

Locators (Part 2)

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 started with locators. In this tutorial we will continue looking at them!

What you will Learn:

Css syntax tagname.classname

Xpath regular expression locator to identify element

Css regular expression locator to identify element

Css syntax tagname.classname:

Let us continue with the 'Facebook' class that we created in our previous tutorial. In line#14, we had used By.id locator to identify the input field. Let us comment line#14

```
5 public class Facebook {
6
7= public static void main(String[] args) {
8     System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TRAINING\\Software\\chromedriver.exe");
9
10    WebDriver driver = null;
11
12    driver = new ChromeDriver();
13    driver.get("https://facebook.com");
14    //driver.findElement(By.id("email")).sendKeys("dummy@gmail.com");
15    driver.findElement(By.name("pass")).sendKeys("dummy");
16
17    //driver.findElement(By.xpath("//input[@value='Log In']")).click(); //xpath method 1: //tagname[@attribute='value']
18    //driver.findElement(By.xpath("//*[@value='Log In']")).click(); //xpath method 2: //*[@attribute='value']
19
20    //driver.findElement(By.cssSelector("input[value='Log In']")).click(); //css method 1: tagname[attribute='value']
21    //driver.findElement(By.cssSelector("[value='Log In']")).click(); //css method 2: [attribute='value']
22    //driver.findElement(By.cssSelector("input#u_0_2")).click(); //css method 3: tagname#id
23    //driver.findElement(By.cssSelector("#u_0_2")).click(); //css method 4: #id
24    //driver.findElement(By.linkText("Forgotten account?")).click();
25 }
```

Figure 1

Let us now study about another css syntax **tagname.classname**

This syntax can be used only if the element has 'class' attribute. Let us inspect the email id field on facebook page. It is represented by 'input' tag. You see that it has 'class' attribute. The value of class attribute contains 2 classes: inputtext and login_form_input_box

Since there are 2 classes, we can crate 2 css paths based on above css syntax viz

tagname.classname1 and **tagname.classname2**

So, the first css path would be: **input.inputtext**

The second css path can be: **input.login_form_input_box**

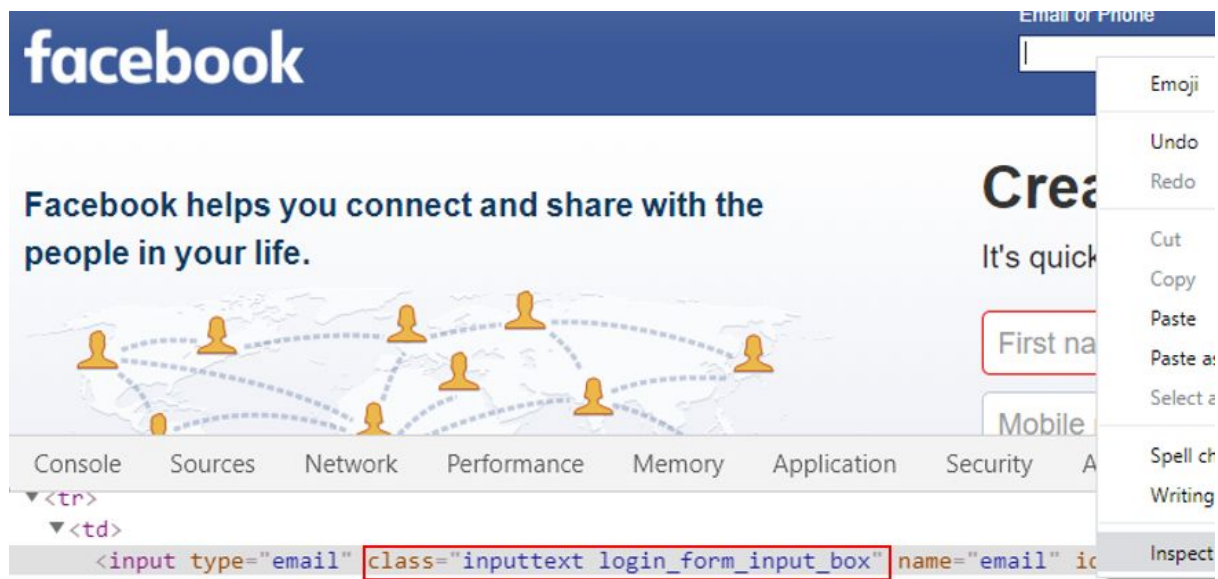


Figure 2

To enter text in input field, let us use By.cssSelector with first css path **input.inputtext**

```

5  lic class Facebook {
6
7  public static void main(String[] args) {
8      System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TRAINING\\S
9
10     WebDriver driver = null;
11
12     driver = new ChromeDriver();
13     driver.get("https://facebook.com");
14     //driver.findElement(By.id("email")).sendKeys("dummy@gmail.com");
15     driver.findElement(By.cssSelector("input.inputtext")).sendKeys("dummy@gmail.com");
16     driver.findElement(By.name("pass")).sendKeys("dummy");

