Selenium 7 Day Revise Module By:- Anjani

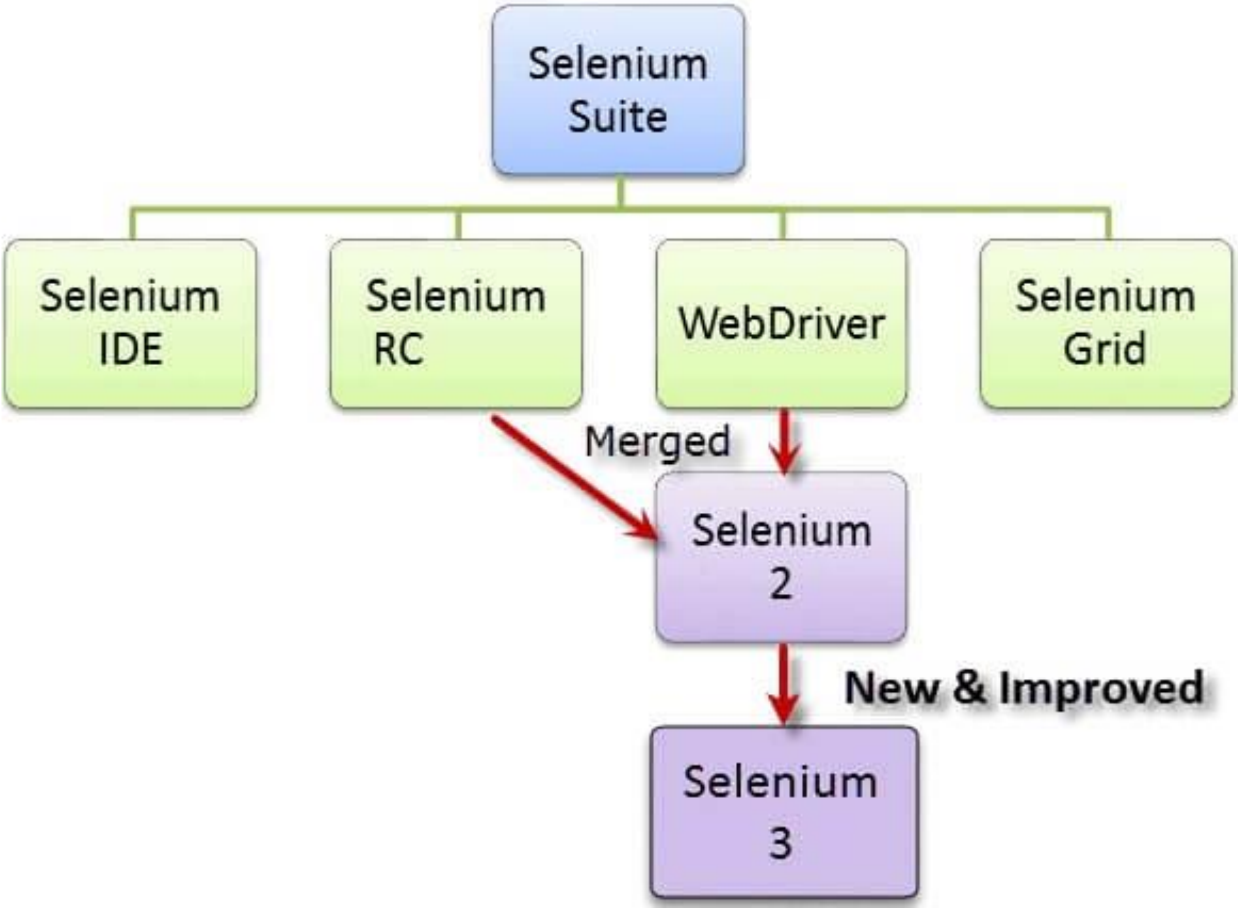
Introduction to Selenium

What is Selenium?

Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. It is quite similar to HP Quick Test Pro (QTP now UFT) only that Selenium focuses on automating web-based applications. Testing done using Selenium tool is usually referred as Selenium Testing.

Selenium is not just a single tool but a suite of software's, each catering to different testing needs of an organization. **It has four components.**

- Selenium Integrated Development Environment (IDE)
- Selenium Remote Control (RC)
- WebDriver
- Selenium Grid



At the moment, Selenium RC and WebDriver are merged into a single framework to form **Selenium 2**. Selenium 1, by the way, refers to Selenium RC.

Who developed Selenium?

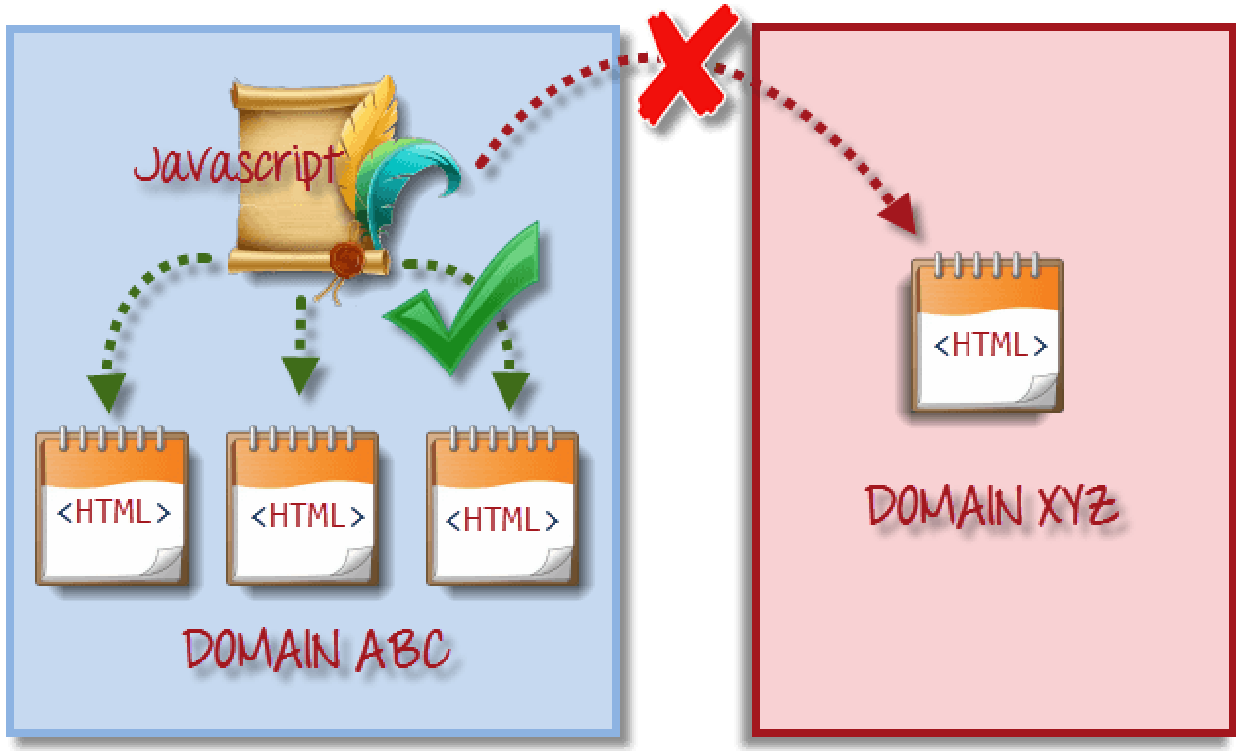
Since Selenium is a collection of different tools, it had different developers as well. Below are the key persons who made notable contributions to the Selenium Project

Primarily, Selenium was **created by Ja**
Huggins in 2004. An engineer at Thou
he was working on a web application th
frequent testing. Having realized that th
repetitious Manual Testing of their appl
was becoming more and more inefficie
created a JavaScript program that woul
automatically control the browser's acti
named this program as the
"**JavaScriptTestRunner**."

Seeing potential in this idea to help aut
web applications, he made JavaScriptR
open-source which was later re-named
as **Selenium Core**.

The Same Origin Policy Issue

Same Origin policy prohibits JavaScript code from accessing elements from a domain that is different from where it was launched. Example, the HTML code in www.google.com uses a JavaScript program "randomScript.js". The same origin policy will only allow randomScript.js to access pages within google.com such as google.com/mail, google.com/login, or google.com/signup. However, it cannot access pages from different sites such as yahoo.com/search or guru99.com because they belong to different domains.



Under Same Origin Policy, a Javascript program can only access pages on the same domain where it belongs. It cannot access pages from different domains

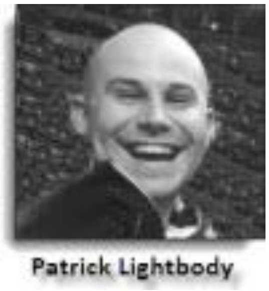
This is the reason why prior to Selenium RC, testers needed to install local copies of both Selenium Core (a JavaScript program) and the web server containing the web application being tested so they would belong to the same domain

Birth of Selenium Remote Control (Selenium RC)

Paul Hammant

Unfortunately; testers using Selenium Core had to install the whole application under test and the web server on their own local computers because of the restrictions imposed by the **same origin policy.** So another ThoughtWork's engineer, **Paul Hammant**, decided to create a server that will act as an HTTP proxy to "trick" the browser into believing that Selenium Core and the web application being tested come from the same domain. This system became known as the **Selenium Remote Control** or **Selenium 1**.

Birth of Selenium Grid



Selenium Grid was developed by **Patrick Lightbody** to address the need of minimizing test execution times as much as possible. He initially called the system "**Hosted QA**." It was capable of capturing browser screenshots during significant stages, and also of **sending out Selenium commands to different machines simultaneously.**

Birth of Selenium IDE

Shinya Kasatani of Japan created **Selenium IDE**, a Firefox extension that can automate the browser through a record-and-playback feature. He came up with this idea to further increase the speed in creating test cases. He donated Selenium IDE to the Selenium Project in **2006**.

Birth of WebDriver

Simon Stewart

Simon Stewart created WebDriver circa **2006** when browsers and web applications were becoming more powerful and more restrictive with JavaScript programs like Selenium Core. **It was the first cross-platform testing framework that could control the browser from the OS level.**

Birth of Selenium 2

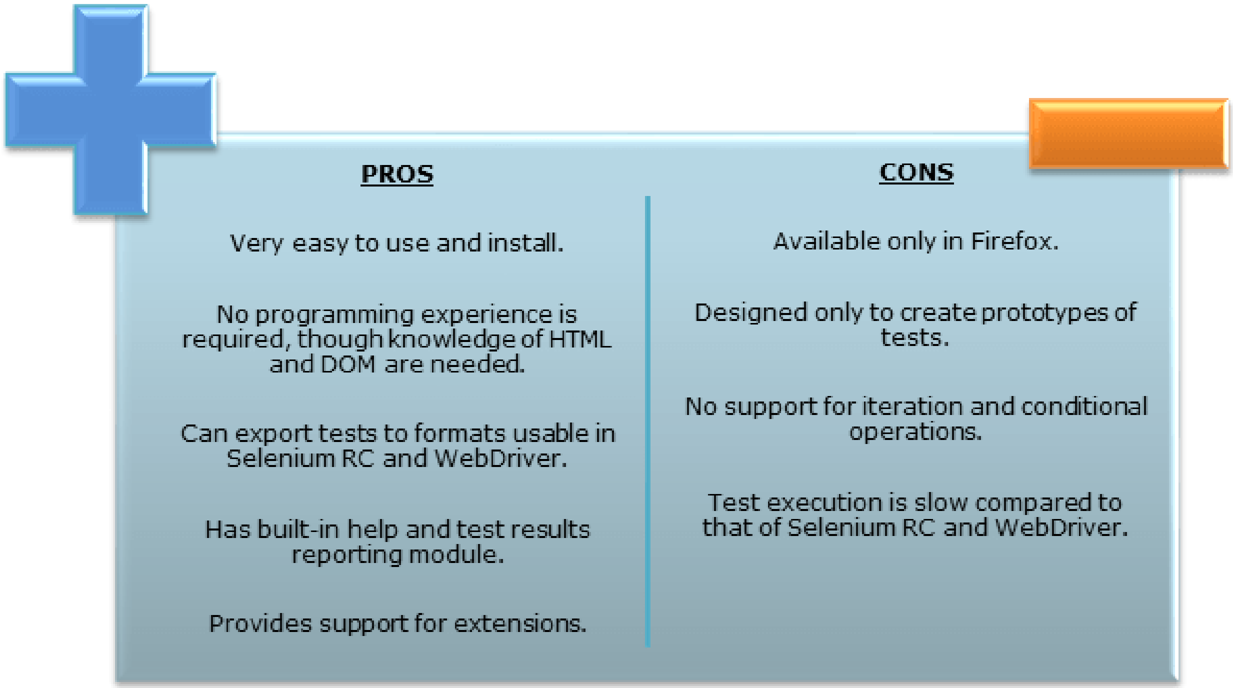
In **2008**, the whole Selenium Team decided to merge WebDriver and Selenium RC to form a more powerful tool called **Selenium 2**, with **WebDriver being the core**. Currently, Selenium RC is still being developed but only in maintenance mode. Most of the Selenium Project's efforts are now focused on Selenium 2.

So, Why the Name Selenium?

It came from a joke which Jason cracked one time to his team. Another automated testing framework was popular during Selenium's development, and it was by the company called **Mercury Interactive** (yes, the company who originally made QTP before it was acquired by HP). Since Selenium is a well-known antidote for Mercury poisoning, Jason suggested that name. His teammates took it, and so that is how we got to call this framework up to the present.

Brief Introduction Selenium IDE

Selenium Integrated Development Environment (IDE) is the **simplest framework** in the Selenium suite and is **the easiest one to learn**. It is a **Firefox plugin** that you can install as easily as you can with other plugins. However, because of its simplicity, Selenium IDE should only be used as a **prototyping tool**. If you want to create more advanced test cases, you will need to use either Selenium RC or WebDriver.

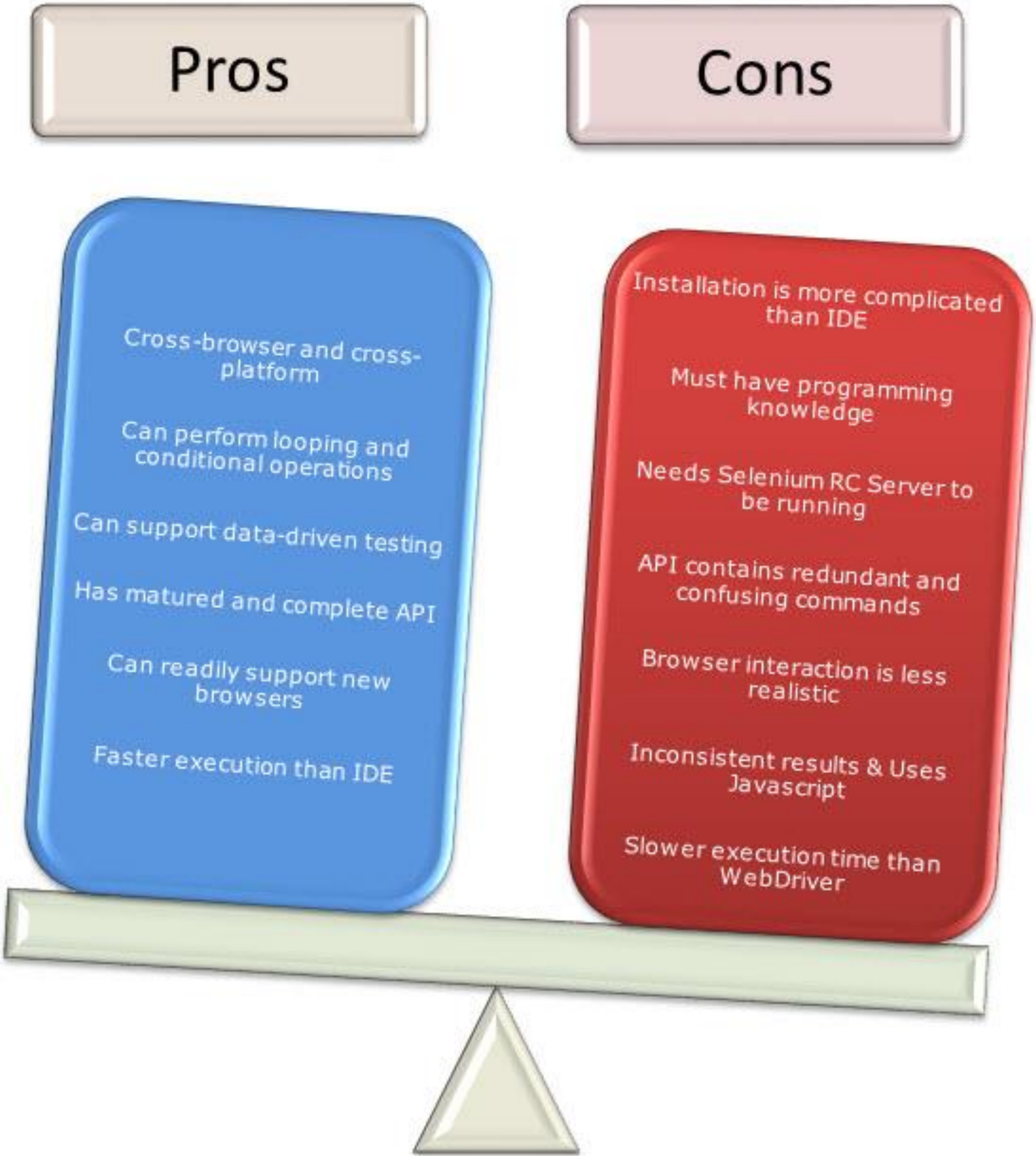


PROS	CONS
Very easy to use and install.	Available only in Firefox.
No programming experience is required, though knowledge of HTML and DOM are needed.	Designed only to create prototypes of tests.
Can export tests to formats usable in Selenium RC and WebDriver.	No support for iteration and conditional operations.
Has built-in help and test results reporting module.	Test execution is slow compared to that of Selenium RC and WebDriver.
Provides support for extensions.	

