Checkboxes, Radiobuttons, links

This is the next tutorial in selenium-java series. Please go through the previous tutorials before you start this one. In the last few tutorials, we learned about locators. In this tutorial we will see how to identify checkboxes, radio buttons, links!

What you will Learn:
1. Identify checkboxes
2. Identify radio buttons
3. Identify links on a page

Identify Checkboxes:
Checkbox is straightforward. Go to https://www.goair.in/ and inspect 'Armed Forces' checkbox

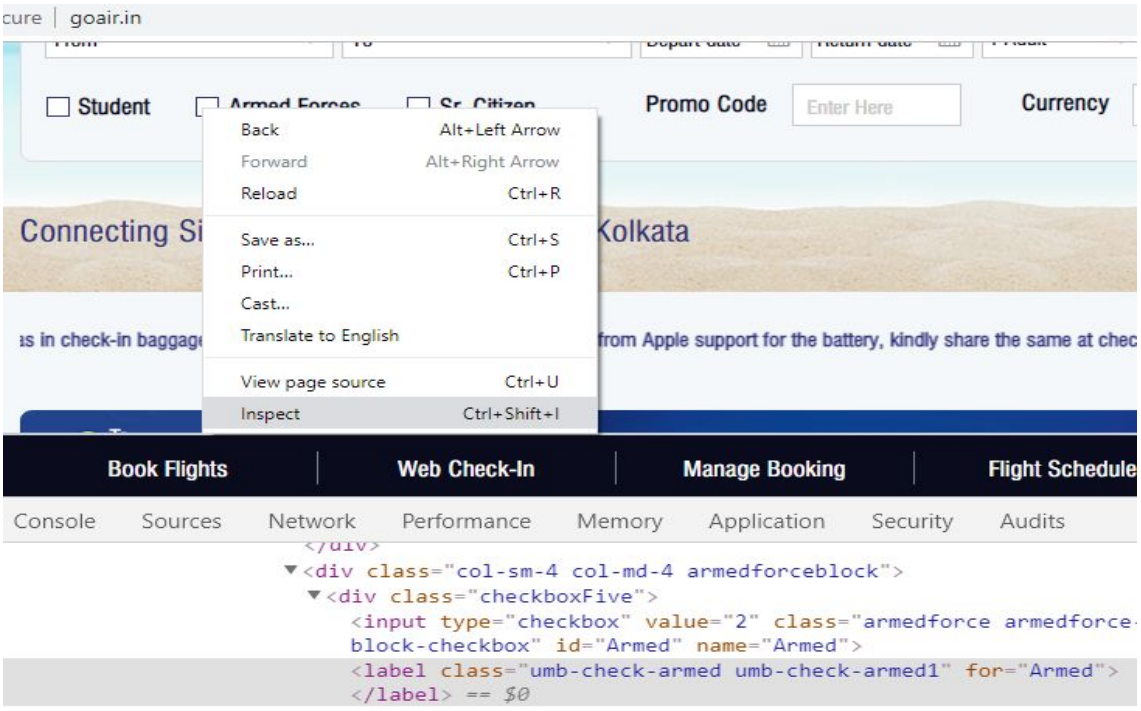

Figure 1

So it is represented by label and we can write an xpath

```
12          driver = new ChromeDriver();
13
14          driver.get("http://goair.in/");
15          driver.findElement(By.xpath("//label[@for='Armed']")).click();
```