```

Figure 3

Run script, notice that dummy email id gets typed in the input field

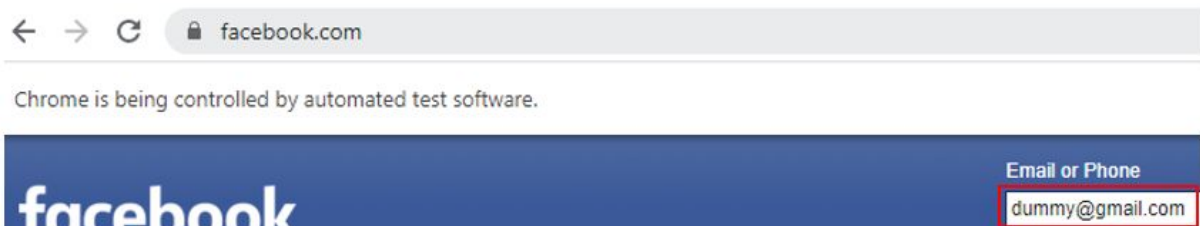


Figure 4

Comment line 15

```

7  public static void main(String[] args) {
8      System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TRAINING\\S
9
10     WebDriver driver = null;
11
12     driver = new ChromeDriver();
13     driver.get("https://facebook.com");
14     //driver.findElement(By.id("email")).sendKeys("dummy@gmail.com");
15     //driver.findElement(By.cssSelector("input.inputtext")).sendKeys("dummy@gmail.com");

```

Figure 5

Let us now enter text in input field by using second css path **input.login_form_input_box**

```
8      System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TRAINING\\Software\\c
9
10     WebDriver driver = null;
11
12     driver = new ChromeDriver();
13     driver.get("https://facebook.com");
14     //driver.findElement(By.id("email")).sendKeys("dummy@gmail.com");
15     //driver.findElement(By.cssSelector("input.inputtext")).sendKeys("dummy@gmail.com"); //css met
16     driver.findElement(By.cssSelector("input.login_form_input_box")).sendKeys("dummy@gmail.com");
17     driver.findElement(By.name("pass")).sendKeys("dummy");
```

Figure 6

Run script, notice that dummy email id gets typed in the input field

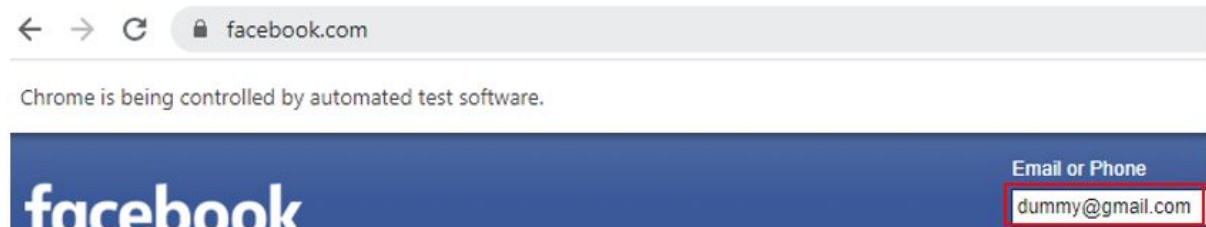


Figure 7

Using regular expression xpath to identify element:

We can also use regular expressions to identify an element. Sometimes the value of an attribute might dynamically change. For example, the value of 'id' attribute might be 123mai456 in one session. In another session it might be 789mai543. Notice that start and end of these values are changing dynamically. However we have a static content in both of these 'mai'. So we can use this in our regex. We can use 'contains' keyword to form a regular expression. So the regular expression syntax for xpath would be **//tagname[contains(@attribute,'value')]**

Open a new session of chrome and inspect 'Log In' button. Note down the value of 'id' attribute (u_0_b)

