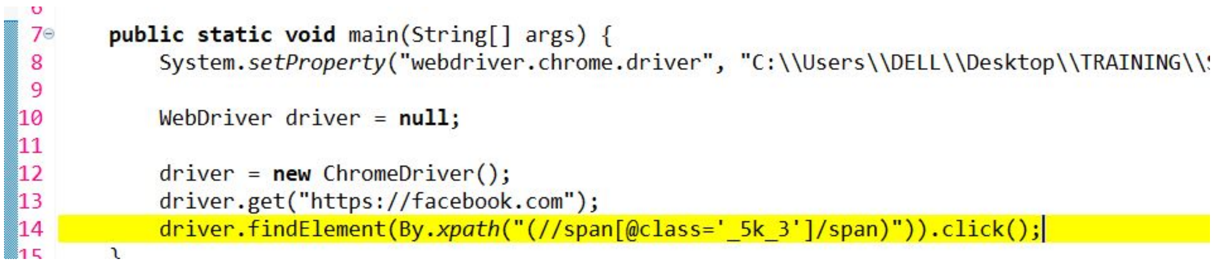
Locators (Part 4)

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:
Custom xpath (continued…)
Parent to child xpath
Custom css
What is relative xpath?
What is absolute xpath?

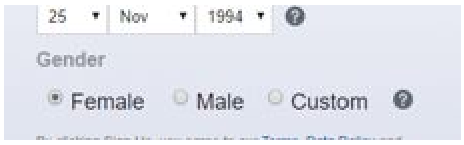
Custom xpath (continued…):
This is in continuation with where we left our last tutorial. Let us change span[2] to simply 'span', see line#14 below. Note that 'span' defaults to span[1]. Since selenium scans the page from left to right & top to bottom, as soon as it finds the first span tag viz first radio button (Female), it stops further scanning & click it



```
 7      public static void main(String[] args) {
 8          System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TRAINING\\
 9
10          WebDriver driver = null;
11
12          driver = new ChromeDriver();
13          driver.get("https://facebook.com");
14          driver.findElement(By.xpath("(//span[@class='_5k_3']/span)")).click();
15      }
```

Figure 1

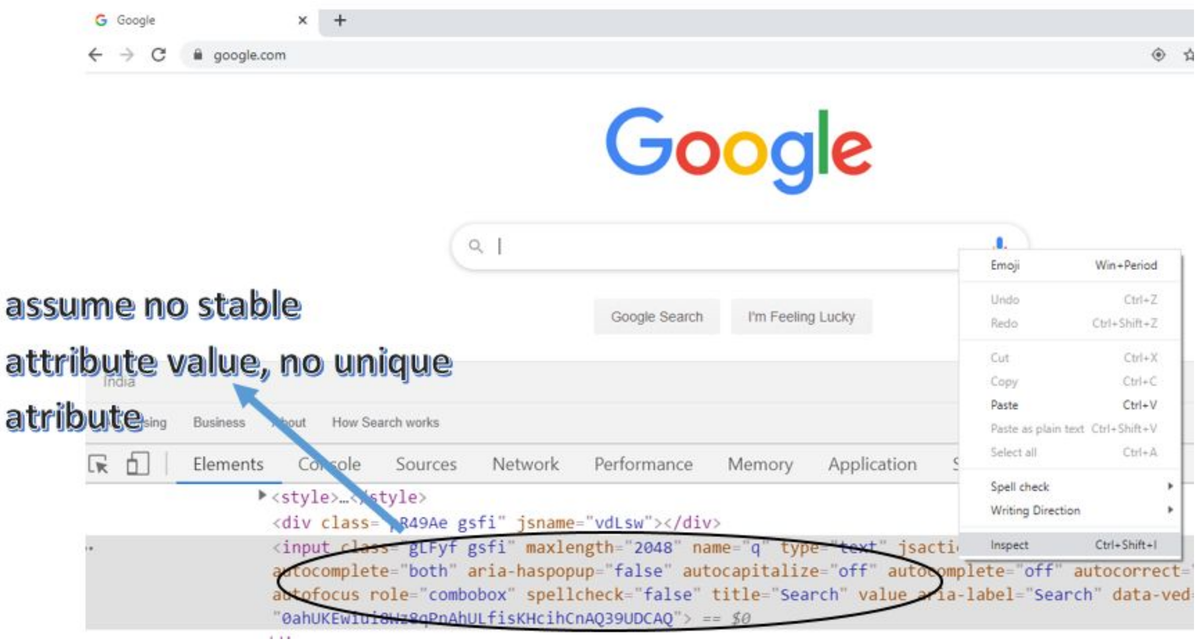
Run, notice that first radio button 'Female' is selected



```
25 ▾  Nov ▾  1994 ▾
Gender
● Female  ○ Male  ○ Custom
```

Figure 2

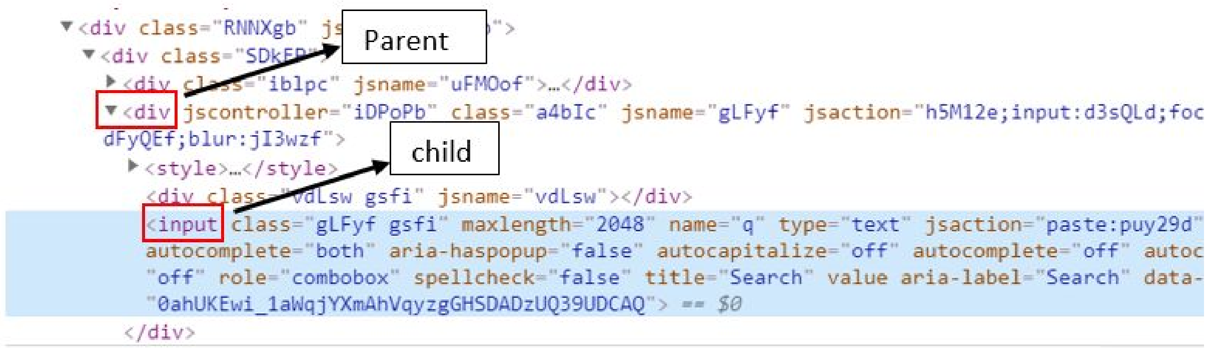
Parent to child xpath:
Let's say there is no unique attribute to identify an element. Let's suppose in google.com, the google search field does not have any unique attribute to identify this field, let's assume we only have 'data-ved' attribute and that too is dynamic (entire value changes with every session). So in these cases parent to child relationship xpath comes into picture. So if child does not have a stable attribute, we look for one of its parent that has stable attribute. Once we find a parent that has stable attribute, we can write an xpath starting from that parent and keep traversing till the child (that has no unique attribute) with the help of tagnames



```
▶<style>…</style>
 <div class="RNNXgb R49Ae gsfi" jsname="vdLsw"></div>
 <input class="gLFyf gsfi" maxlength="2048" name="q" type="text" jsacti
 autocomplete="both" aria-haspopup="false" autocapitalize="off" autocomplete="off" autocorrect=
 autofocus role="combobox" spellcheck="false" title="Search" value aria-label="Search" data-ved
 "0ahUKEwiui8Wz8qPnAhULfisKHcihCnAQ39UDCAQ"> == $0
```

