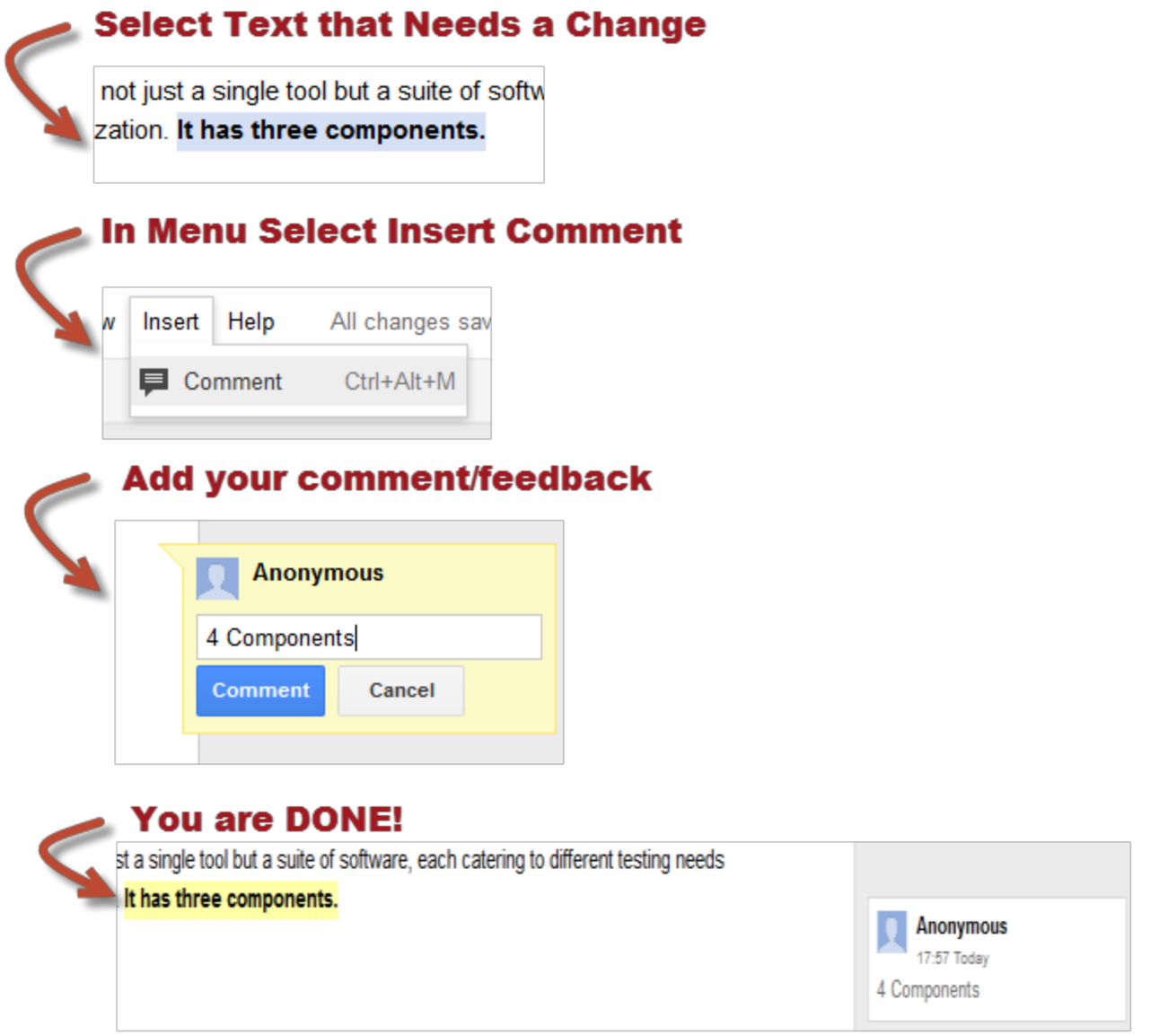
This is Article 1 of an 11 part tutorial series on Selenium. Video will be after text tutorials are finalized

We would love to hear your feedback to gauge that we are moving in the right direction with our course creation. To leave a Comment



What should I Review?

- Do you understand the contents of the tutorials? This is most important. Highlight sections you find confusing.
- Mark information that seems superfluous and is a drag to read. You do not want boring tutorials right?
- Mark Grammatical Errors & Spelling Mistakes
- Does the article follow a Logical Flow? Or are the contents organized haphazardly
- Do you feel a need of an image/snapshot to explain a particular topic better
- Is the article poorly researched? Is the data presented correct? Are any important sub-topics / data /facts left out?

