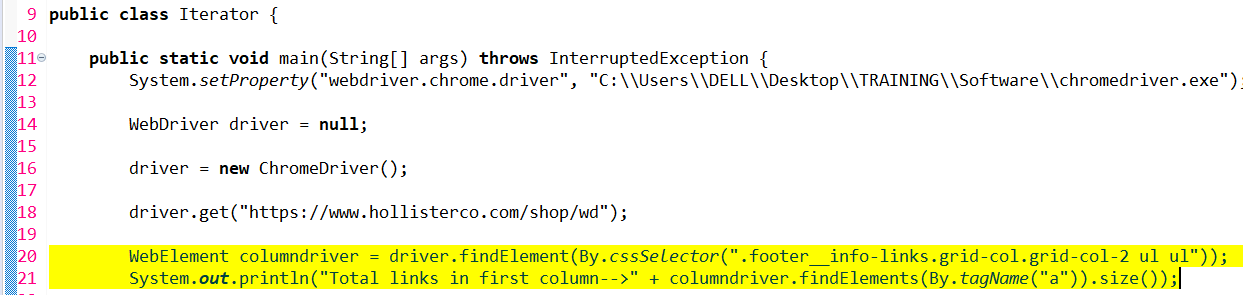
**Selenium Java - Tutorial 11- Window Tabs & Iterator**

This is the next tutorial in selenium-java series. Please go through the previous tutorials before you start this one. In the last tutorial, we learned how to handle checkboxes, radio buttons, links. In this tutorial we will see how to handle window tabs using iterator!

**What you will Learn:**1. Open links in different window tabs  
2. Handling window tabs using iterator  
3. Tabs Exercise 1  
4. Tabs Exercise 2

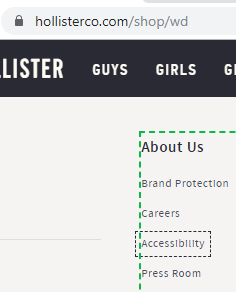
**Open links in different window tabs**

Create a new class & just copy the code (refer previous tutorial) that counts links in the first column

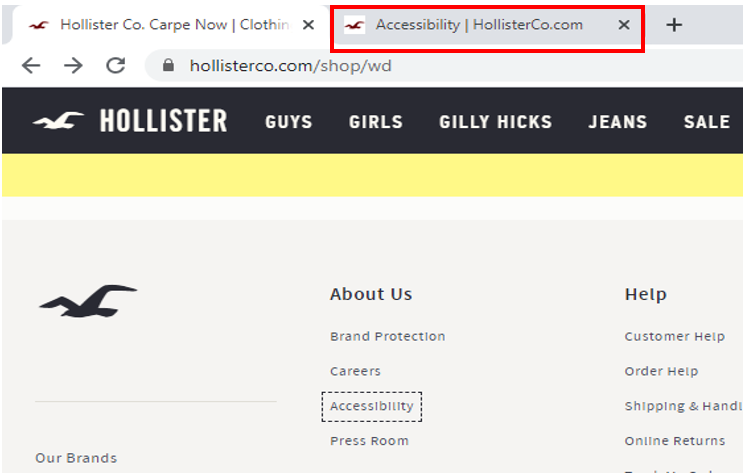
  
***Figure 1***

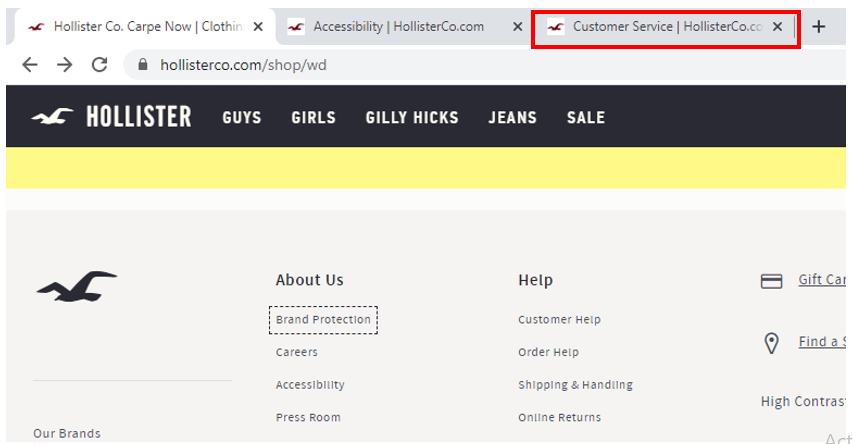
When we run this script, the total number of links in first column gets printed (4)