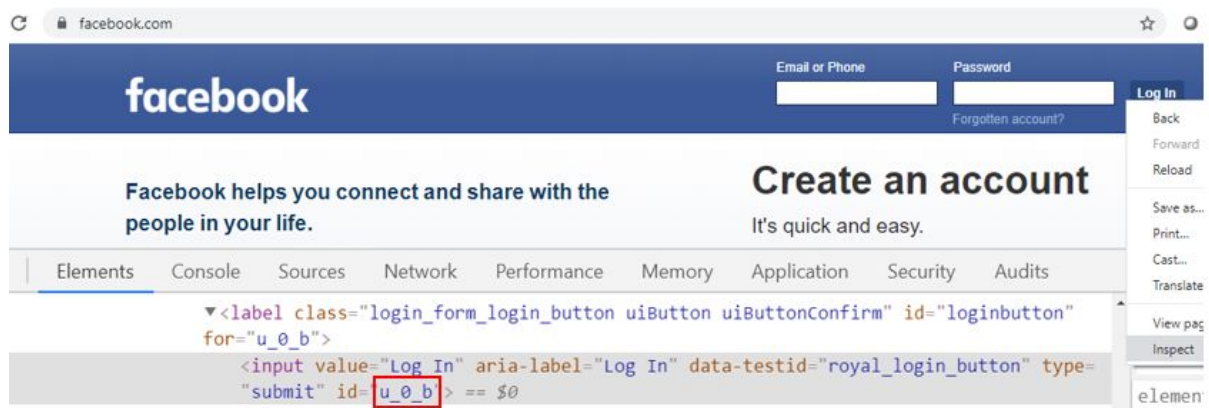


Figure 8

Close the above session. Open a new session of chrome and inspect 'Log In' button. The value if 'id' attribute changes. Note down the value of 'id' attribute (u_0_2). The static part in these two cases is: u_0_

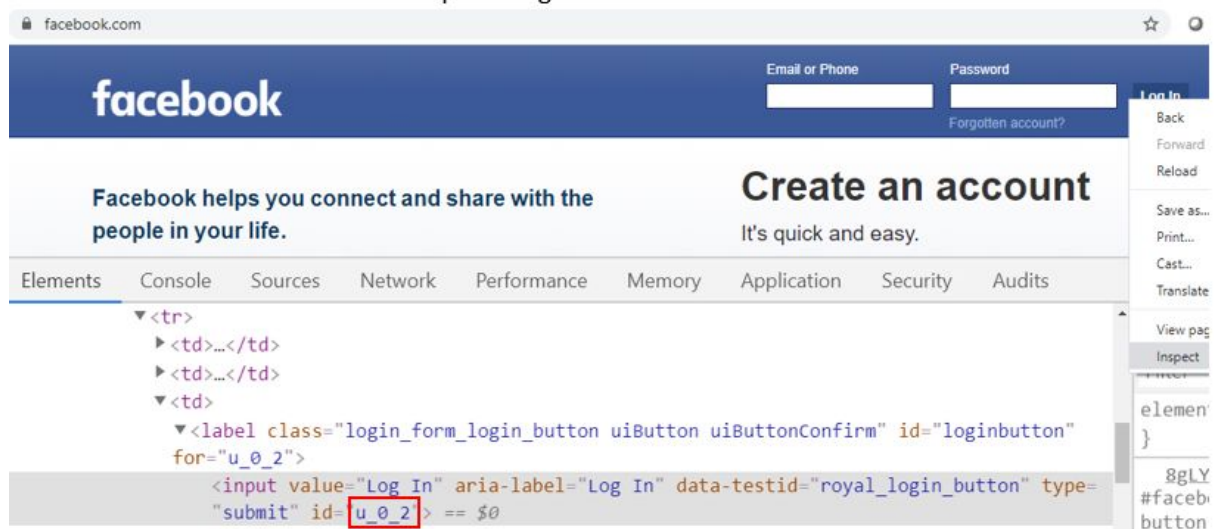


Figure 9

Similarly if you open another session and inspect, the value changes. To handle this scenario, let us create a new class to demonstrate the use of regex and write line#17. So we are using the static value of the attribute

```

5 public class Regex {
6
7     public static void main(String[] args) {
8         System.setProperty("webdriver.chrome.driver", "C:\\\\Users\\DELL\\Desktop
9
10        WebDriver driver = null;
11
12        driver = new ChromeDriver();
13
14        driver.get("https://facebook.com");
15        driver.findElement(By.id("email")).sendKeys("dummy@gmail.com");
16        driver.findElement(By.name("pass")).sendKeys("dummy");
17        driver.findElement(By.xpath("//input[contains(@id,'u_0_')]")).click();