Figure 3

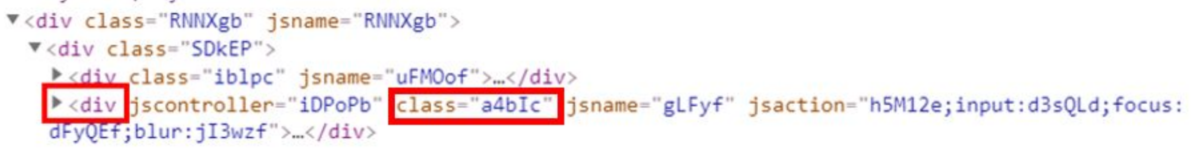
Now how do we identify a parent? Simply close the parent tag, if you don't see the child, then it means it's the correct parent. So let's say below is the child tag



```
▼<div class="RNNXgb" js…">
  ▼<div class="SDkEP">
    ▶<div class="iblpc" jsname="uFMOof">…</div>
    ▼<div jscontroller="iDPoPb" class="a4bIc" jsname="gLFyf" jsaction="h5M12e;input:d3sQLd;foc
    dFyQEf;blur:jI3wzf">
      ▶<style>…</style>
       <div class="vdLsw gsfi" jsname="vdLsw"></div>
       <input class="gLFyf gsfi" maxlength="2048" name="q" type="text" jsaction="paste:puy29d"
       autocomplete="both" aria-haspopup="false" autocapitalize="off" autocomplete="off" autoc
       "off" role="combobox" spellcheck="false" title="Search" value aria-label="Search" data-
       "0ahUKEwi_1aWqjYXmAhVqyzgGHSDADzUQ39UDCAQ"> == $0
     </div>
```

Figure 4

If we close below 'div' tag then child tag shown in above figure is also getting closed. So it means that this 'div' tag is the correct parent of child tag

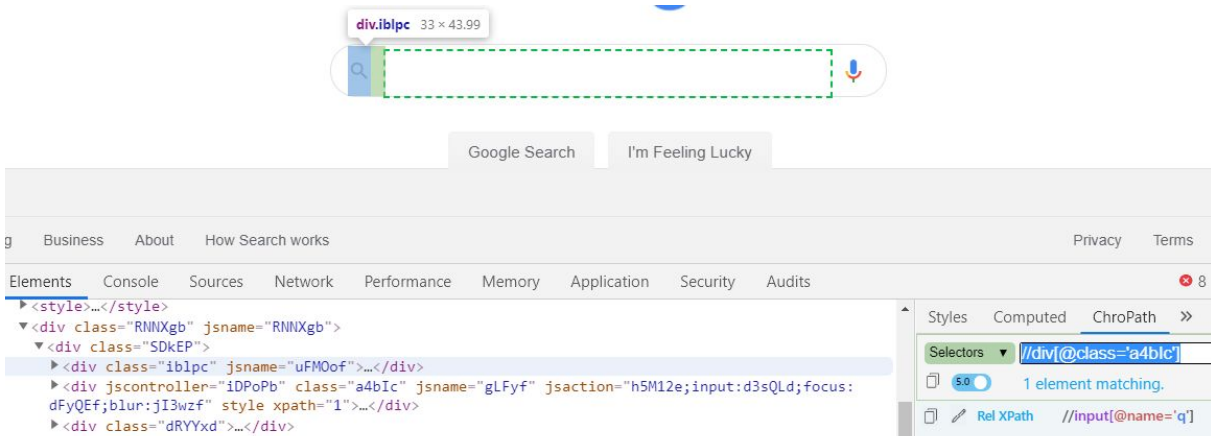


```
▼<div class="RNNXgb" jsname="RNNXgb">
  ▼<div class="SDkEP">
    ▶<div class="iblpc" jsname="uFMOof">…</div>
    ▶<div jscontroller="iDPoPb" class="a4bIc" jsname="gLFyf" jsaction="h5M12e;input:d3sQLd;focus:
    dFyQEf;blur:jI3wzf">…</div>
```

Figure 5

Now we will check if this parent 'div' tag has any unique attribute or not. The 'class' attribute seen in above figure seems to be unique. So we have found a parent having unique attribute. If we don't find any unique attribute than we can go one level up and find a parent that has unique attribute. We will keep going one level up till we find a parent having unique attribute.

So, in this case, we have found a parent that is the 'div' tag having 'class' attribute with a value 'a4bIc'. So the custom xpath of parent is **//div[@class='a4bIc']** . Let us check if this xpath is correct or not. Just copy this xpath & paste it in 'Selectors' field under ChroPath. Hit enter. We see that the outer body of google search field gets highlighted. So it means that the parent xpath is correct.

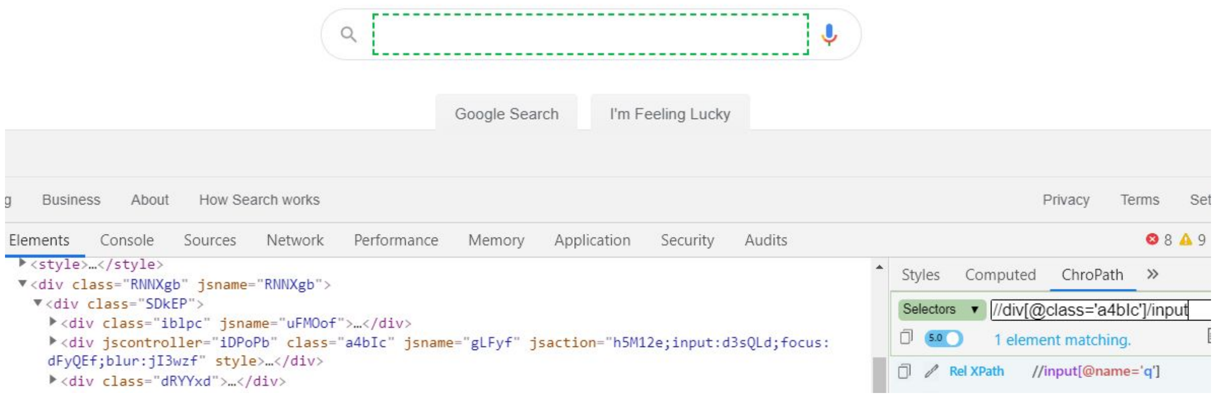


```
▸<style>…</style>
▾<div class="RNNXgb" jsname="RNNXgb">
  ▾<div class="SDkEP">
    ▸<div class="iblpc" jsname="uFMOof">…</div>
    ▸<div jscontroller="iDPoPb" class="a4bIc" jsname="gLFyf" jsaction="h5M12e;input:d3sQLd;focus:
    dFyQEf;blur:jI3wzf" style xpath="1">…</div>
    ▸<div class="dRYYxd">…</div>
```

```
Selectors  //div[@class='a4bIc']
1 element matching.
Rel XPath  //input[@name='q']
```

Figure 6

Now from this parent tag, we will traverse to our desired child tag. Child tag is the place where you will be entering the text viz child identifies the actual search field. In our case, the child is represented by 'input' tag. So the xpath up to the child path becomes **//div[@class='a4bIc']/input** . Let us put this in Chropath & hit enter. So we have now identified the google search text field. So we are able to identify a child based upon the parent-child relationship.