Figure 2

Run the script, notice below that 'Armed Forces' checkbox gets selected

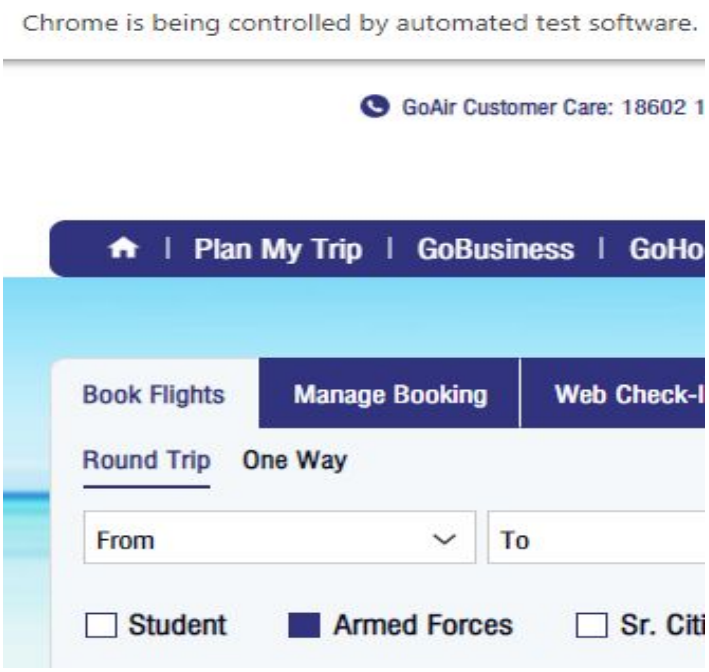

Figure 3

Similarly you can try out select/de-select checkboxes on various other sites.

Identify radio buttons:

We will now see how to identify radio button and will see the usage of findelements and GetAttribute methods. Navigate to http://echoecho.com/htmlforms10.htm and come to the page bottom where you see the radio buttons. Inspect the radio buttons. You will notice that there are 2 groups:

group1 contains 3 elements: Milk, Butter and cheese

group2 contains 3 elements: Water, Beer and Wine

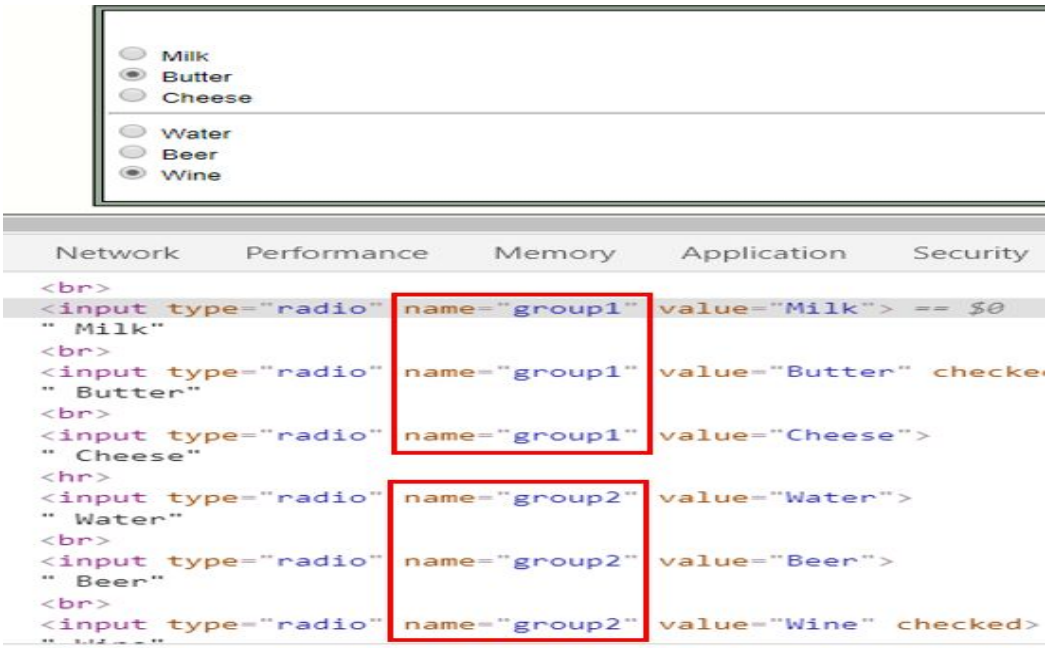

Figure 4

We can use “findelement**s**” method to find all the 3 elements of group1. Look at the figure below. Using ‘for’ loop, we will click the radio buttons one by one. In line 16, we are storing the integer value returned by ‘size’ method inside ‘count’ variable. We will than use the ‘count’ variable in ‘for’ loop. Notice that we are using ‘findElements’ method since there is more than element in a group viz 3 radio buttons. So in line#19, we are fetching/getting all the 3 radio buttons one by one and then clicking them.

```
14          driver.get("http://echoecho.com/htmlforms10.htm");
15
16          int count = driver.findElements(By.xpath("//input[@name='group1']")).size();
17
18          for(int i=0; i<count; i++) {
19          driver.findElements(By.xpath("//input[@name='group1']")).get(i).click();
20          }
21
```