```

Figure 10

Run, the script enters userid/password and clicks the 'Log In' button, below page comes up

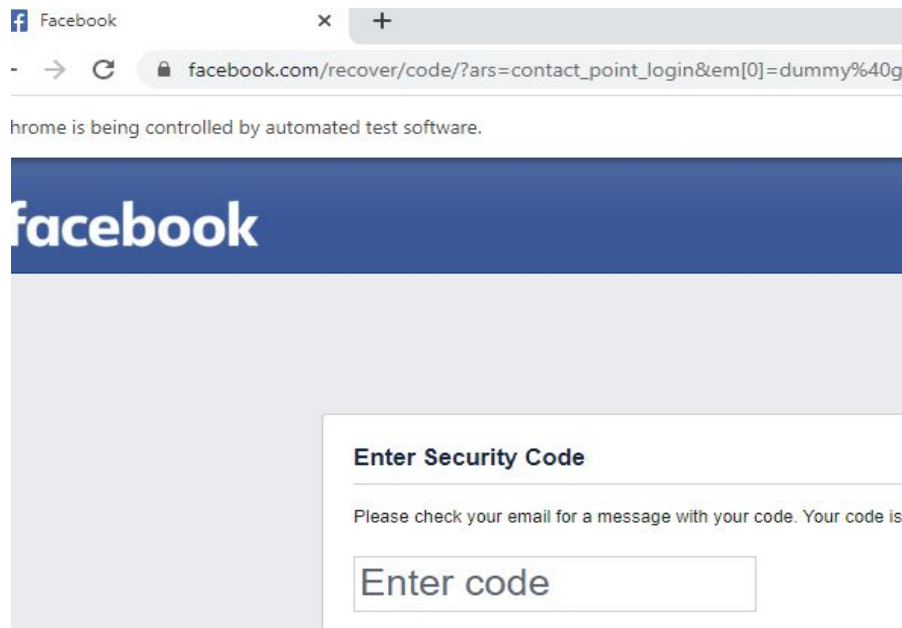


Figure 11

Comment line#17. Now, instead of regular expression, let us try to use the entire value of 'id' attribute, see line#18

```

5 public class Regex {
6
7     public static void main(String[] args) {
8         System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\
9
10        WebDriver driver = null;
11
12        driver = new ChromeDriver();
13
14        driver.get("https://facebook.com");
15        driver.findElement(By.id("email")).sendKeys("dummy@gmail.com");
16        driver.findElement(By.name("pass")).sendKeys("dummy");
17        //driver.findElement(By.xpath("//input[contains(@id,'u_0_')]")).click();
18        driver.findElement(By.xpath("//input[@id='u_0_2']")).click();

```

Figure 12

Run, you will notice 'NoSuchElementException' because the value of 'id' attribute changes at runtime and hence the selenium script is not able to find any element on the webpage that has the value of 'id' attribute as 'u_0_2'. Also notice in the figure below that the 'Log In' button does not get clicked.

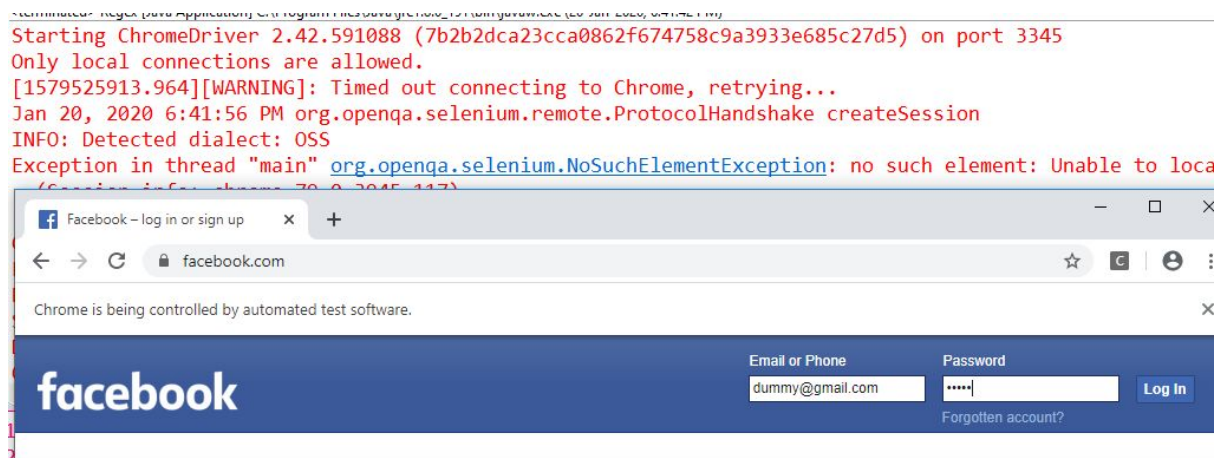


Figure 13

Another example: Go to google.com & inspect the search field. Notice the value of 'data-ved' attribute. It is dynamic in nature

