**To download and install Selenium**

* Click <https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/>
* Click **+ Add to Firefox**
* Click **Install**
* Click **Restart**  browser
* Upper R-corner toolbar, click **selenium** icon .
* Ctrl+Alt+S

Video-

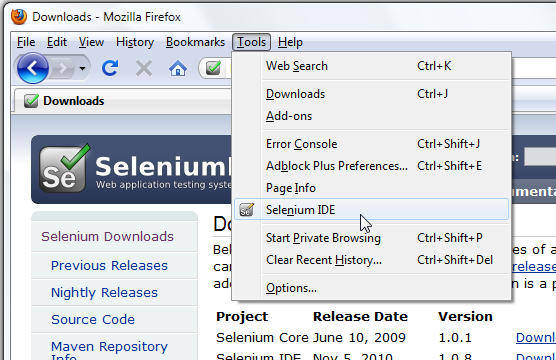
<http://qtpselenium.com/selenium-training/selenium-tutorial/selenium-training-java-part-1?gclid=Cj0KEQjwnv27BRCmuZqMg_Ddmt0BEiQAgeY1l6OTn6Rp2eb6vHUzLNqUIfZ_ceBZCiMFLWn754jf4pwaAojc8P8HAQ>  
<https://www.youtube.com/watch?v=GolxCQvGKdE>

<https://www.youtube.com/watch?v=3Yz9hzTueWM>

<http://qtpselenium.com/home/course/training/selenium-tutorial>

<http://www.softwaretestingclass.com/selenium-training-series-getting-started-with-selenium-ide/>

After installing Selenium IDE, it is listed under the Firefox Tools menu.

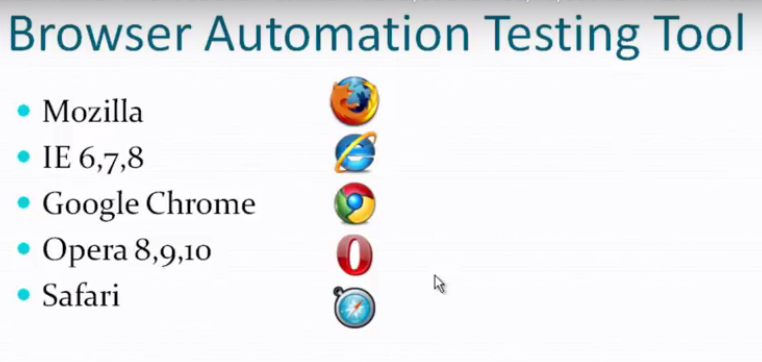


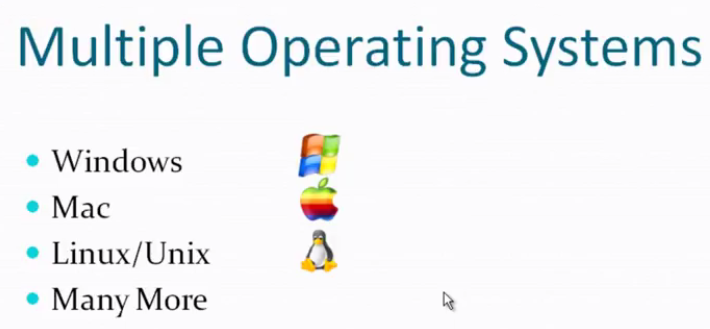
<http://www.seleniumhq.org/>

<http://www.seleniumhq.org/docs/02_selenium_ide.jsp>

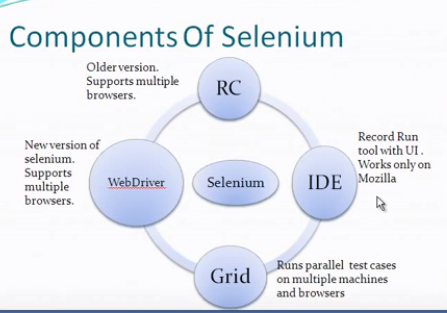
Selenium IDE <http://www.seleniumhq.org/projects/ide/>

