1. **What is Selenium?**

* Acceptance Testing tool for web-apps
* Tests run directly in browser
* Selenium can be deployed on Windows, Linux, and Macintosh.
* Implemented entirely using browser technologies -
* JavaScript
* DHTML
* Frames

1. **Selenium Components**

* Selenium IDE
* Selenium Core
* Selenium RC
* Selenium Grid

1. **Overview of Selenium IDE:**

* Test Case Pane
* Toolbar
* Menu Bar
* D.Log/Reference/UI-Element/Rollup Pane

**3.A Test Case Pane:**

Your script is displayed in the test case pane. It has two tabs, One for displaying the command (source) and their parameters in a readable “table” format.



**3.B Toolbar:**

The toolbar contains buttons for controlling the execution of your test cases, including a step feature.

**3.C Menu Bar:**

* File Menu: The File menu allows you to create, open and save test case  and test suite files.
* Edit Menu: The Edit menu allows copy, paste, delete, undo and select all operations for editing the commands in your test case.
* Options Menu: The Options menu allows the changing of settings. You can set the timeout value for certain commands, add user-defined user extensions to the base set of Selenium commands, and specify the format (language) used when saving your test cases.

1. **Introducing Selenium Commands**

The command set is often called selenese. Selenium commands come in three “flavors”.

**Actions, Accessory and Assertions.**

1. **Actions:**

User actions on application / Command the browser to do something.  Actions are commands that generally manipulate the state of the application.

* Click link- click / Clickandwait
* Selecting items

1. **Accessors:**

Accessors examine the state of the application and store the results in variables, e.g. "storeTitle".

1. **Assertions:**

For validating the application we are using Assertions

* For verifying the web pages
* For verifying the text
* For verifying alerts

Assertions can be used in 3 modes:

* assert
* verify
* waitFor

Example: "assertText","verifyText" and "waitForText".

1. **Commonly Used Selenium Commands**

These are probably the most commonly used commands for building test.

* **open** - opens a page using a URL.
* **click/clickAndWait** - performs a click operation, and optionally waits for a new page to load.
* **verifyTitle/assertTitle** - verifies an expected page title. verifyTextPresent- verifies expected text is somewhere on the page.
* **verifyElementPresent** -verifies an expected UI element, as defined by its HTML tag, is present on the page.
* **verifyText** - verifies expected text and it’s corresponding HTML tag are present on the page.
* **verifyTable** - verifies a table’s expected contents.
* **waitForPageToLoad** -pauses execution until an expected new page loads. Called automatically when clickAndWait is used.
* **waitForElementPresent** -pauses execution until an expected UI element, as defined by its HTML tag, is present on the page.

1. **Recording and Run settings**

When Selenium-IDE is first opened, the record button is ON by default. During recording, Selenium-IDE will automatically insert commands into your test case based on your actions.

* Remember Base URL MODE
* Record Absolute recording mode

**Running Test Cases**

**Run a Test Case** Click the Run button to run the currently displayed test case.

**Run a Test** Suite Click the Run All button to run all the test cases in the currently loaded test suite.

**Stop and Start** The Pause button can be used to stop the test case while it is running. The icon of this button then changes to indicate the Resume button. To continue click Resume.