  
***Figure 2***

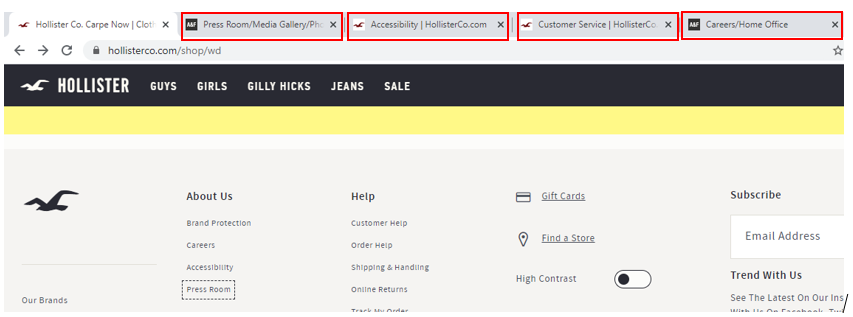
The 4 links that we see over here are: Brand Protection, Careers, Accessibility, Press room. Now our requirement is that, we have to click each of these 4 links, one by one. When we click these links, the links should open in a new window tab. They should not open in the same browser window.  
  
***Figure 3***

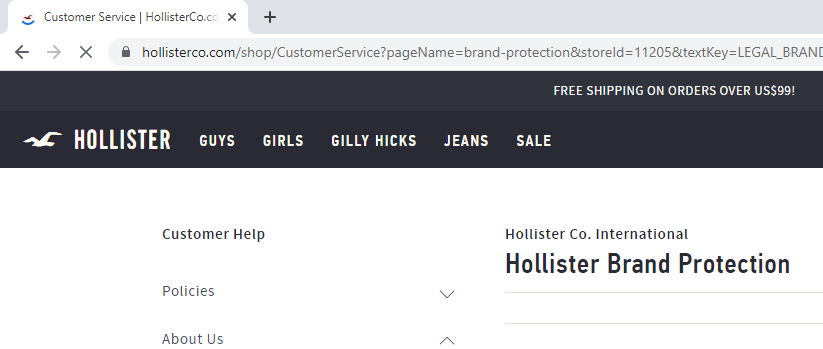
To simulate this requirement manually, move the mouse to any link (as seen above), press the control button and hit enter. Notice below that a new window tab opens having the title ‘Accessibility’

  
***Figure 4***

Similarly move the mouse to ‘Brand Protection’ link (as seen below), press the control button and hit enter. Notice that a new window tab opens having the title ‘Customer Service’.   
  
***Figure 5***

Repeat the same simulation for other 2 links as well. So we see 4 different window tabs open for each of the 4 links.

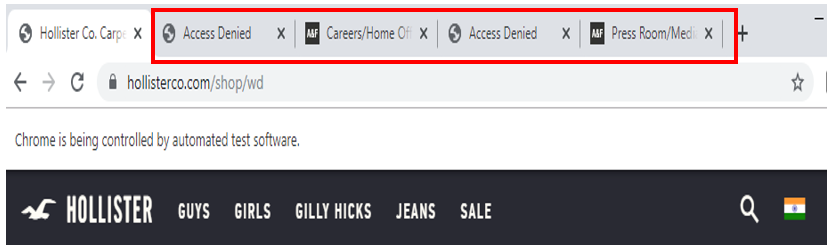
  
***Figure 6***

If you do not use the control button, than the link will open in the same page, like seen below. We do not want the link to open in the same page  
  