==========

Article Starts Here

==========

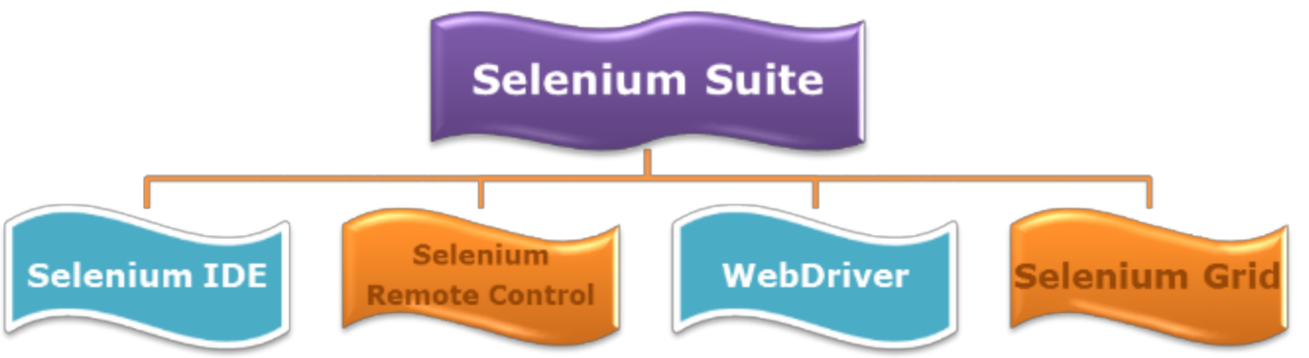
Introduction to Selenium

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) only that Selenium focuses on automating web-based applications.

Selenium is not just a single tool but a suite of software, 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 being 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 creating different features. Below are the key persons who made notable contributions to the Selenium Project.

Birth of Selenium Core

Primarily, Selenium was **created by Jason Huggins in 2004**. An engineer at ThoughWorks, he was working on a web application that required frequent testing. Having realized that the repetitious manual testing of their application

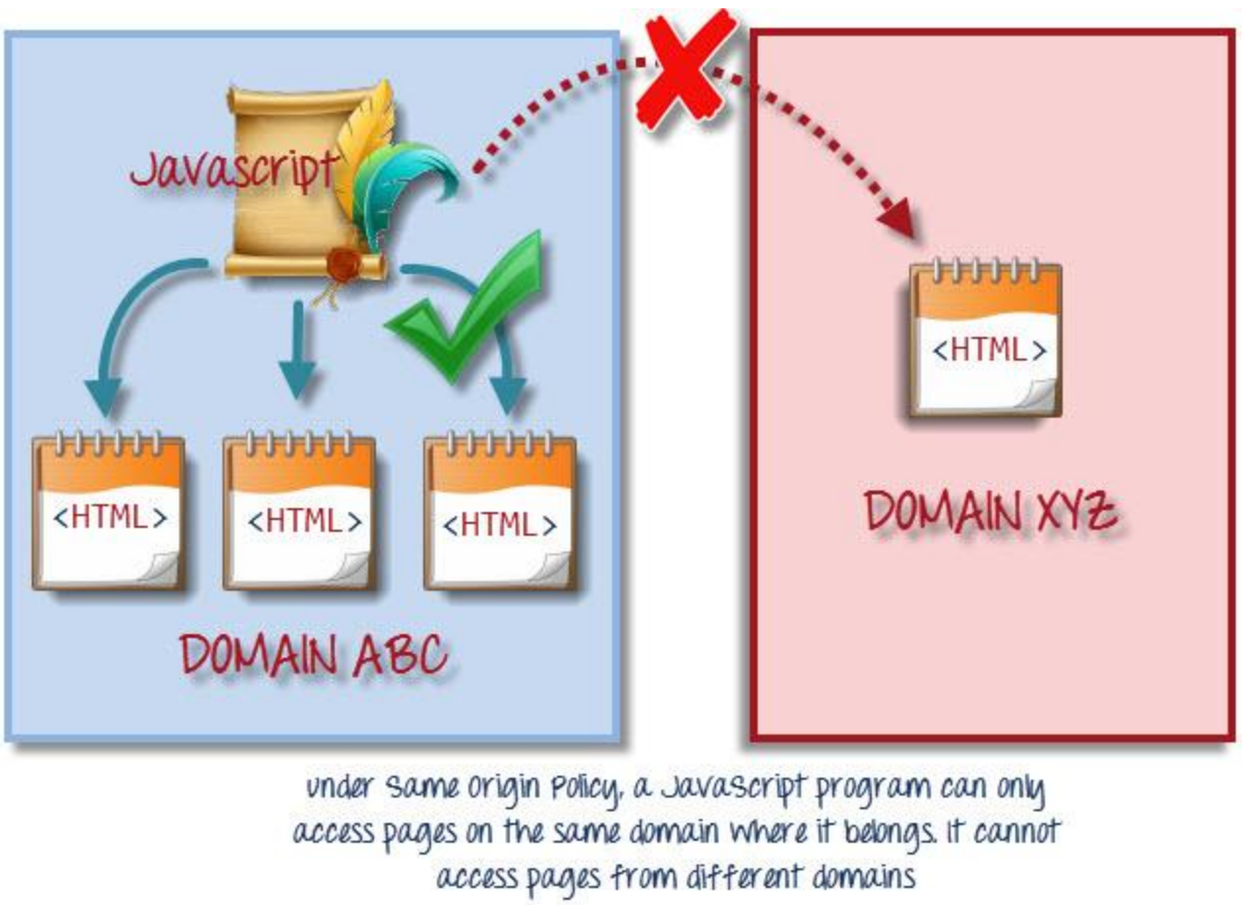
was becoming more and more inefficient, he created a JavaScript program that would automatically control the browser's actions. He named this program as the "**JavaScriptTestRunner**."