Figure 5

Run the script, you will see all the 3 buttons getting clicked one by one

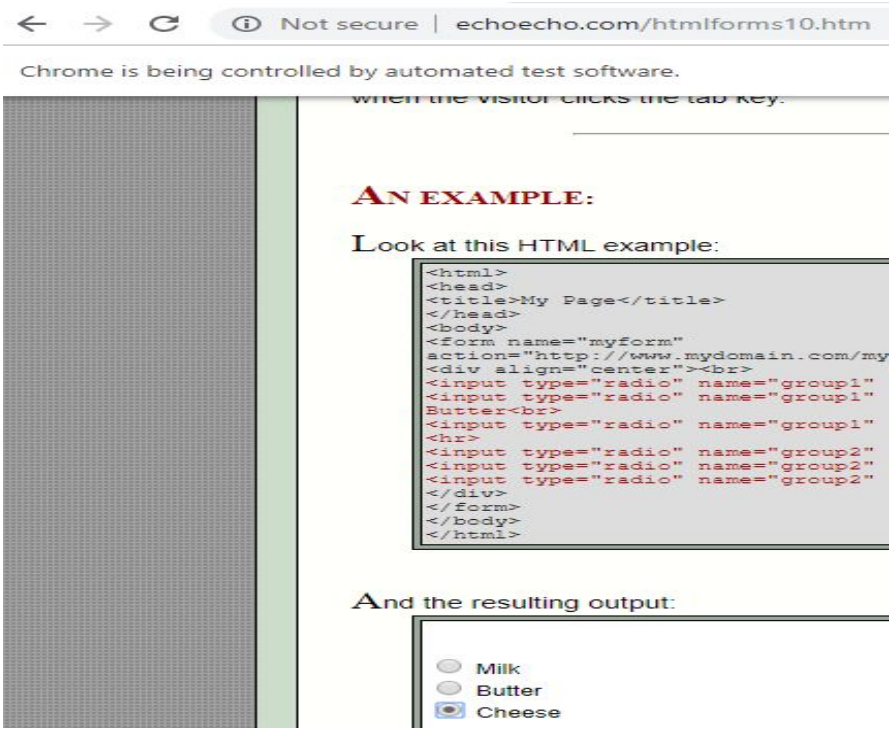

Figure 6

We can also print the values of all the 3 radios by using getAttribute method. See line#19. We are using an SOP to print the value of all the radio buttons one by one. In line#20, the actual click happens.

```
11
12          driver = new ChromeDriver();
13
14          driver.get("http://echoecho.com/htmlforms10.htm");
15
16          int count = driver.findElements(By.xpath("//input[@name='group1']")).size();
17
18          for(int i=0; i<count; i++) {
19              System.out.println(driver.findElements(By.xpath("//input[@name='group1']")).get(i).getAttribute("value"));
20              driver.findElements(By.xpath("//input[@name='group1']")).get(i).click();
```

Figure 7

So when we run the script, the value of a radio button gets printed and then it gets clicked. This repeats for all the radio buttons.

```
Milk
Butter
Cheese
```

Figure 8

If you would have noticed, when you launch this site, by default, 'Butter' radio button is seen as selected