Brief Introduction Selenium Remote Control (Selenium RC)

Selenium RC was the **flagship testing framework** of the whole Selenium project for a long time. This is the first automated web testing tool that **allowed users to use a programming language they prefer**. As of version 2.25.0, RC can support the following programming languages:

- Java
- C#
- PHP
- Python
- Perl
- Ruby

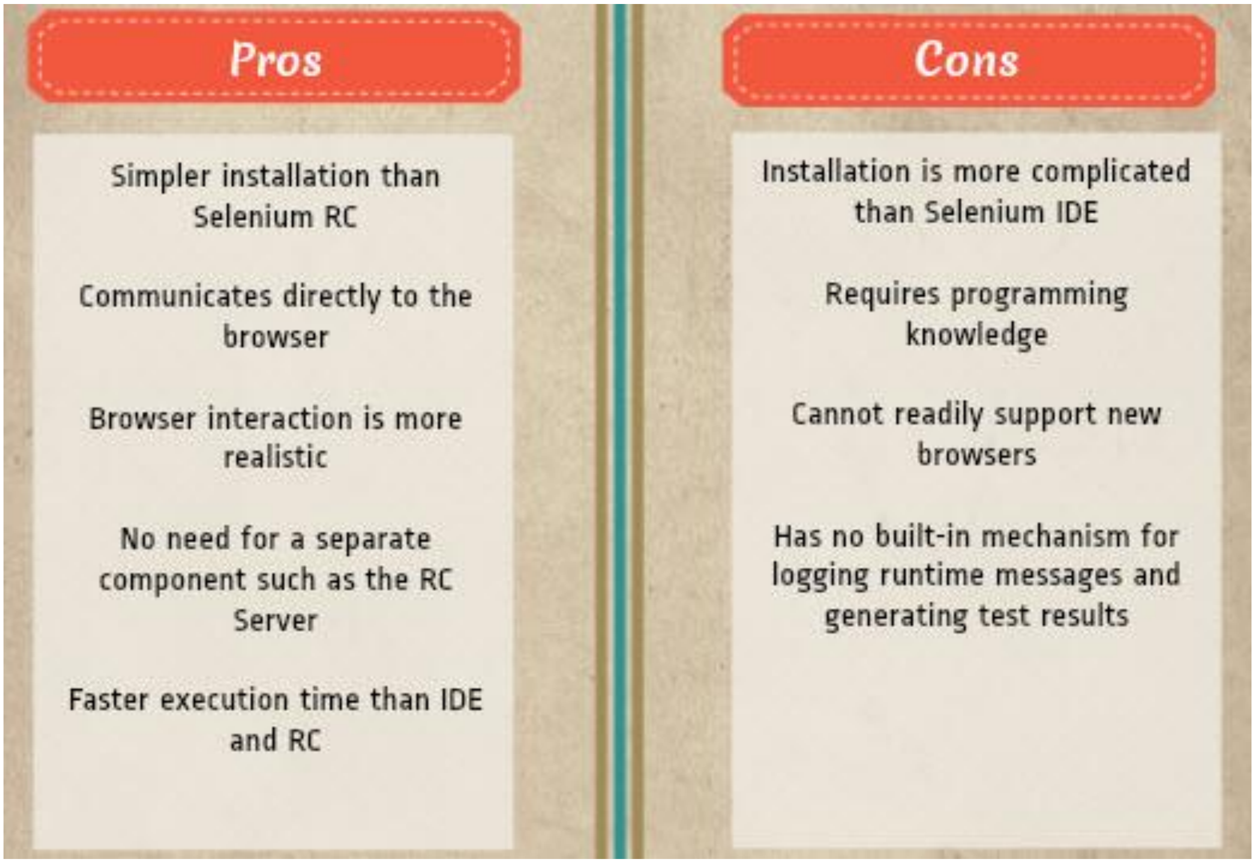

Pros
Cons
Cross-browser and cross-platform
Can perform looping and conditional operations
Can support data-driven testing
Has matured and complete API
Can readily support new browsers
Faster execution than IDE
Installation is more complicated than IDE
Must have programming knowledge
Needs Selenium RC Server to be running
API contains redundant and confusing commands
Browser interaction is less realistic
Inconsistent results & Uses Javascript
Slower execution time than WebDriver

Brief Introduction WebDriver

The WebDriver proves itself to be **better than both Selenium IDE and Selenium RC** in many aspects. It implements a more modern and stable approach in automating the browser's actions. WebDriver, unlike Selenium RC, does not rely on JavaScript for Automation. **It controls the browser by directly communicating with it.**

The supported languages are the same as those in Selenium RC.

- Java
- C#
- PHP
- Python
- Perl
- Ruby



Pros	Cons
Simpler installation than Selenium RC	Installation is more complicated than Selenium IDE
Communicates directly to the browser	Requires programming knowledge
Browser interaction is more realistic	Cannot readily support new browsers
No need for a separate component such as the RC Server	Has no built-in mechanism for logging runtime messages and generating test results
Faster execution time than IDE and RC	

Selenium Grid

Selenium Grid is a tool **used together with Selenium RC to run parallel tests** across different machines and different browsers all at the same time. Parallel execution means running multiple tests at once.

Features:

- Enables **simultaneous running of tests** in **multiple browsers and environments.**
- **Saves time** enormously.
- Utilizes the **hub-and-nodes** concept. The hub acts as a central source of Selenium commands to each node connected to it.

Note on Browser and Environment Support

Because of their architectural differences, Selenium IDE, Selenium RC, and WebDriver support different sets of browsers and operating environments.

	Selenium IDE	**WebDriver**
BrowserSupport	Mozilla Firefox	Internet Explorer versions 6 to 11, both 32 and 64-bit Microsoft Edge version 12.10240 & above (partial support some functionalities under development) Firefox 3.0 and above Google Chrome 12.0. and above Opera 11.5 and above Android - 2.3 and above for phones and tablets (devices & emulators) iOS 3+ for phones (devices & emulators) and 3.2+ for tablets (devices & emulators) HtmlUnit 2.9 and above
Operating System	Windows,Mac OS X, Linux	All operating systems where the browsers above can run.

Note: Selenium WebDriver is termed as the successor of Selenium RC which has been deprecated & officially announced by SeleniumHQ.

How to Choose the Right Selenium Tool for Your Need

Tool	**Why Choose?**
Selenium IDE	• To learn about concepts on automated testing and Selenium, including: • Selenese commands such as type, open, clickAndWait, assert, verify, etc. • Locators such as id, name, xpath, css selector, etc. • Executing customized JavaScript code using runScript • Exporting test cases in various formats. • To create tests with little or no prior knowledge in programming. • To create simple test cases and test suites that you can export later to RC or WebDriver. • To test a web application against Firefox only.
Selenium RC	• To design a test using a more expressive language than Selenese • To run your test against different browsers (except HtmlUnit) on different operating systems. • To deploy your tests across multiple environments using Selenium Grid.

Tool	Why Choose?
	• To test your application against a new browser that supports JavaScript. • To test web applications with complex AJAX-based scenarios.
WebDriver	• To use a certain programming language in designing your test case. • To test applications that are rich in AJAX-based functionalities. • To execute tests on the HtmlUnit browser. • To create customized test results.
Selenium Grid	• To run your Selenium RC scripts in multiple browsers and operating systems simultaneously. • To run a huge test suite, that needs to complete in the soonest time possible.

A Comparison between Selenium and QTP(now UFT)

Quick Test Professional(QTP) is a proprietary automated testing tool previously owned by the company **Mercury Interactive** before it was **acquired by Hewlett-Packard in 2006**. The Selenium Tool Suite has many advantages over QTP as detailed below -

Advantages of Selenium over QTP

Selenium	QTP
Open source, **free to use**, and **free of charge.**	**Commercial**.
Highly extensible	Limited add-ons
Can run tests across **different browsers**	Can only run tests in **Firefox**, **Internet Explorer**and **Chrome**
Supports **various operating systems**	Can only be used in **Windows**
Supports **mobile devices**	QTP Supports Mobile app test automation (iOS & Android) using HP solution called - HP Mobile Center
Can execute tests **while** the **browser is minimized**	Needs to have the application under test to be visible on the desktop
Can execute tests **in parallel**.	Can only execute in parallel but using Quality Center which is again a paid product.

Advantages of QTP over Selenium

Advantages of QTP over Selenium

QTP	Selenium
Can test **both web and desktop applications**	Can only test web applications
Comes with a **built-in object repository**	Has no built-in object repository
Automates faster than Selenium because it is a fully featured IDE.	Automates at a slower rate because it does not ha native IDE and only third party IDE can be used development
Data-driven testing is easier to perform because **it has built-in global and local data tables**.	Data-driven testing is more cumbersome since yo rely on the programming language's capabilities f values for your test data
Can access controls within the browser(such as the Favorites bar, Address bar, Back and Forward buttons, etc.)	Cannot access elements outside of the web applic under test
Provides professional **customer support**	No official user support is being offered.
Has native capability to **export test data** into external formats	Has no native capability to export runtime data o external formats
Parameterization Support is built	Parameterization can be done via programming b difficult to implement.
Test Reports are generated automatically	No native support to generate test /bug reports.

Though clearly, QTP has more advanced capabilities, Selenium outweighs QTP in three main areas:

- **Cost**(because Selenium is completely free)
- **Flexibility**(because of a number of programming languages, browsers, and platforms it can support)
- **Parallel testing**(something that QTP is capable of but only with use of Quality Center)

Summary

- The entire Selenium Tool Suite is comprised of four components:
  - **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.
- Selenium RC and WebDriver was merged to form **Selenium 2**.
- Selenium is more advantageous than QTP in terms of **costs and flexibility**. It also allows you to **run tests in parallel**, unlike in QTP where you are only allowed to run tests sequentially.

•

Day 1 - Part 2

How to Download & Install Selenium IDE for Firefox

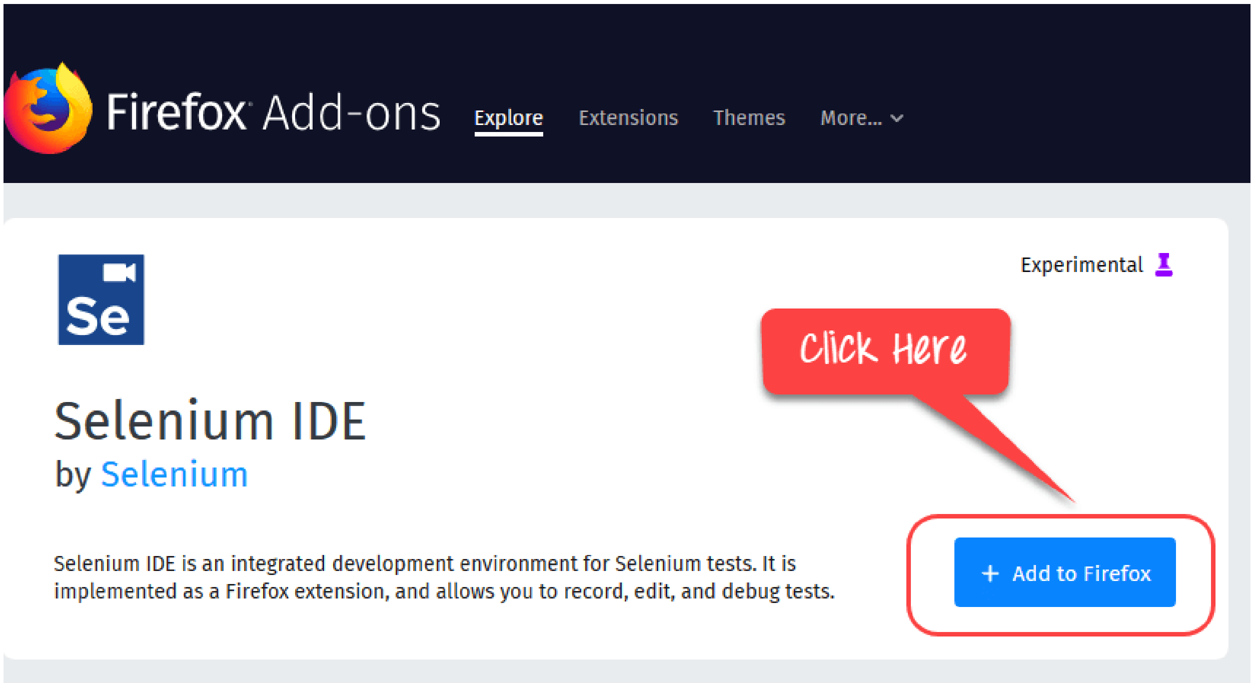
Installation of Selenium IDE

What you need

- Mozilla Firefox
- Active Internet Connection

If you do not have Mozilla Firefox yet, you can download it from http://www.mozilla.org/en-US/firefox/new.

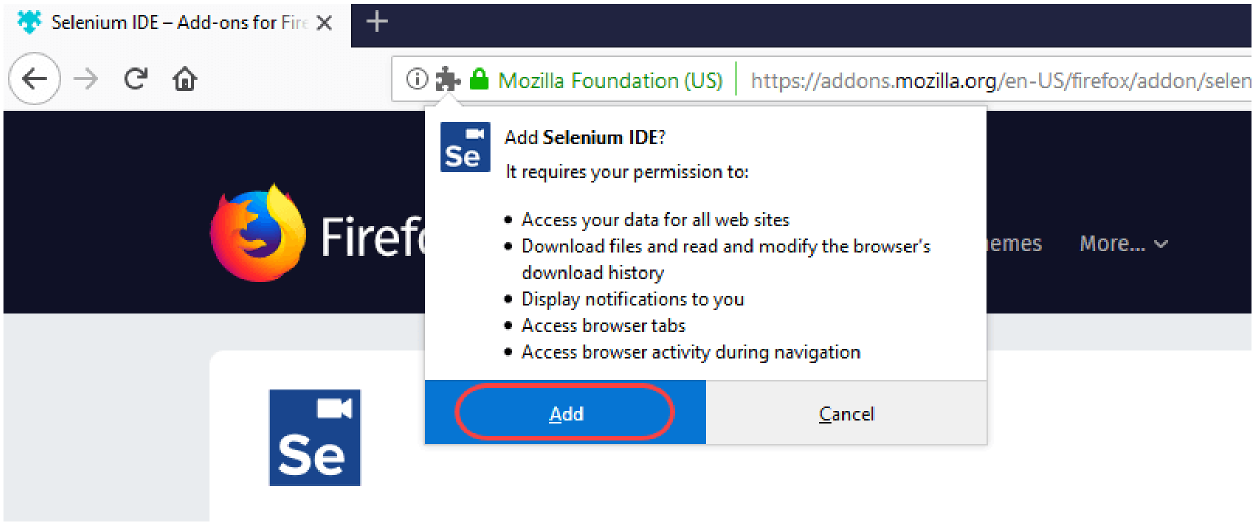
Steps 1) Launch Firefox and navigate to https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/. Click on Add to Firefox



Selenium IDE
by Selenium

Selenium IDE is an integrated development environment for Selenium tests. It is implemented as a Firefox extension, and allows you to record, edit, and debug tests.

Steps 2) Wait until Firefox completes the download and then click "**Add.**"



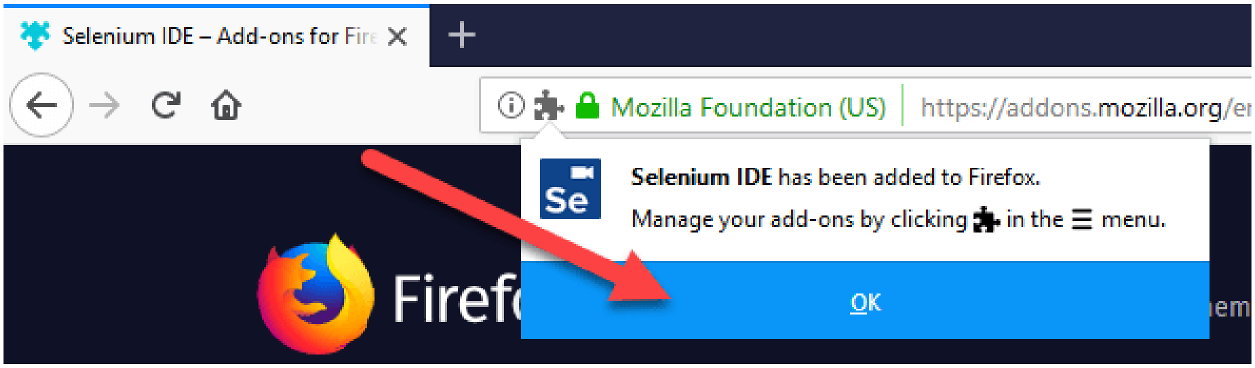
https://addons.mozilla.org/en-US/firefox/addon/selen

Add **Selenium IDE**?