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

The Same Origin Policy

This 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



Birth of Selenium Remote Control (Selenium RC)

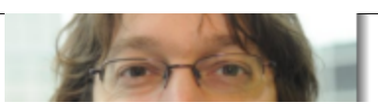
Unfortunately, testers using Selenium Core had to install the whole application under test and the web server on their own local computers because of the restrictions imposed by the **same origin policy.** And so another ThoughtWorks engineer, **Paul Hammant**, decided to create a server that will act 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 as "**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 Kasataniof 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 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 can**

	control the browser from the OS level.

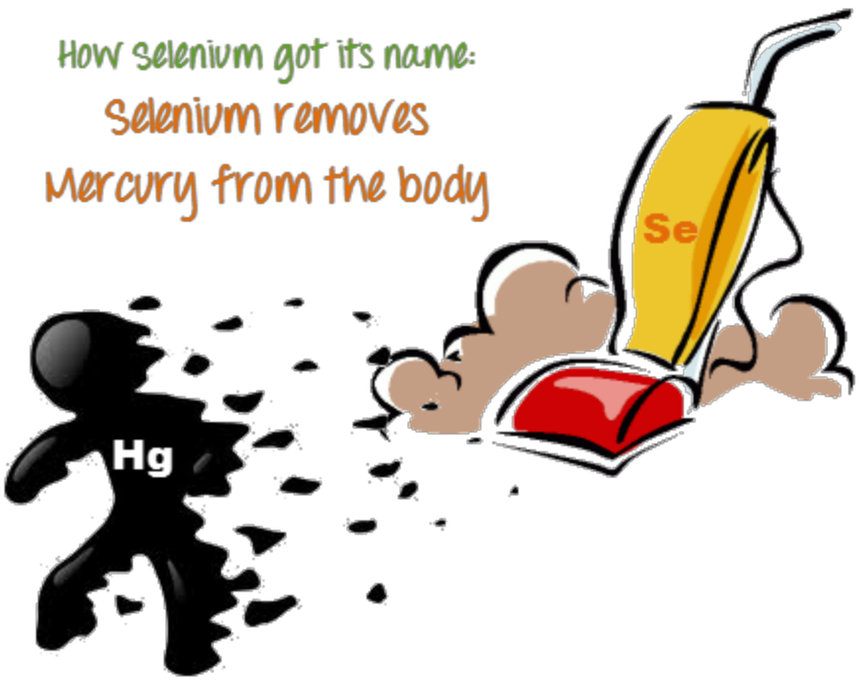
Birth of Selenium 2

In **2008**, the whole Selenium Team decided to merge WebDriver and Selenium RC to form a more powerful framework 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 focused on Selenium 2.

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.

So, Why the Name Selenium?

It came from a joke that 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, that 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 get to call this framework up to the present.