***Figure 7***

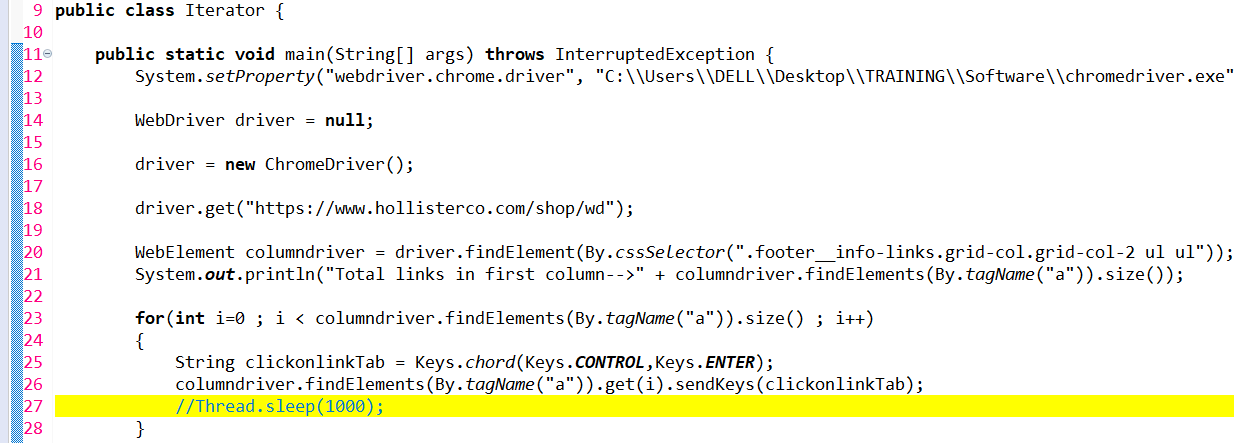
Now let us see how to simulate the same action using selenium script. The same action can be simulated using the 2 methods of ‘Keys’ class: CONTROL and ENTER, see line 25. So we have created a ‘for’ loop. This will click one link at a time. Note that the index ‘i’ of first link would start from 0. The logic of ‘for’ loop is self-explanatory. In line#23, we are initializing the variable/index ‘i’ starting from 0. We are than saying that this index should be less than size (4 in our case). Thus the index should be less than 4 (the range would be 0, 1, 2, and 3). We finally increment the index by 1 using i++ after each iteration. Line#25 captures the 2 keys in a string variable. We than pass this string variable in the sendkeys function in line#26. So in line#26, we are actually doing the click operation. Line#27 introduces a delay of 1 sec before the script clicks the next link.

  
***Figure 8***

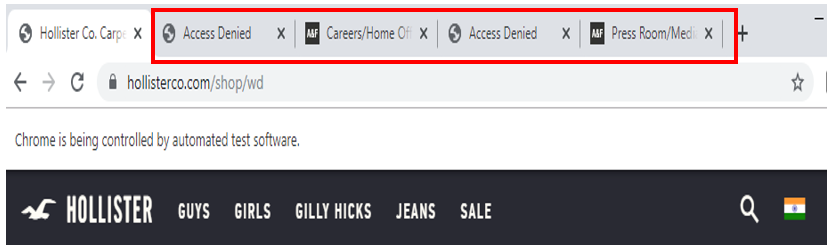
So when we run this script, the 4 links are clicked one by one and 4 new tabs open for each of the 4 links

  
***Figure 9***

Now let us see what happens if we comment the 1 second wait time

**  
*Figure 10***

When you run the script, you will notice that the 4 windows open very fast.