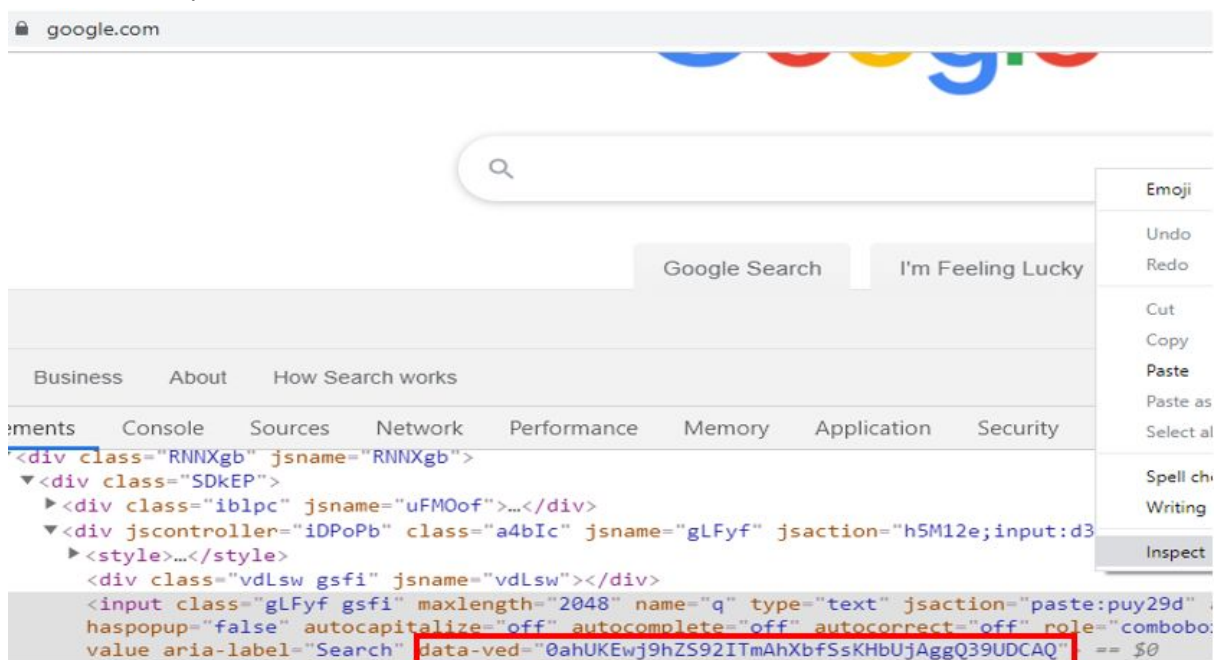


Figure 14

Copy and paste the value in a notepad. Open a new session 2 more times & copy/paste the values.

The red text below is dynamic in nature, the other text is static in every session:

0ahUKEwj-zOWI2ITmAhXXfCsKHREdABAQ39UDCAQ

0ahUKEwjTxvSj2ITmAhXGbSsKHbUjAggQ39UDCAQ

0ahUKEwj9hZS92ITmAhXbfSsKHbUjAggQ39UDCAQ

Let us use any portion of static text in our script (example: 9UDCA). Comment lines 14-18. Add lines 21 & 22. Ensure that you use By.xpath in line#22

```

5 public class Regex {
6
7     public static void main(String[] args) {
8         System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TRAINING\\Sof
9
10        WebDriver driver = null;
11
12        driver = new ChromeDriver();
13        /*
14        driver.get("https://facebook.com");
15        driver.findElement(By.id("email")).sendKeys("dummy@gmail.com");
16        driver.findElement(By.name("pass")).sendKeys("dummy");
17        //driver.findElement(By.xpath("//input[contains(@id,'u_0_')]")).click();
18        driver.findElement(By.xpath("//input[@id='u_0_2']")).click();
19        */
20
21        driver.get("https://google.com");
22        driver.findElement(By.xpath("//input[contains(@data-ved,'9UDCA')]")).sendKeys("home");