Selenium Tool Suite

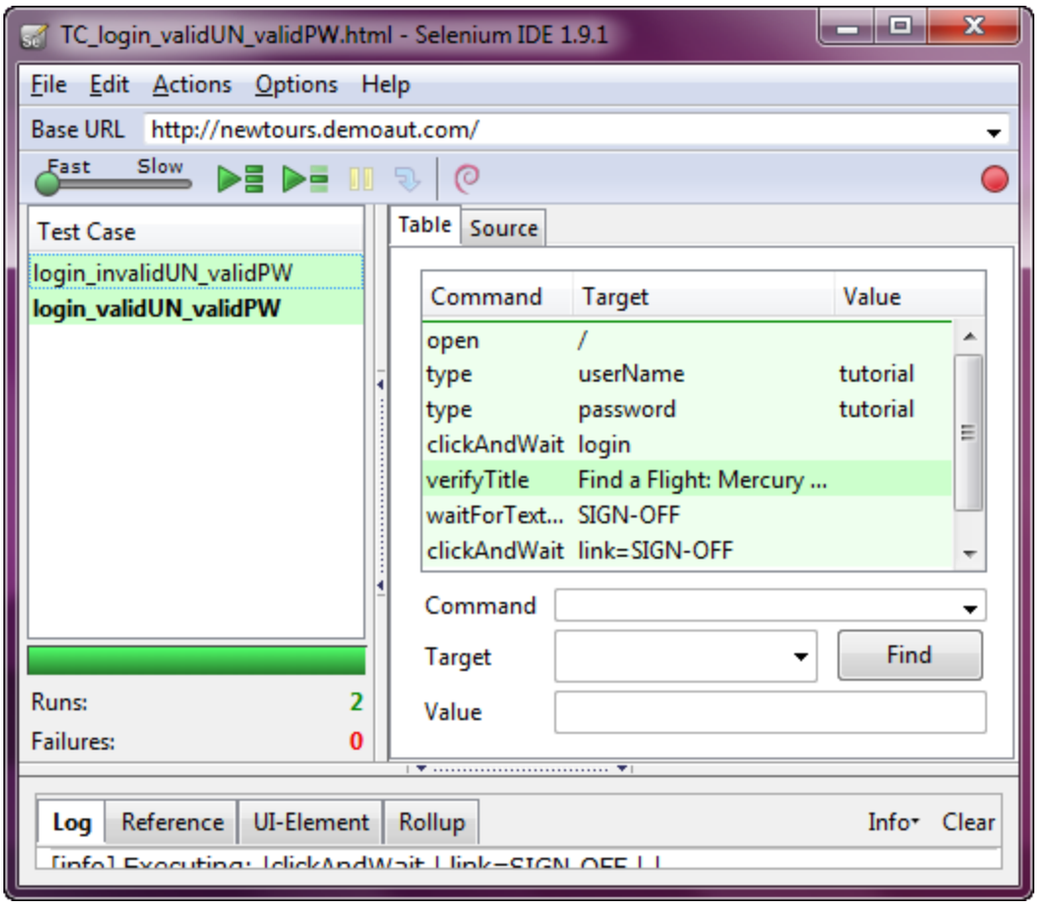
The Selenium Tool Suite is comprised of four components:

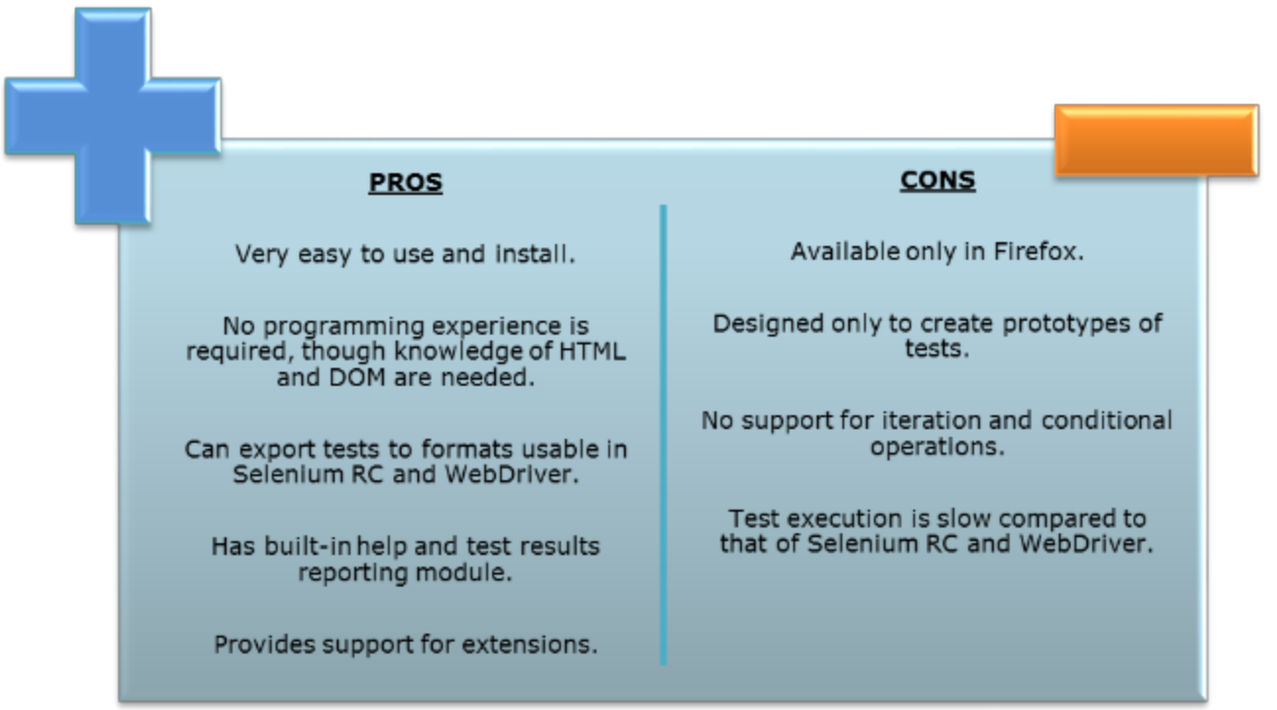
- Selenium Integrated Development Environment (IDE)
- Selenium Remote Control (RC)
- WebDriver
- Selenium Grid

A quick summary of their features are shown below.

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.