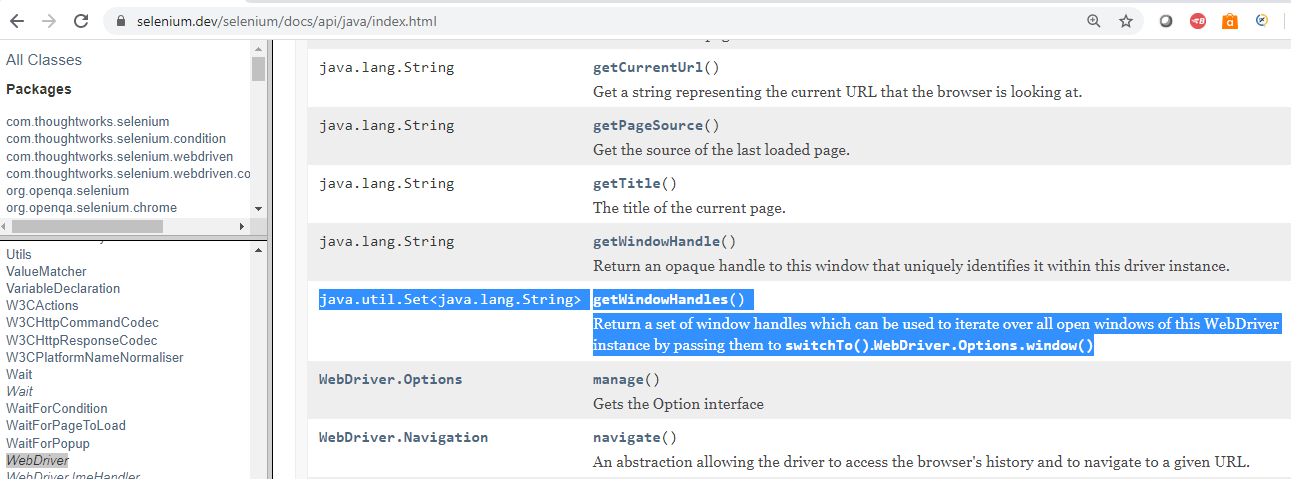
 ***Figure 11***

Let us uncomment back the wait statement

**Handling window tabs using iterator**

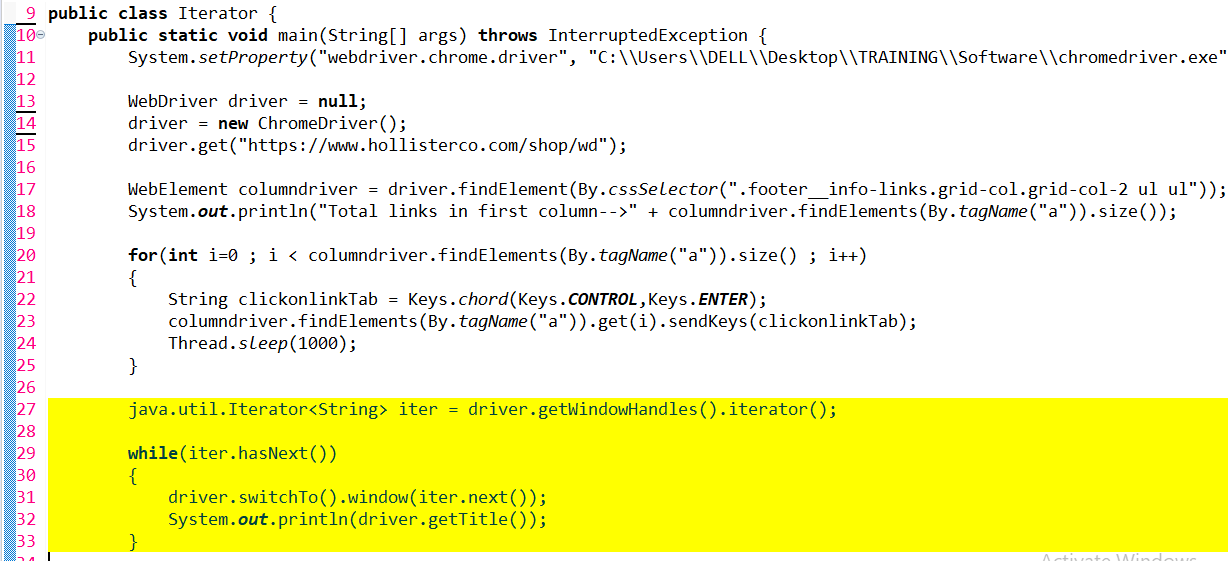
In continuous with previous exercise, our next goal is get the title of each of the 4 pages that the script opens. Now, each separate window that opens, has a unique window handle.

If you go to the selenium-java documentation and click *WebDriver* from the left panel, you would see ‘getWindowHandles()’ method listed, see below. This method returns a set of window handles.

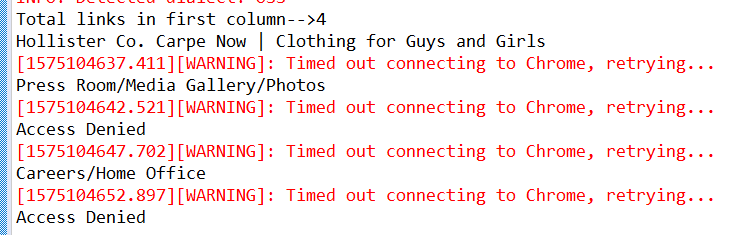
  
***Figure 12***  
  
Let us see how to use this method. To handle all windows that are opened by web driver, we can use "driver.getWindowHandles()".Once we have the window handles, we can easily switch from one window to another in a web application. This method returns Iterator<String>, see line 27

iter.hasNext() tells us whether next index is present or not, line 29  
iter.next() actually moves to next window using switchTo() method, line 31