```

Figure 15

When we run the script, notice below that 'home' text is being entered in the google search textbox

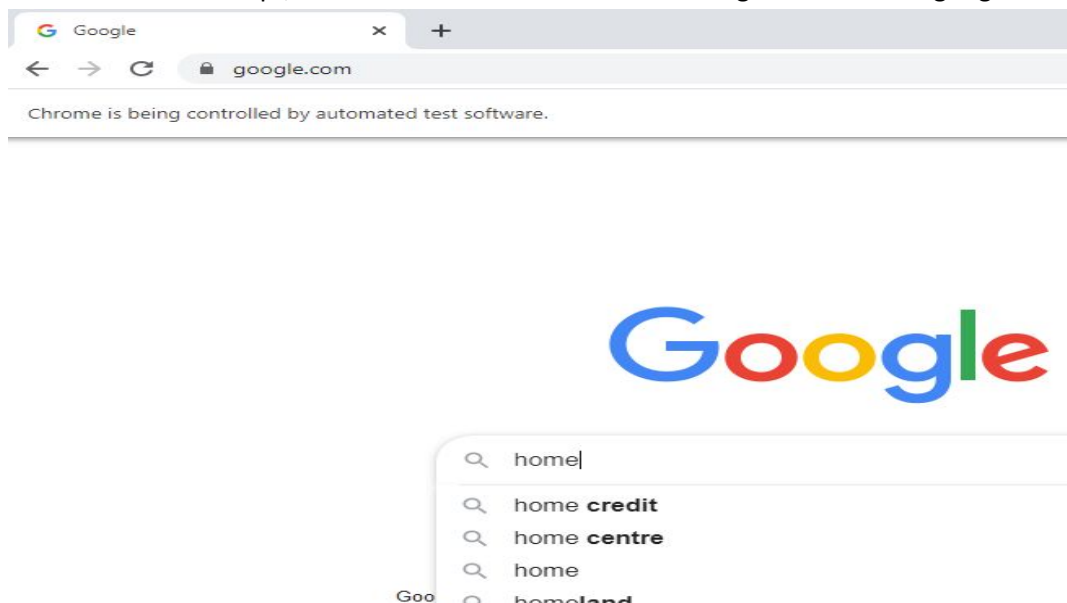


Figure 16

Let us now comment line#22

```

5 public class Regex {
6
7     public static void main(String[] args) {
8         System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TRAINING\\Software\\chromedri
9
10        WebDriver driver = null;
11
12        driver = new ChromeDriver();
13        /*
14        driver.get("https://facebook.com");
15        driver.findElement(By.id("email")).sendKeys("dummy@gmail.com");
16        driver.findElement(By.name("pass")).sendKeys("dummy");
17        //driver.findElement(By.xpath("//input[contains(@id,'u_0_')]")).click();
18        driver.findElement(By.xpath("//input[@id='u_0_2']")).click();
19        */
20
21        driver.get("https://google.com");
22        //driver.findElement(By.xpath("//input[contains(@data-ved,'9UDCA')]")).sendKeys("home");//regex xpath

```

Figure 17

Using regular expression css to identify element:

We have seen above the syntax of xpath regex: `//tagname[contains(@attribute,'value')]`.

The syntax of css regex is slightly simpler. It does not use 'contains' keyword. Instead it uses star * **tagname[attribute*='value']**

Ensure that you use `By.cssSelector` in line#24

```
5 public class Regex {
6
7     public static void main(String[] args) {
8         System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TRAINING\\Software\\chr
9
10        WebDriver driver = null;
11
12        driver = new ChromeDriver();
13        /*
14        driver.get("https://facebook.com");
15        driver.findElement(By.id("email")).sendKeys("dummy@gmail.com");
16        driver.findElement(By.name("pass")).sendKeys("dummy");
17        //driver.findElement(By.xpath("//input[contains(@id,'u_0_')]").click();
18        driver.findElement(By.xpath("//input[@id='u_0_2']")).click();
19        */
20
21        driver.get("https://google.com");
22        //driver.findElement(By.xpath("//input[contains(@data-ved,'9UDCA')]").sendKeys("home");//regex
23
24        driver.findElement(By.cssSelector("input[data-ved*='9UDCA']")).sendKeys("selenium");//regex css
```

Figure 18

When we run the script, notice below that 'selenium' text is being entered in the google search textbox

