# Selenium IDE with Scripts & Commands

**Introduction**: Selenium commands are also known as “Selenese”.

* Selenese commands can have up to a maximum of two parameters: **target** and **value**.
* Parameters are not required all the time. It depends on how many the command will need.

**Types of Commands:**

1. **Actions:** These are commands that directly interact with page elements. The "click" command is an action because you directly interact with the element you are clicking at. The "type" command is also an action because you are putting values into a text box, and the text box shows them to you in return. There is a two-way interaction between you and the text box.
2. **Assessors:** They are commands that allow you to store values to a variable. The "storeTitle" command is an accessor because it only "reads" the page title and saves it in a variable. It does not interact with any element on the page.
3. **Assertions:** They are commands that verify if a certain condition is met. Three Types of Assertions:-

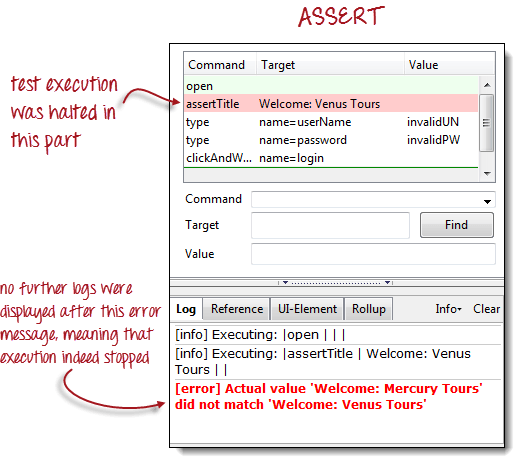
* **Assert**. When an "assert" command fails, the test is stopped immediately.
* **Verify**. When a "verify" command fails, Selenium IDE logs this failure and continues with the test execution.
* **WaitFor**. Before proceeding to the next command, "waitFor" commands will first wait for a certain condition to become true.

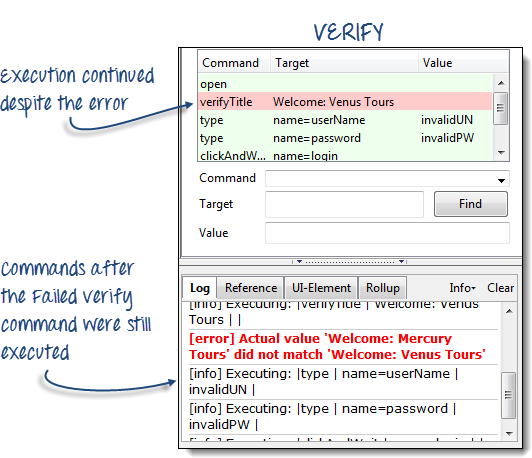
If the condition becomes true within the waiting period, the step passes.

If the condition does not become true, the step fails. Failure is logged, and test execution proceeds to the next command.

By default, the timeout value is set to 30 seconds. You can change this in the Selenium IDE Options dialog under the General tab.

**Assert vs. Verify**

[](https://www.guru99.com/images/assert.png)

[](https://www.guru99.com/images/verify.png)

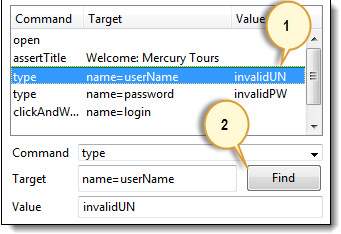
## Common Commands

|  |  |  |
| --- | --- | --- |
| **Command** | **Number of Parameters** | **Description** |
| Open | 0 - 2 | Opens a page using a URL. |
| click/clickAndWait | 1 | Clicks on a specified element. |
| type/typeKeys | 2 | Types a sequence of characters. |
| verifyTitle/assertTitle | 1 | Compares the actual page title with an expected value. |
| verifyTextPresent | 1 | Checks if a certain text is found within the page. |
| verifyElementPresent | 1 | Checks the presence of a certain element. |
| verifyTable | 2 | Compares the contents of a table with expected values. |
| waitForPageToLoad | 1 | Pauses execution until the page is loaded completely. |
| waitForElementPresent | 1 | Pauses execution until the specified element becomes present. |

