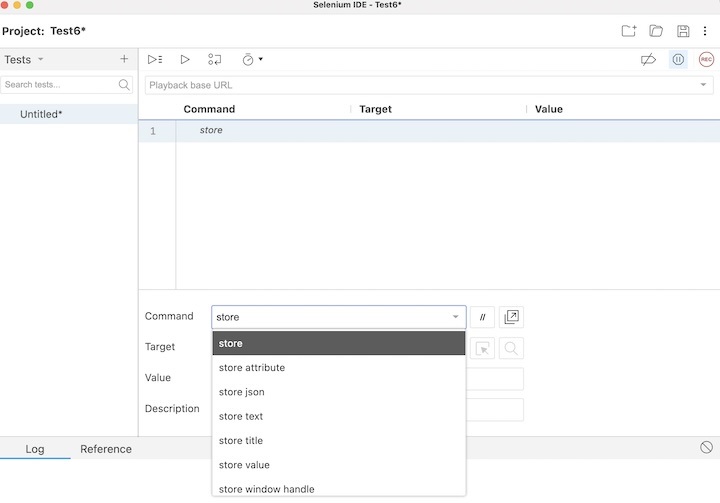
**Lab Exercise 8 - Selenium IDE – Accessors Commands**

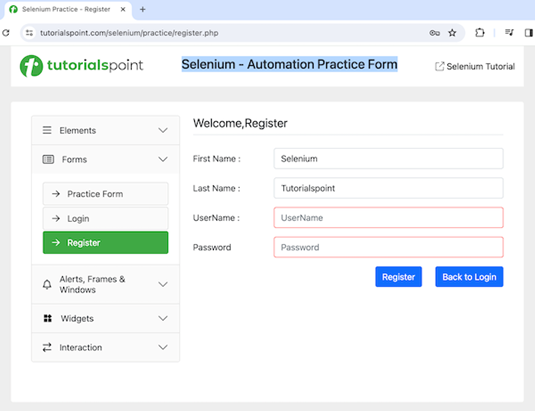
Selenium IDE consists of three kinds of commands known as the **Actions**, Accessors, and **Assertions**. An accessor command is used to check the application state and then store the outcome in a variable.

We can get hold of the Accesor commands in Selenium IDE, by entering the store within the **Command** field. Once entered, all the commands related to the store will be available for selection.



**Hands-On 1**

Let us take an example of the below page, where we would enter the text **Selenium** beside the **First Name:** label and **Tutorialspoint** beside **Last Name:** label. Also, we would obtain the text **Selenium - Automation Practice Form** using the Selenium IDE taking the help of the Accessors commands.

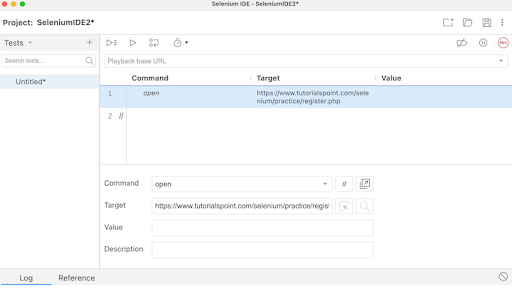


The steps to be followed are listed below −

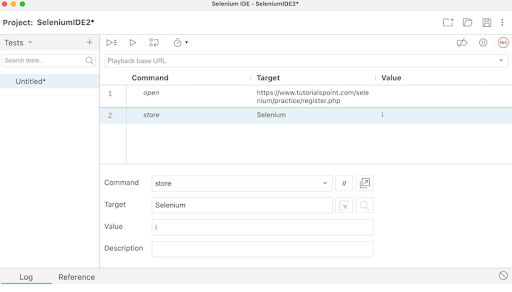
**Step 1** − Repeat the Steps 1 to 4 as described in the link **Selenium IDE Store Variables**.

**Step 2** − The workspace is now ready in the Selenium IDE.

**Step 3** − Enter open in the Command field, and [**Selenium Automation Practice Form**](https://www.tutorialspoint.com/selenium/practice/register.php) in the Target field, to launch the application.



