**Why Selenium?**

As the current industry trends have shown that there is mass movement towards automation testing. The cluster of repetitive manual testing scenarios has raised a demand to bring in the practice of automating these manual scenarios.

**The benefits of implementing automation test are many; let us take a look at them:**

* Supports execution of repeated test cases
* Aids in testing a large test matrix
* Enables parallel execution
* Encourages unattended execution
* Improves accuracy thereby reducing human generated errors
* Saves time and money

**All this results in to the following:**

* High ROI
* Faster GoTo market

Automation testing benefits are many and well understood and largely talked about in the software test industry.

One of the most commonly asked question comes with this is –

* What is the best tool for me to get my tests automated?
* Is there a cost involved?
* Is it easy to adapt?

One of the best answers to all the above questions for automating web based applications is Selenium. Because:

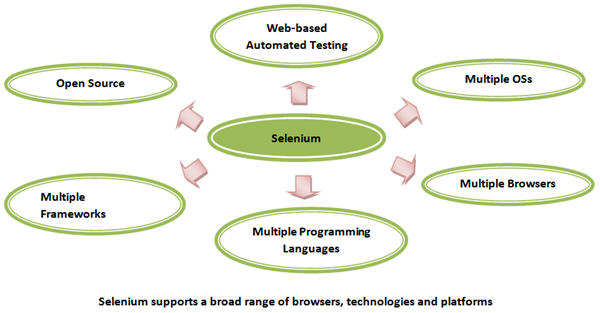
* It’s open source
* have a large user base and helping communities
* have multi browser and platform compatibility
* has active repository developments
* supports multiple language implementations

**First glance at Selenium**

Selenium is one of the most popular automated testing suites. Selenium is designed in a way to support and encourage automation testing of functional aspects of web based applications and a wide range of browsers and platforms. Due to its existence in the open source community, it has become one of the most accepted tools amongst the testing professionals.

**Selenium supports a broad range of browsers, technologies and platforms.**

***(Click on image for enlarged view)***

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-intro-1-new.jpg)

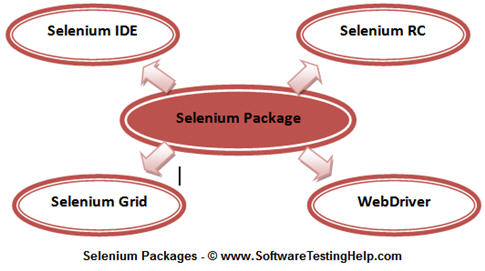
**Selenium Components**

Selenium is not just a single tool or a utility, rather a package of several [testing tools](http://www.softwaretestinghelp.com/category/software-testing-tools/) and for the same reason it is referred to as a Suite. Each of these tools is designed to cater different testing and [test environment requirements](http://www.softwaretestinghelp.com/test-bed-test-environment-management-best-practices/).

**The suite package constitutes of the following sets of tools:**

* Selenium Integrated Development Environment (IDE) [Selenium intro 2](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-intro-2.jpg)
* Selenium Remote Control (RC) [Selenium intro 3](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-intro-3.jpg)
* Selenium WebDriver
* Selenium Grid [Selenium intro 4](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-intro-4.jpg)

Selenium RC and WebDriver, in a combination are popularly known as Selenium 2. Selenium RC alone is also referred as Selenium 1.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-intro-5.jpg)

**Brief Introduction to Selenium tools**

**Selenium Core**

Selenium is a result of continuous efforts by an engineer at ThoughtWorks, named as Jason Huggins. Being responsible for the testing of an internal Time and Expenses application, he realized the need for an automation testing tool so as to get rid of repetitive manual tasks without compromising with the quality and accuracy.

------------

As a result, he built a JavaScript program, named as “JavaScriptTestRunner” in early 2004 that could automatically control the browser’s actions which seemed very much similar to that of a user communicating with the browser.

Henceforth, Jason started demoing the tool to the vast audience. Eventually the discussions were laid out to categorize this tool in the open source category as well as its potential to grow as a re-usable testing framework for other web based applications.