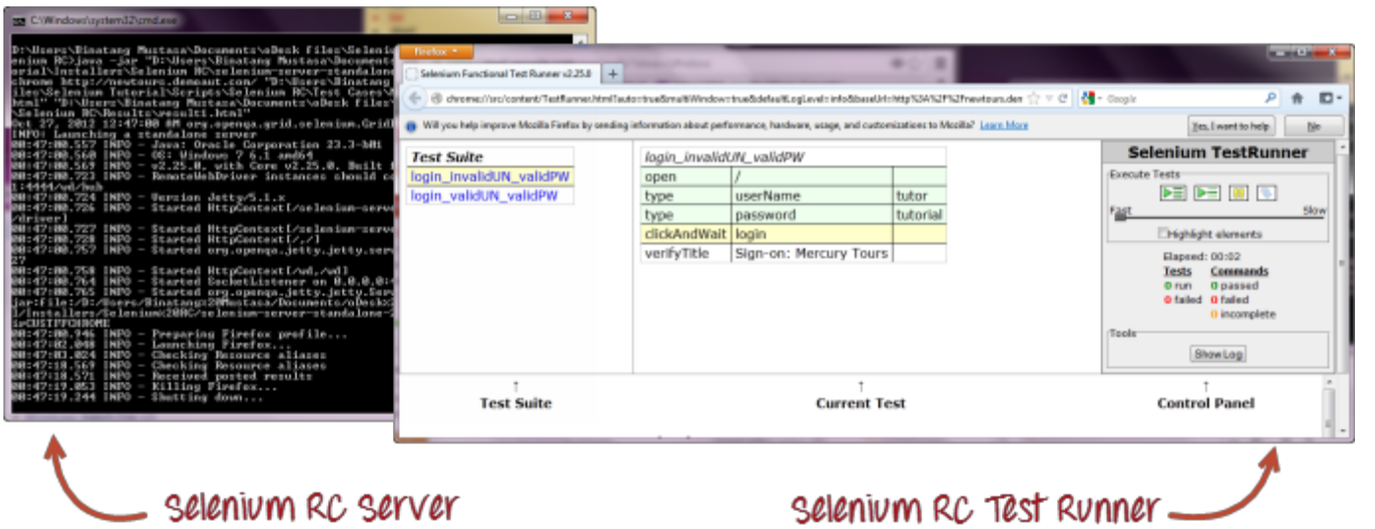


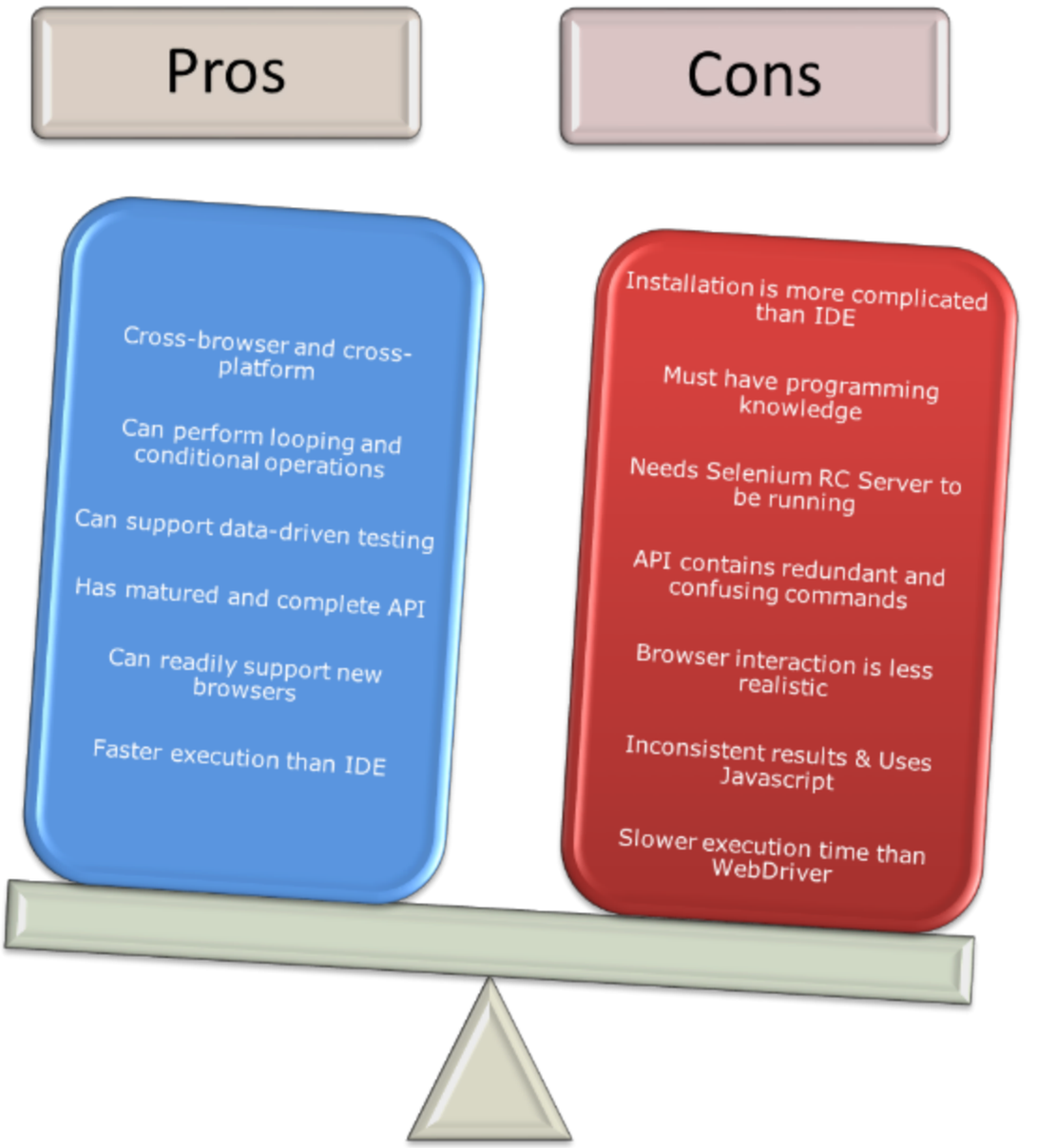
Selenium Remote Control (Selenium RC)