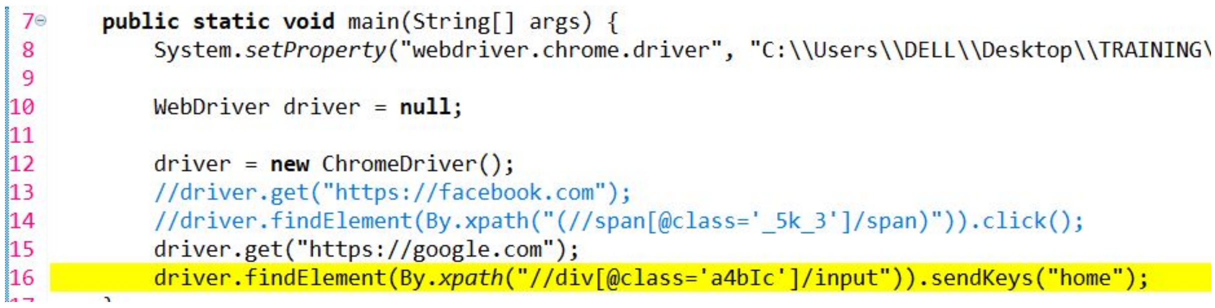


```
▸<style>…</style>
▾<div class="RNNXgb" jsname="RNNXgb">
  ▾<div class="SDkEP">
    ▸<div class="iblpc" jsname="uFMOof">…</div>
    ▸<div jscontroller="iDPoPb" class="a4bIc" jsname="gLFyf" jsaction="h5M12e;input:d3sQLd;focus:
    dFyQEf;blur:jI3wzf" style>…</div>
    ▸<div class="dRYYxd">…</div>
```

```
Selectors  //div[@class='a4bIc']/input
1 element matching.
Rel XPath  //input[@name='q']
```

Figure 7

So let us use this xpath & enter some value in the text field.



```
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TRAINING\

        WebDriver driver = null;

        driver = new ChromeDriver();
        //driver.get("https://facebook.com");
        //driver.findElement(By.xpath("(//span[@class='_5k_3']/span)")).click();
        driver.get("https://google.com");
        driver.findElement(By.xpath("//div[@class='a4bIc']/input")).sendKeys("home");
    }
```

Figure 8

Run, we see 'home' getting typed in the search field

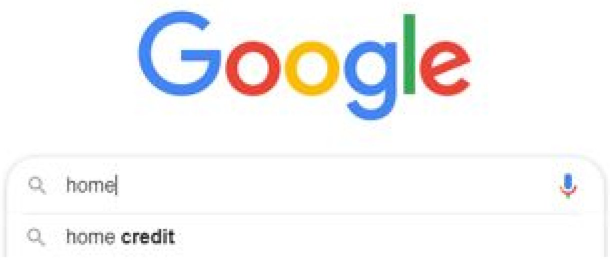
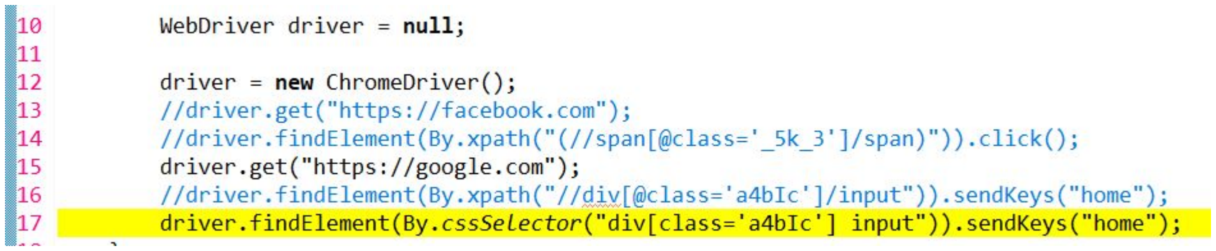


Figure 9

Custom css:
Let us now convert the custom xpath **//div[@class='a4bIc']/input** to custom css. To do that, we have to simply remove // @. Also, the forward slash / will be replaced by white space. So our custom css would become **div[class='a4bIc'] input**
Let us now put it in selector field & hit enter, we see that google search text field gets highlighted

Figure 10

Comment line 16. Add line#17 using custom css



```
10          WebDriver driver = null;
11  
12          driver = new ChromeDriver();
13          //driver.get("https://facebook.com");
14          //driver.findElement(By.xpath("(//span[@class='_5k_3']/span)")).click();
15          driver.get("https://google.com");
16          //driver.findElement(By.xpath("//div[@class='a4bIc']/input")).sendKeys("home");
17          driver.findElement(By.cssSelector("div[class='a4bIc'] input")).sendKeys("home");
```

Figure 11

Run, we see 'home' getting typed in the search field

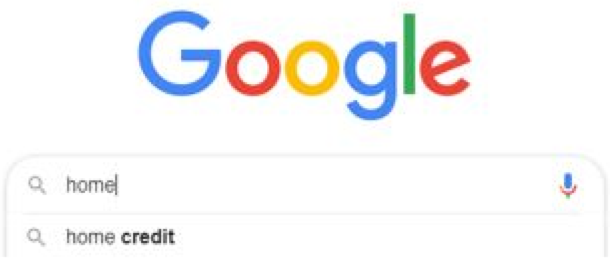
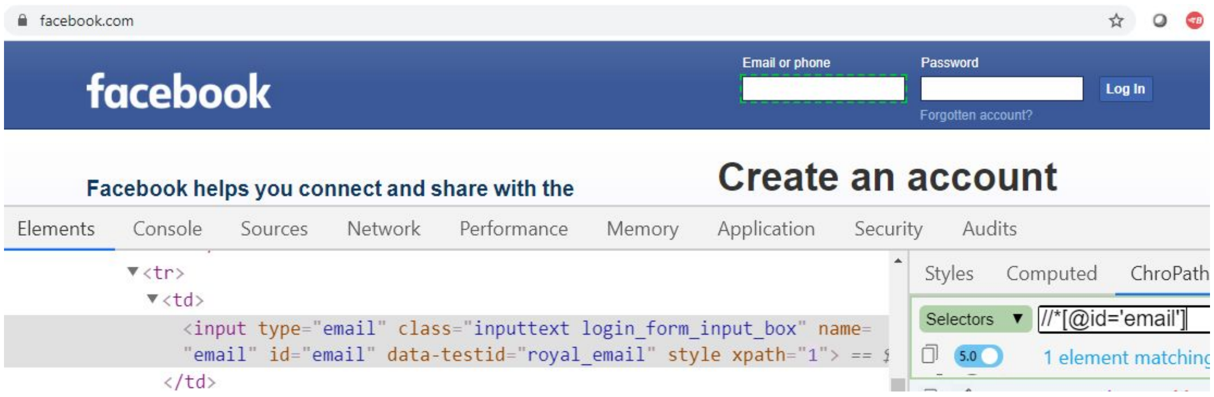


Figure 12

What is relative xpath?