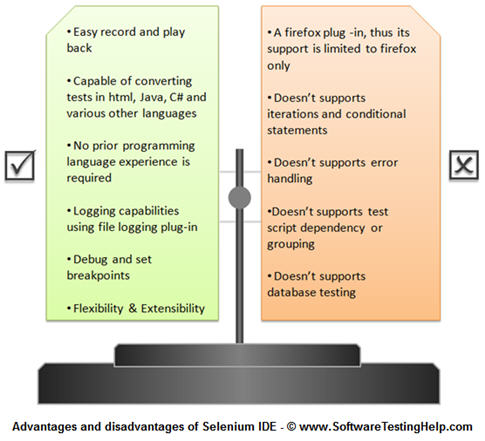
The tool was later on acclaimed with the name “Selenium Core”.

**Selenium IDE (Selenium Integrated Development Environment)**

Selenium IDE was developed by Shinya Kasatani. While studying Selenium Core, he realized that this JavaScript code can be extended to create an integrated development environment (IDE) which can be plugged into Mozilla Firefox. This IDE was capable of recording and playing back the user actions on a Firefox instance to which it was plugged-in. Later on Selenium IDE became a part of Selenium Package in the year 2006. The tool turned out a great value and potential to the community.

Selenium IDE is the simplest and easiest of all the tools within the Selenium Package. Its record and playback feature makes it exceptionally easy to learn with minimal acquaintances to any programming language. With several advantages, a few disadvantages accompanied Selenium IDE, thus making it inappropriate to be used in cases of more advanced test scripts.

**Advantages and disadvantages of Selenium IDE:**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-intro-6.jpg)

The disadvantages of IDE are in reality not disadvantages of selenium, rather just limitations to what IDE could achieve. These limitations can be overcome by using Selenium RC or WebDriver.

**Selenium RC (Selenium Remote Control)**

Selenium RC is a tool which is written in java that allows a user to construct test scripts for a web based application in which ever programming language he/she chooses. Selenium RC came as result to overcome various disadvantages incurred by Selenium IDE or Core.

Loopholes and restrictions which were imposed while using Selenium Core made it difficult for the user to leverage the benefits of the tool to its totality. Thus it made the testing process a cumbersome and a far reaching task.

One of the crucial restrictions was **same origin policy.**

**Problem of same origin policy:**

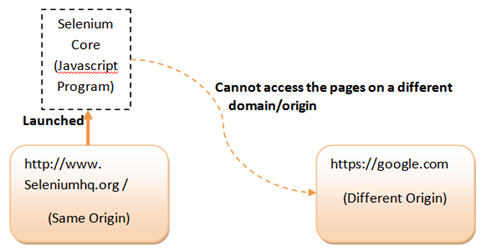
The problem of same origin policy disallows to access the DOM of a document from an origin that is different from the origin we are trying to access the document.

Origin is a sequential combination of scheme, host and port of the URL. For example, for a URL http://www.seleniumhq.org/projects/, the origin is a combination of http, seleniumhq.org, 80 correspondingly.

Thus the Selenium Core (JavaScript Program) cannot access the elements from an origin that is different from where it was launched.

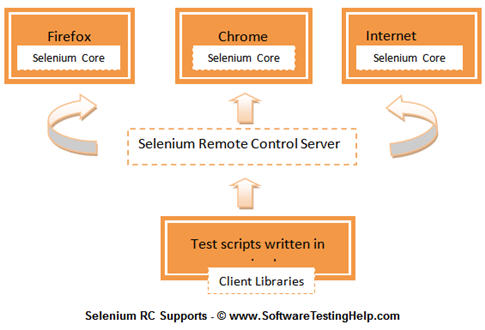
For Example, if I have launched the JavaScript Program from “http://www.seleniumhq.org/”, then I would be able to access the pages within the same domain such as “http://www.seleniumhq.org/projects/” or “http://www.seleniumhq.org/download/”. The other domains like google.com, yahoo.com would no more be accessible.