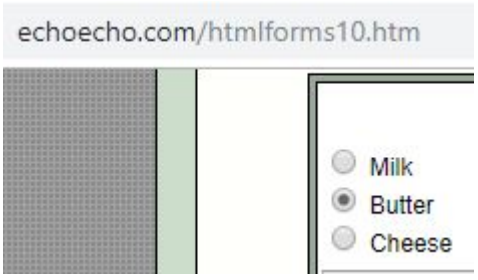

Figure 9

Now what we would like to do is that, during runtime, we would like to select 'Cheese' radio button. The logic will be that, if the value returned by getAttribute method is 'Cheese', we would like to click the 'Cheese' radio button.
To achieve this, we will store the output of line#20 in a string variable and write an 'if' condition. Comment line#19. Write an 'if' condition (line#22)

```
14          driver.get("http://echoecho.com/htmlforms10.htm");
15
16          int count = driver.findElements(By.xpath("//input[@name='group1']")).size();
17
18          for(int i=0; i<count; i++) {
19              //System.out.println(driver.findElements(By.xpath("//input[@name='group1']")).get(i).getAttribute("value"));
20              String text = driver.findElements(By.xpath("//input[@name='group1']")).get(i).getAttribute("value");
21
22              if(text.equals("Cheese"))
23              driver.findElements(By.xpath("//input[@name='group1']")).get(i).click();
24          }
```

Figure 10

Run the script, see that 'Cheese' is clicked

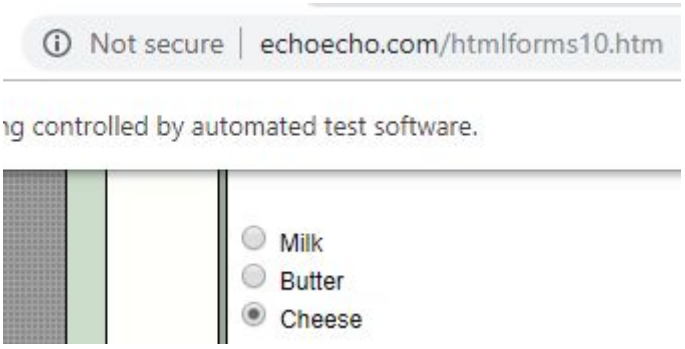

Figure 11

Identify links on a page:
Go to https://www.hollisterco.com/shop/wd

We know that a link is represented by tagname <a>. So we can count all the links on a page by using the 'size' method. The 'findElement**s**' method will find all the elements on the page having tagname 'a'

```
17        driver.get("https://www.hollisterco.com/shop/wd");
18
19        System.out.println("Total links on entire page-->" + driver.findElements(By.tagName("a")).size());
```

Figure 12

Run, it will print the total number of links in entire page.

```
INFO: Detected dialect: OSS
Total links on entire page-->233
```

Figure 13

Now, how do we get the total number of links ONLY in the footer section of this website (containing 'About Us' and 'Help' columns, the 2 columns in the figure below)? It is not straightforward to inspect the footer containing these 2 columns. You have to keep moving the mouse inside the 'Elements' section, till you see only the section comprising these 2 columns gets highlighted. So below, when the mouse is kept on the highlighted on the <div tag, the focus goes to these 2 columns in the footer section. So this means that, we can now create our own custom css path.

Figure 14a

This footer (dotted section in the figure below) is represented by 'class' attribute & this 'class' attribute has 3 class names. So we can use dot to separate the 3 classes. Try to inspect **.footer__info-links.grid-col.grid-col-2** in the chropath. Notice that there is only '1 element matching' for this csspath