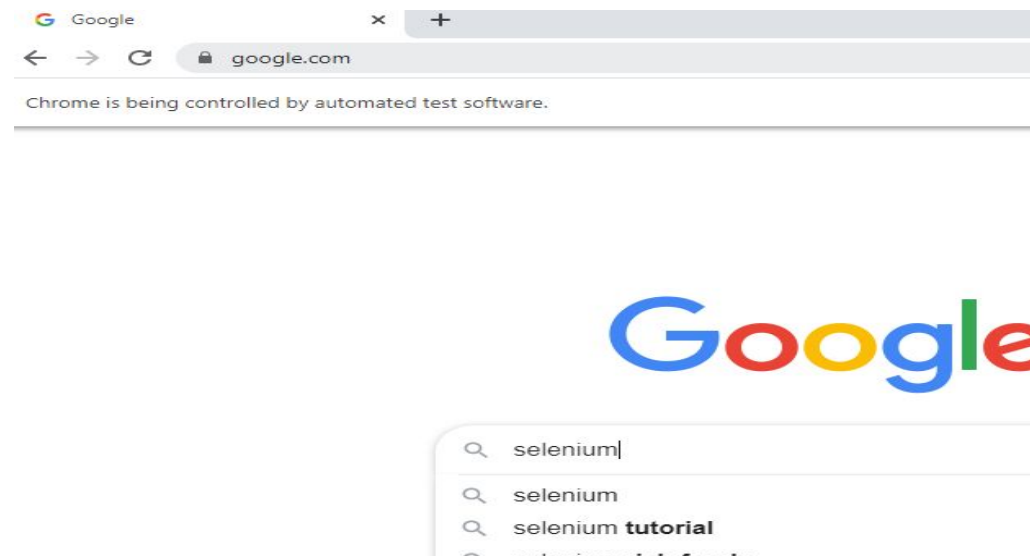


Figure 19

Let us see another example. Inspect 'Advertising' link on google.com.

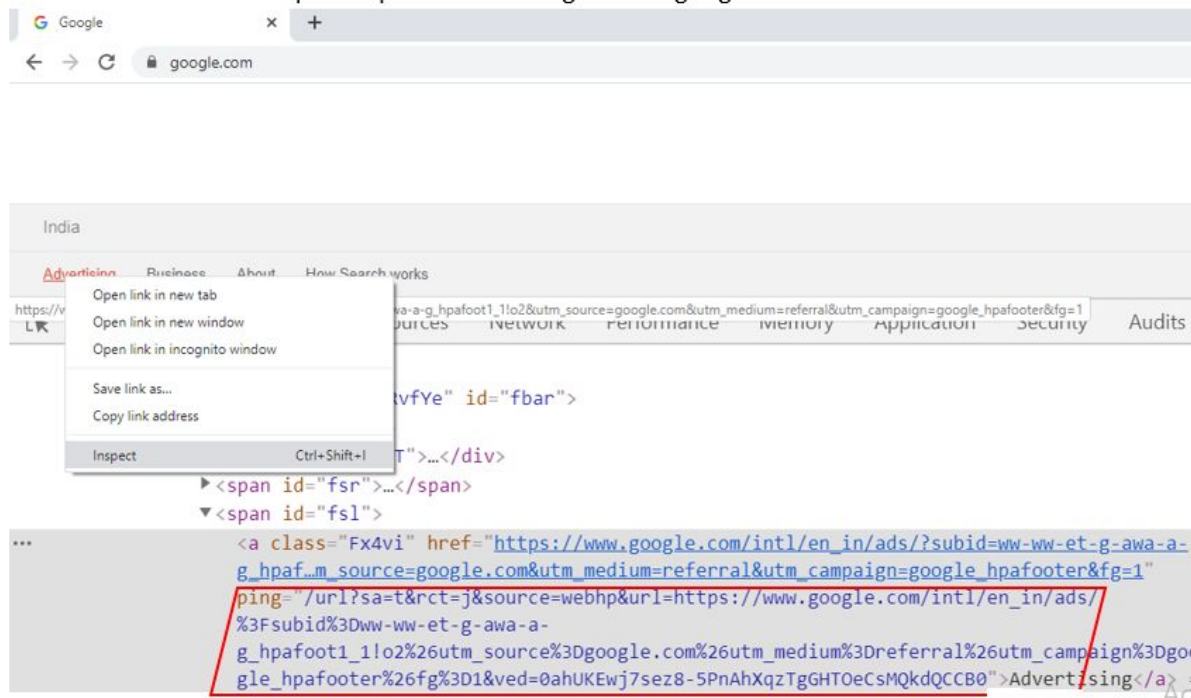


Figure 20

Look at the value of 'ping' attribute, copy it and paste it in notepad

```
/url?sa=t&rct=j&source=webhp&url=
https://www.google.com/intl/en_in/ads/%3Fsubid%3Dww-ww-et-g-awa-a-g_hpafoot1_1!o2%26utm_sour
ce%3Dgoogle.com%26utm_medium%3Dreferral%26utm_campaign%3Dgoogle_hpafooter%26fg%3D1&ved=0ahUK
Ewj7sez8-5PnAhXqzTgGHTOeCsMQkdQCCB0
```