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

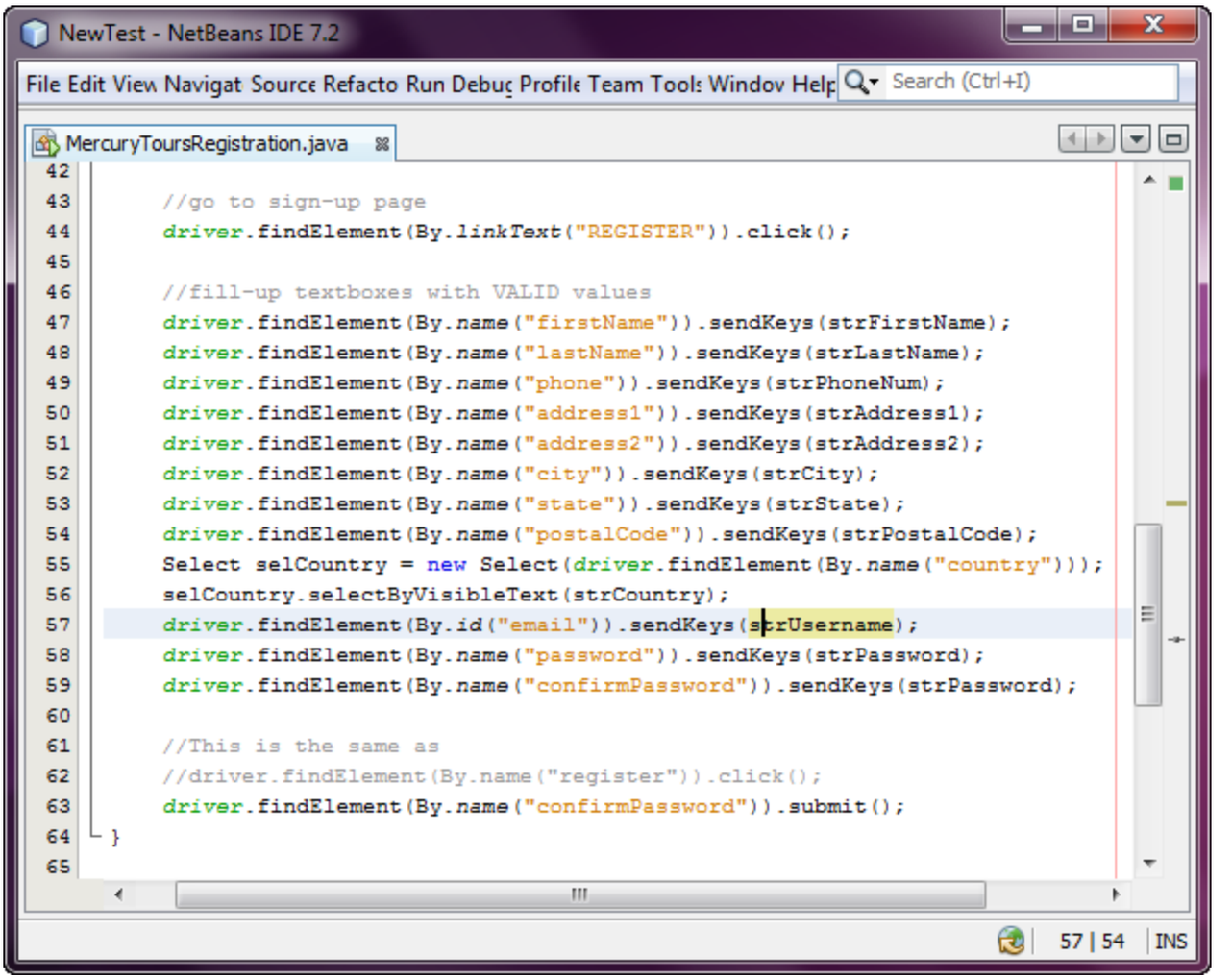
- C#
- PHP
- Python
- Perl
- Ruby





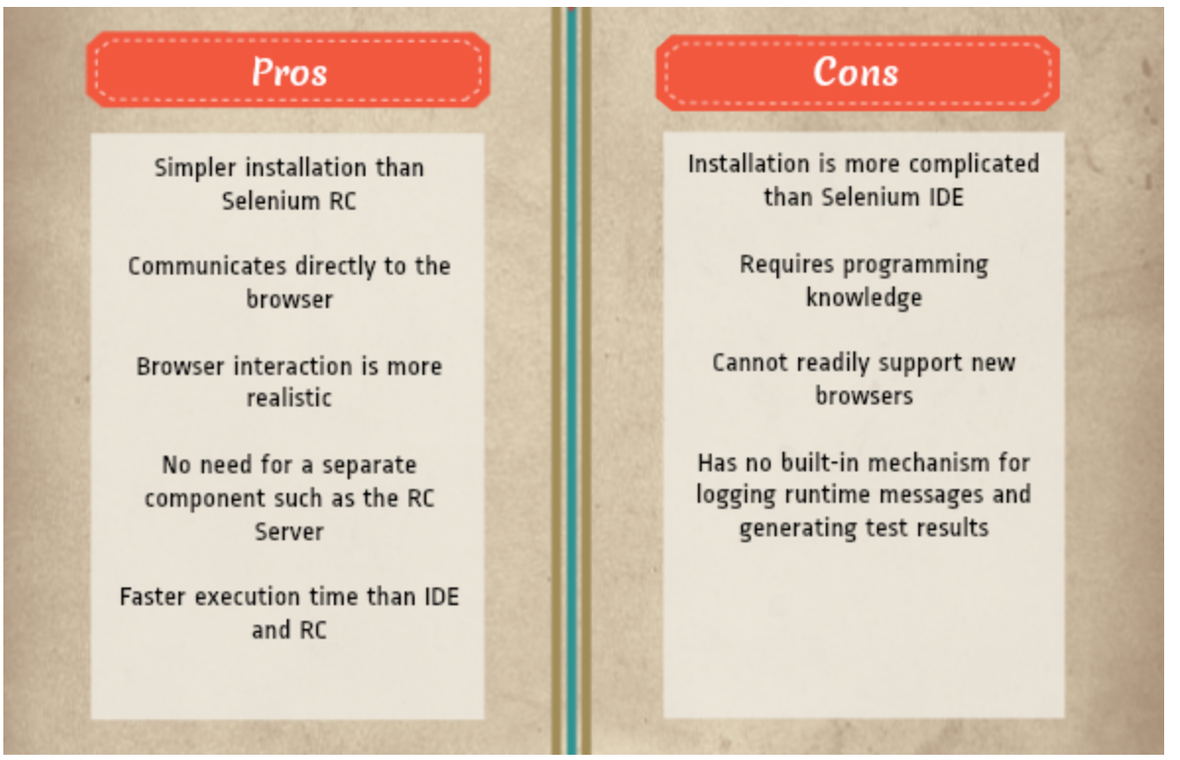
WebDriver

The WebDriver proves itself to be **better than both Selenium IDE and Selenium RC** in many respects. 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 to it.**



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

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



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.
- Utilizesthe **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	Selenium RC	WebDriver
Browser Support	Mozilla Firefox	Mozilla Firefox Internet Explorer Google Chrome Safari Opera Konqueror Others	Internet Explorer versions 6 to 9, both 32 and 64-bit Firefox 3.0, 3.5, 3.6, 4.0, 5.0, 6, 7 and above (current version is 16.0.1) Google Chrome 12.0.712.0 and above (current version is 22.0.1229.94 m) Opera 11.5 and above (current version is 12.02) 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 (current version is 2.10)
Operating System	Windows Mac OS X Linux	Windows Mac OS X Linux Solaris	All operating systems where the browsers above can run.

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. • 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, and you need to complete it the soonest time possible.

A Comparison between Selenium and QTP

Quick Test Professional(QTP)is a proprietary automated testing tool previously owned by the company **Mercury Interactive** before it was **acquired by Hewlett-Packardin2006**.The Selenium Tool Suite is similar to QTP in many respects, some of which are enumerated below:

Similarities:

- They both perform automated testing
- Both are capable of handling scalable test suites
- Selenium IDE and QTP both have record and playback feature
- They can facilitate data-driven, keyword-driven, model-based, and modularity-driven testing
- Both frameworks support JavaScript