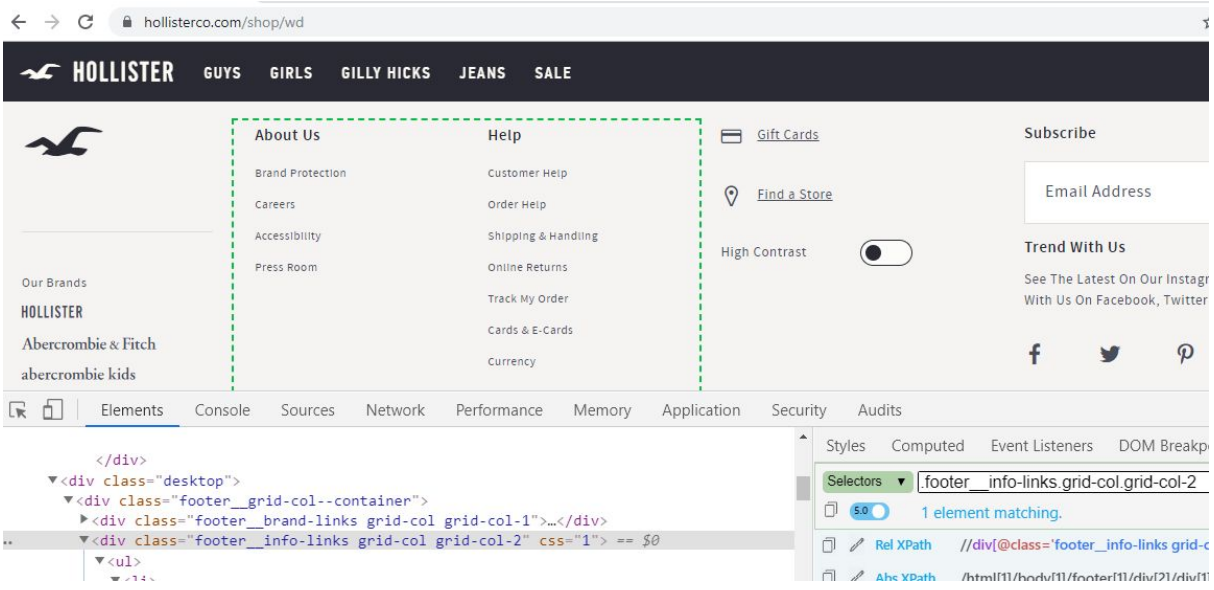

Figure 14b

So we will limit the webdriver scope to only this section, currently the global driver scope is on entire page. We will create mini driver only for this footer section.

So below code will help us find our footer element

```
driver.findElement(By.cssSelector(".footer__info-links.grid-col.grid-col-2"));
```

Figure 15

Now the 'findElement' method above returns a WebElement. Let's name this WebElement as 'footerdriver'

```
WebElement footerdriver = driver.findElement(By.cssSelector(".footer__info-links.grid-col.grid-col-2"));
```

Figure 16

Now we can find all the links within this 'footerdriver' & count them. This 'footerdriver' becomes our mini driver that focusses only on the footer section. It does not focus on the entire page unlike 'driver'

```
System.out.println("Total links on footer section-->" + footerdriver.findElements(By.tagName("a")).size());
```

Figure 17

The entire code is

```
15        driver.get("https://www.hollisterco.com/shop/wd");
16
17        System.out.println("Total links on entire page-->" + driver.findElements(By.tagName("a")).size());
18
19        WebElement footerdriver = driver.findElement(By.cssSelector(".footer__info-links.grid-col.grid-col-2"));
20
21        System.out.println("Total links on footer section-->" + footerdriver.findElements(By.tagName("a")).size());
```

Figure 18

Run, notice that there are 11 links in the footer section

```
Total links on entire page-->233
Total links on footer section-->11
```

Figure 19

The count is correct, as you can see below, there are 11 links in the footer section

