17

Run the script, 'Female' radio button gets selected (since selenium scans the page from left to right and finds the first match)

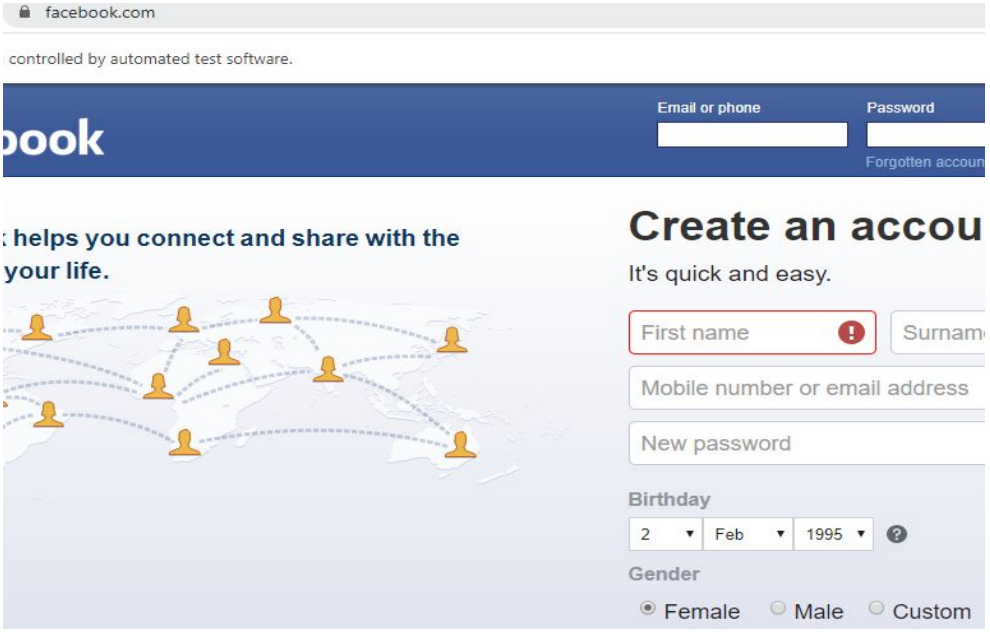

Figure 18

Identify element based on text:
Sometimes in an agile world, the html code might be in the development phase and the html tags may not yet be decided. In this situation, we can still identify elements based on text, example, **//*[text()='Men']**

* is a regular expression which means we still don't know the tag (under development). Instead of any attribute, we can use text(). Go to the website https://www.aeo.in/ and inspect //*[text()='Men']. Notice below, there is 1 element matching and we see the focus around 'Men'. Thus we are able to identify 'Men' link based on the text 'Men'

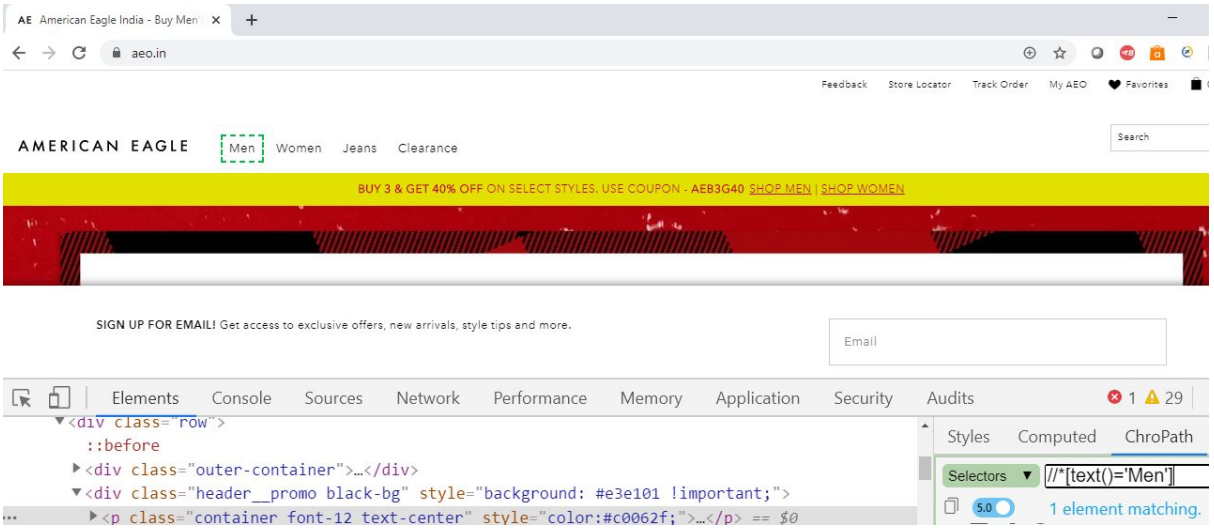

Figure 19

Line#13 serves this purpose

```
11          driver = new ChromeDriver();
12          driver.get("https://aeo.in");
13          driver.findElement(By.xpath("//*[text()='Men']")).click();
```