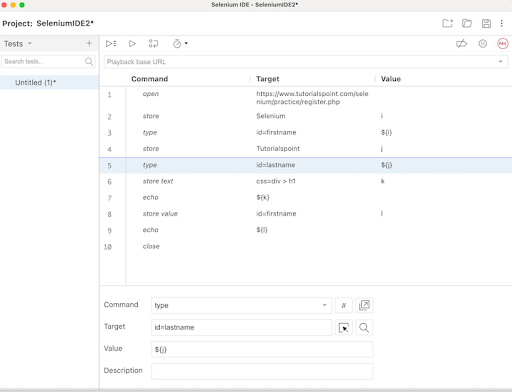
**Step 4** − Enter store in the Command field, Selenium in the Target field, and i in the Value field. Please note, Selenium is the value we would like to enter using the variable i.



**Step 5** − Enter type in the Command field, id=firstname in the Target field, and ${i} in the Value field. Please note, id=firstname is the locator value of the input box and ${i} is used to refer to the variable value we would enter using the variable i.

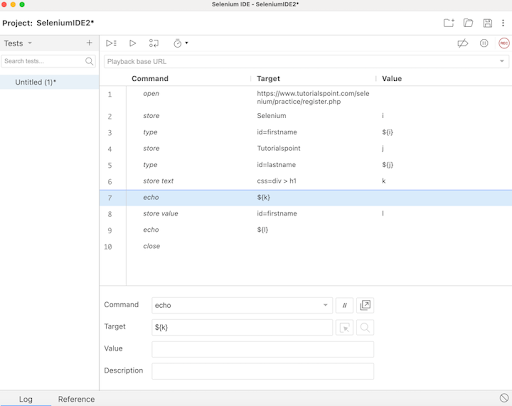
**Step 6** − Enter store in the Command field, Tutorialspoint in the Target field, and j in the Value field. Please note, Tutorialspoint is a value we would like to enter using the variable j.

**Step 7** − Enter type in the Command field, id=lastname in the Target field, and ${j} in the Value field. Please note, id=lastname is the locator value of the input box and ${j} is used to refer to the variable value we would enter using the variable j.



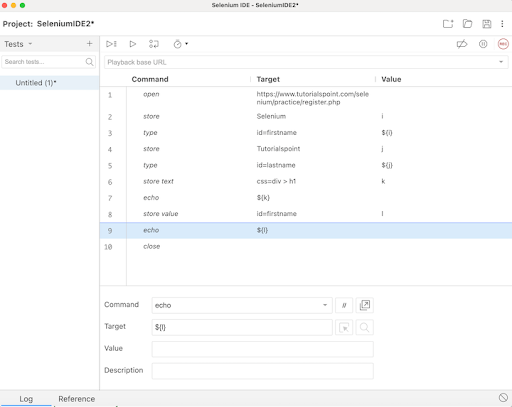
**Step 8** − Enter store Text in the Command field, css=div > h1 in the Target field, and k in the Value field. Please note, css=div > h1 is the locator value of the text that we would obtain and k is used to refer to the variable value where we would store the text obtained.

**Step 9** − Enter echo in the Command field, ${k} in the Target field. Please note, this would print the text we accessed in Step10 on the web page under the Log in Selenium IDE.

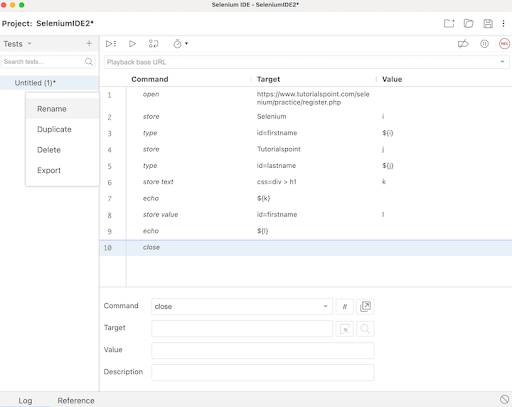


**Step 10** − Enter store value in the Command field, id=firstname in the Target field, and l in the Value field. Please note, id=firstname is the locator value of the input box and l is used to refer to the variable we would use to obtain the value entered in the input box.

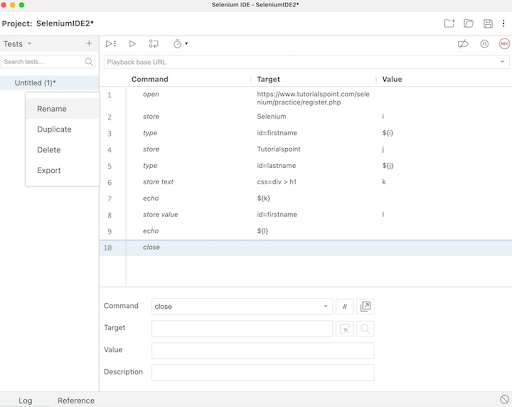
**Step 11** − Enter echo in the Command field, ${l} in the Target field. Please note, this would print the value entered in Step7 on the web page under the Log in Selenium IDE.