There are 2 types of xpath: Relative & Absolute. In relative xpath, we jump directly to specific desired element rather than traversing through the nodes. We do not take any help of parent node to identify an element. We directly jump to the desired element with the help of **'id'** attribute. The relative xpath that starts with //. So if you see below, the relative xpath **//*[@id='email']** can be used to identify the 'Email' field. If you notice, we haven't used any of the parent tag.



```
▼<tr>
  ▼<td>
     <input type="email" class="inputtext login_form_input_box" name=
     "email" id="email" data-testid="royal_email" style xpath="1"> ==
   </td>
```

//*[@id='email']

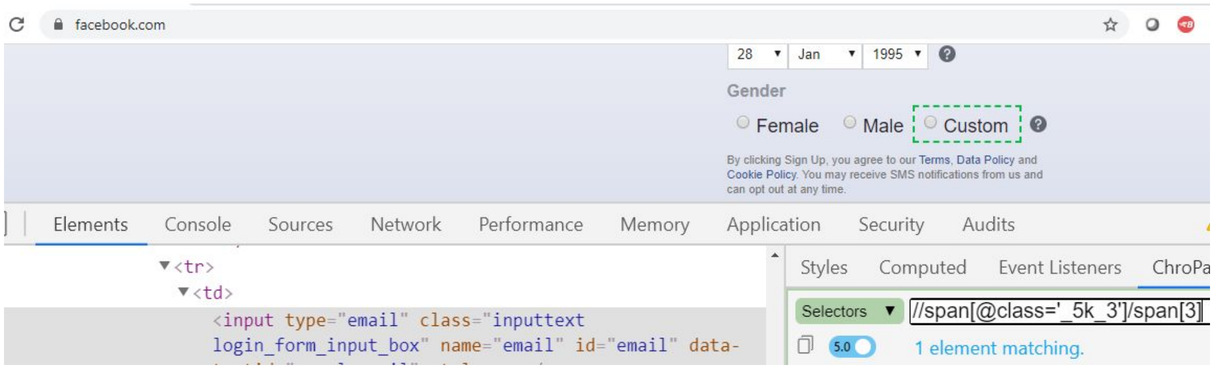
1 element matching

Figure 13

What is absolute xpath?

In the case of absolute xpath, we will traverse to the desired element with the help of parent nodes. The hierarchy will start from initial html tag. The absolute xpath starts with single slash /. So we would be traversing like: /parent/child1/child2/.....

Sometime back, we had used custom xpath **//span[@class='_5k_3']/span[3]** to identify 'Custom' radio button



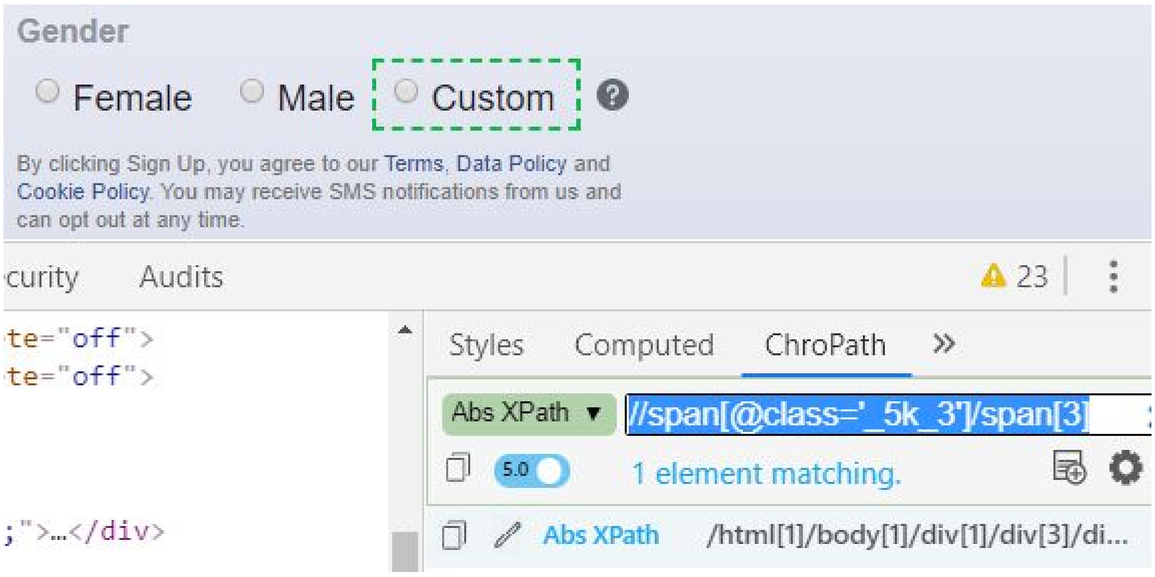
```
▼<tr>
  ▼<td>
     <input type="email" class="inputtext
     login_form_input_box" name="email" id="email" data-
```

//span[@class='_5k_3']/span[3]

1 element matching.

Figure 14

Select 'Abs Xpath' from selector dropdown as seen below


Gender
Female
Male
Custom
By clicking Sign Up, you agree to our Terms, Data Policy and Cookie Policy. You may receive SMS notifications from us and can opt out at any time.
curity
Audits
23
Styles
Computed
ChroPath
Abs XPath
5.0
Abs XPath

```
te="off">
te="off">
;">…</div>
```

//span[@class='_5k_3']/span[3]

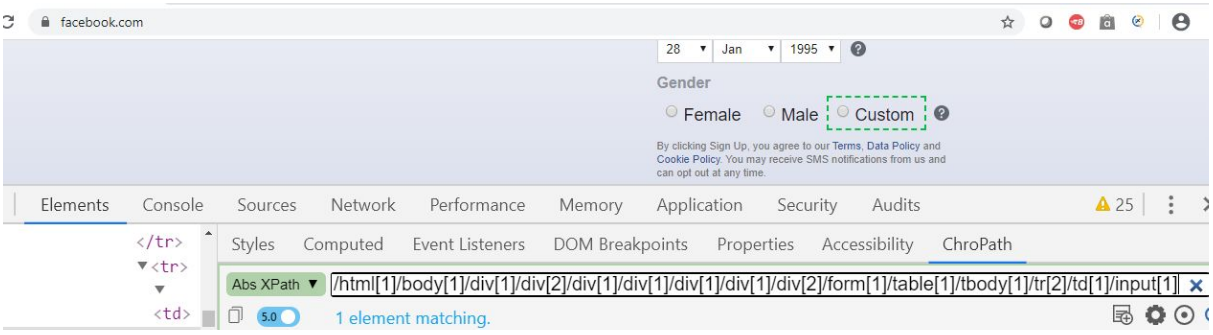
1 element matching.