Figure 20

Run, notice that 'Men' link gets clicked and below page comes up

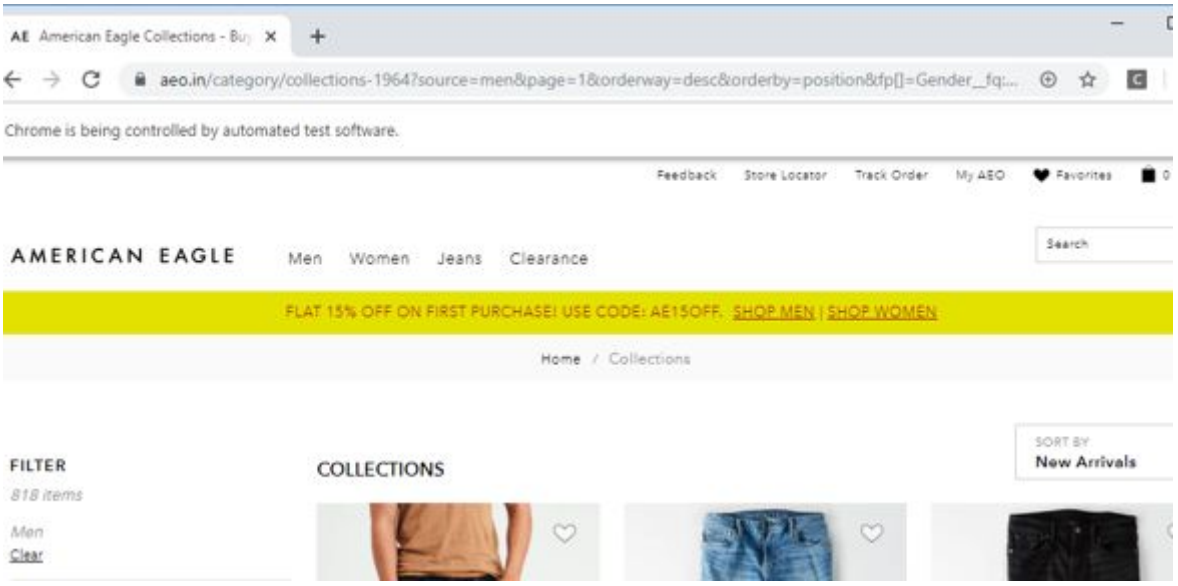

Figure 21

How to treat classnames that have spaces in them?

Let us now see how to treat the class names, using css, if the classnames have space in between. Let us inspect email id field, see below. The value of 'class' attribute contains 2 classes having white space in between:

inputtext and login_form_input_box

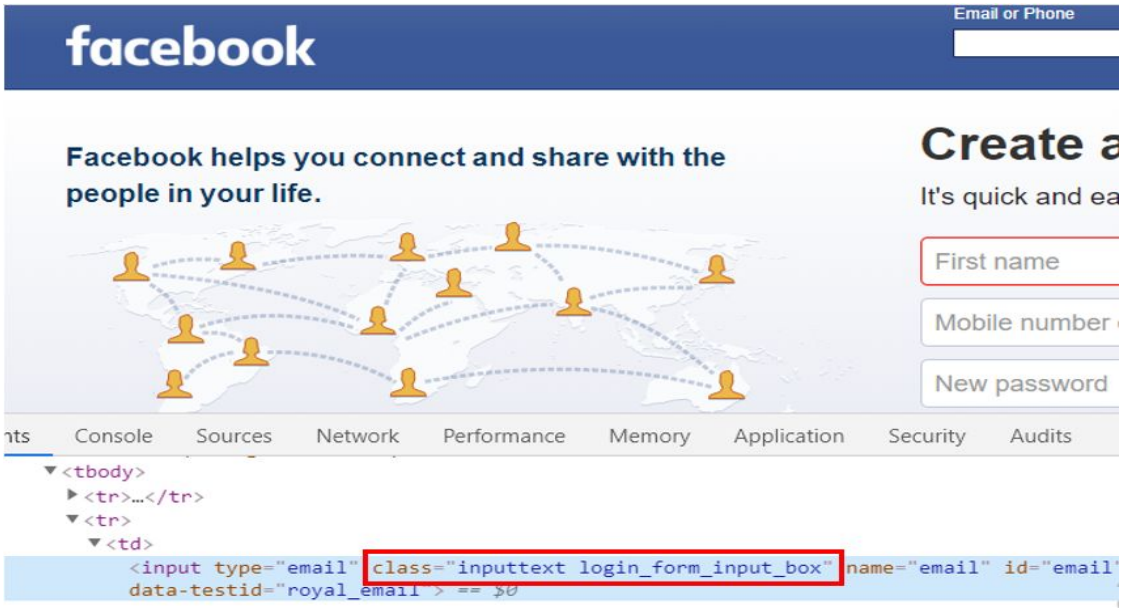

Figure 22

There are few rules that you can keep in mind in such cases:
Rule 1# In css selector, you can put a dot as a prefix before the classname. So we can say **.inputtext** and **.login_form_input_box**

Rule#2 In css selector, we can replace white space in between the classes with a dot. So our css selector becomes:
.inputtext.login_form_input_box

So let us inspect **.inputtext.login_form_input_box** in the Chropath. Notice that there are 2 matching results (email and password boxes are highlighted)

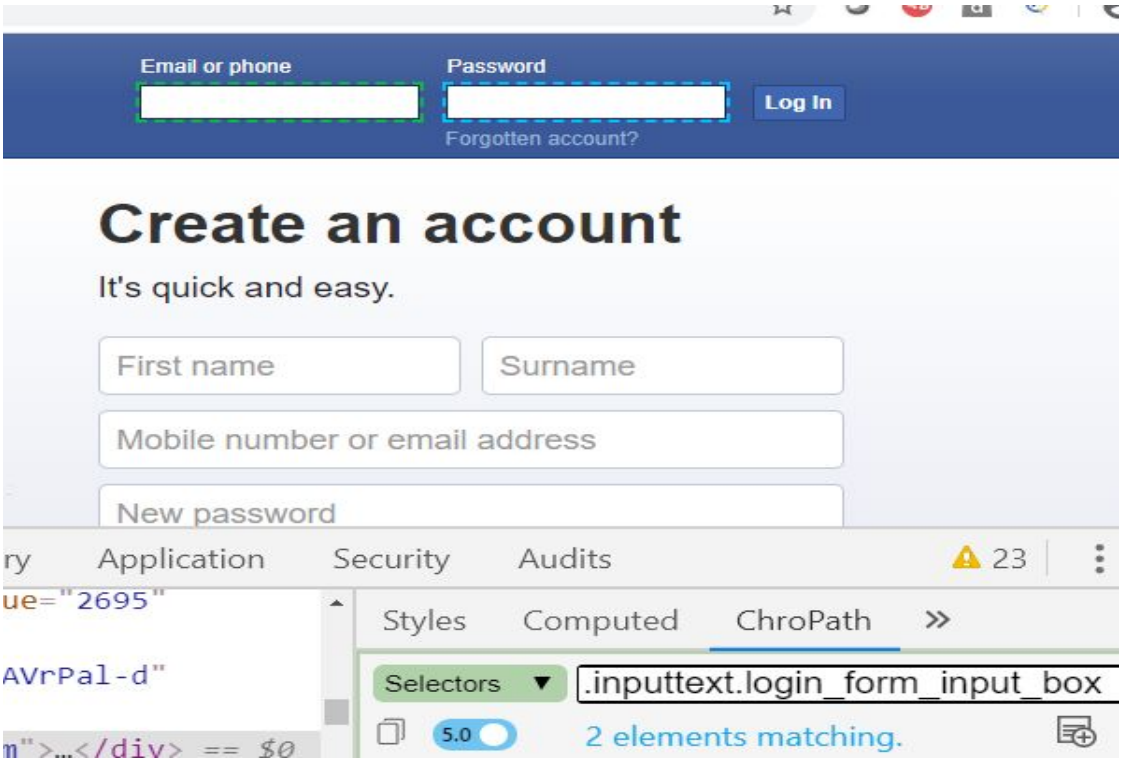

Figure 23

We can now use this css path with By.cssSelector

```
12          driver = new ChromeDriver();
13          driver.get("https://facebook.com");
14          driver.findElement(By.cssSelector(".inputtext.login_form_input_box")).sendKeys("dummy@gmail.com");
```

Figure 24

Run, notice that text gets entered in 'Email' field

Figure 25

With this, we are done with locators. Thank you for reading!