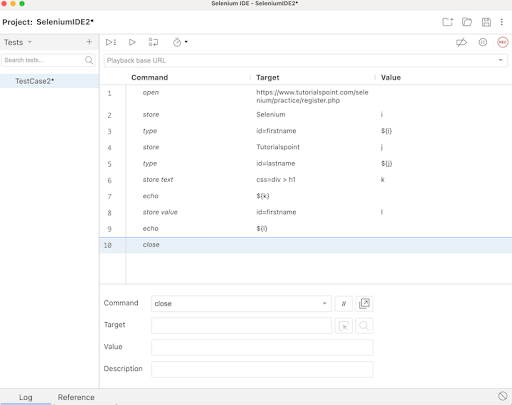
**Step 12** − Enter close in the Command field to close the browser window.



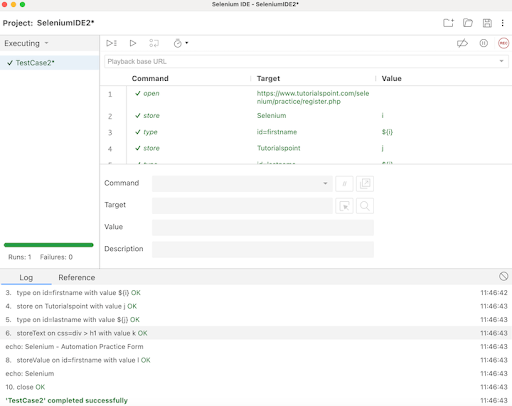
**Step 13** − Click on the three dots appearing at the left pane, then select the Rename option.



**Step 14** − Enter a name, say TestCase2 under the Rename Test case field, then click on Rename. The entered name would appear on the left of Selenium IDE.



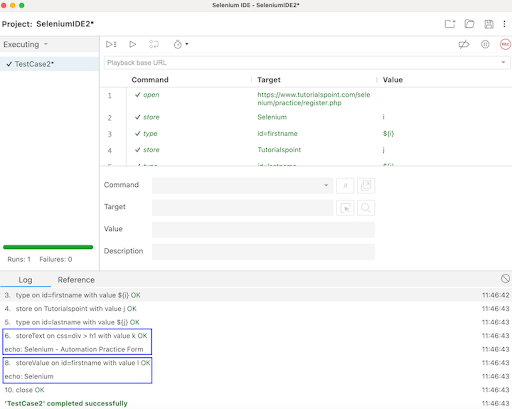
**Step 15** − Click on the Run all tests from the top, and wait for the test execution to complete. In our example, we would see Runs: 1, Failures: 0, and a green bar denoting the test ran successfully without any failures. Also, the message TestCase2 completed successfully appeared under the Log.



In the example above, we obtained the text **Selenium - Automation Practice Form** which appeared on the web page, and also retrieved the value entered in the input box beside the **First Name:** label - **Selenium** under the Log. Besides, we got a green tick beside the test case name - **TestCase2** at the top, signifying a passed test.

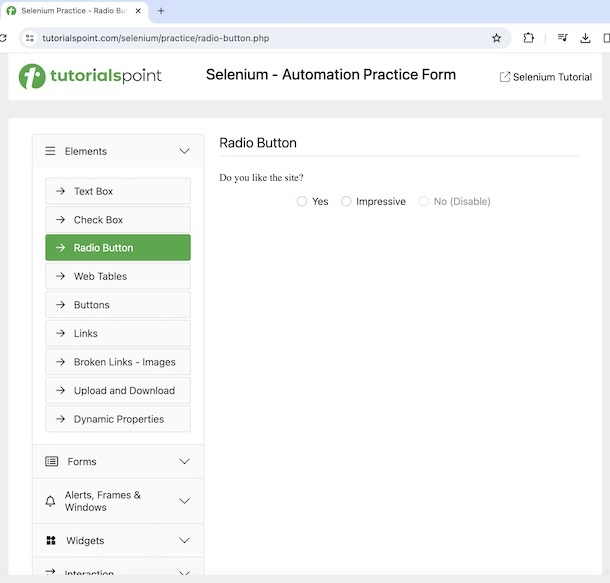
Thus, in Selenium IDE, we would get the information about the test results both at the test step level and only if all the steps would be executed successfully, we would get a PASS at the test case level as well.