It requires your permission to:

- Access your data for all web sites
- Download files and read and modify the browser's download history
- Display notifications to you
- Access browser tabs
- Access browser activity during navigation

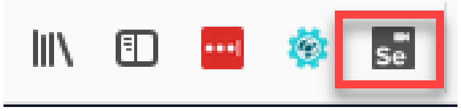
Steps 3) Once install is complete, you will get a confirmation message. Click **"OK"**



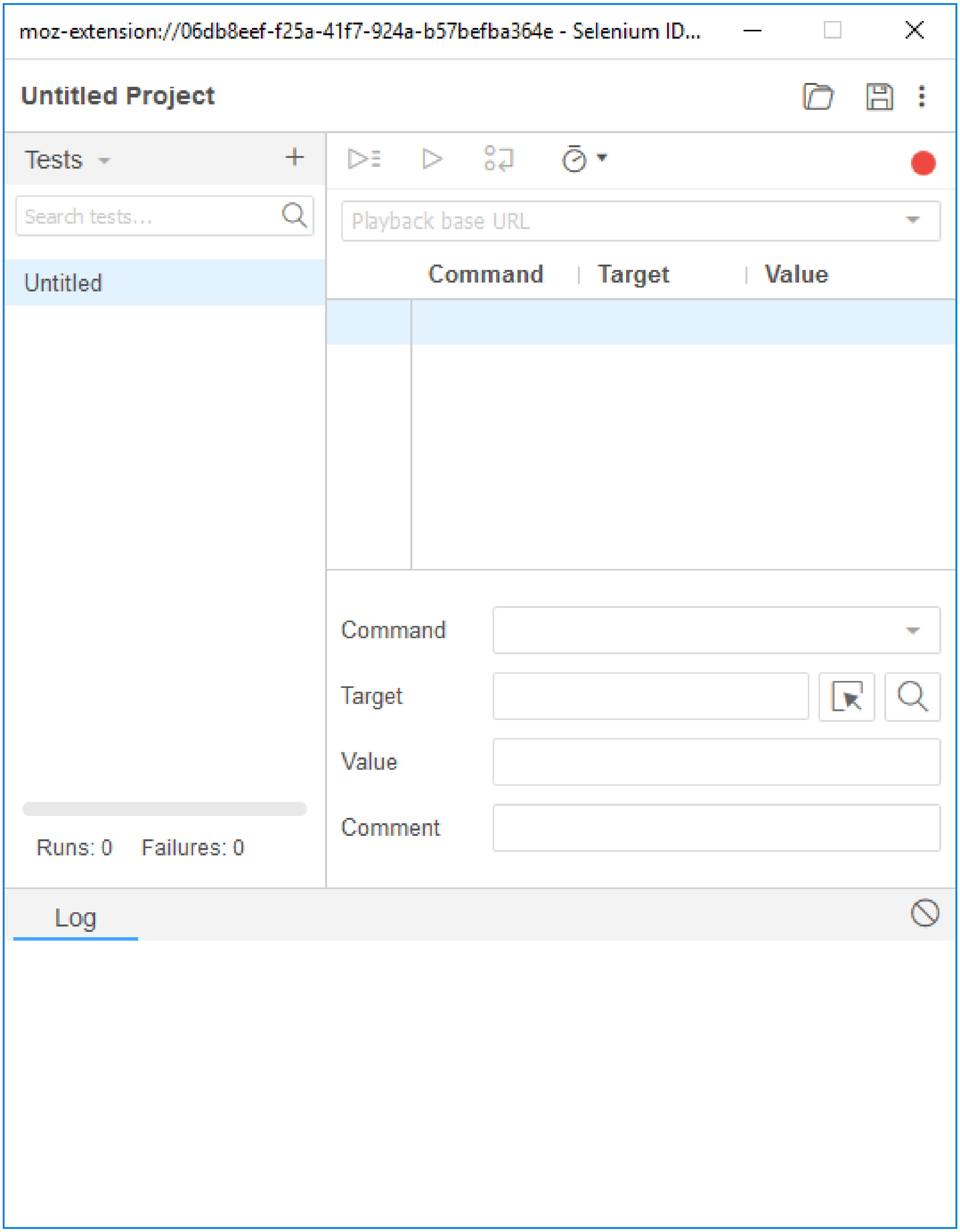
Selenium IDE has been added to Firefox.

Manage your add-ons by clicking 🧩 in the ☰ menu.

Steps 4) Click on the Selenium IDE icon



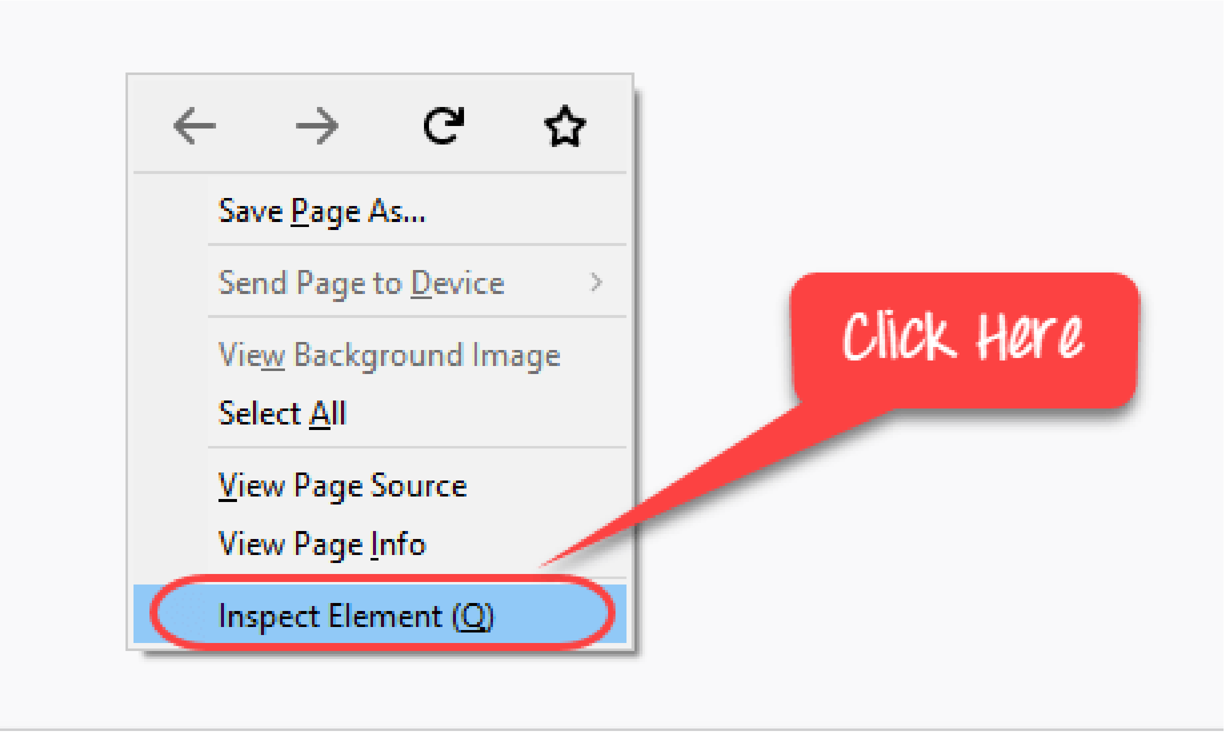
Selenium IDE will open



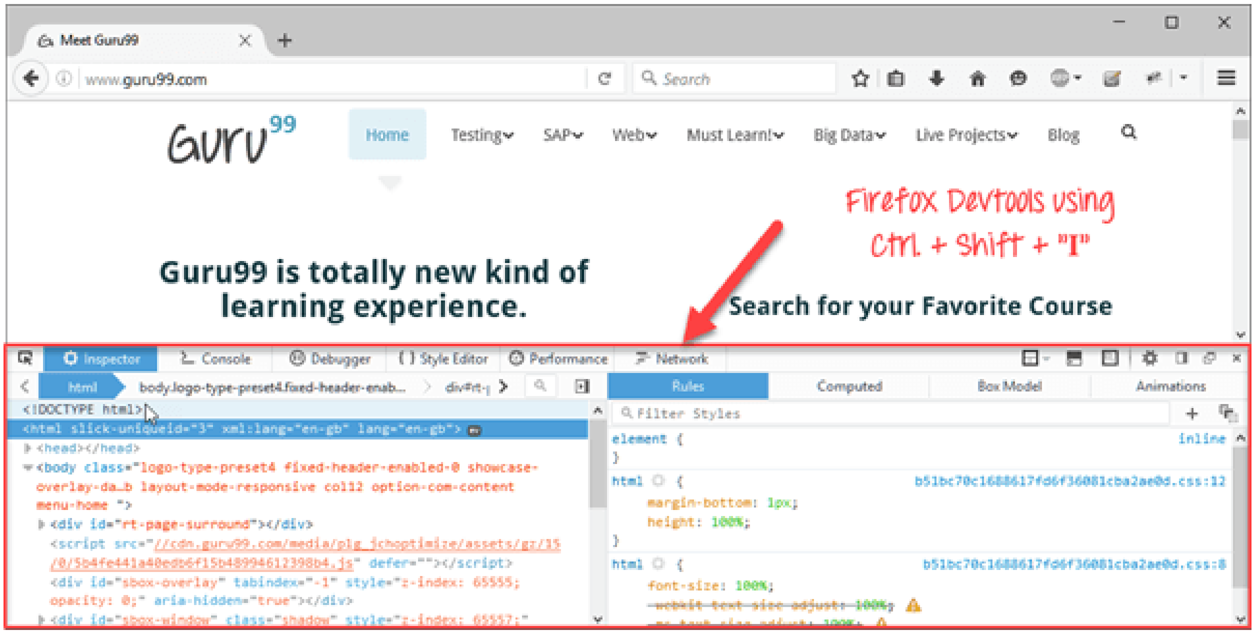
Firefox DevTools in Firefox

Firefox DevTools is a Firefox feature that we will use to **inspect the HTML elements** of the web application under test. It will provide us the name of the element that our Selenese command would act upon.

Step 1) Right click anywhere on the page and select Inspect Element. You can also use shortcut Cntrl + Shift + I



Step 2) You will see the Interface

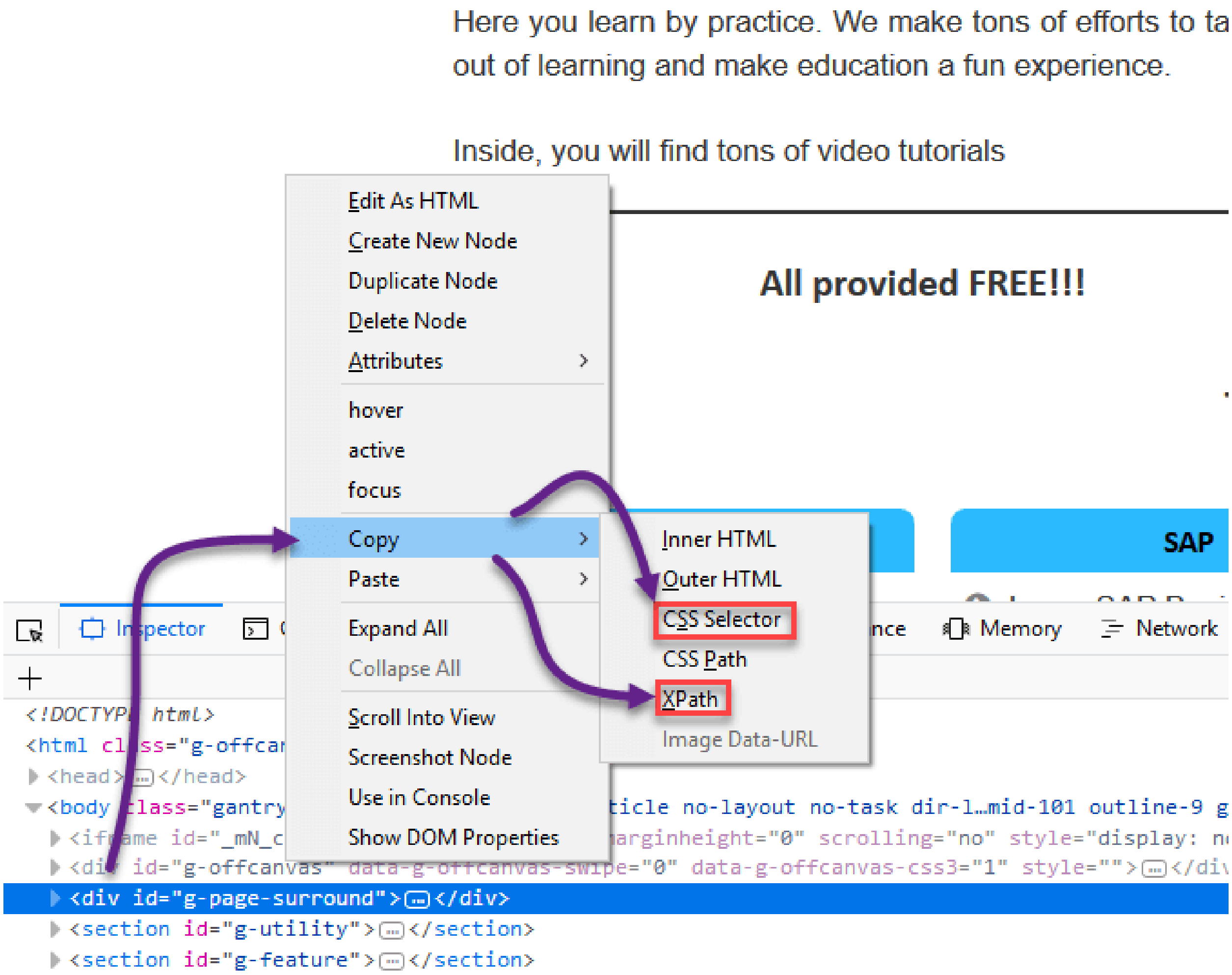


Firefox Devtools using Ctrl + Shift + "I"

```
<!DOCTYPE html>
<html slick-uniqueid="3" xml:lang="en-gb" lang="en-gb">
<head></head>
<body class="logo-type-preset4 fixed-header-enabled-0 showcase-
overlay-da…b layout-mode-responsive col12 option-com-content
menu-home ">
  <div id="rt-page-surround"></div>
  <script src="//cdn.guru99.com/media/plg_jchoptimize/assets/gz/15
  /0/5b4fe441a40edb6f15b48994612398b4.js" defer=""></script>
  <div id="sbox-overlay" tabindex="-1" style="z-index: 65555;
  opacity: 0;" aria-hidden="true"></div>
  <div id="sbox-window" class="shadow" style="z-index: 65557;"
```

```
element {                                                        inline
}
html {                          b51bc70c1688617fd6f36081cba2ae0d.css:12
    margin-bottom: 1px;
    height: 100%;
}
html {                           b51bc70c1688617fd6f36081cba2ae0d.css:8
    font-size: 100%;
    ~~-webkit-text-size-adjust: 100%;~~
```

Step 3) You can right click on an element and chose CSS or XPath. This is useful in object identification



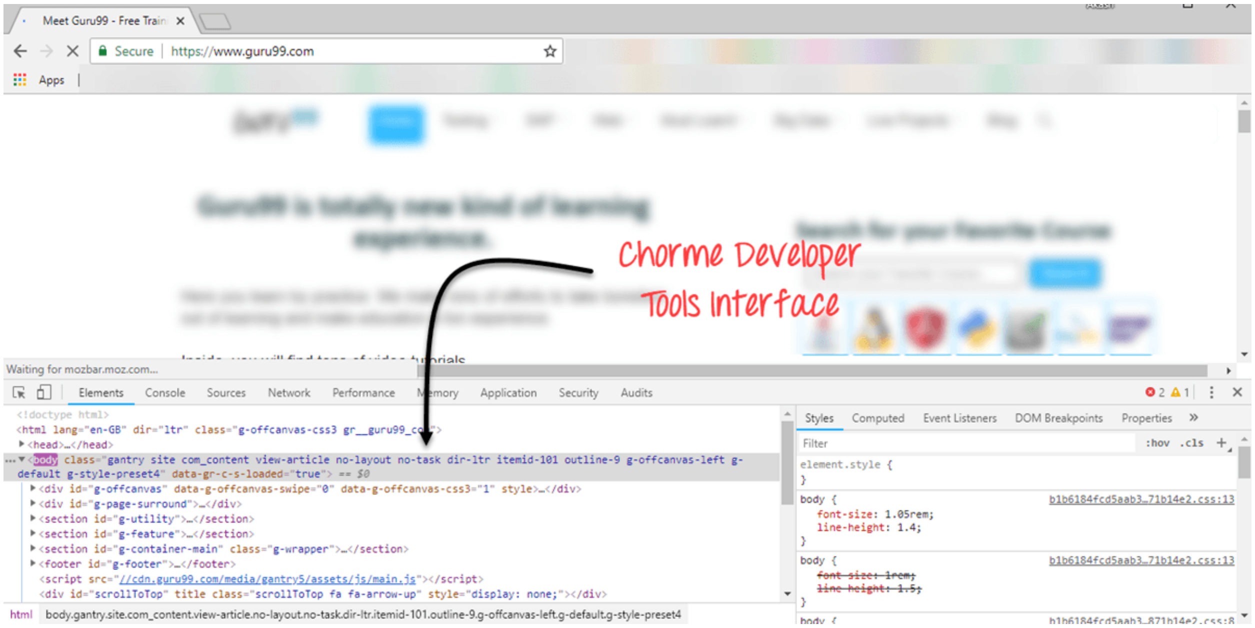
Here you learn by practice. We make tons of efforts to ta out of learning and make education a fun experience.

Inside, you will find tons of video tutorials

All provided FREE!!!

```
<!DOCTYP  html>
<html cl ss="g-offcar
 <head>…</head>
 <body class="gantry                              ticle no-layout no-task dir-l…mid-101 outline-9 g
   <if ame id="_mN_c                              arginheight="0" scrolling="no" style="display: n
   <di  id="g-offcanvas" data-g-offcanvas-swipe="0" data-g-offcanvas-css3="1" style="">…</di
   <div id="g-page-surround">…</div>
   <section id="g-utility">…</section>
   <section id="g-feature">…</section>
```

Note: Likewise, you can also use Developer Tools in Chrome to identify object properties



Chorme Developer Tools Interface

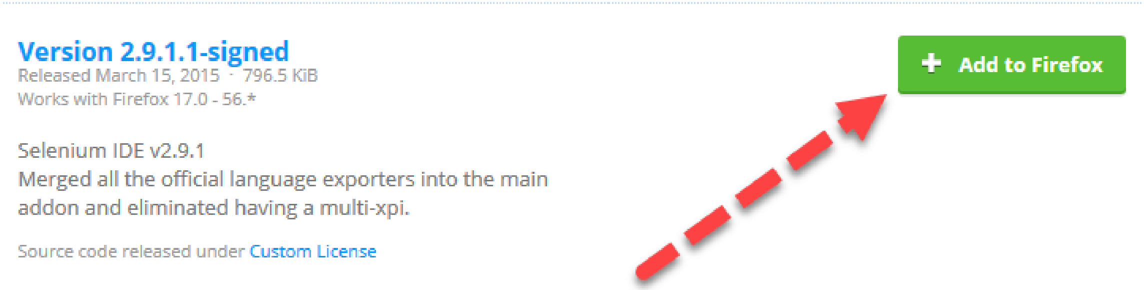
```
<!doctype html>
<html lang="en-GB" dir="ltr" class="g-offcanvas-css3 gr__guru99_co">
▶<head>…</head>
…▼<body class="gantry site com_content view-article no-layout no-task dir-ltr itemid-101 outline-9 g-offcanvas-left g-
default g-style-preset4" data-gr-c-s-loaded="true"> == $0
  ▶<div id="g-offcanvas" data-g-offcanvas-swipe="0" data-g-offcanvas-css3="1" style>…</div>
  ▶<div id="g-page-surround">…</div>
  ▶<section id="g-utility">…</section>
  ▶<section id="g-feature">…</section>
  ▶<section id="g-container-main" class="g-wrapper">…</section>
  ▶<footer id="g-footer">…</footer>
   <script src="//cdn.guru99.com/media/gantry5/assets/js/main.js"></script>
   <div id="scrollToTop" title class="scrollToTop fa fa-arrow-up" style="display: none;"></div>
html  body.gantry.site.com_content.view-article.no-layout.no-task.dir-ltr.itemid-101.outline-9.g-offcanvas-left.g-default.g-style-preset4
```

```
element.style {
}
body {                                   b1b6184fcd5aab3…71b14e2.css:13
   font-size: 1.05rem;
   line-height: 1.4;
}
body {                                   b1b6184fcd5aab3…71b14e2.css:13
   ~~font-size: 1rem;~~
   ~~line-height: 1.5;~~
}
body {                                   b1b6184fcd5aab3…871b14e2.css:8
```

Selenium IDE was deprecated, and the development had stopped. Only recently the project has been resurrected. The new Selenium lacks many features compared to the deprecated IDE. Features are being added but at a slow pace. To explore all the features of Selenium IDE, we recommend you use the old version. To use the old version of IDE

Step 1) Use Firefox 54 Portable Version check here

Step 2) Visit Selenium IDE version https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/versions/and install



Version 2.9.1.1-signed
Released March 15, 2015 · 796.5 KiB
Works with Firefox 17.0 - 56.*

Selenium IDE v2.9.1
Merged all the official language exporters into the main addon and eliminated having a multi-xpi.