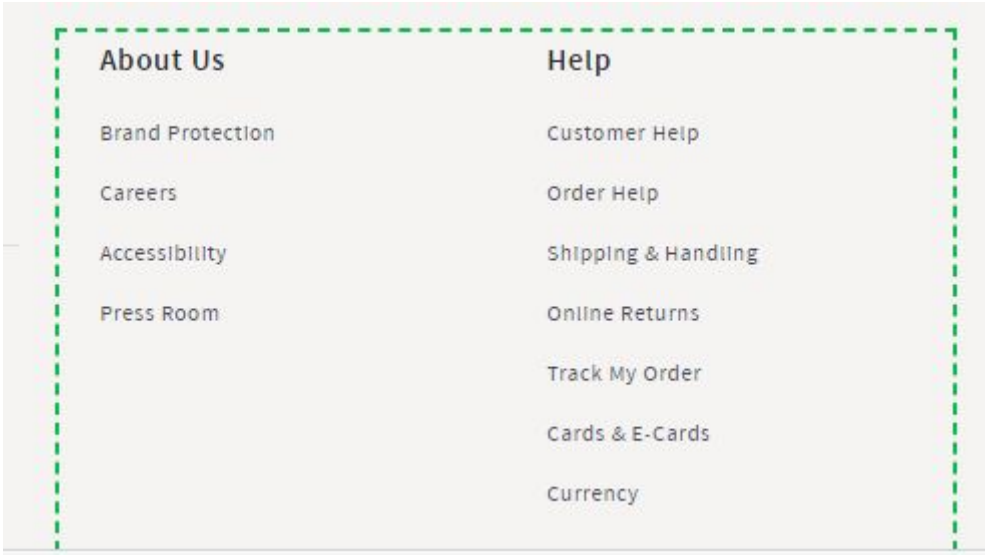

Figure 20

If for some reason, selenium throws an error that it cannot find the footer element OR if the footer hidden (since it is at the bottom of the page), than you can use 'Actions' class to scroll to the bottom of the page using below syntax:

```
Actions actions = new Actions(driver);
actions.sendKeys(Keys.CONTROL.END).perform();
```

Similarly we can further narrow down to count the links in only first column. If we hover over the mouse over the <ul tag seen in the red box (in the figure below), you will notice that the links of 'About' column get highlighted. Now the parent of this <ul> tag is <ul> tag and the parent of this <ul> tag is a <div. tag comprising of 3 classes. So our custom csspath would become:
.footer__info-links.grid-col.grid-col-2 ul ul