Moreover, we would infer that variables used in one step can be accessed further in other steps using the syntax: ${<variable name>}. In the Steps 6, 7, 8, 9, 10, 11, 12, and 13, we followed this approach in our test. Also, we had used store and store related commands like store Text, store Value, and so on to store the variables. The echo command was used to print those values under the Log tab.



Example 2

Let us take another example of the below page, where we would capture the page title Selenium Practice - Radio Button taking the help of the Accessors commands.



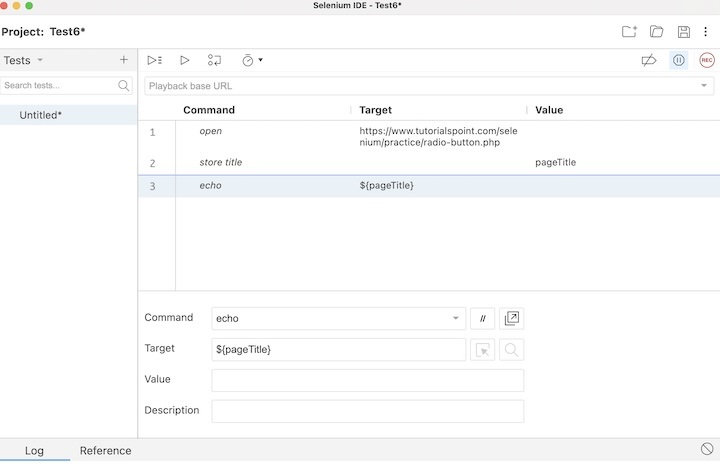
The steps to capture the browser title are listed below −

**Step 1** − Follow steps 1 to 4 from the previous example.

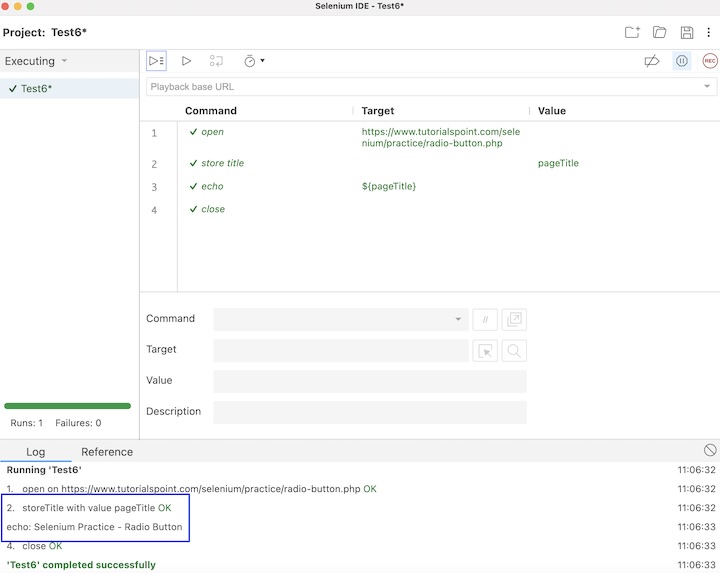
**Step 2** − Enter open in the Command field, and [**Selenium Automation Practice Form**](https://www.tutorialspoint.com/selenium/practice/radio-button.php) in the Target field, to launch the application.

**Step 3** − Enter store title in the Command field, and pageTitle in the Value field. Please note, pageTitle is variable to capture the title of the page launched.

**Step 4** − Enter echo in the Command field, ${pageTitle} in the Target field. Please note, this would print the text we obtained in Step3 under the Log in Selenium IDE.



**Step 5** − Follow steps 14 to 17 from the previous example.



In the example above, we obtained the page title **Selenium Practice - Radio Button** under the Log. Besides, we got a green tick beside the test case name - **Test6** at the top, signifying a passed test.

Moreover, we would infer that variables used in one step can be accessed further in other step using the syntax: ${<variable name>}. In the Steps 2, and 3, we followed this approach in our test along with the store title command to get the page. The echo command was used to print that value under the Log tab.

**Conclusion**

This concludes our comprehensive take on the tutorial on Selenium IDE Accessors Commands. Weve started with describing how to access Accessor commands in Selenium IDE, and walked through the different Accessor commands like store, store text, store title, and so on and illustrated how to use them along with Selenium. This equips you with in-depth knowledge of the Accessors Commands in Selenium IDE. It is wise to keep practicing what youve learned and exploring others relevant to Selenium to deepen your understanding and expand your horizons.