Source code released under Custom License

The following features may not be available in latest IDE version. We will keep updating the tutorials as the new version is updated.

Plugins

Selenium IDE can support additional Firefox add-ons or plugins created by other users. You can visit here for a list of Selenium add-ons available to date. Install them just as you do with other Firefox add-ons.

By default, Selenium IDE comes bundled with 4 plugins:

1. Selenium IDE: C# Formatters

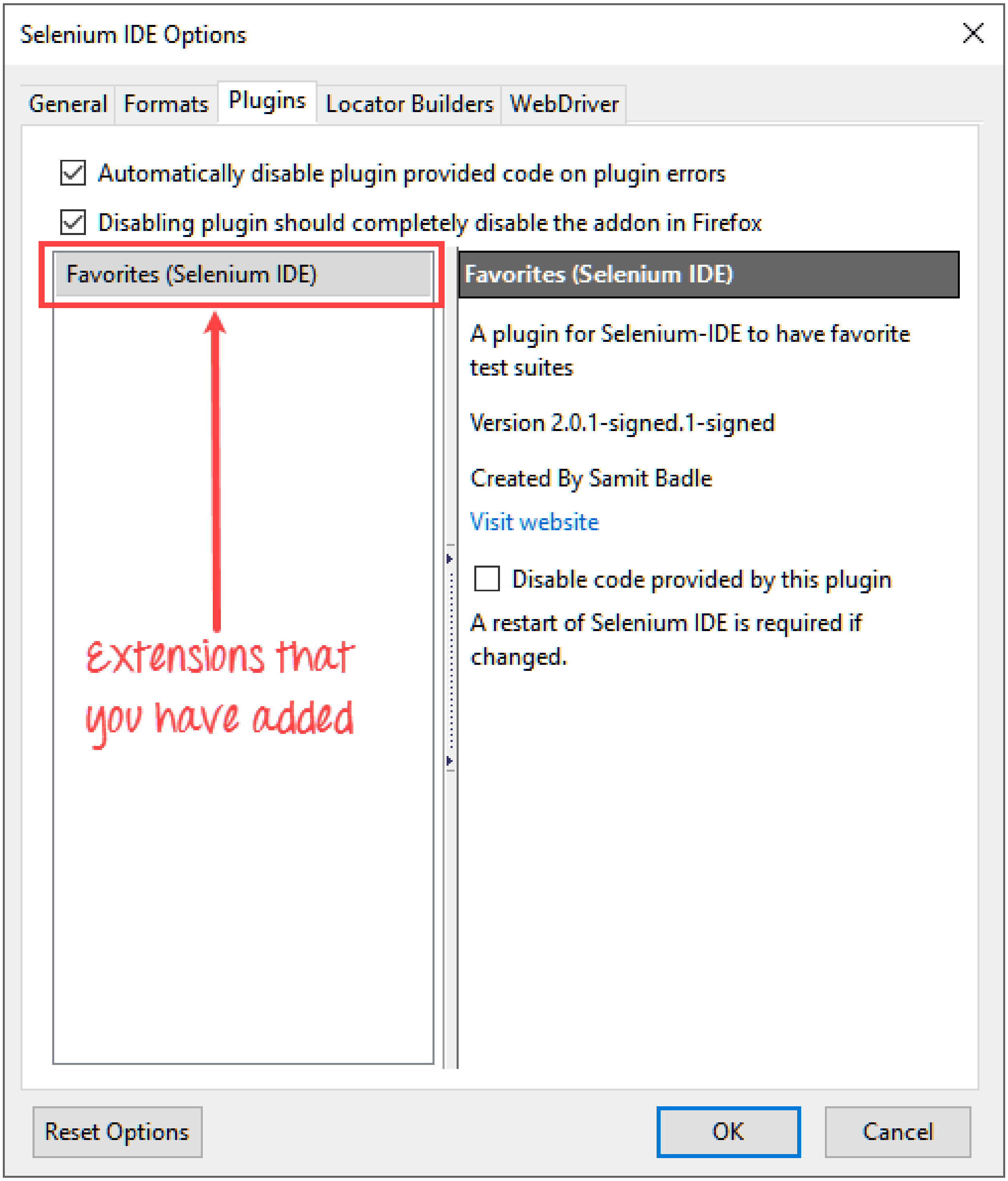
2. Selenium IDE: Java Formatters

3. Selenium IDE: Python Formatters

4. Selenium IDE: Ruby Formatters

These four plugins are required by Selenium IDE to convert Selenese into different formats.

The Plugins tab shows a list of all your installed add-ons, together with the version number and name of the creator of each.



☑ Automatically disable plugin provided code on plugin errors

☑ Disabling plugin should completely disable the addon in Firefox

Favorites (Selenium IDE)

A plugin for Selenium-IDE to have favorite test suites

Version 2.0.1-signed.1-signed

Created By Samit Badle

Visit website

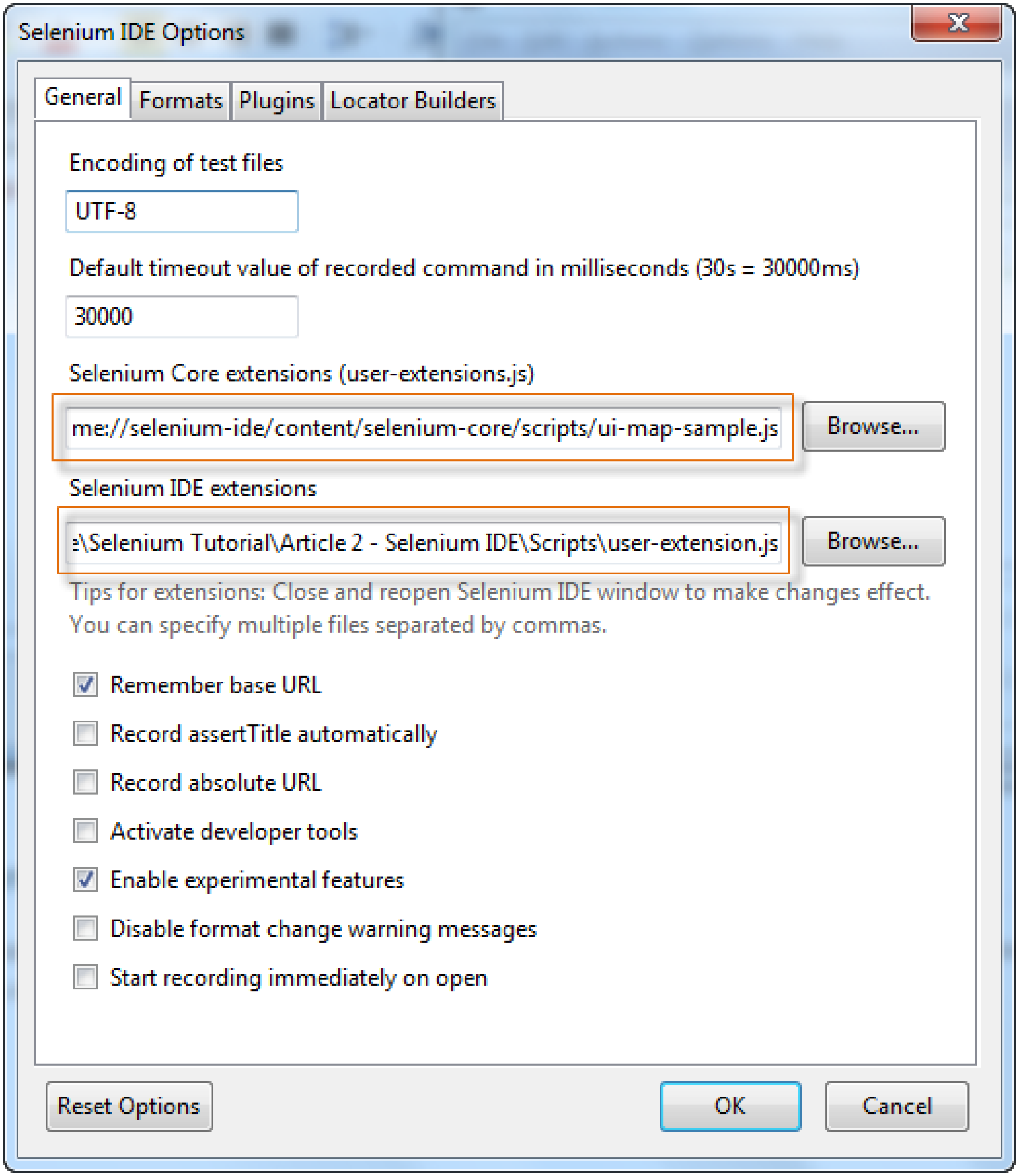
□ Disable code provided by this plugin

A restart of Selenium IDE is required if changed.

User Extensions

Selenium IDE can support user extensions to provide advanced capabilities. User extensions are in the form of JavaScript files. You install them by specifying their absolute path in either of these two fields in the Options dialog box.

- Selenium Core extensions (user-extensions.js)
- Selenium IDE extensions



Encoding of test files

UTF-8

Default timeout value of recorded command in milliseconds (30s = 30000ms)

30000

Selenium Core extensions (user-extensions.js)

me://selenium-ide/content/selenium-core/scripts/ui-map-sample.js

Selenium IDE extensions

:\Selenium Tutorial\Article 2 - Selenium IDE\Scripts\user-extension.js

Tips for extensions: Close and reopen Selenium IDE window to make changes effect. You can specify multiple files separated by commas.

☑ Remember base URL

□ Record assertTitle automatically

□ Record absolute URL

□ Activate developer tools

☑ Enable experimental features

□ Disable format change warning messages

□ Start recording immediately on open

You will be able to find tons of user extensions here.

Day 1 - Part 3

Introduction to Selenium IDE

Selenium IDE (Integrated Development Environment) is the simplest tool in the Selenium Suite. It is a Firefox add-on that creates tests very quickly through its record-and-playback functionality. This feature is similar to that of QTP. It is effortless to install and easy to learn.

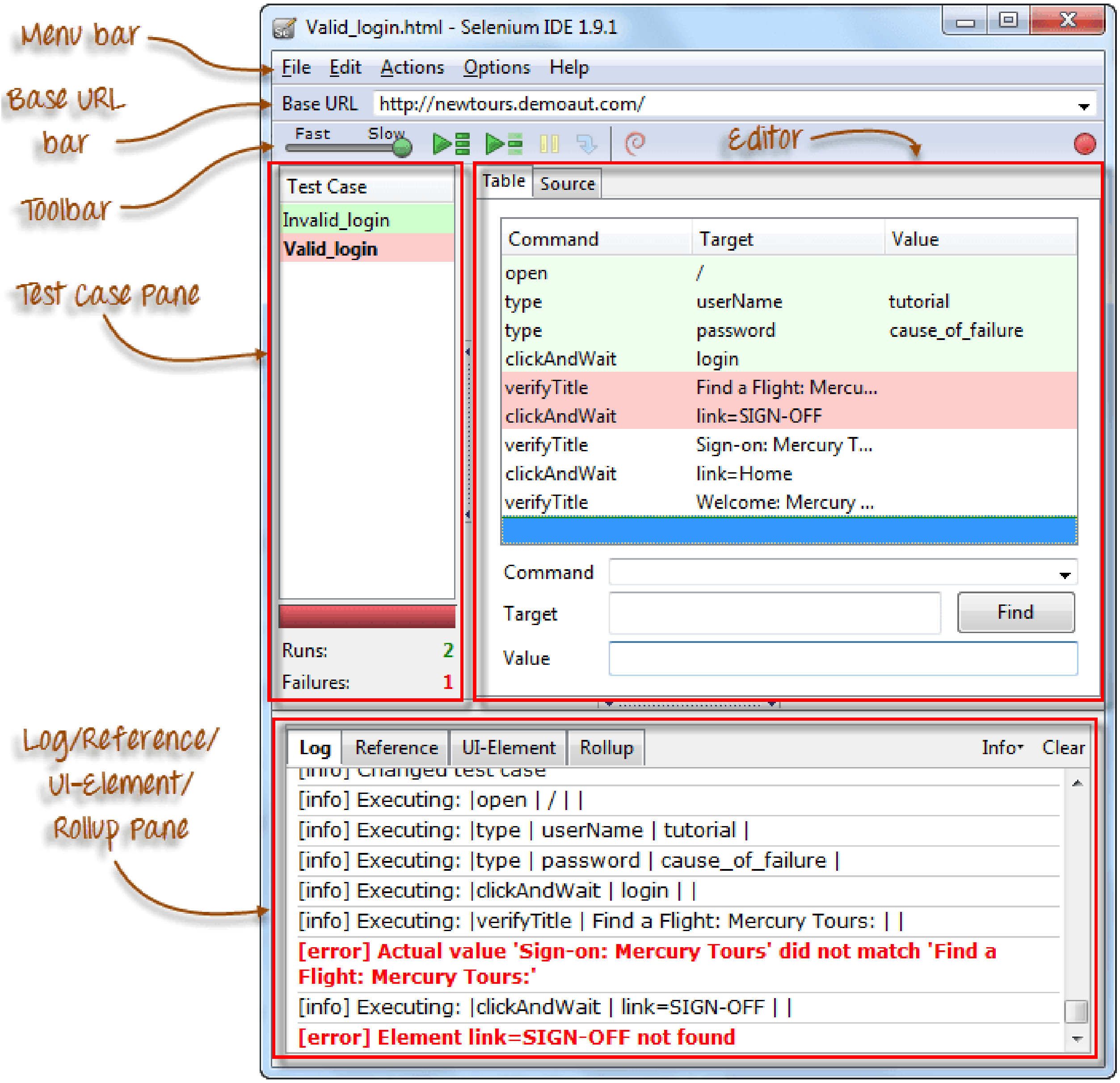
Because of its simplicity, Selenium IDE should only be used as a prototyping tool, not an overall solution for developing and maintaining complex test suites.

Though you will be able to use Selenium IDE without prior knowledge in programming, you should at least be familiar with HTML, JavaScript, and the DOM (Document Object Model) to utilize this tool to its full potential. Knowledge of JavaScript will be required when we get to the section about the Selenese command "**runScript."**

Selenium IDE supports autocomplete mode when creating tests. This feature serves two purposes:

- It helps the tester to enter commands more quickly.
- It restricts the user from entering invalid commands.

Features of Selenium IDE



Base URL http://newtours.demoaut.com/

Test Case
Invalid_login
Valid_login

Runs: 2

Failures: 1

Command	Target	Value
open	/	
type	userName	tutorial
type	password	cause_of_failure
clickAndWait	login	
verifyTitle	Find a Flight: Mercu...	
clickAndWait	link=SIGN-OFF	
verifyTitle	Sign-on: Mercury T...	
clickAndWait	link=Home	
verifyTitle	Welcome: Mercury ...	

Command

Target

Value

```
[info] Changed test case
[info] Executing: |open | / | |
[info] Executing: |type | userName | tutorial |
[info] Executing: |type | password | cause_of_failure |
[info] Executing: |clickAndWait | login | |
[info] Executing: |verifyTitle | Find a Flight: Mercury Tours: | |
[error] Actual value 'Sign-on: Mercury Tours' did not match 'Find a
Flight: Mercury Tours:'
[info] Executing: |clickAndWait | link=SIGN-OFF | |
[error] Element link=SIGN-OFF not found
```

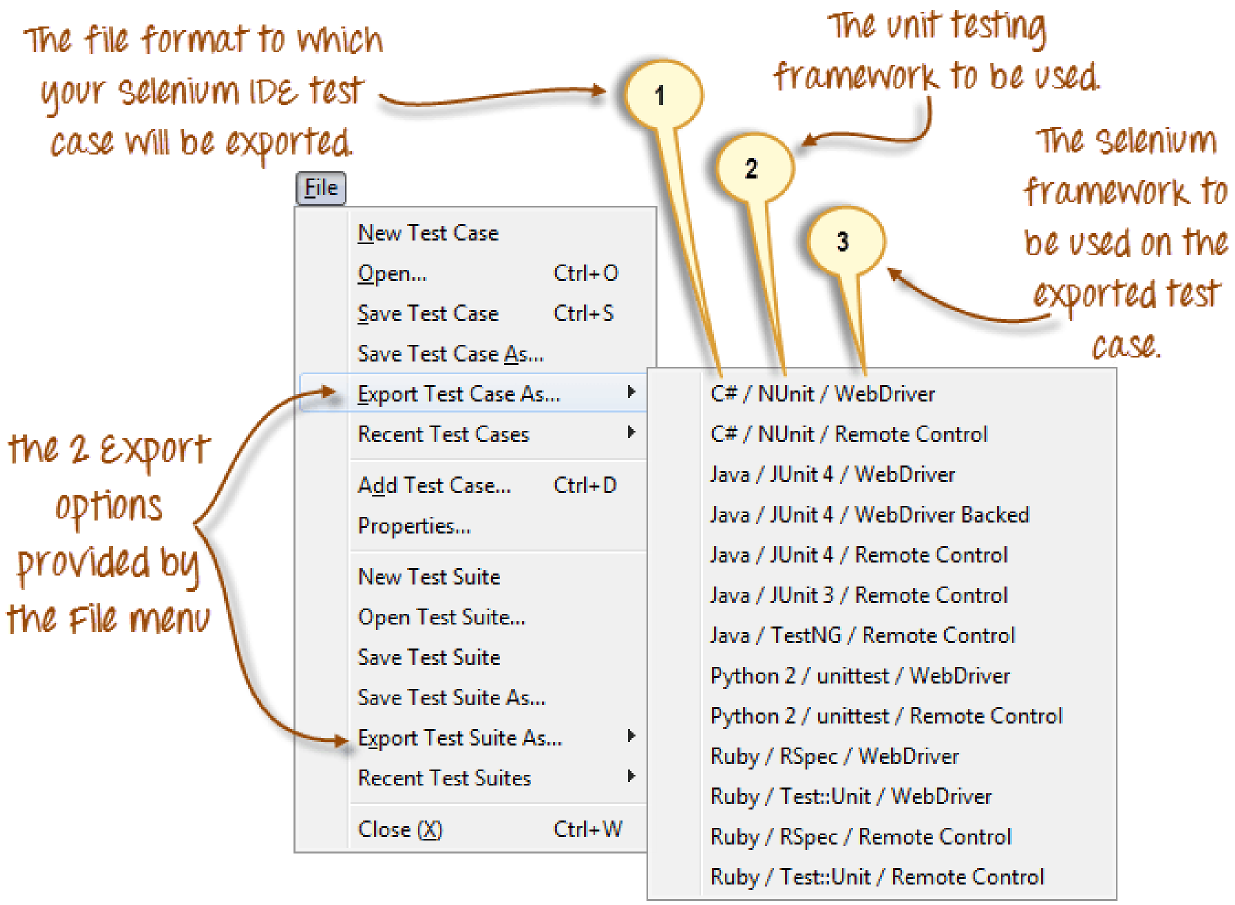
Menu Bar

It is located at the **top most portion** of the IDE. The most commonly used menus are the File, Edit, and Options menus.