/html[1]/body[1]/div[1]/div[3]/di...

Figure 15

Click pencil icon

Notice that the entire absolute path comes up in the selector field.



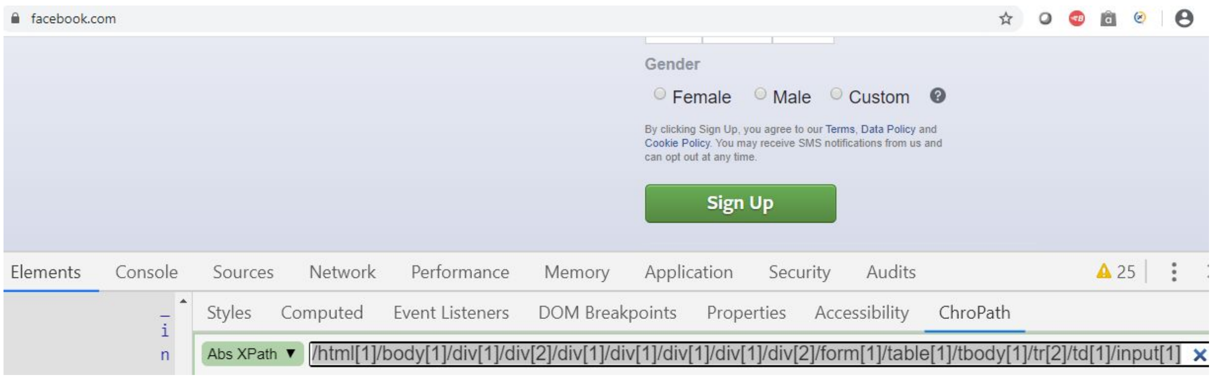
/html[1]/body[1]/div[1]/div[2]/div[1]/div[1]/div[1]/div[1]/div[2]/form[1]/table[1]/tbody[1]/tr[2]/td[1]/input[1]

1 element matching.

Figure 16

Select entire absolute xpath, copy it & paste it in notepad. Notice that it starts with /html
/html[1]/body[1]/div[1]/div[2]/div[1]/div[1]/div[1]/div[1]/div[2]/form[1]/table[1]/tbody[1]/tr[2]/td[1]/input[1]

Now refresh facebook.com and come to the page bottom where you see the radio buttons



/html[1]/body[1]/div[1]/div[2]/div[1]/div[1]/div[1]/div[1]/div[2]/form[1]/table[1]/tbody[1]/tr[2]/td[1]/input[1]

Figure 17

Click ✕ that you see at the end of 'Abs Xpath'. Below should be seen

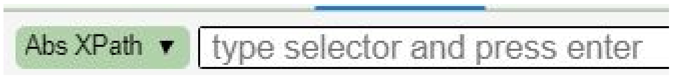


Figure 18

Select 'Selectors' from the dropdown

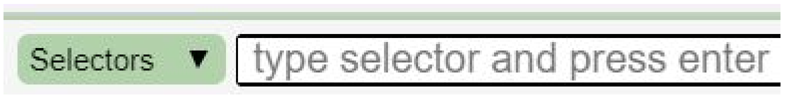
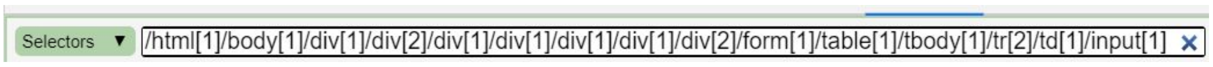


Figure 19

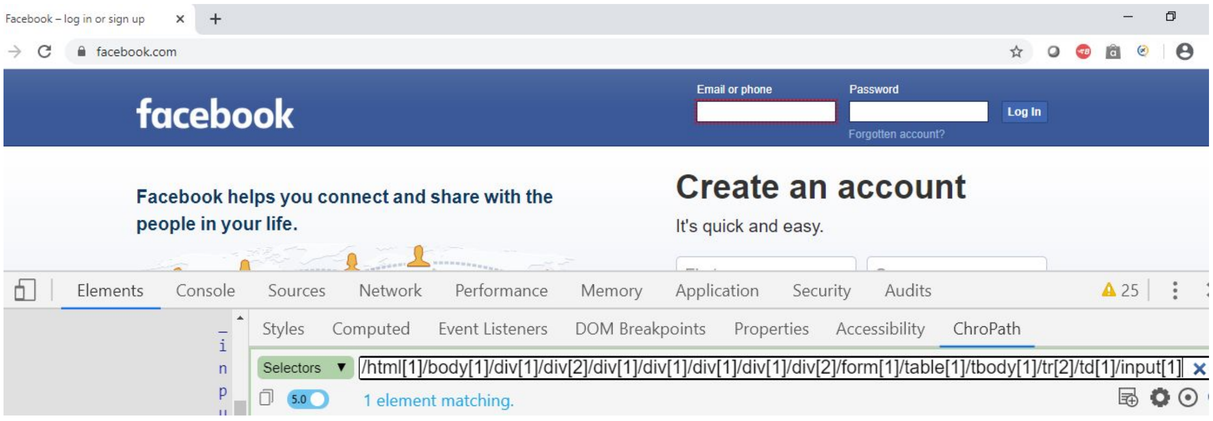
Copy the absolute xpath from notepad and paste it in 'Selectors' field



Selectors ▼ /html[1]/body[1]/div[1]/div[2]/div[1]/div[1]/div[1]/div[1]/div[2]/form[1]/table[1]/tbody[1]/tr[2]/td[1]/input[1]

Figure 20

Hit enter. Notice that 'Email' field got selected instead of 'Custom'. Ideally this absolute xpath should have selected 'Custom' since that is what we originally started with. So the lesson learnt is that, we cannot blindly convert the custom xpath to absolute xpath using the above method. We have to double check if the absolute xpath identifies the correct element or not.



Selectors ▼ /html[1]/body[1]/div[1]/div[2]/div[1]/div[1]/div[1]/div[1]/div[2]/form[1]/table[1]/tbody[1]/tr[2]/td[1]/input[1]

1 element matching.