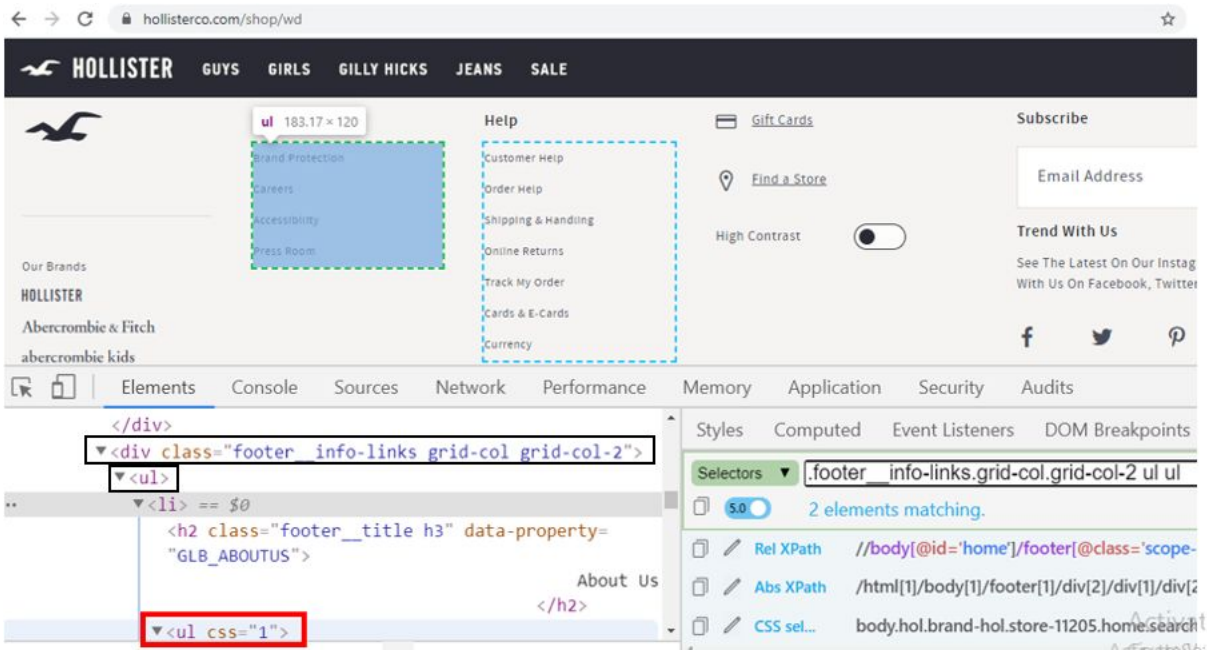

Figure 21

When we inspect this custom csspath, we find 2 elements matching. When we run our script, selenium will scan from left to right & hence will find the links of 'About' column. So let us create our mini driver 'columndriver' reference pointing to only first column (About Us).

```
8  public class Links {
9
10     public static void main(String[] args) {
11         System.setProperty("webdriver.chrome.driver", "C:\\Users\\DELL\\Desktop\\TRAINING\\Software\\chromedriver.exe");
12
13         WebDriver driver = null;
14
15         driver = new ChromeDriver();
16
17         driver.get("https://www.hollisterco.com/shop/wd");
18
19         System.out.println("Total links on entire page-->" + driver.findElements(By.tagName("a")).size());
20
21         WebElement footerdriver = driver.findElement(By.cssSelector(".footer__info-links.grid-col.grid-col-2"));
22         System.out.println("Total links on footer section-->" + footerdriver.findElements(By.tagName("a")).size());
23
24         WebElement columndriver = driver.findElement(By.cssSelector(".footer__info-links.grid-col.grid-col-2 ul ul"));
25         System.out.println("Total links in first column-->" + columndriver.findElements(By.tagName("a")).size());
26     }
```

Figure 22

When we run script, we see the total number of links in first column as 4

```
Total links on entire page-->220
Total links on footer section-->11
Total links in first column-->4
```

Figure 23

The count is correct, as you can see below, there are 4 links in the first column

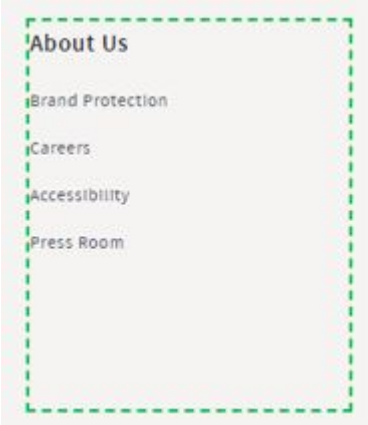

Figure 24

As an exercise, try to count the number of links in the highlighted section seen below