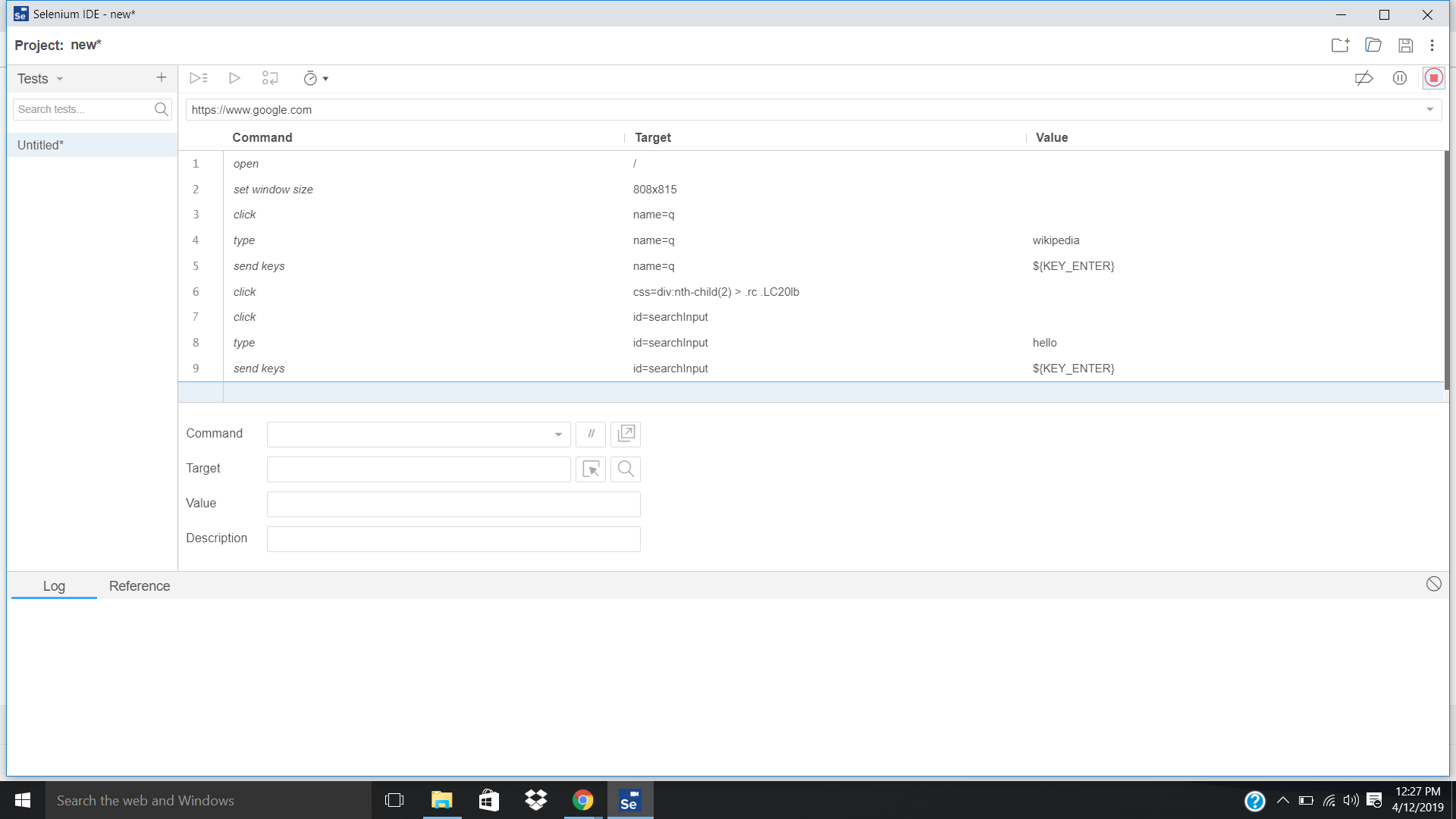
**Stop in the Middle** You can set a breakpoint in the test case to cause it to stop on a particular command. This is useful for debugging your test case. To set a breakpoint, select a command, right-click, and from the context menu select Toggle Breakpoint.

**Start from the Middle** You can tell the IDE to begin running from a specific command in the middle of the test case. This also is used for debugging. To set a startpoint, select a command, right-click, and from the context menu select Set/Clear Start Point.

**Run Any Single Command** Double-click any single command to run it by itself. This is useful when writing a single command. It lets you immediately test a command you are constructing, when you are not sure if it is correct. You can double-click it to see if it runs correctly. This is also available from the context menu.