Figure 21

A better way out to identify the absolute xpath is: simply right click the desired element and inspect it. Below we are inspecting 'Custom' radio button

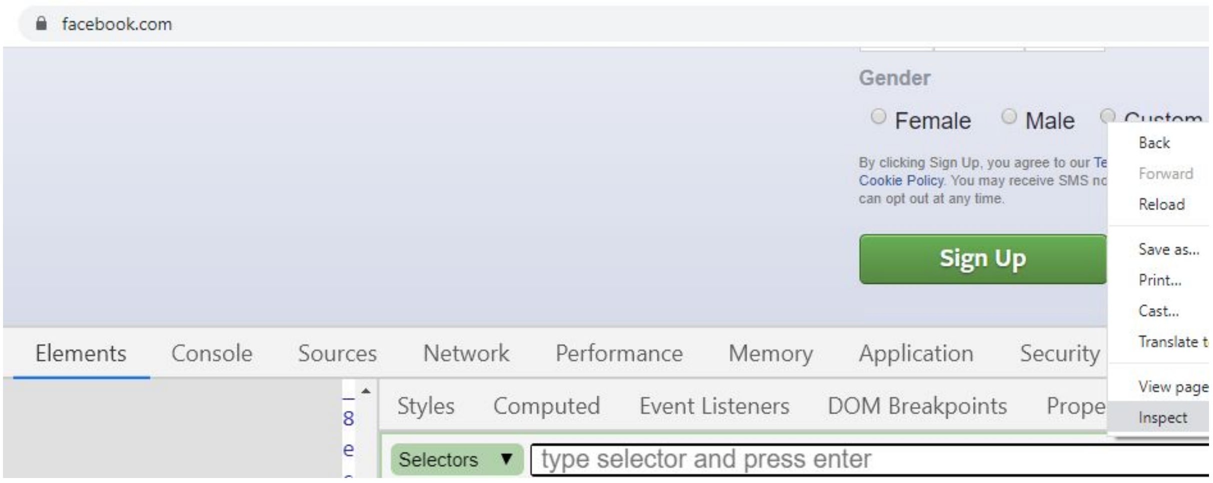
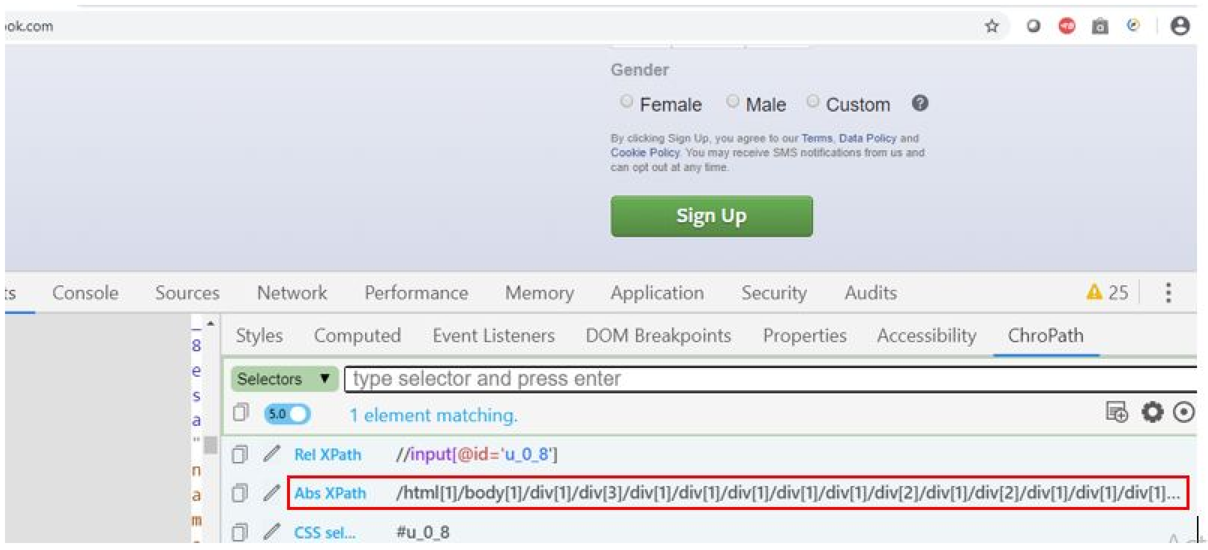


Figure 22

After you inspect, you would see 'Abs Xpath' at the bottom, see below figure



By clicking Sign Up, you agree to our Terms, Data Policy and Cookie Policy. You may receive SMS notifications from us and can opt out at any time.

Rel XPath //input[@id='u_0_8']

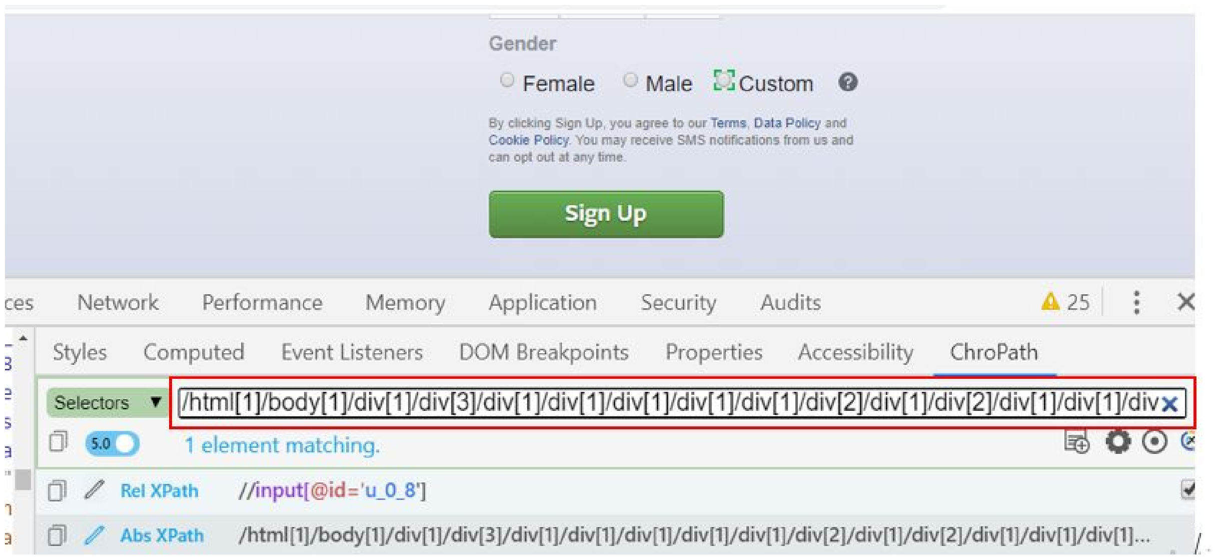
Abs XPath /html[1]/body[1]/div[1]/div[3]/div[1]/div[1]/div[1]/div[1]/div[1]/div[1]/div[2]/div[1]/div[2]/div[1]/div[1]/div[1]...

CSS sel... #u_0_8

Figure 23

Click the pencil icon seen along with the 'Abs XPath'

You would than see the absolute xpath in the 'Selectors' field, see below



By clicking Sign Up, you agree to our Terms, Data Policy and Cookie Policy. You may receive SMS notifications from us and can opt out at any time.