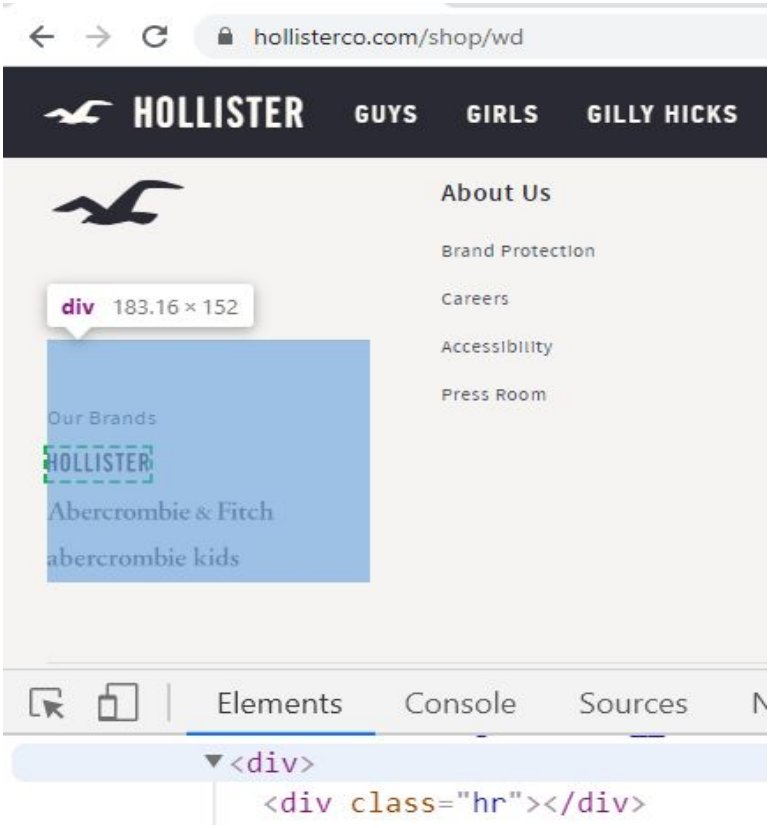

Figure 25

Next, try to count the number of links in the highlighted section seen below

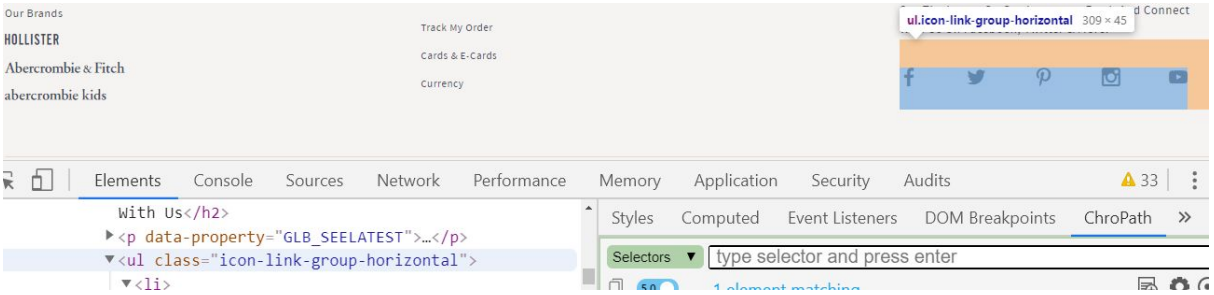

Figure 26

Next, try to count the number of links in the highlighted section seen below

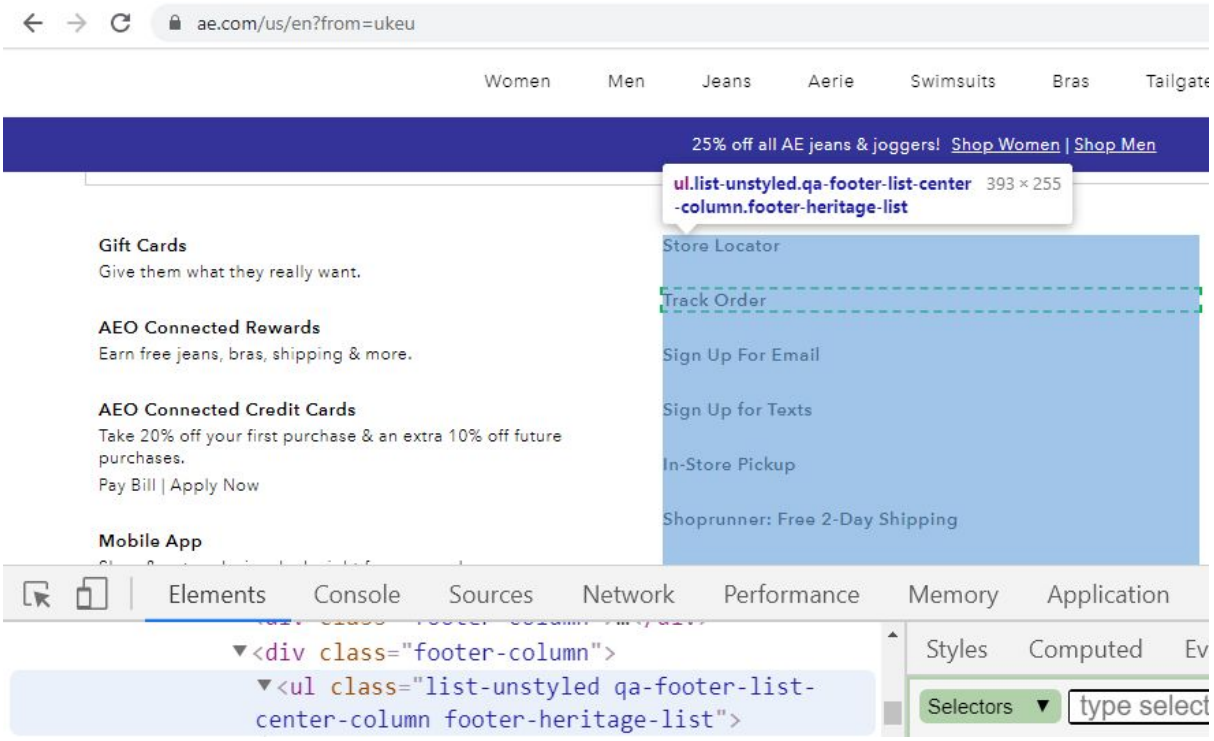

Figure 27

Thank you for reading!