Figure 21

Similarly, open 2 more new sessions of chrome and paste the value of 'ping' attribute from each of these sessions. So, including the above session (figure 21), we have 3 sessions. The text inside the red block in each of these 3 sessions is dynamic in nature, it changes with every new session.

```
/url?sa=t&rct=j&source=webhp&url=
https://www.google.com/intl/en_in/ads/%3Fsubid%3Dww-ww-et-g-awa-a-g_hpafoot1_1!o2%26utm_sour
ce%3Dgoogle.com%26utm_medium%3Dreferral%26utm_campaign%3Dgoogle_hpafooter%26fg%3D1&ved=0ahUK
Ewj7sez8-5PnAhXqzTgGHTOeCsMQkdQCCB0

/url?sa=t&rct=j&source=webhp&url=
https://www.google.com/intl/en_in/ads/%3Fsubid%3Dww-ww-et-g-awa-a-g_hpafoot1_1!o2%26utm_sour
ce%3Dgoogle.com%26utm_medium%3Dreferral%26utm_campaign%3Dgoogle_hpafooter%26fg%3D1&ved=0ahUK
Ewj7sez8-5PnAhXqzTgGHTOeCsMQkdQCCB0

/url?sa=t&rct=j&source=webhp&url=
https://www.google.com/intl/en_in/ads/%3Fsubid%3Dww-ww-et-g-awa-a-g_hpafoot1_1!o2%26utm_sour
ce%3Dgoogle.com%26utm_medium%3Dreferral%26utm_campaign%3Dgoogle_hpafooter%26fg%3D1&ved=0ahUK
Ewj7sez8-5PnAhXqzTgGHTOeCsMQkdQCCB0
```

Figure 22

See below. Comment line#24. Add line#25 wherein we are using a static text in our regex

```

5 public class Regex {
6
7     public static void main(String[] args) {
8         System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TRAINING\\Software\\chrome
9
10        WebDriver driver = null;
11
12        driver = new ChromeDriver();
13        /*
14        driver.get("https://facebook.com");
15        driver.findElement(By.id("email")).sendKeys("dummy@gmail.com");
16        driver.findElement(By.name("pass")).sendKeys("dummy");
17        //driver.findElement(By.xpath("//input[contains(@id,'u_0_')]")).click();
18        driver.findElement(By.xpath("//input[@id='u_0_2']")).click();
19        */
20
21        driver.get("https://google.com");
22        //driver.findElement(By.xpath("//input[contains(@data-ved,'9UDCA')]")).sendKeys("home");//regex xpa
23
24        //driver.findElement(By.cssSelector("input[data-ved*='9UDCA']")).sendKeys("selenium");//regex css
25        driver.findElement(By.cssSelector("a[ping*='QCC']")).click();//regex css

```

Figure 23

Run the script, notice that 'Advertising' link gets clicked & below page comes up

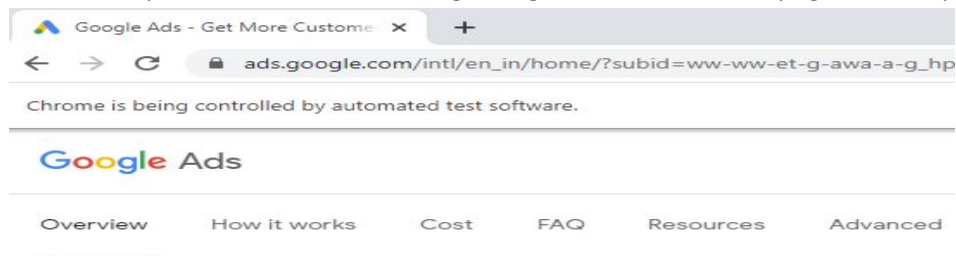


Figure 24

In our next tutorial, we will continue learning more locators. Thank you for reading!