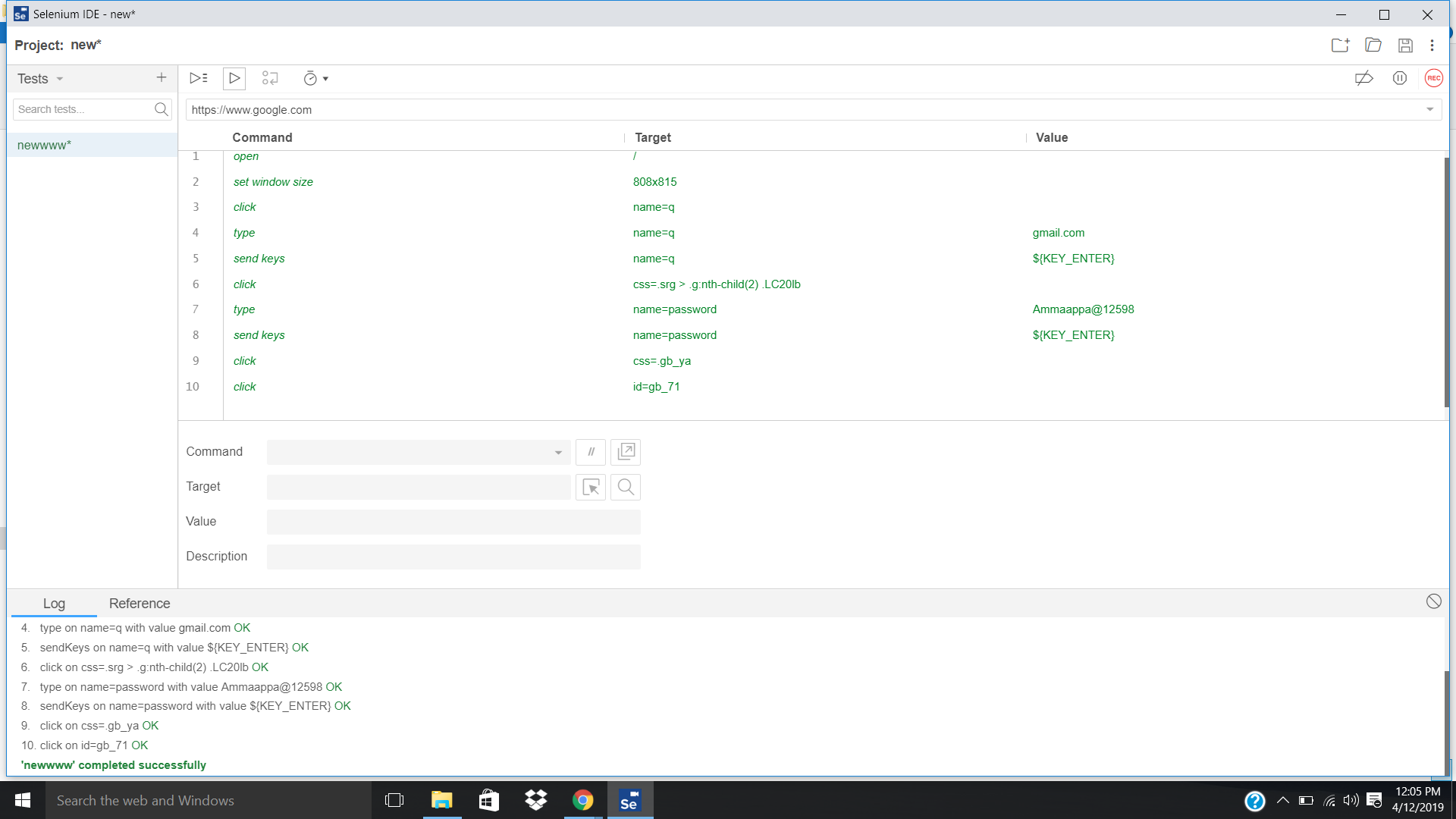
**SNAPSHOTS**

1. TEST CASES



1. TEST CASE EXECUTION

2.A SUCCESS



2.B FAILURE

