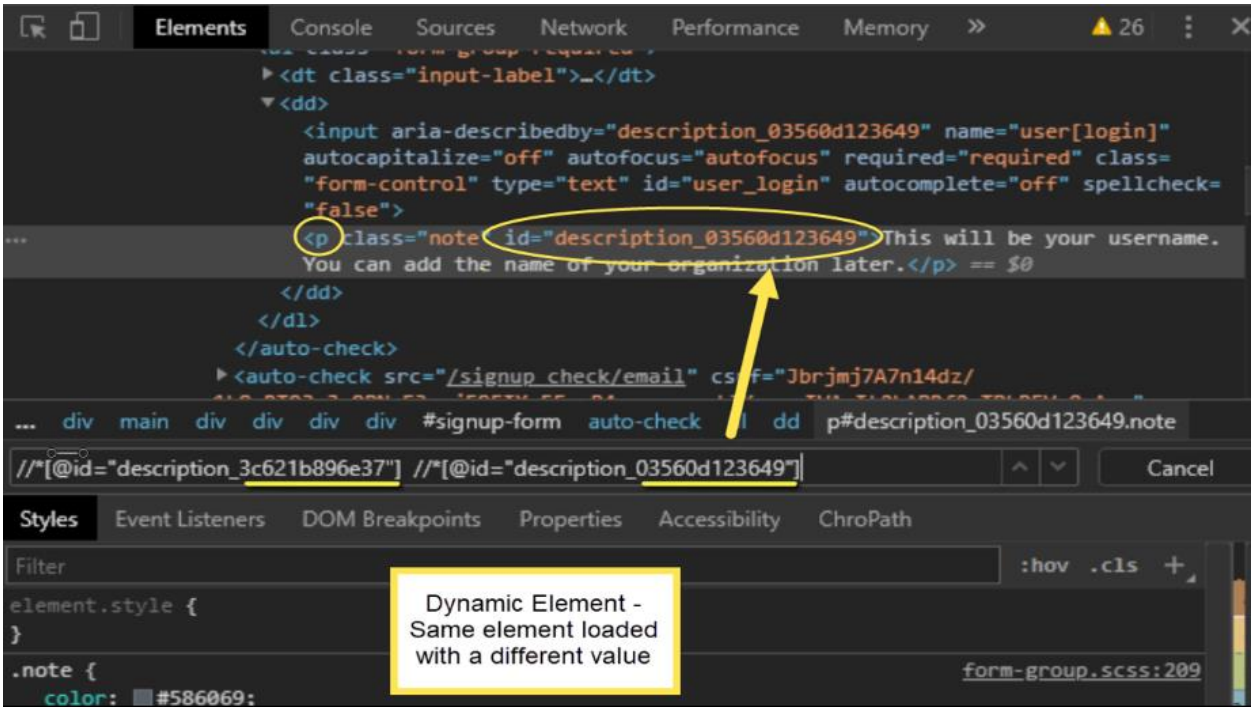
Rex Jones II

Selenium 4
Multiple Relative Locators

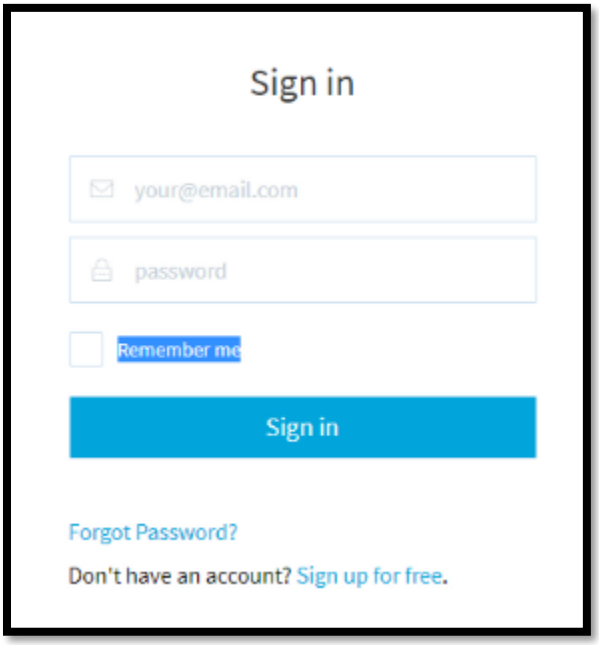
Selenium 4 Video Playlist
https://www.youtube.com/playlist?list=PLfp-cJ6BH8u_4AMzeLVizVfqn4SCywSTJ

Above-ToRightOf (Relative Locators)

In this session, we will look at the benefit of using more than 1 Relative Locator. I mentioned in the previous session that Relative Locators are good for Dynamic Elements. Here's an example of a dynamic element that I covered in my xpath video 87 talking about "Why We Should Never Copy XPath Values". There are 2 id values. On the left, the value ends with 37 and on the right, the value ends with 49. In this screenshot, the arrow is pointing to the most recent id value that ends with 49. The value changed but the tagName remained at p for paragraph.