File menu

- It contains options to create, open, save and close tests.
- 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**

- **"Export Test Case As..."** will export only the currently opened test case.
- **"Export Test Suite As..."** will export all the test cases in the currently opened test suite.



The file format to which your selenium IDE test case will be exported

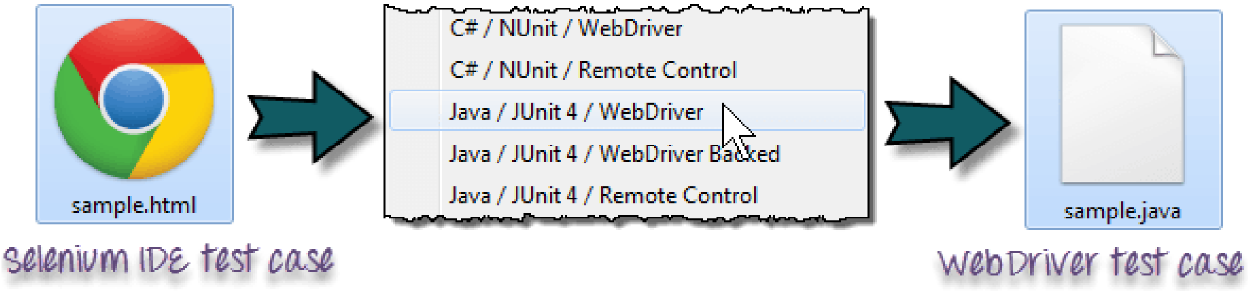
The unit testing framework to be used

The selenium framework to be used on the exported test case.

the 2 export options provided by the File menu

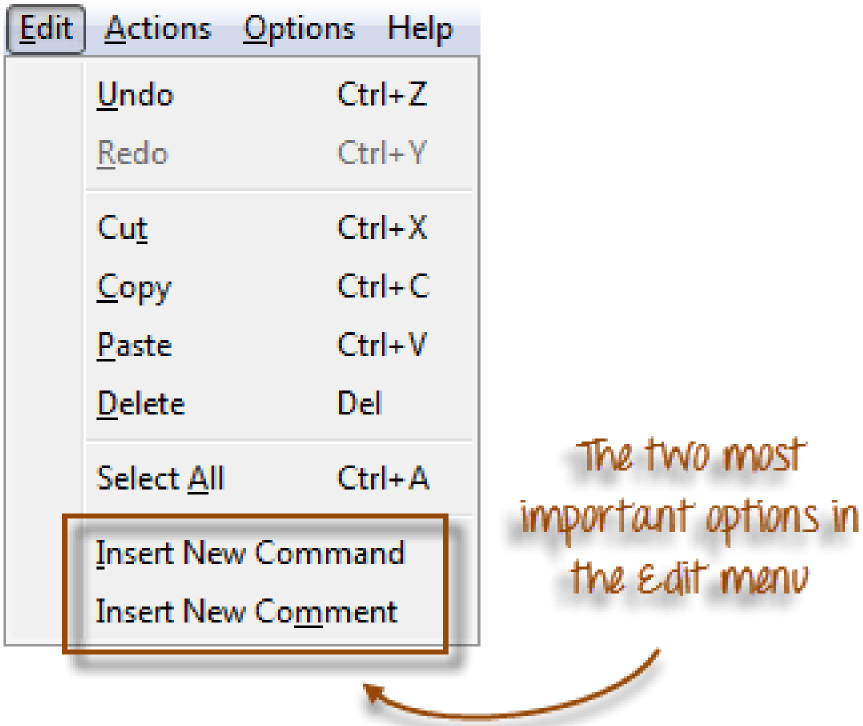
1
2
3
File
New Test Case
Open... Ctrl+O
Save Test Case Ctrl+S
Save Test Case As...
Export Test Case As...
Recent Test Cases
Add Test Case... Ctrl+D
Properties...
New Test Suite
Open Test Suite...
Save Test Suite
Save Test Suite As...
Export Test Suite As...
Recent Test Suites
Close (X) Ctrl+W
C# / NUnit / WebDriver
C# / NUnit / Remote Control
Java / JUnit 4 / WebDriver
Java / JUnit 4 / WebDriver Backed
Java / JUnit 4 / Remote Control
Java / JUnit 3 / Remote Control
Java / TestNG / Remote Control
Python 2 / unittest / WebDriver
Python 2 / unittest / Remote Control
Ruby / RSpec / WebDriver
Ruby / Test::Unit / WebDriver
Ruby / RSpec / Remote Control
Ruby / Test::Unit / Remote Control

- As of **Selenium IDE v1.9.1**, test cases can be exported only to the following formats:
- .cs (C# source code)
- .java (Java source code)
- .py (Python source code)
- .rb (Ruby source code)

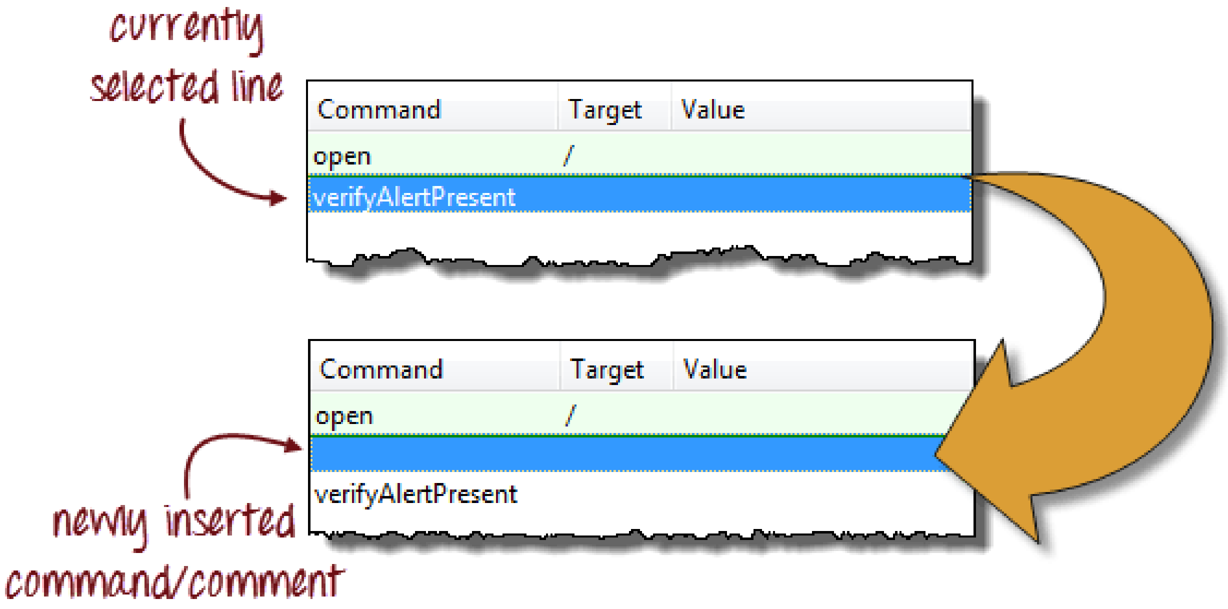

sample.html
C# / NUnit / WebDriver
C# / NUnit / Remote Control
Java / JUnit 4 / WebDriver
Java / JUnit 4 / WebDriver Backed
Java / JUnit 4 / Remote Control
sample.java
Selenium IDE test case
WebDriver test case

Edit Menu

- It contains usual options like Undo, Redo, Cut, Copy, Paste, Delete, and Select All.
- The two most important options are the "**Insert New Command**" and "**Insert New Comment**".



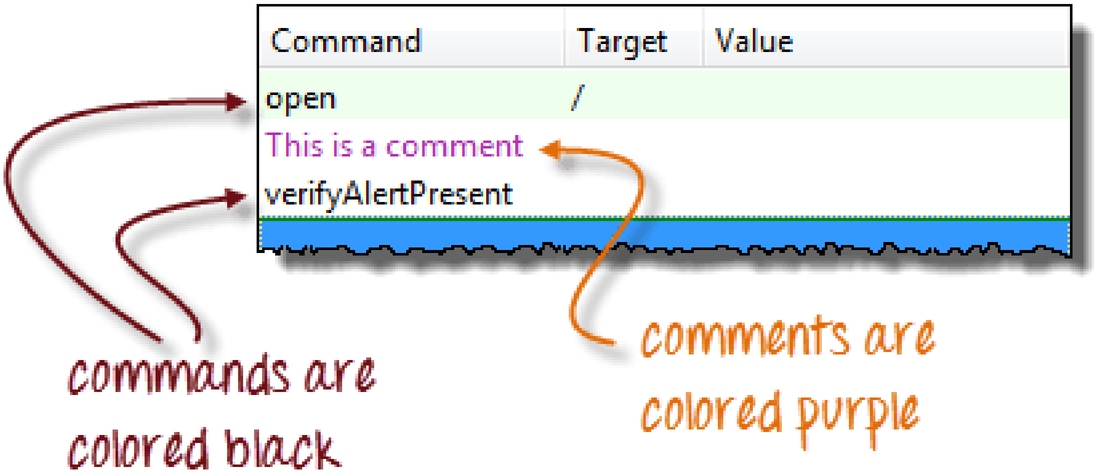
- The newly inserted command or comment **will be placed on top of the currently selected line**.



Command	Target	Value
open	/	
verifyAlertPresent		

Command	Target	Value
open	/	
verifyAlertPresent		

- **Commands** are colored **black**.
- **Comments** are colored **purple.**

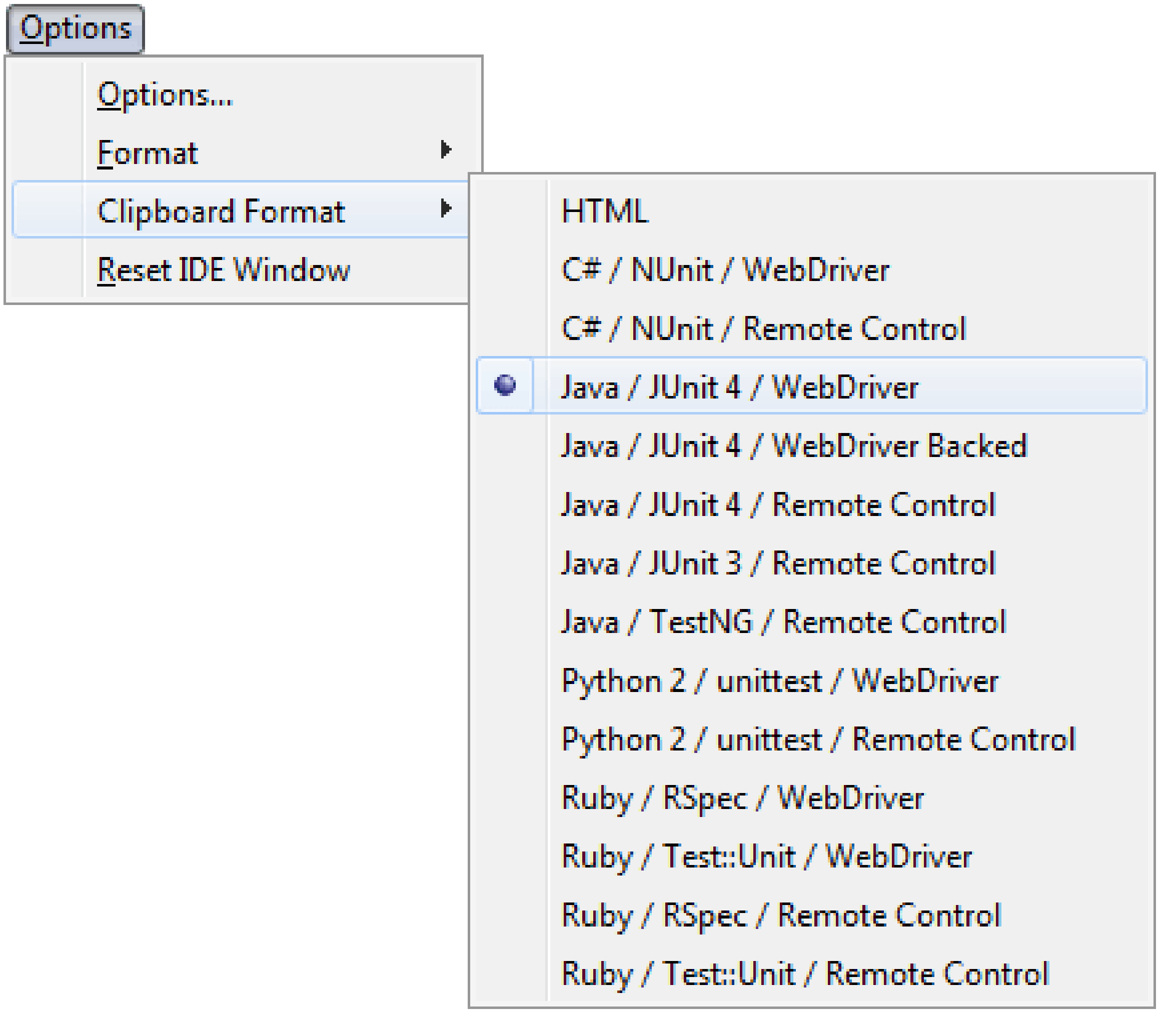


Command	Target	Value
open	/	
This is a comment		
verifyAlertPresent		

Options menu

It provides the **interface for configuring various settings** of Selenium IDE.

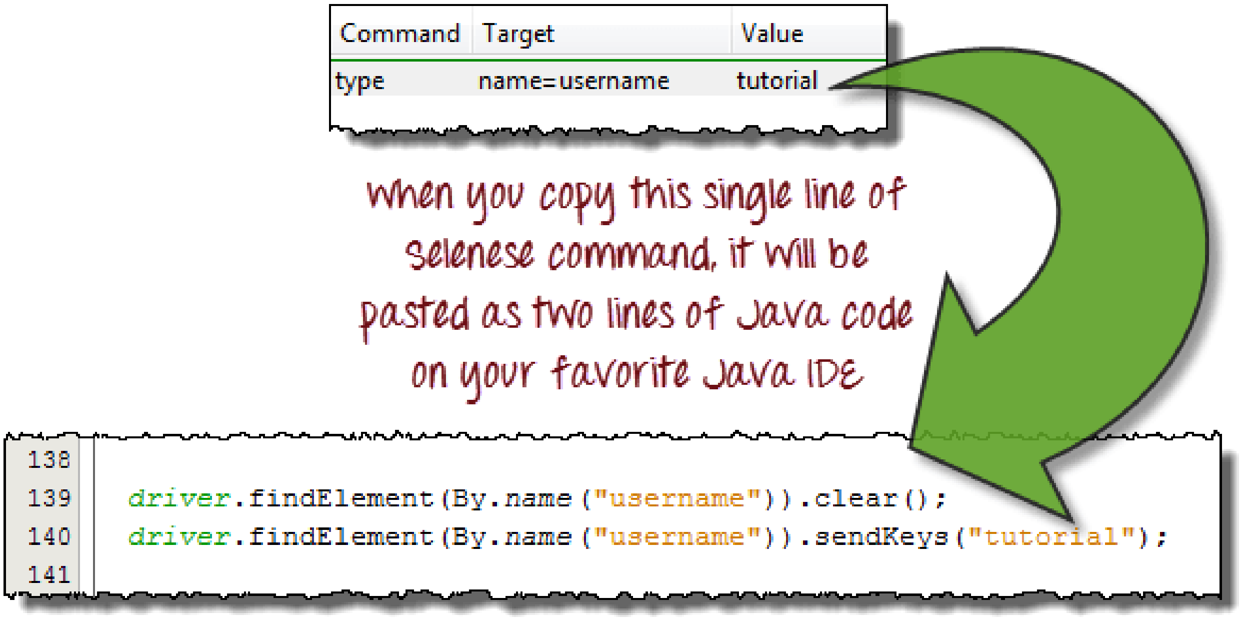
We shall concentrate on the **Options** and **Clipboard Format** options.


Options
Options...
Format
Clipboard Format
Reset IDE Window
HTML
C# / NUnit / WebDriver
C# / NUnit / Remote Control
● Java / JUnit 4 / WebDriver
Java / JUnit 4 / WebDriver Backed
Java / JUnit 4 / Remote Control
Java / JUnit 3 / Remote Control
Java / TestNG / Remote Control
Python 2 / unittest / WebDriver
Python 2 / unittest / Remote Control
Ruby / RSpec / WebDriver
Ruby / Test::Unit / WebDriver
Ruby / RSpec / Remote Control
Ruby / Test::Unit / Remote Control

Clipboard Format

- **The Clipboard allows you to c Selenese comn from the edito paste it as a co snippet**.
- The format of t code follows th option you sele here in Clipboa Format's list.
- **HTML is the de selection.**

For example, when you choose **Java/JUnit 4/WebDriver** as your clipboard format, every Selenese command you copy from Selenium IDE's editor will be pasted as **Java code**. See the illustration below.