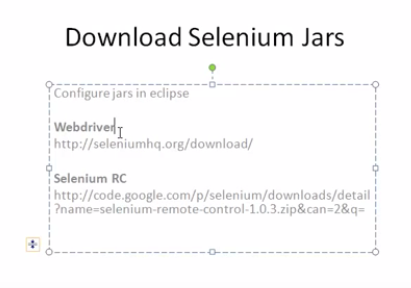
* The Selenium-IDE (Integrated Development Environment) is the tool you use to develop your Selenium test cases.
* It’s an easy-to-use Firefox plug-in and is generally the most efficient way to develop test cases.
* Using Firefox, first, download the IDE from the SeleniumHQ [downloads page](http://seleniumhq.org/download/)
* Select Install Now.
* Restart Firefox. After Firefox reboots you will find the Selenium-IDE listed under the Firefox Tools menu.
* To run the Selenium-IDE, simply select it from the Firefox Tools menu. It opens as follows with an empty script-editing window and a menu for loading, or creating new test cases.













<http://www.seleniumhq.org/docs/01_introducing_selenium.jsp>

Selenium IDE is used for test cases and build test scripts.

## IDE Features

**Menu Bar**

The File menu has options for Test Case and Test Suite (suite of Test Cases). Using these you can add a new Test Case, open a Test Case, save a Test Case, export --Test Case in a language of your choice. You can also open the recent Test Case. All these options are also available for Test Suite.

The Edit menu allows copy, paste, delete, undo, and select all operations for editing the commands in your test case.

The Actions menu allows to do execution that is in toolbar.

The Options menu allows the changing of settings. You can set options, 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.

The Help menu is the standard Firefox Help menu; only one item on this menu–UI-Element Documentation–pertains to Selenium-IDE.

**Open test case:**  Click **File** > **Open…** > Select **Test case** or **Test suite**.  
Selenium IDE allows many options for running your tests.  
- Run a single Test Case.  
- Run a Test Suite.  
- Pause test case execution and resume it later using Pause and resume buttons.  
- Stop test case execution at a required step using a breakpoint.  
- Start test case execution from a required step using Set Start point.  
- Run any single command step by step

### **Toolbar**

The toolbar contains buttons for controlling the execution of your test cases, including a step feature for debugging your test cases. The right-most button, the one with the red-dot, is the record button.

_images/chapt3_img06_IDE_features.png

speed control

Speed Control: controls how fast your test case runs.

run all

Run All: Runs the entire test suite when a test suite with multiple test cases is loaded.

run

Run: Runs the currently selected test. When only a single test is loaded this button and the Run All button have the same effect.

pauseresume

Pause/Resume: Allows stopping and re-starting of a running test case.

step

Step: Allows you to “step” through a test case by running it one command at a time. Use for debugging test cases.

testrunner

TestRunner Mode: Allows you to run the test case in a browser loaded with the Selenium-Core TestRunner. The TestRunner is not commonly used now and is likely to be deprecated. This button is for evaluating test cases for backwards compatibility with the TestRunner. Most users will probably not need this button.

rollup

Apply Rollup Rules: This advanced feature allows repetitive sequences of Selenium commands to be grouped into a single action. Detailed documentation on rollup rules can be found in the UI-Element Documentation on the Help menu.

record

Record: Records the user’s browser actions.

### **Test Case Pane/ Test Case**

A test case is represented by a table view.  
Each row in table represents a step and it contains 3 columns elements.   
All column/fields are case sensitive. Be careful when passing values.  
-Command – Specifies the action to be performed by step  
-Target – Specifies the web element for the action  
-Value – Specifies the input data or value for the step.

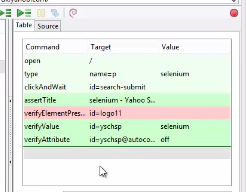
Your script is displayed in the test case pane. It has two tabs, one for displaying the command and their parameters in a readable “table” format.