/html[1]/body[1]/div[1]/div[3]/div[1]/div[1]/div[1]/div[1]/div[1]/div[2]/div[1]/div[2]/div[1]/div[1]/div

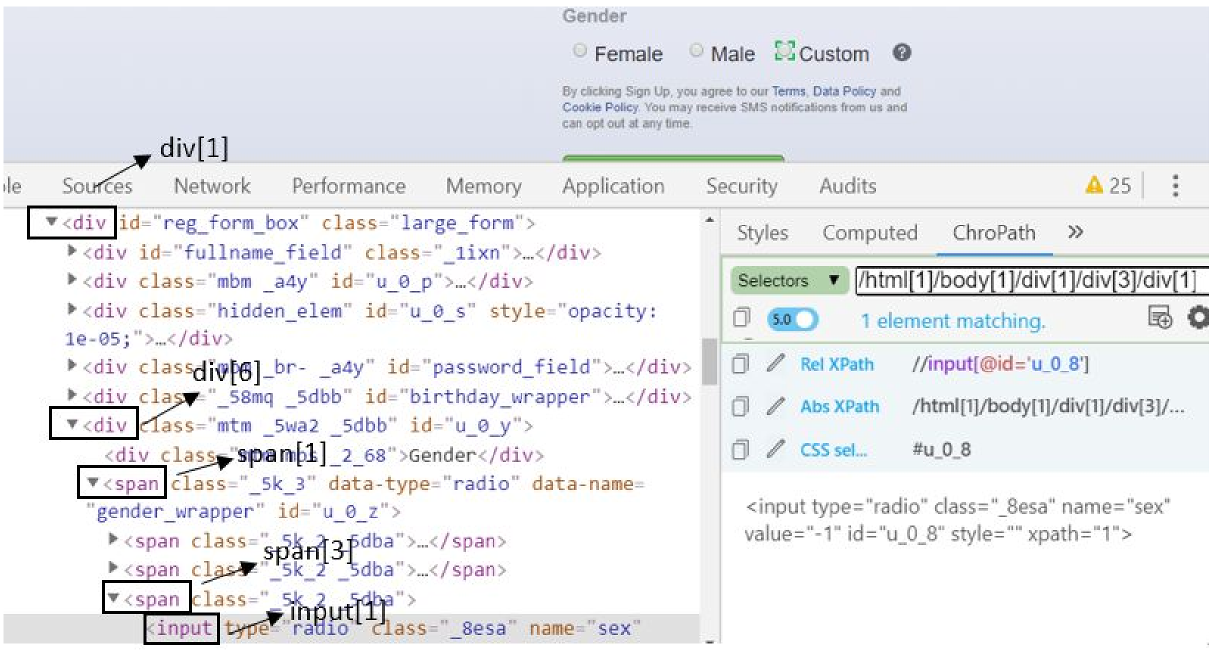
Rel XPath //input[@id='u_0_8']

Abs XPath /html[1]/body[1]/div[1]/div[3]/div[1]/div[1]/div[1]/div[1]/div[1]/div[2]/div[1]/div[2]/div[1]/div[1]/div[1]...

Figure 24

Copy the absolute xpath and paste it, see below. Notice that absolute path starts with /html:
/html[1]/body[1]/div[1]/div[3]/div[1]/div[1]/div[1]/div[1]/div[1]/div[2]/div[1]/div[2]/div[1]/div[1]/div[1]/div[1]/form[1]/**div[1]/div[6]/span[1]/span[3]/input[1]**

Let us try to understand a portion of above absolute xpath. Look at the **bold** portion above. This can be mapped to the figure below. So, under div[1], we are expanding the 6th div tag/child viz div[6]. Under div[6], we are expanding the 1st span tag/child viz span[1]. Under span[1], we are expanding the 3rd span tag/child viz span[3]. Under span[3], we are focussing 1st input tag viz input[1]. So this is how we can construct the absolute xpath.



```
▼<div id="reg_form_box" class="large_form">
  ▶<div id="fullname_field" class="_1ixn">…</div>
  ▶<div class="mbm _a4y" id="u_0_p">…</div>
  ▶<div class="hidden_elem" id="u_0_s" style="opacity:
  1e-05;">…</div>
  ▶<div class="mbm _br- _a4y" id="password_field">…</div>
  ▶<div class="_58mq _5dbb" id="birthday_wrapper">…</div>
  ▼<div class="mtm _5wa2 _5dbb" id="u_0_y">
     <div class="mtm mbs _2_68">Gender</div>
    ▼<span class="_5k_3" data-type="radio" data-name=
    "gender_wrapper" id="u_0_z">
      ▶<span class="_5k_2 _5dba">…</span>
      ▶<span class="_5k_2 _5dba">…</span>
      ▼<span class="_5k_2 _5dba">
         <input type="radio" class="_8esa" name="sex"
```

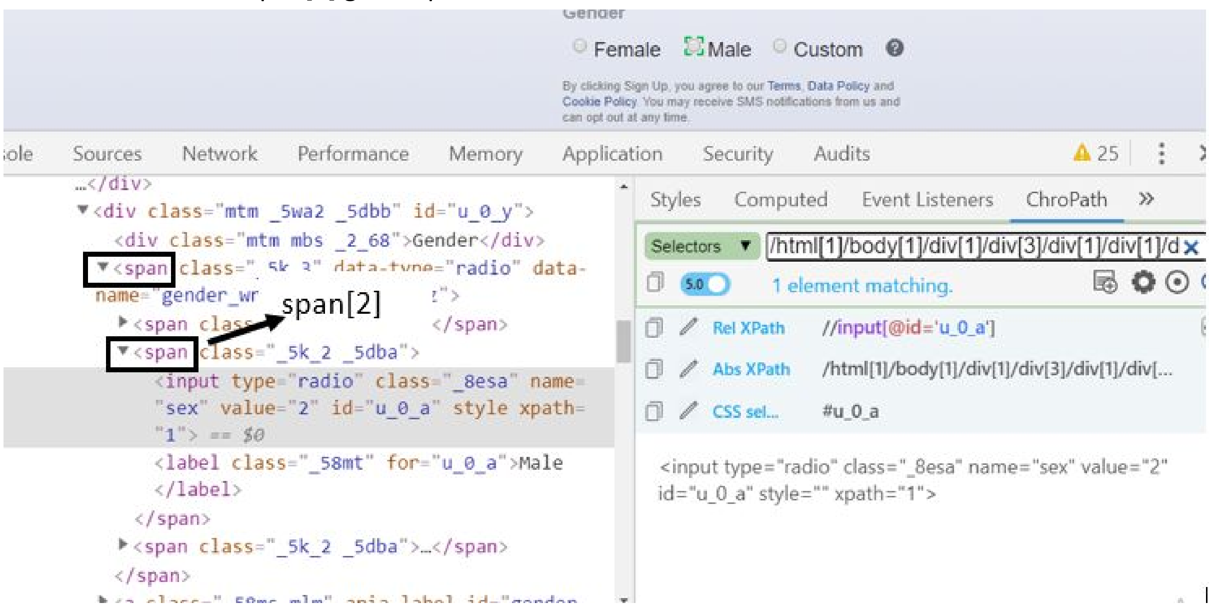
```
/html[1]/body[1]/div[1]/div[3]/div[1]
Rel XPath   //input[@id='u_0_8']
Abs XPath   /html[1]/body[1]/div[1]/div[3]/...
CSS sel...  #u_0_8

<input type="radio" class="_8esa" name="sex"
value="-1" id="u_0_8" style="" xpath="1">
```