Thus, to test the application using Selenium Core, one has to install the entire application on the Selenium Core as well as web server to overcome the problem of same origin policy.

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-intro-7.jpg)

So, In order to govern the same origin policy without the need of making a separate copy of Application under test on the Selenium Core, Selenium Remote Control was introduced. While Jason Huggins was demoing Selenium, another fellow colleague at ThoughtWorks named Paul Hammant suggested a work around of same origin policy and a tool that can be wired up with a programming language of our choice. Thus Selenium RC came into existence.

Unlike selenium IDE, selenium RC supports a wide range of browsers and platforms.

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-intro-8.jpg)

**Workflow Description**

* User creates test scripts in a desired programming language.
* For every programming language, there is a designated client library.
* Client library deports the test commands to the selenium server.
* Selenium server deciphers and converts the test commands into JavaScript commands and sends them to the browser.
* Browser executes the commands using selenium core and sends results back to the selenium server
* Selenium server delivers the test results to the client library.

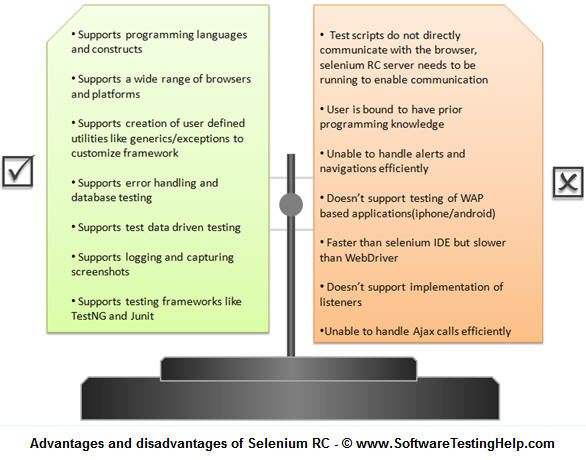
There are a few pre-requisites to be in place before creating Selenium RC scripts:

* A Programming Language – Java, C#, Python etc.
* An Integrated Development Environment –Eclipse, Netbeans etc.
* A Testing Framework (optional) – JUnit, TestNG etc.
* And Selenium RC setup off course

**Advantages and disadvantages of selenium RC:**

Coming on to the advantages and disadvantages of selenium RC, refer the following figure.

***(Click on image for enlarged view)***

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-intro-9.jpg)

**Selenium Grid**

With selenium RC, life of a tester has always been positive and favorable until the emerging trends raised a demand to execute same or different test scripts on multiple platforms and browsers concurrently so as to achieve distributed test execution, testing under different environments and saving execution time remarkably. Thus, catering these requirements selenium grid was brought into the picture.

Selenium Grid was introduced by Pat Lightbody in order to address the need for executing the test suites on multiple platforms simultaneously.

**Selenium WebDriver**

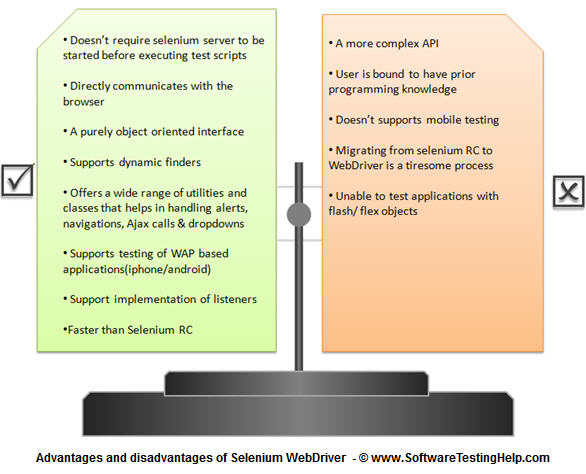
Selenium WebDriver was created by yet another engineer at ThoughtWorks named as Simon Stewart in the year 2006. WebDriver is also a web-based testing tool with a subtle difference with Selenium RC. Since, the tool was built on the fundamental where an isolated client was created for each of the web browser; no JavaScript Heavy lifting was required. This led to a compatibility analysis between Selenium RC and WebDriver. As a result a more powerful automated testing tool was developed called ***Selenium 2***.