Command	Target	Value
type	name=username	tutorial

when you copy this single line of
selenese command, it will be
pasted as two lines of Java code
on your favorite Java IDE

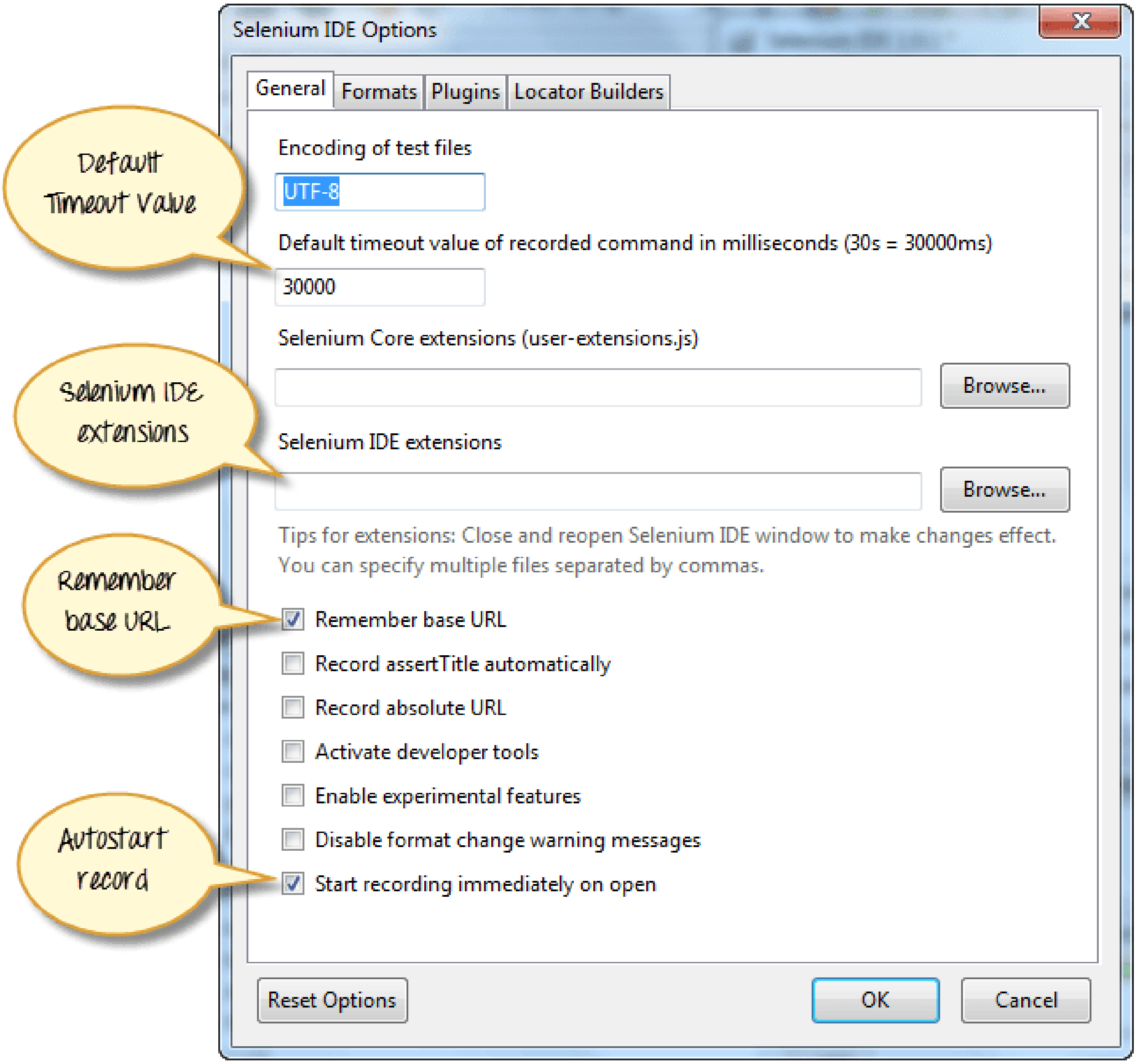
```java

driver.findElement(By.name("username")).clear();
driver.findElement(By.name("username")).sendKeys("tutorial");

```

Selenium IDE Options dialog box

You can launch the Selenium IDE Options dialog box by clicking Options > Options... on the menu bar. Though there are many settings available, we will concentrate on the few important ones.



Encoding of test files

UTF-8

Default timeout value of recorded command in milliseconds (30s = 30000ms)

30000

Selenium Core extensions (user-extensions.js)

Selenium IDE extensions

Tips for extensions: Close and reopen Selenium IDE window to make changes effect. You can specify multiple files separated by commas.

☑ Remember base URL

□ Record assertTitle automatically

□ Record absolute URL

□ Activate developer tools

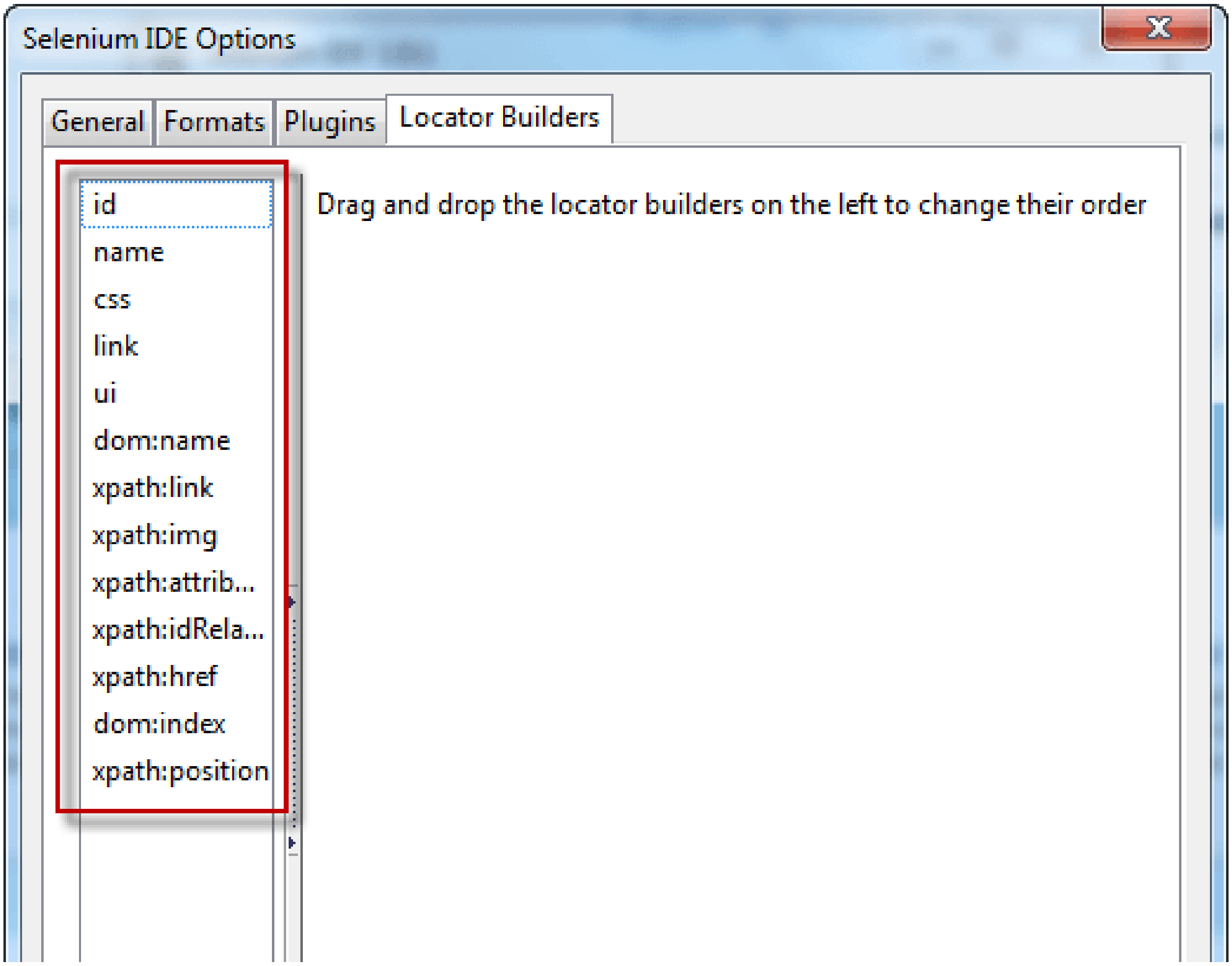
□ Enable experimental features

□ Disable format change warning messages

☑ Start recording immediately on open

- **Default Timeout Value**. This refers to the time that Selenium has to wait for a certain element to appear or become accessible before it generates an error. **Default timeout value is 30000ms**.
- **Selenium IDE extensions**. This is where you specify the extensions you want to use to extend Selenium IDE's capabilities. You can visit http://addons.mozilla.org/en-US/firefox/and use "Selenium" as a keyword to search for the specific extensions.
- **Remember base URL.** Keep this checked if you want Selenium IDE to remember the Base URL every time you launch it. If you uncheck this, Selenium IDE will always launch with a blank value for the Base URL.

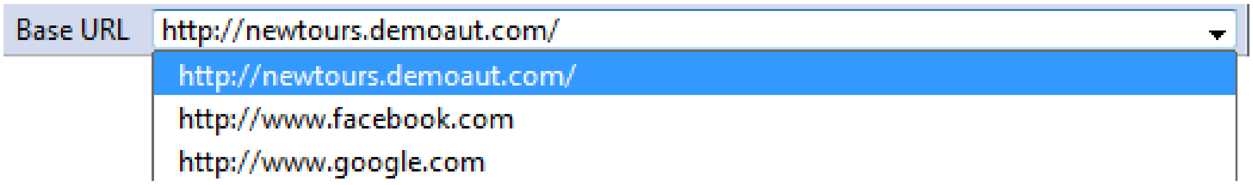
- **Autostart record.** If you check this, Selenium IDE will immediately record your browser actions upon startup.
- **Locator builders.** This is where you specify the order by which locators are generated while recording. **Locators are ways to tell Selenium IDE which UI element should a Selenese command act upon**. In the setup below, when you click on an element with an ID attribute, that element's ID will be used as the locator since "id" is the first one in the list. If that element does not have an ID attribute, Selenium will next look for the "name" attribute since it is second in the list. The list goes on and on until an appropriate one is found.



id
name
css
link
ui
dom:name
xpath:link
xpath:img
xpath:attrib...
xpath:idRela...
xpath:href
dom:index
xpath:position

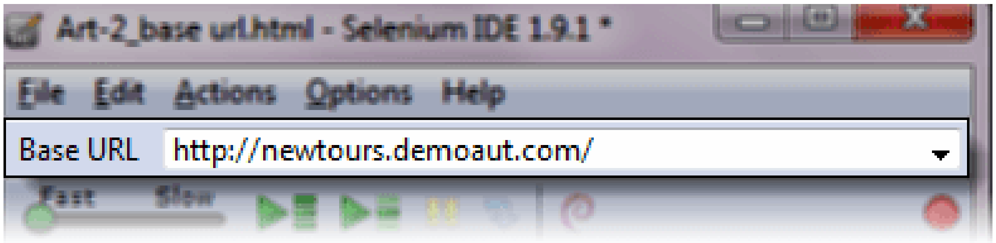
Drag and drop the locator builders on the left to change their order

Base URL Bar

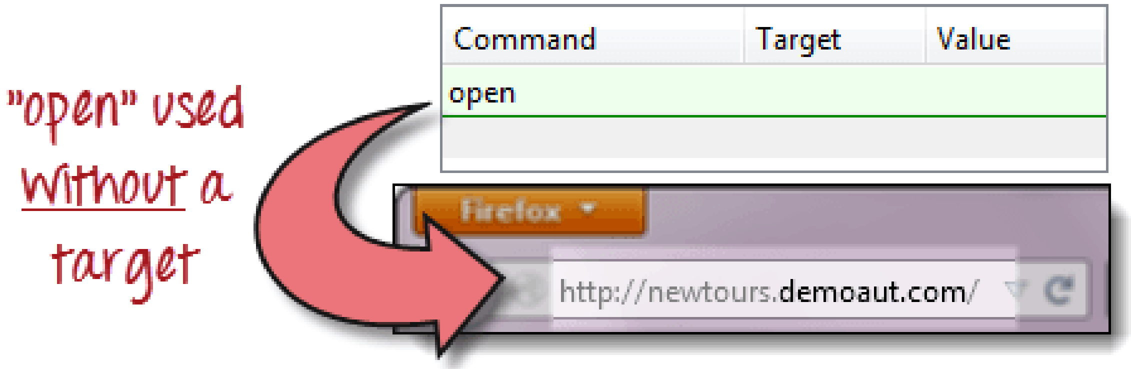


Base URL http://newtours.demoaut.com/

- It has **a dropdown menu that remembers all previous values** for easy access.
- The Selenese command **"open" will take you to the URL that you specified in the Base URL**.
- In this tutorial series, we will be using http://newtours.demoaut.com as our Base URL. It is the site for Mercury Tours, a web application maintained by HP for web Testing purposes. We shall be using this application because it contains a complete set of elements that we need for the succeeding topics.
- **The Base URL is very useful in accessing relative URLs**. Suppose that your Base URL is set to http://newtours.demoaut.com. When you execute the command "open" with the target value "signup," Selenium IDE will direct the browser to the sign-up page. See the illustration below.

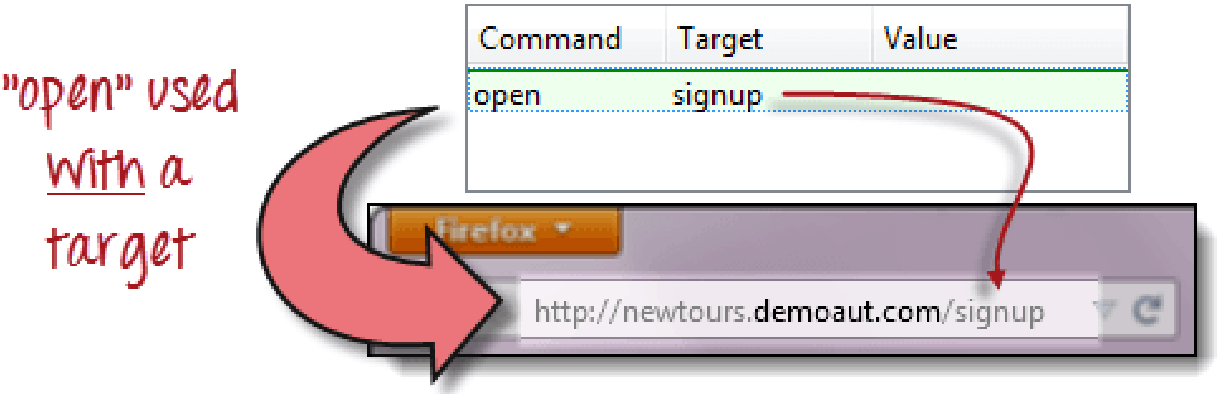


Base URL http://newtours.demoaut.com/



Command	Target	Value
open		

http://newtours.demoaut.com/



Command	Target	Value
open	signup	

http://newtours.demoaut.com/signup

Toolbar

Playback Speed. This controls the speed of your Test Script Execution.

Record. This starts/ends your recording session. Each browser action is entered as a Seler command in the Editor.

Play entire test suite. This will sequentially play all the test cases listed in the Test Case Pa

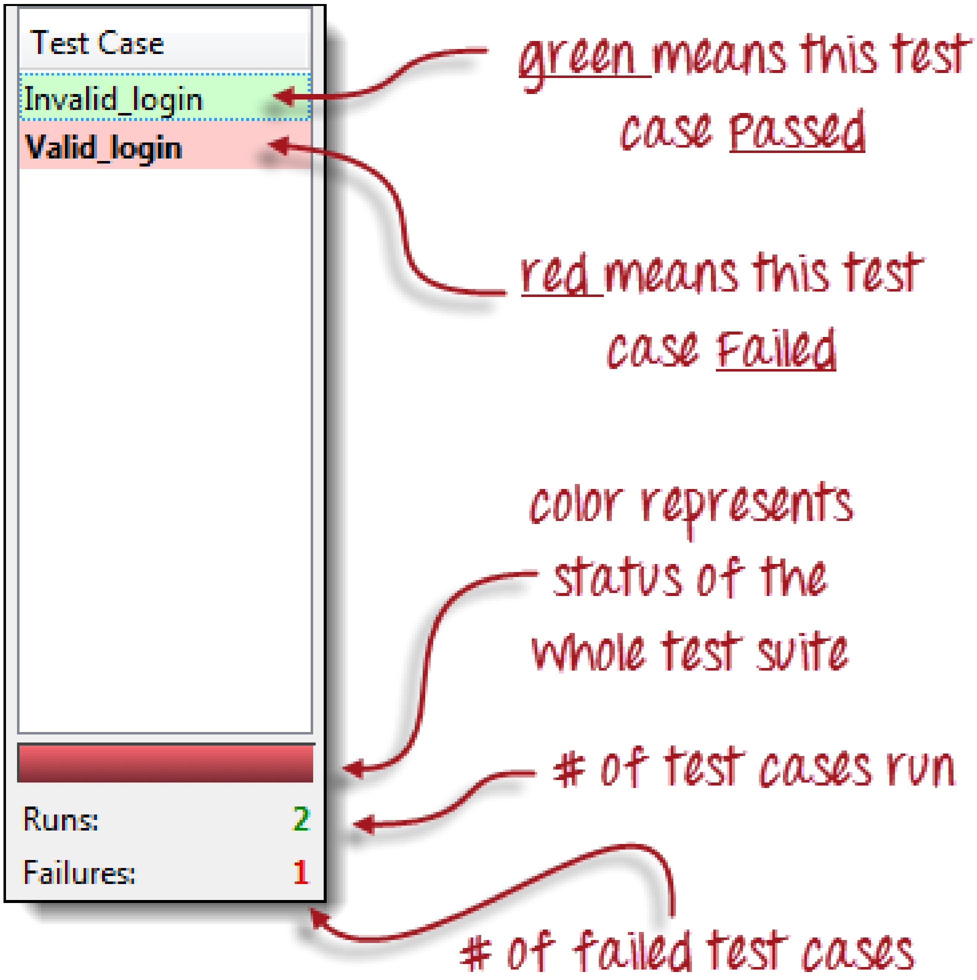
Play current test case. This will play only the currently selected test case in the Test Case I

Pause/Resume. This will pause or resume your playback.