- Selenium IDE, Selenium RC, and QTP all have built-in test reporting and event-logging capabilities
- Both are capable of capturing screenshots of the application under test.
- Both frameworks support regular expressions

On the other hand, enumerated below are their differences:

Advantages of Selenium over QTP

Selenium	QTP
• **Open source**, **free to use**, and **free of charge.**	• **Proprietary**, so it isn't free.
• **Highly extensible**	• Limited add-ons
• Can run tests across **different browsers**	• Can only run tests in **Firefox** and **Internet Explorer**.
• Supports **various operating systems**	• Can only be used in **Windows**
• Supports**mobile devices**	• Cannot be used in mobile devices.
• **Allows you to use an IDE of your choice** (NetBeans, Eclipse, IDEA, etc.)	• All test scripts can only be created **within QTP**
• 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 tests **sequentially**

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 has fewer components undergoing fewer processes.	• Automates at a slower rate because it has more components undergoing more complicated processes.
• Data-driven testing is easier to perform because **it has built-in global and local data tables**.	• Data-driven testing is more cumbersome since you have to rely on the programming language's capabilities for setting 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 application under test
• Can compare **bitmap comparison** between images	• Cannot perform bitmap comparison
• 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 onto external formats

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 only partially capable of)

Summary

- **Jason Huggins** pioneered Selenium in 2004.**Paul Hammant**led the development of Selenium Remote Control. **Patrick Lightbody** developed Selenium Grid.**Shinya Kasatani** created Selenium IDE.**Simon Stewart** developed WebDriver.
- 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 WebDriverwas 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.