In line 32 we are fetching the title of the page

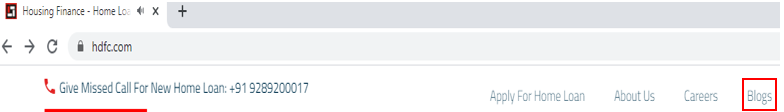
  
***Figure 13***

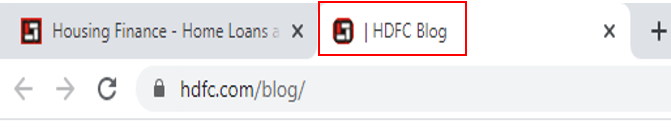
Run the script, notice that the title of each window (including the parent window) gets printed

  
***Figure 14***

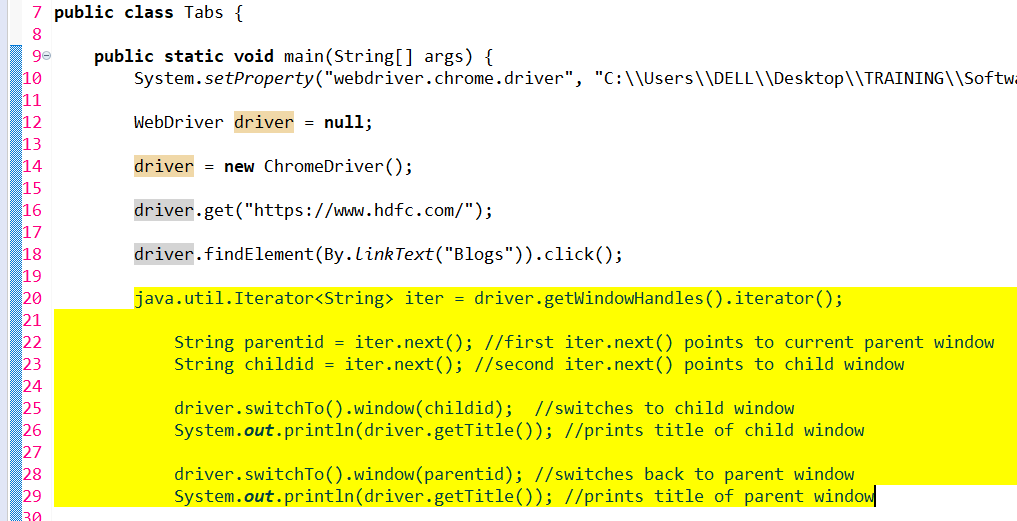
**Tabs Exercise 1**

Let us see another example. Navigate to <https://www.hdfc.com/>   
You can see the link ‘Blogs’

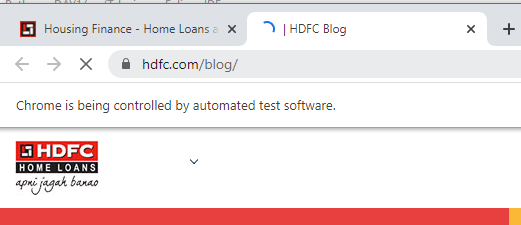
  
***Figure 15***

When we click this link, we see that a new window tab opens ‘HDFC Blog’  
  
***Figure 16***

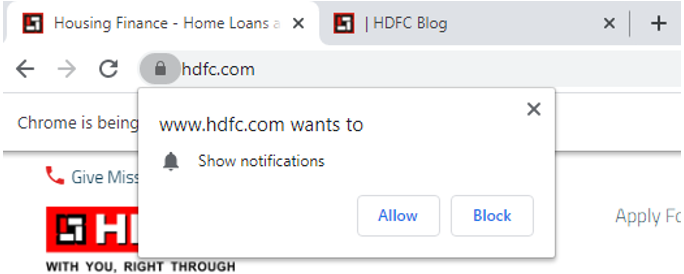
So just like we had done previously, we will perform the below steps to print the titles of parent and child windows:  
🡪we will get the window handles  
🡪we will use the iter.next(), the first iter.next() would point to current parent window, store this id in string var   
🡪the next iter.next would point to first child window, store this id in string var   
🡪we than switch to child window & print the page title   
🡪we than switch to parent window & print the page title