Step. This button will allow you to step into each command in your test script.

Apply rollup rules. This is an advanced functionality. It allows you to group Selenese comm
together and execute them as a single action.

Test Case Pane



Test Case

Invalid_login

Valid_login

Runs: 2

Failures: 1

green means this test case passed

red means this test case Failed

color represents status of the whole test suite

# of test cases run

# of failed test cases

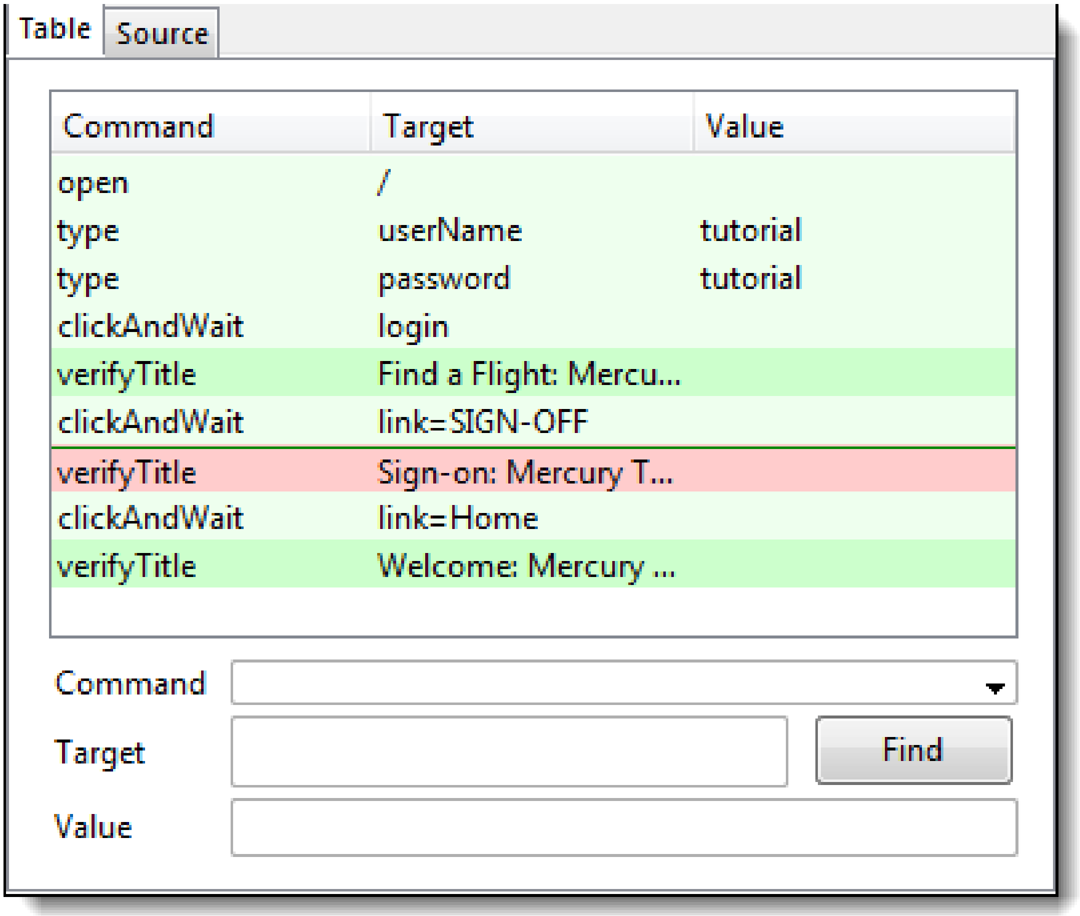
- In Selenium IDE, you can op
than one test case at a tim
- **The test case pane shows y
of currently opened test ca**
- When you open a test suite
case pane will **automatical
the test cases**contained in
- The test case written in **bol**
the **currently selected test**
- After playback, **each test ca
coded**to represent if it pass
failed.
    - Green color means
    - Red color means "F
- At the bottom portion is a s
the number of test cases th
run and failed.

Editor

You can think of the editor as **the place where all the action happens**. It is available in two views: Table and Source.

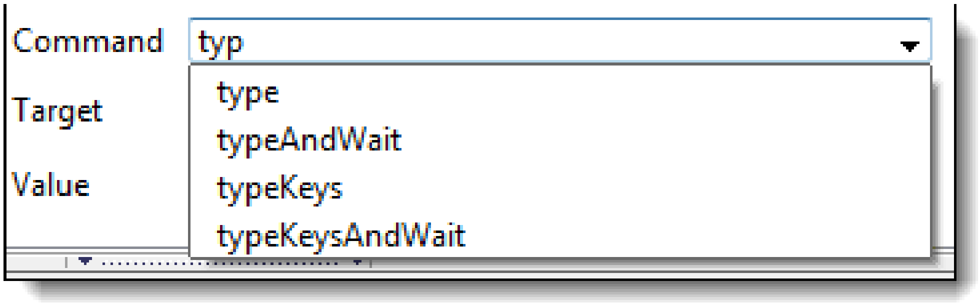
Table View

- Most of the time, you will work on Selenium IDE using the **Table View**.
- This is **where you create and modify Selenese commands.**
- After playback, each step is color-coded.



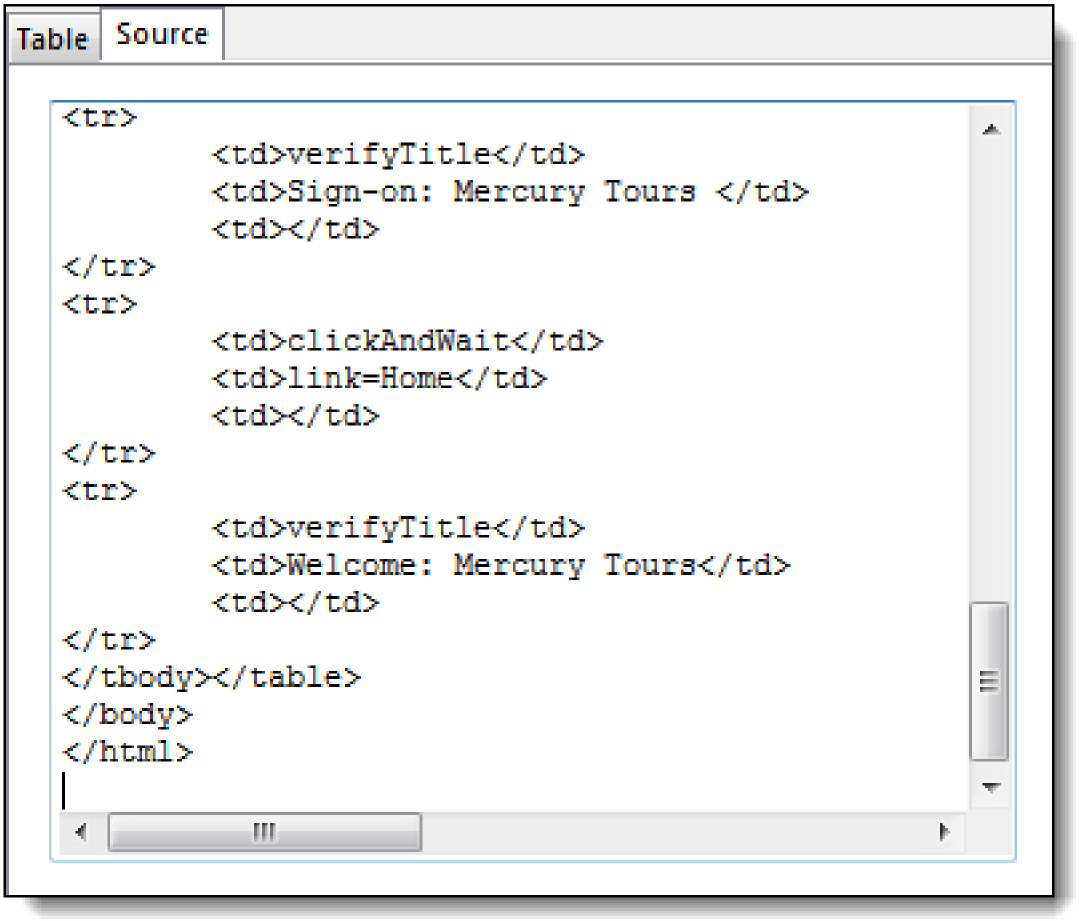
Command	Target	Value
open	/	
type	userName	tutorial
type	password	tutorial
clickAndWait	login	
verifyTitle	Find a Flight: Mercu...	
clickAndWait	link=SIGN-OFF	
verifyTitle	Sign-on: Mercury T...	
clickAndWait	link=Home	
verifyTitle	Welcome: Mercury ...	

- To create steps, type the name of the command in the "Command" text box.
- **It displays a dropdown list of commands** that match with the entry that you are currently typing.
- Target is any parameter (like username, password) for a command and Value is the input value (like tom, 123pass) for those Targets.


Command: typ

Source View

- It displays the steps in HTML (default) format.
- It also allows you to edit your script just like in the Table View.



```
<tr>
        <td>verifyTitle</td>
        <td>Sign-on: Mercury Tours </td>
        <td></td>
</tr>
<tr>
        <td>clickAndWait</td>
        <td>link=Home</td>
        <td></td>
</tr>
<tr>
        <td>verifyTitle</td>
        <td>Welcome: Mercury Tours</td>
        <td></td>
</tr>
</tbody></table>
</body>
</html>
```

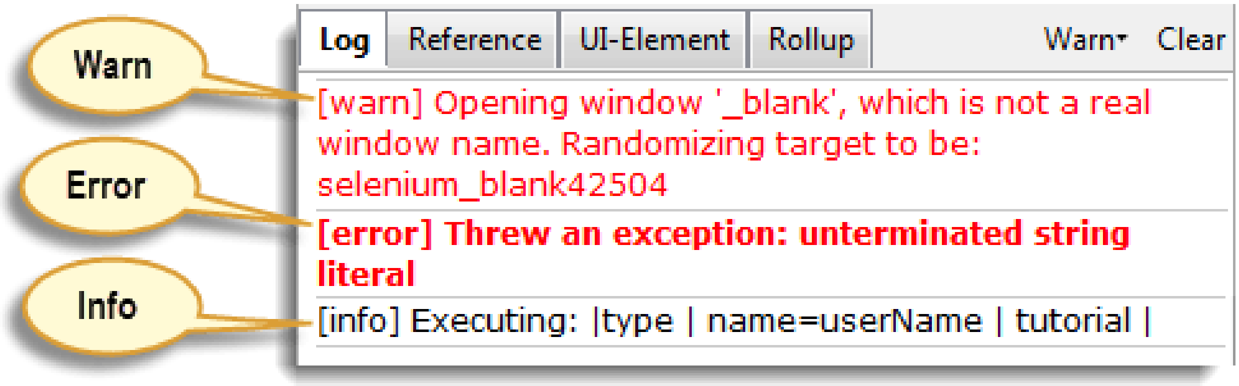
Log Pane

The Log Pane displays runtime messages during execution. It provides real-time updates as to what Selenium IDE is doing.

Logs are categorized into four types:

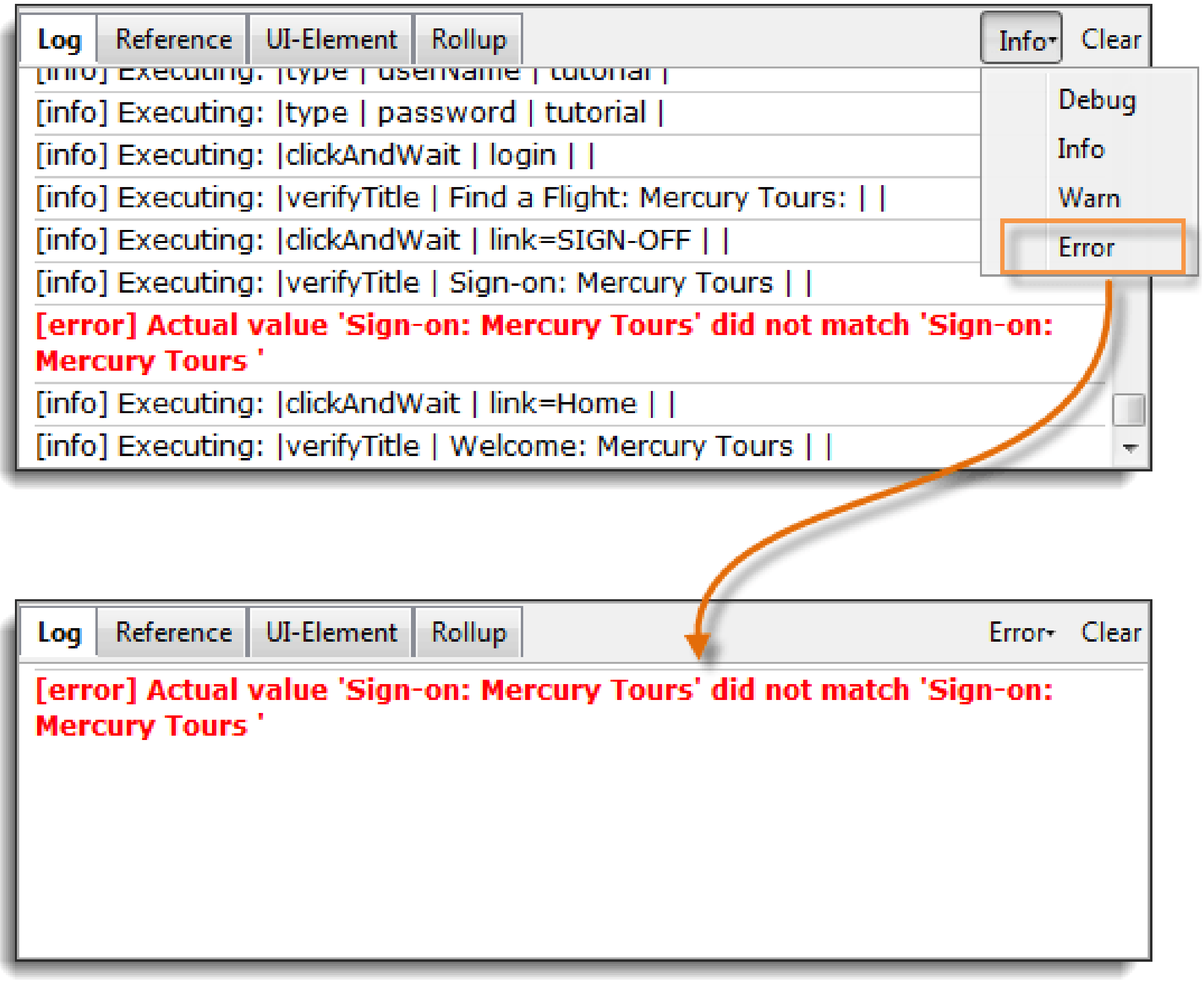
- Debug - By default, Debug messages are not displayed in the log panel. They show up only when you filter them. They provide technical information about what Selenium IDE is doing behind the scenes. It may display messages such as a specific module has done loading, a certain function is called, or an external JavaScript file was loaded as an extension.
- Info - It says which command Selenium IDE is currently executing.

- Warn - These are warning messages that are encountered in special situations.
- Error - These are error messages generated when Selenium IDE fails to execute a command, or if a condition specified by "verify" or "assert" command is not met.



```
[warn] Opening window '_blank', which is not a real window name. Randomizing target to be: selenium_blank42504
[error] Threw an exception: unterminated string literal
[info] Executing: |type | name=userName | tutorial |
```

Logs can be filtered by type. For example, if you choose to select the "Error" option from the dropdown list, the Log Pane will show error messages only.

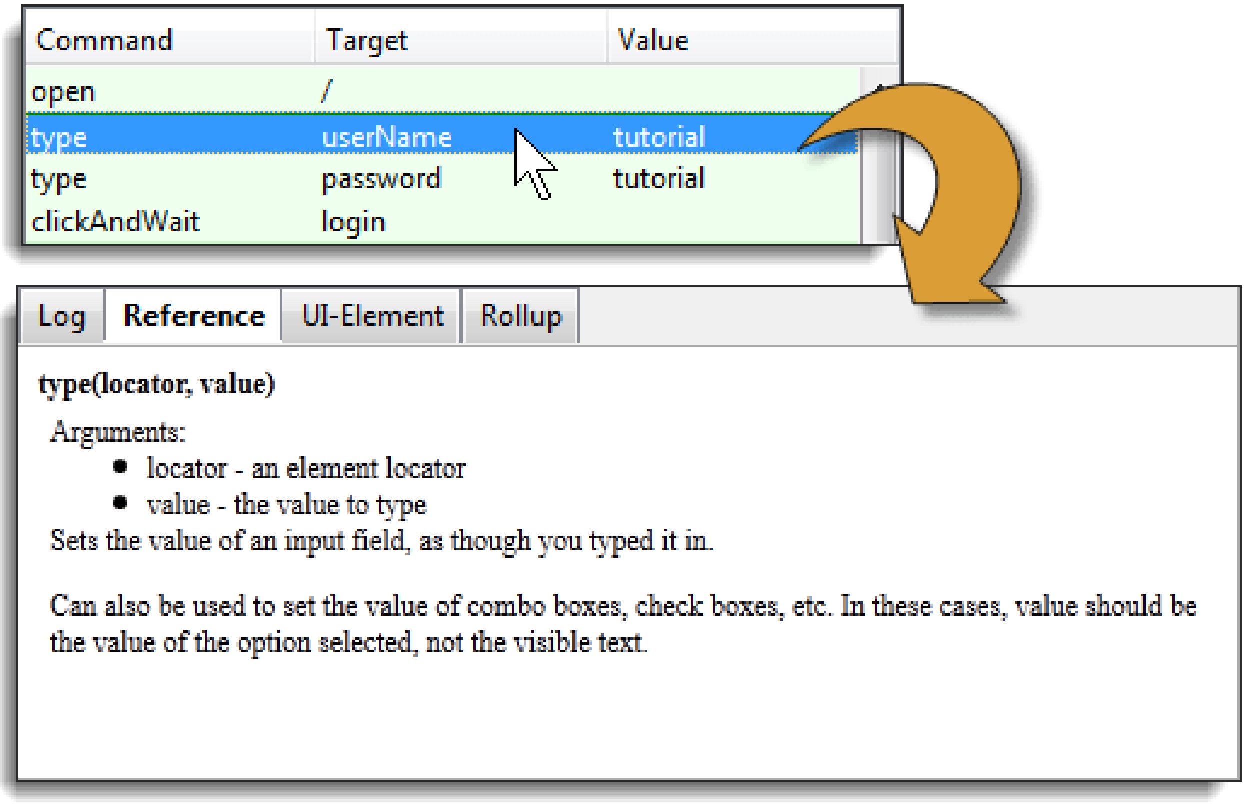


```
[info] Executing: |type | userName | tutorial |
[info] Executing: |type | password | tutorial |
[info] Executing: |clickAndWait | login | |
[info] Executing: |verifyTitle | Find a Flight: Mercury Tours: | |
[info] Executing: |clickAndWait | link=SIGN-OFF | |
[info] Executing: |verifyTitle | Sign-on: Mercury Tours | |
[error] Actual value 'Sign-on: Mercury Tours' did not match 'Sign-on:
Mercury Tours '
[info] Executing: |clickAndWait | link=Home | |
[info] Executing: |verifyTitle | Welcome: Mercury Tours | |
```

```
[error] Actual value 'Sign-on: Mercury Tours' did not match 'Sign-on:
Mercury Tours '
```

Reference Pane

The Reference Pane shows a concise description of the currently selected Selenese command in the Editor. It also shows the **description about the locator and value** to be used on that command.



