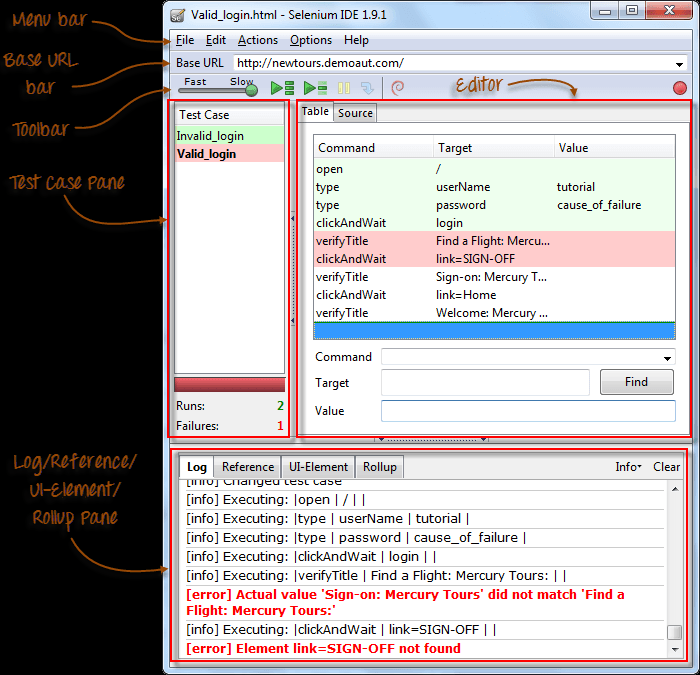
**Selenium IDE**, a Firefox add-on that you can only use in creating relatively simple test cases and test suites.

**Selenium Remote Control**, also known as Selenium 1, which is the first Selenium tool that allowed users to use programming languages in creating complex tests.

**WebDriver**, the newer breakthrough that allows your test scripts to communicate directly to the browser, thereby controlling it from the OS level.

**Selenium Grid** is also a tool that is used with Selenium RC to execute parallel tests across different browsers and operating systems.

**Features of Selenium IDE**

Tests are saved in HTML format. 

The most useful option is **"Export"** because it allows you to turn your Selenium IDE test cases into file formats that can run on Selenium Remote Control and WebDriver

"Insert New Command" and "Insert New Comment".

**The newly inserted command or comment will be placed on top of the currently selected line.** **Commands are colored black.**

**Comments are colored purple**

**Clipboard Format**

**The Clipboard Format allows you to copy a Selenese command from the editor and paste it as a code snippet.**

[**http://www.guru99.com/introduction-selenuim-ide.html**](http://www.guru99.com/introduction-selenuim-ide.html) **read from Locator builders**

**3 Types of Commands**

|  |  |
| --- | --- |
| **Actions** | These are commands that directly interact with page elements.  Example: 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. |
| **Accessors** | They are commands that allow you to store values to a variable.  Example: 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. |
| **Assertions** | They are commands that verify if a certain condition is met.  **3 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. |

### ****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. |

## Execute Command

**This allows you to execute any single command without running the whole test case**.

press "X" on your keyboard.

## Start point

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

## Breakpoints

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

**Web driver commands:**

1. **driver.get**("http://newtours.demoaut.com");
2. **driver.findElement**(By.*name*("userName")).**getTagName();**
3. System.***out***.println("current url\t"+**driver.getCurrentUrl());**
4. System.***out***.println("page source\t"+**driver.getPageSource());**
5. **driver.navigate().to*("https://www.google.co.in");***
6. **driver.navigate().back();**
7. **driver.navigate().forward();**
8. **driver.navigate().refresh();**