WebDriver is clean and a purely object oriented framework. It utilizes the browser’s native compatibility to automation without using any peripheral entity. With the increasing demand it has gained a large popularity and user base.

**Advantages and disadvantages of Selenium WebDriver:**

Refer the following figure for the advantages and disadvantages of WebDriver.

***(Click on image for enlarged view)***

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-intro-10.jpg)

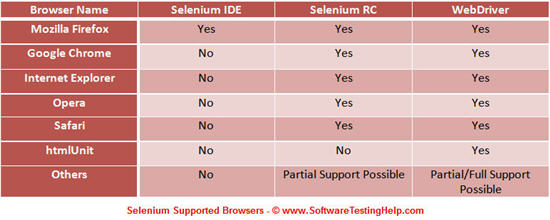
**Selenium 3**

Selenium 3 is an advance version of Selenium 2. It is a tool focused for automation of mobile and web applications. Stating that it supports mobile testing, we mean to say that the WebDriver API has been extended to address the needs of mobile application testing. The tool is expected to be launched soon in the market.

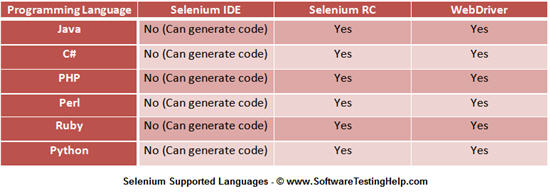
**Environment and Technology Stack**

With the advent and addition of each new tool in the selenium suite, environments and technologies became more compatible. Here is an exhaustive list of environments and technologies supported by selenium tool set.

**Supported Browsers**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-intro-11.jpg)

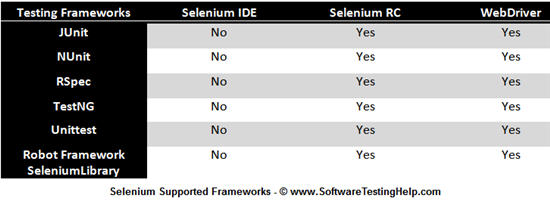
**Supported Programming Languages**

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-intro-12.jpg)

**Supported Operating Systems**

[](http://cdn2.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-intro-13.jpg)

**Supported Testing Frameworks**

[](http://cdn.softwaretestinghelp.com/wp-content/qa/uploads/2014/10/Selenium-intro-14.jpg)

**Conclusion**

In this tutorial, we tried to make you acquainted with the Selenium suite describing its various components, their usages and their advantages over each other.

**Here are the cruxes of this article.**

* Selenium is a suite of several automated testing tools, each of them catering to different testing needs.
* All these tools fall under the same umbrella of open source category and supports only web based testing.
* Selenium suite is comprised of 4 basic components; Selenium IDE, Selenium RC, WebDriver, Selenium Grid.
* User is expected to choose wisely the right Selenium tool for his/her needs.
* Selenium IDE is distributed as a Firefox plug-in. It is easier to install and use. User is not required to possess prior programming knowledge. Selenium IDE is an ideal tool for a naive user.
* Selenium RC is a server that allows user to create test scripts in a desired programming language. It also allows executing test scripts within the large spectrum of browsers.
* Selenium Grid brings out an additional feature to Selenium RC by distributing its test script on different platforms and browsers at the same time for execution, thus implementing the master slave architecture.
* WebDriver is a different tool altogether that has various advantages over Selenium RC. The fusion of Selenium RC and WebDriver is also known as Selenium 2. WebDriver directly communicates with the web browser and uses its native compatibility to automate.
* Selenium 3 is the most anticipated inclusion in the Selenium suite which is yet to be launched in the market. Selenium 3 strongly encourages mobile testing.