Locating Strategies: <http://www.w3schools.com/Xpath/>  
- Identifier: Default. First element with matching id attribute is used. Ex: identifier=login  
- ID: This is more explicit and used when an element’s id attribute is known. Ex: id=login  
- Name: Locator locate the first element with matching name attribute. Ex: name=username   
- Hyperlink: Locating hyperlink in webpage uses the text link. Ex: link=SignIn  
- XPath: Used when suitable name or id attribute for the element you want to locate.   
 Ex: //html/body/div[2]/div/div[2]/div/fieldset/form/div[1]/input[1]  
(select field > R-click and copy XPath > paste in Target in Selenium > gives the absolute XPath)  
Ex: //input[@id=’user\_password’ and @placedholder=’Password’]  
 //input[contains(@id,’userID’)]  
 //input[starts\_with(@name,’user\_login’)]  
- CSS:   
- DOM:

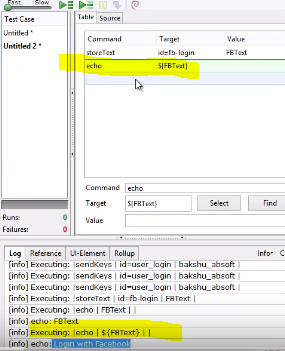
Recommended Locating Strategies: Id, Name, LinkText for links. XPath Attributes, XPath Relative using Immediate Parent, Target field to get locator.

**Selenium IDE Command:** <https://www.youtube.com/watch?v=3Yz9hzTueWM>  
- Input data to AUT: type, check, uncheck, select, sendKeys, click, submit.  
- Retreive data from AUT: store commands  
- Wait for AUT events: waitFor commands such as waitForText, waitForElementPresent, waitForAttribute

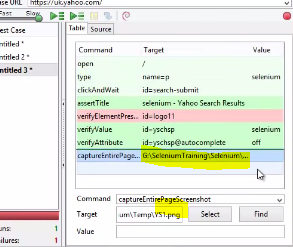
### - Assert/Verify AUT behavior or state: assert/verify commands



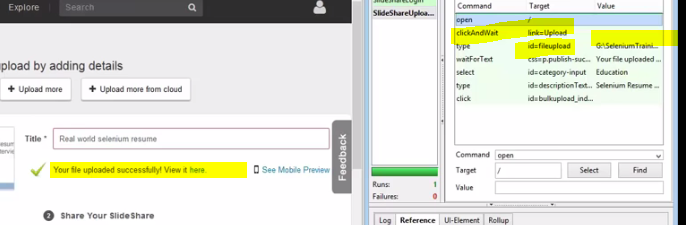
### **Print in the logs: echo command**

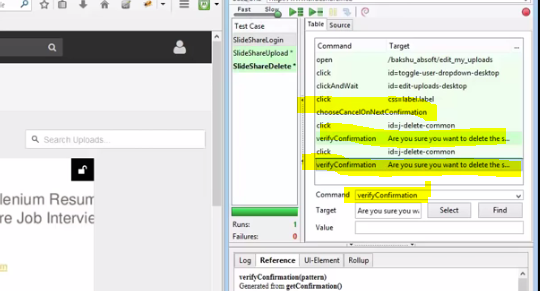


### **Capturing screenshots command:** captureEntirePageScreenshot and in Target field copy the filepath from file explorer where it will save and name the screenshot YS1.png extension.

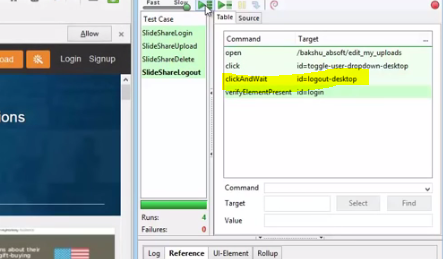


**Upload Button:** <https://www.youtube.com/watch?v=N06S26yo8jg>



**Cancel/Delete Button**

**Logout Button**



### 

### **Log/Reference/UI-Element/Rollup Pane**

The bottom pane is used for four different functions–Log, Reference, UI-Element, and Rollup–depending on which tab is selected.