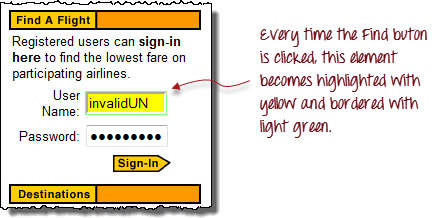
## Find Button

**The Find button in Selenium IDE is used to verify if what we had put in the Target text box is indeed the correct UI element.**

Let us use the Invalid\_login test case that we created in the previous sections. Click on any command with a Target entry, say, the third commands.



Click on the Find button. Notice that the User Name text box within the Mercury Tours page becomes highlighted for a second.

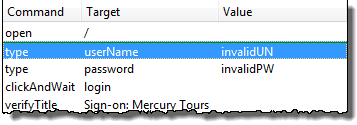
[](https://www.guru99.com/images/firefox_-_user_name_highlighted(1).png)

This indicates that Selenium IDE was able to detect and access the expected element correctly. If the Find button highlighted a different element or no element at all, then there must be something wrong with your script.

**Execute** **Command**

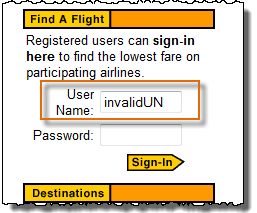
**This allows you to execute any single command without running the whole test case.** Just click on the line you wish to execute and then either click on "Actions > Execute this command" from the menu bar or simply press "X" on your keyboard.

**Step1.**Make sure that your browser is on the Mercury Tours homepage. Click on the command you wish to execute. In this example, click on the "type | userName | invalidUN" line.

[](https://www.guru99.com/images/execute1.png)

**Step2.**Press "X" on your keyboard.

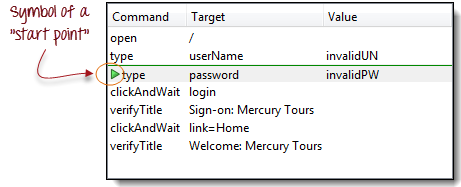
**Step3.**Observe that the text box for username becomes populated with the text "invalidUN"

[](https://www.guru99.com/images/execute2.png)

**Executing commands this way is highly dependent on the page that Firefox is currently displaying.** This means that if you try the example above with the Google homepage displayed instead of Mercury Tours', then your step will fail because there is no text box with a "userName" attribute within Google's homepage.

## Start point

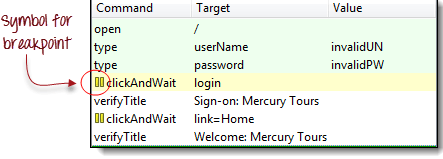
**A start point is an indicator that tells Selenium IDE which lines the execution will start. Its shortcut key is "S".**

[](https://www.guru99.com/images/startpoint.png)

**You can only have one start point in a single test script.** The start point is similar to Execute Command in such that they are dependent on the currently displayed page. The start point will fail if you are on the wrong page.

## Breakpoints

Breakpoints are indicators that tell Selenium IDE where to automatically pause the test. **The shortcut key is "B".**

[](https://www.guru99.com/images/BreakPoint.png)

The yellow highlight means that the current step is pending. This proves that Selenium IDE has paused execution on that step. **You can have multiple breakpoints in one test case.**

## Step

It allows you to execute succeeding commands one at a time after pausing the test case.

|  |  |
| --- | --- |
| [How to use Selenium IDE with Scripts & Commands (Assertions, Actions)](https://www.guru99.com/images/Stepa.png) | **Before clicking "Step."**  The test case pauses at the line "clickAndWait  | login". |
| [How to use Selenium IDE with Scripts & Commands (Assertions, Actions)](https://www.guru99.com/images/Stepb.png) | **After clicking "Step."**  The "clickAndWait | login" line is run and pauses to the next command (verifyTitle | Sign-on: Mercury Tours). |