Figure 25

Now in our absolute xpath, let us change **span[3]** to **span[2]** viz
/html[1]/body[1]/div[1]/div[3]/div[1]/div[1]/div[1]/div[1]/div[1]/div[2]/div[1]/div[2]/div[1]/div[1]/div[1]/div[1]/form[1]/div[1]/div[6]/span[1]/**span[2]**/input[1]

Inspect the above absolute xpath, notice that 'Male' radio button gets highlighted this time. Also notice this time that span[2] gets expanded



```
  …</div>
 ▼<div class="mtm _5wa2 _5dbb" id="u_0_y">
    <div class="mtm mbs _2_68">Gender</div>
   ▼<span class="_5k_3" data-type="radio" data-
   name="gender_wr            ">
     ▶<span class                </span>
     ▼<span class="_5k_2 _5dba">
        <input type="radio" class="_8esa" name=
        "sex" value="2" id="u_0_a" style xpath=
        "1"> == $0
        <label class="_58mt" for="u_0_a">Male
        </label>
      </span>
     ▶<span class="_5k_2 _5dba">…</span>
    </span>
```

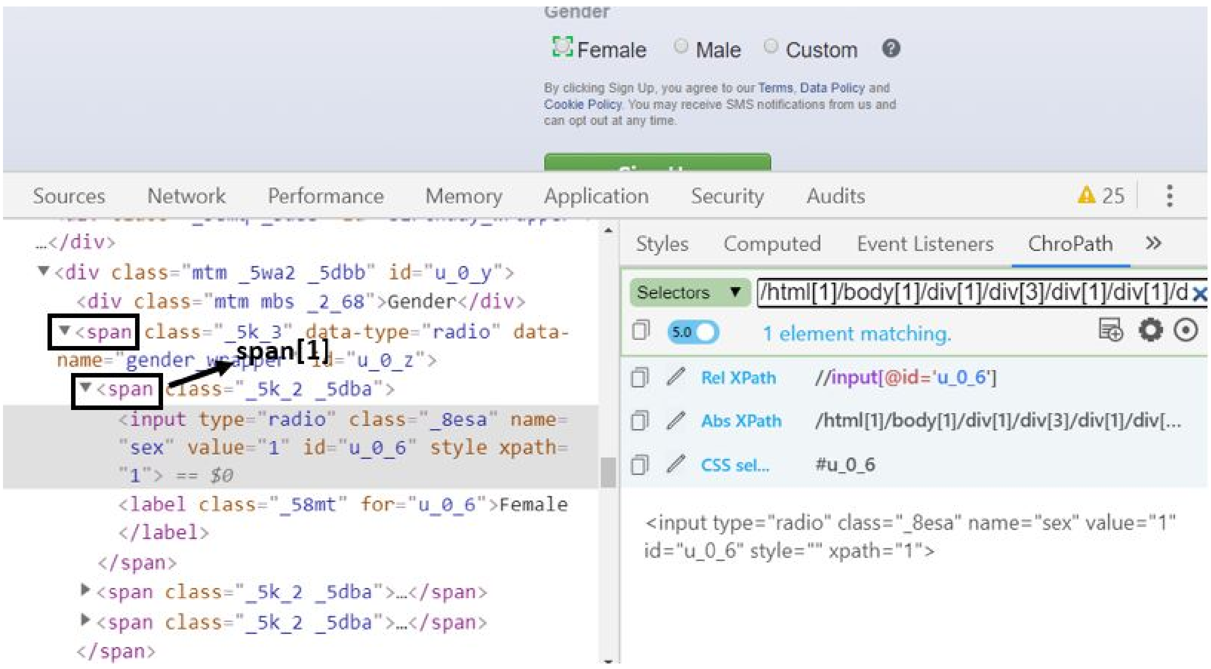
```
/html[1]/body[1]/div[1]/div[3]/div[1]/div[1]/d
Rel XPath   //input[@id='u_0_a']
Abs XPath   /html[1]/body[1]/div[1]/div[3]/div[1]/div[...
CSS sel...  #u_0_a

<input type="radio" class="_8esa" name="sex" value="2"
id="u_0_a" style="" xpath="1">
```

Figure 26

Now in our absolute xpath, let us change **span[2]** to **span[1]** viz
/html[1]/body[1]/div[1]/div[3]/div[1]/div[1]/div[1]/div[1]/div[1]/div[2]/div[1]/div[2]/div[1]/div[1]/div[1]/div[1]/form[1]/div[1]/div[6]/span[1]/**span[1]**/input[1]

Inspect the above absolute xpath, notice that 'Female' radio button gets highlighted this time. Also notice this time that span[1] gets expanded


Gender
Female
Male
Custom
By clicking Sign Up, you agree to our Terms, Data Policy and Cookie Policy. You may receive SMS notifications from us and can opt out at any time.
Sources
Network
Performance
Memory
Application
Security
Audits
25
Styles
Computed
Event Listeners
ChroPath
Selectors
5.0
span[1]

```
…</div>
▼<div class="mtm _5wa2 _5dbb" id="u_0_y">
   <div class="mtm mbs _2_68">Gender</div>
  ▼<span class="_5k_3" data-type="radio" data-
  name="gender_wrapper" id="u_0_z">
    ▼<span class="_5k_2 _5dba">
        <input type="radio" class="_8esa" name=
        "sex" value="1" id="u_0_6" style xpath=
        "1"> == $0
        <label class="_58mt" for="u_0_6">Female
        </label>
     </span>
    ▶<span class="_5k_2 _5dba">…</span>
    ▶<span class="_5k_2 _5dba">…</span>
   </span>
```

/html[1]/body[1]/div[1]/div[3]/div[1]/div[1]/d

1 element matching.

Rel XPath	//input[@id='u_0_6']
Abs XPath	/html[1]/body[1]/div[1]/div[3]/div[1]/div[...
CSS sel...	#u_0_6

```
<input type="radio" class="_8esa" name="sex" value="1"
id="u_0_6" style="" xpath="1">
```

Figure 27

So this is how we construct absolute xpaths. There is a disadvantage of using absolute xpath. Let us consider below absolute xpath. If the developer changes any tag in the middle of this xpath, the script will fail to identify the element. So absolute xpath is not advised.

/html[1]/body[1]/div[1]/div[3]/div[1]/div[1]/div[1]/div[1]/div[1]/div[2]/div[1]/div[2]/div[1]/div[1]/div[1]/div[1]/form[1]/div[1]/div[6]/span[1]/span[1]/input[1]

We will continue with locators in our next tutorial. Thank you for reading!