#### **Log**

When you run your test case, error messages and information messages showing the progress are displayed in this pane automatically, even if you do not first select the Log tab. These messages are often useful for test case debugging. Notice the Clear button for clearing the Log. Also notice the Info button is a drop-down allowing selection of different levels of information to log.  
**Reference**

The Reference tab is the default selection whenever you are entering or modifying [Selenese](http://www.seleniumhq.org/docs/02_selenium_ide.jsp#selenese) commands and parameters in Table mode. In Table mode, the Reference pane will display documentation on the current command. When entering or modifying commands, whether from Table or Source mode, it is critically important to ensure that the parameters specified in the Target and Value fields match those specified in the parameter list in the Reference pane. The number of parameters provided must match the number specified, the order of parameters provided must match the order specified, and the type of parameters provided must match the type specified. If there is a mismatch in any of these three areas, the command will not run correctly.  
**UI-Element and Rollup**

Detailed information on these two panes (which cover advanced features) can be found in the UI-Element Documentation on the Help menu of Selenium-IDE.

## Building Test Cases

There are three primary methods for developing test cases. Frequently, a test developer will require all three techniques.

### **Recording**

Many first-time users begin by recording a test case from their interactions with a website. When Selenium-IDE is first opened, the record button is ON by default. If you do not want Selenium-IDE to begin recording automatically you can turn this off by going under Options > Options... and deselecting “Start recording immediately on open.”

During recording, Selenium-IDE will automatically insert commands into your test case based on your actions. Typically, this will include:

* clicking a link - *click* or *clickAndWait* commands
* entering values - *type* command
* selecting options from a drop-down listbox - *select* command
* clicking checkboxes or radio buttons - *click* command

Here are some “gotchas” to be aware of:

* The *type* command may require clicking on some other area of the web page for it to record.
* Following a link usually records a *click* command. You will often need to change this to *clickAndWait* to ensure your test case pauses until the new page is completely loaded. Otherwise, your test case will continue running commands before the page has loaded all its UI elements. This will cause unexpected test case failures.

2. **Adding Verifications and Asserts With the Context Menu**

Your test cases will also need to check the properties of a web-page. This requires *assert* and *verify* commands. We won’t describe the specifics of these commands here; that is in the chapter on [Selenium Commands – “Selenese”](http://www.seleniumhq.org/docs/02_selenium_ide.jsp#selenium-commands-selenese). Here we’ll simply describe how to add them to your test case.

With Selenium-IDE recording, go to the browser displaying your test application and right click anywhere on the page. You will see a context menu showing *verify* and/or *assert*commands.

3. **Editing**

#### **Insert Command**

##### **Table View**

Select the point in your test case where you want to insert the command. To do this, in the Test Case Pane, left-click on the line where you want to insert a new command. Right-click and select Insert Command; the IDE will add a blank line just ahead of the line you selected. Now use the command editing text fields to enter your new command and its parameters.

##### **Source View**

Select the point in your test case where you want to insert the command. To do this, in the Test Case Pane, left-click between the commands where you want to insert a new command, and enter the HTML tags needed to create a 3-column row containing the Command, first parameter (if one is required by the Command), and second parameter (again, if one is required to locate an element) and third parameter(again, if one is required to have a value). Example:

<tr>  
 <td>Command</td>  
 <td>target (locator)</td>  
 <td>Value</td>  
 </tr>

#### **Insert Comment**

Comments may be added to make your test case more readable. These comments are ignored when the test case is run.

Table View or Source View

#### **Edit a Command or Comment**

##### **Table View**

Simply select the line to be changed and edit it using the Command, Target, and Value fields.

##### **Source View**

Since Source view provides the equivalent of a WYSIWYG (What You See is What You Get) editor, simply modify which line you wish–command, parameter, or comment.

4. **Opening and Saving a Test Case**

Like most programs, there are Save and Open commands under the File menu.

## Running Test Cases

The IDE allows many options for running your test case. You can run a test case all at once, stop and start it, run it one line at a time, run a single command you are currently developing, and you can do a batch run of an entire test suite. Execution of test cases is very flexible in the IDE.

**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.