Command	Target	Value
open	/	
type	userName	tutorial
type	password	tutorial
clickAndWait	login	

type(locator, value)

Arguments:

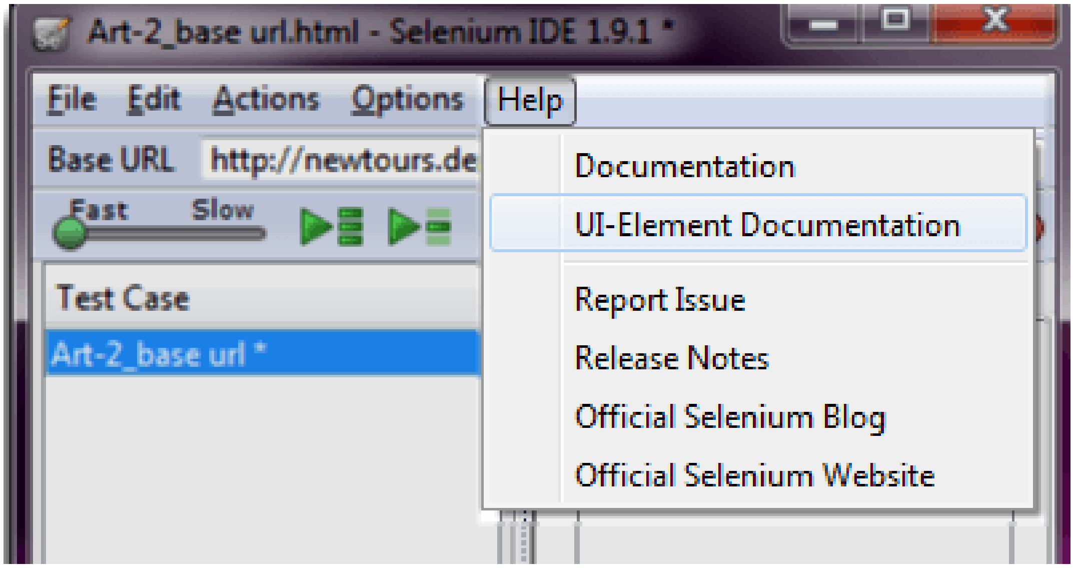
- locator - an element locator
- value - the value to type

Sets the value of an input field, as though you typed it in.

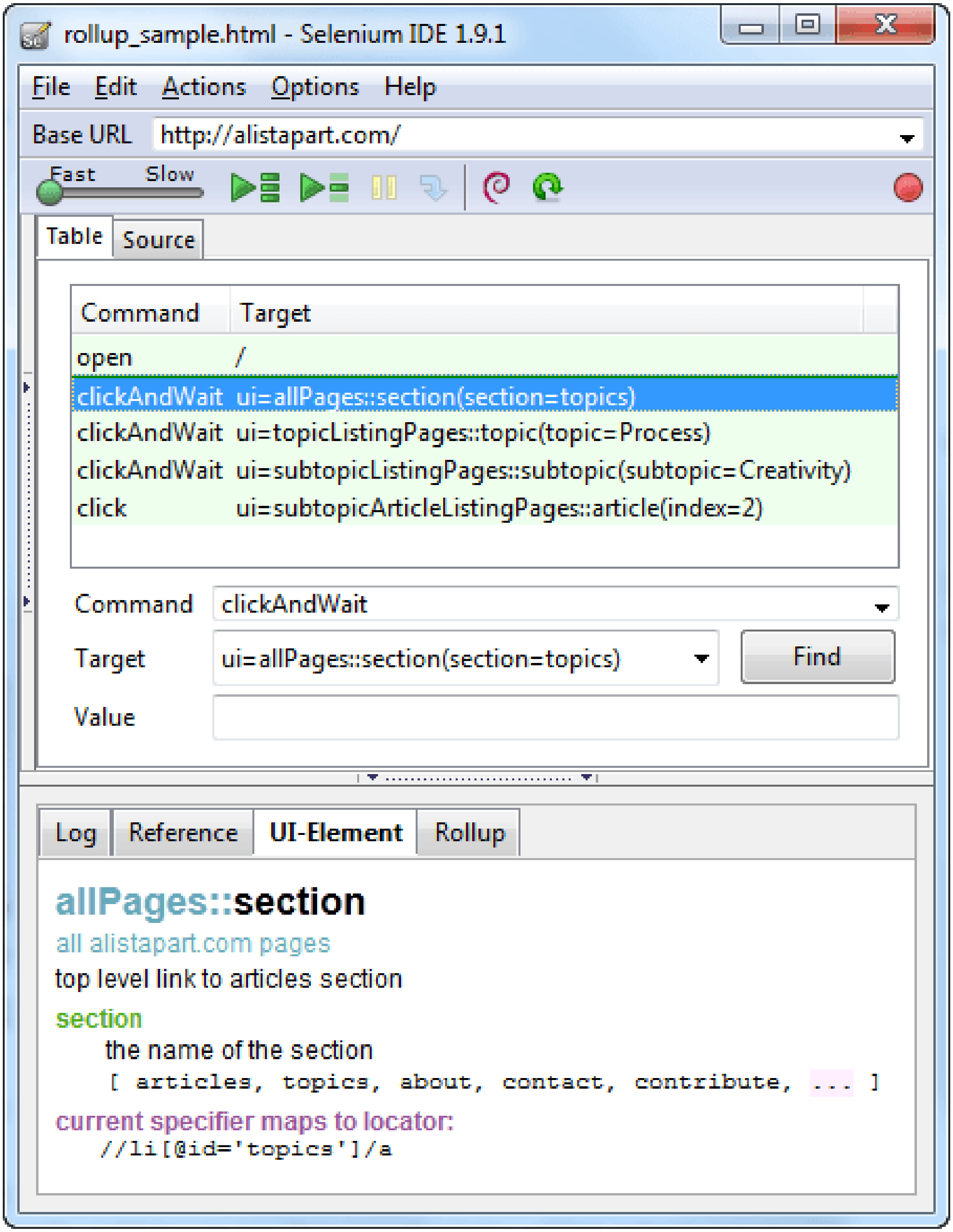
Can also be used to set the value of combo boxes, check boxes, etc. In these cases, value should be the value of the option selected, not the visible text.

UI-Element Pane

The UI-Element is for advanced Selenium users. **It uses JavaScript Object Notation (JSON) to define element mappings.** The documentation and resources are found in the "UI Element Documentation" option under the Help menu of Selenium IDE.



An example of a UI-element screen is shown below.



Base URL http://alistapart.com/

Command	Target
open	/
clickAndWait	ui=allPages::section(section=topics)
clickAndWait	ui=topicListingPages::topic(topic=Process)
clickAndWait	ui=subtopicListingPages::subtopic(subtopic=Creativity)
click	ui=subtopicArticleListingPages::article(index=2)

Command clickAndWait

Target ui=allPages::section(section=topics)

Value

allPages::section

all alistapart.com pages

top level link to articles section

section

the name of the section

```
[ articles, topics, about, contact, contribute, ... ]
```

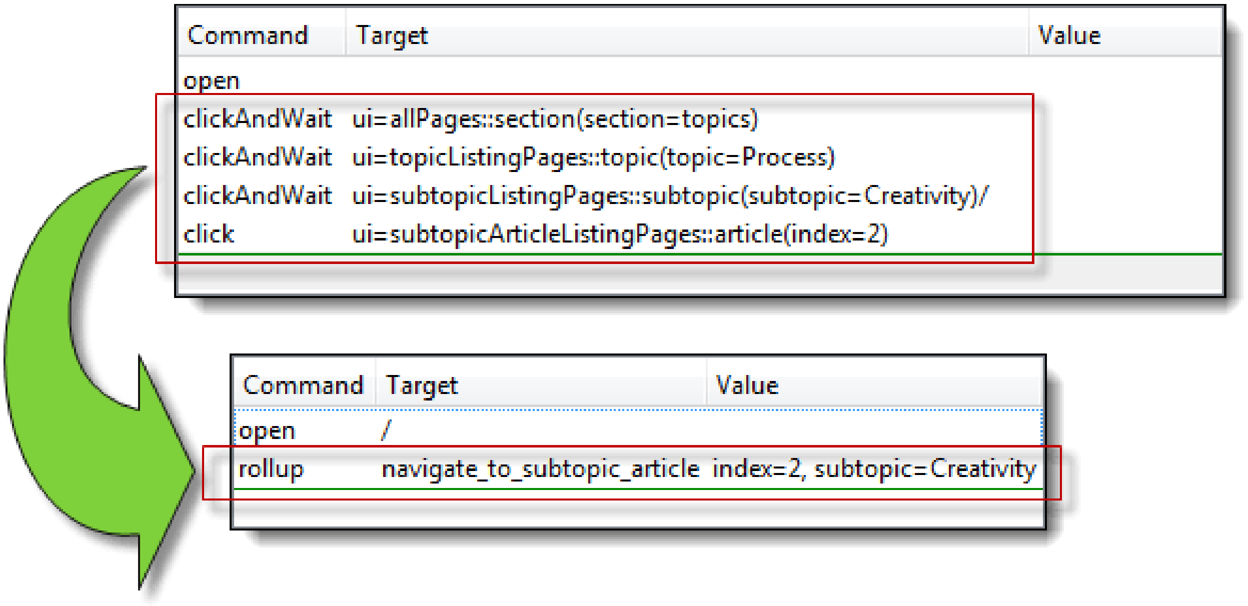
current specifier maps to locator:

```
//li[@id='topics']/a
```

Rollup Pane

Rollup allows you to execute a group of commands in one step. A group of commands is simply called as a "rollup." It employs heavy use of JavaScript and UI-Element concepts to formulate a collection of commands that is similar to a "function" in programming languages.

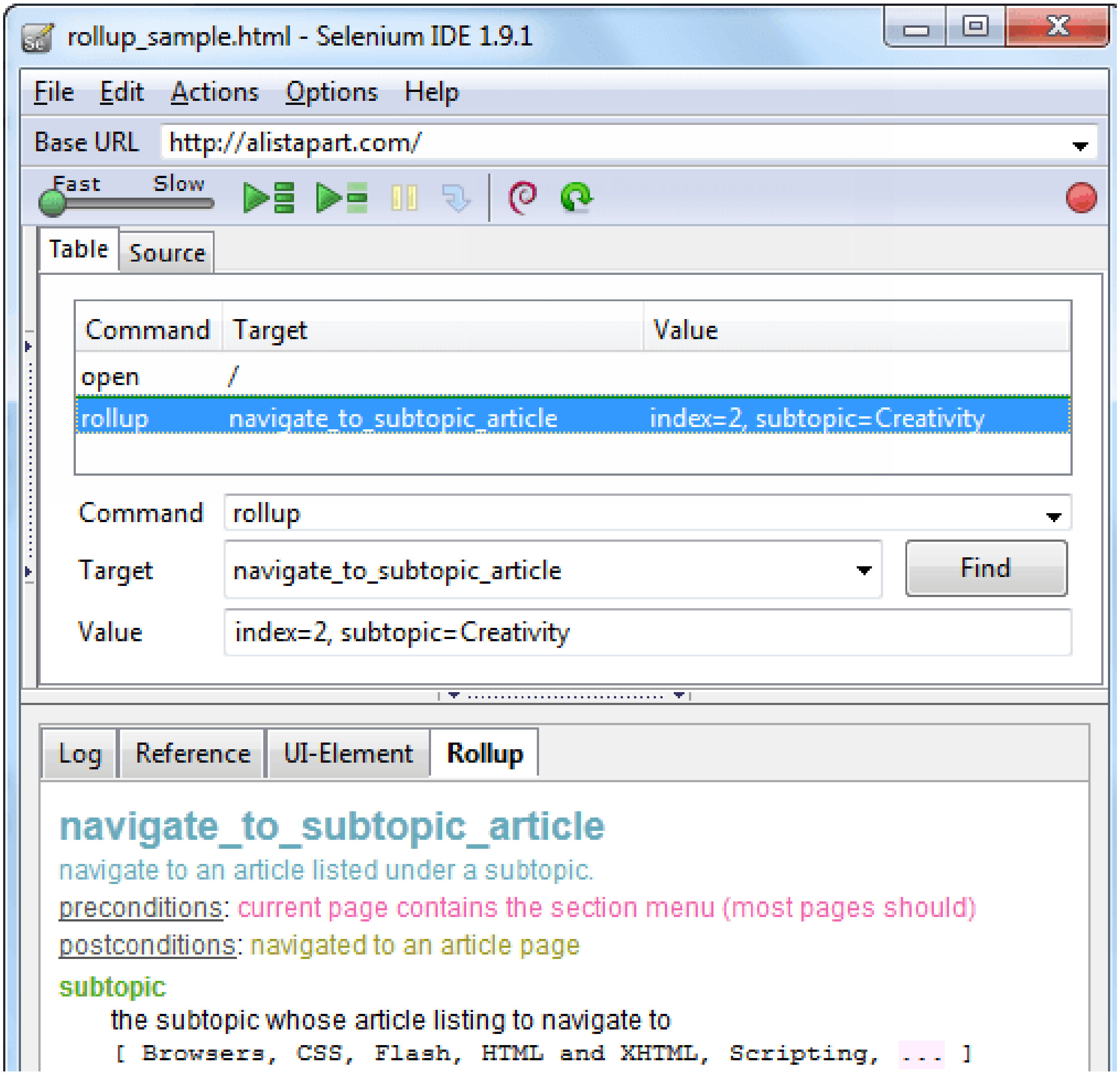
Rollups are reusable; meaning, they can be used multiple times within the test case. Since rollups are groups of commands condensed into one, they contribute a lot in shortening your test script.



Command	Target	Value
open		
clickAndWait	ui=allPages::section(section=topics)	
clickAndWait	ui=topicListingPages::topic(topic=Process)	
clickAndWait	ui=subtopicListingPages::subtopic(subtopic=Creativity)/	
click	ui=subtopicArticleListingPages::article(index=2)	

Command	Target	Value
open	/	
rollup	navigate_to_subtopic_article	index=2, subtopic=Creativity

An example of how the contents of the rollup tab look like is shown below.



Base URL http://alistapart.com/

Command	Target	Value
open	/	
rollup	navigate_to_subtopic_article	index=2, subtopic=Creativity

Command rollup

Target navigate_to_subtopic_article

Value index=2, subtopic=Creativity

navigate_to_subtopic_article

navigate to an article listed under a subtopic.

preconditions: current page contains the section menu (most pages should)

postconditions: navigated to an article page

subtopic

the subtopic whose article listing to navigate to

```
[ Browsers, CSS, Flash, HTML and XHTML, Scripting, ... ]
```

Summary

- Selenium IDE (Integrated Development Environment) **is the simplest tool** in the Selenium Suite.
- It must only be used as a **prototyping tool**.
- **Knowledge of JavaScript and HTML is required for intermediate topics** such as executing the "runScript" and "rollup" commands.A **rollup** is a collection of commands that you can reuse to shorten your test scripts

significantly. **Locators** are identifiers that tell Selenium IDE how to access an element.

- **Firebug** (or any similar add-on) is used to obtain locator values**.**
- The **menu bar** is used in creating, modifying, and exporting test cases into formats useable by Selenium RC and WebDriver.
- The **default format for Selenese commands is HTML**.
- The **"Options" menu provides access to various configurations** for Selenium IDE.
- The **Base URL** is useful in accessing **relative URLs**.

- The **Test Case Pane** shows the list of currently opened test cases and a concise summary of test runs.
- The **Editor** provides the **interface for your test scripts**.
- The **Table View** shows your script **in tabular format** with "Command", "Target", and "Value" as the columns.
- The **Source View** shows your script **in HTML format**.
- The **Log** and **Reference** tabs give feedback and other useful information when executing tests.
- The **UI-Element and Rollup** tabs are **for advanced Selenium IDE users only**. They both require considerable effort in coding JavaScript.
- **UI-Element** allows you to **conveniently map UI elements** using JavaScript Object Notation (JSON).

The following table summarizes the release history for the Selenium IDE.

Major version	**Release date**
1.0.10	06-Dec-10
1.5.0	15-Dec-11
1.8.1	01-Jun-12
2.1.0	30-Jun-13
2.2.0	06-Jul-13

2.3.0	09-Aug-13
2.5.0	02-Jan-14
2.8.0	29-Sep-14
2.9.0	09-Mar-15
2.9.1	- to be released

End For Today We Will Back Tomorrow