We can include Relative Locators to the other locators such as xpath and cssSelector for finding Dynamic values. For this session, let's use the same Sign In page as the previous session but this time get the text of Remember me which is above the Sign In button and to the right of the checkbox.



Let's inspect the checkbox and the id value is rememberMe. We also need the tagName of the Remember me verbiage. So, let's it Inspect and the tagName is span. We are going back to IntelliJ and I need to copy the element for the Sign In button and paste in our new Test Script. We also need to go ahead and get the value of the Remember Me checkbox so I'm going to write WebElement rememberMeCheckbox = driver.findElement(By.id("rememberMe"));

Now, let's find the Remember Me text by writing WebElement rememberMeText = driver.findElement(withTagName("span"))
The verbiage is located .above(signInButton));

Print the verbiage sout("Text = " + rememberMeText.getText());

```
@Test
public void testMultipleRelativeLocators_AboveToRightOf () {
  driver.get("https://app.testproject.io/");
  WebElement signInButton = driver.findElement(By.id("tp-sign-in"));
  WebElement rememberMeCheckbox = driver.findElement(By.id("rememberMe"));
  WebElement rememberMeText = driver.findElement(withTagName("span")
          .above(signInButton));
  System.out.println("Text = " + rememberMeText.getText());
}
```

Let's run. We see Text is blank. Why is it blank? We see is blank. Why is it blank? It is blank because the checkbox and text are both located above the Sign In button. Our Test Script found the checkbox.

Inspect the checkbox again and we see it also has a span tag just like the Remember me. The checkbox was selected for the Relative Locator because it's the first span tag.

Recall, the findElement method finds the first WebElement so that's why checkbox was selected. In this scenario, we need to use multiple Relative Locators because it's more accurate than only using 1 Relative Locator. Remove the semi-colon and add the dot. The text was located above the checkbox and .toRightOf(rememberMeCheckbox);.

```
@Test
public void testMultipleRelativeLocators_AboveToRightOf () {
  driver.get("https://app.testproject.io/");
  WebElement signInButton = driver.findElement(By.id("tp-sign-in"));
  WebElement rememberMeCheckbox = driver.findElement(By.id("rememberMe"));
  WebElement rememberMeText = driver.findElement(withTagName("span")
            .above(signInButton)
            .toRightOf(rememberMeCheckbox));
  System.out.println("Text = " + rememberMeText.getText());
}
```

Now. let's run again and see what happens. Now, text has a value equal to Remember Me and it PASSED. Next, let's find a list of WebElements.

ToRightOf (Relative Locator) / Find List Of Elements

Let's get the URL for our next Test Script. I'm going to copy https://addons.testproject.io and paste it to take us to the Addons page. We are going to get the names of each platform. If I inspect each icon, the tagName is img and the names are located in attribute alt. We see Platform Web, Platform Android, and the third icon is Platform iOS. They are located to the right of this textbox. Inspect the textbox and the id value is q.

To find a list of elements using a Relative Locator, we start by writing List then <WebElement> and the name will be allPlatforms = driver.findElements(). Find elements with an 's' find all of the elements (withTagName("img"))
to the right of text box .toRightOf(By.id("q")));

Here's the logic, we must loop through each image then print the name.

for (WebElement platform : allPlatforms) {)
  sout(platform.getAttribute("alt"));

```
@Test
public void testRelativeLocator_FindListOfWebElements () {
  driver.get("https://addons.testproject.io/");
  List<WebElement> allPlatforms = driver.findElements(withTagName("img")
          .toRightOf(By.id("q")));

  for (WebElement platform : allPlatforms) {
    System.out.println(platform.getAttribute( name: "alt"));
  }
```

Remember the names were located in the alt attribute so that's why I wrote getAttribute and not getText. I created a video 12 that explains the difference between getAttribute and getText if you want to check it out. Let's Run. Bingo, we see all of the platform names: Platform Web, Platform Android, Platform iOS and they PASSED. That's it for using more than 1 Relative Locator to find a WebElement and how to use a Relative Locator to find a list of WebElements. I have more videos on the way. You can like, subscribe and click the bell icon. Plus connect with me on LinkedIn, Twitter and Facebook.

Contact

- ✔ YouTube https://www.youtube.com/c/RexJonesII/videos
- ✔ Facebook http://facebook.com/JonesRexII
- ✔ Twitter https://twitter.com/RexJonesII
- ✔ GitHub https://github.com/RexJonesII/Free-Videos
- ✔ LinkedIn https://www.linkedin.com/in/rexjones34/