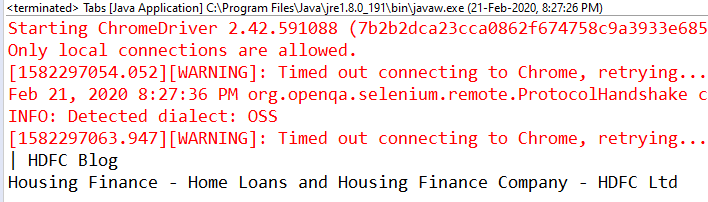
  
***Figure 17***

Run the script, notice that ‘HDFC Blog’ child window tab opens and the control goes to this window. This is due to line#25 in above figure where we are switching to child window

  
***Figure 18***

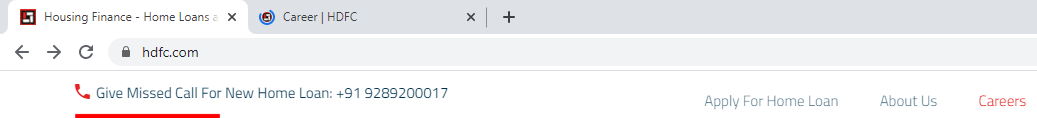
The control than comes back to parent window, see below. This is due to line#28 in above figure where we are switching back to parent window

  
***Figure 19***

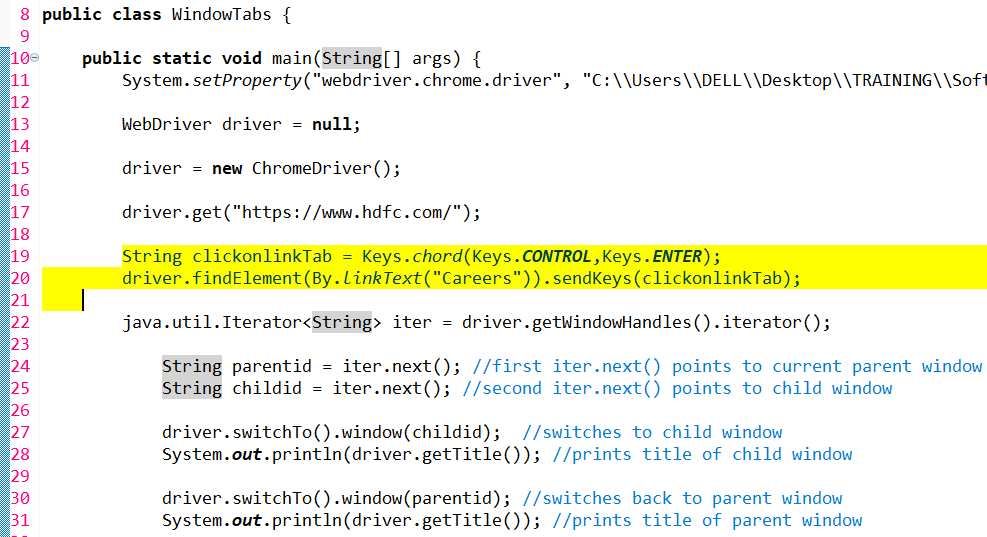
The console first prints the title of child window & than the title of parent window, see below  
   
***Figure 20***

**Tabs Exercise 2**

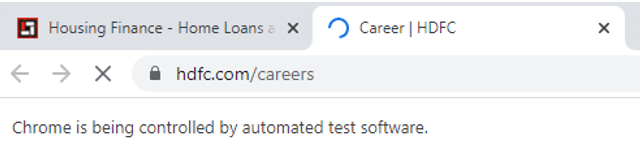
The behaviour of ‘Careers’ link is different. This links opens in a new tab only if you click ‘Ctrl + Enter’ key

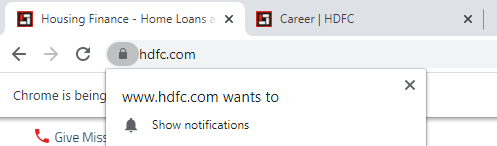
  
***Figure 21***

So we can simulate this action using CONTROL and ENTER keys like we have seen before

  
***Figure 22***

Rest of the steps would be same.

When you run the script, the ‘Careers’ link opens in a new child window, the control goes to this child window  
  
***Figure 23***

The control than switches to parent window  
  
***Figure 24***

The console prints the titles of both the pages

  
***Figure 25***

So this is how we deal with window tabs. Thank you for reading!