**Selenium**

**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.

**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.

Note: 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.**

**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.

## Selenium

**Jason Huggins in 2004**

* He was working on a web application that required frequent testing
* Repetitious [Manual Testing](https://www.guru99.com/manual-testing.html) of their application was becoming more and more inefficient, he created a[JavaScript](https://www.guru99.com/interactive-javascript-tutorials.html)program
* It would automatically control the browser's actions. He named this program as the "**Java Script Test Runner**."

**Issue JavaScript Test Runner**

## Same Origin Policy Issue

**It prohibits JavaScript code from accessing elements from a domain that is different from where it was launched**.

**“It cannot access pages from different sites”**

 Example, the HTML code in www.annauniv.com uses a JavaScript program "randomScript.js". The same origin policy will only allow randomScript.js to access pages within anna.com such as annaunniv.com/result,annauniv.com/login, orannauniv.com/signup. However, it cannot access pages from different sites such as bits-pilani.ac.in /search because they belong to 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

## Selenium Remote Control(Selenium RC)

## Selenium Remote Control(Selenium RC)

## **Paul Hammant**

## He 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****

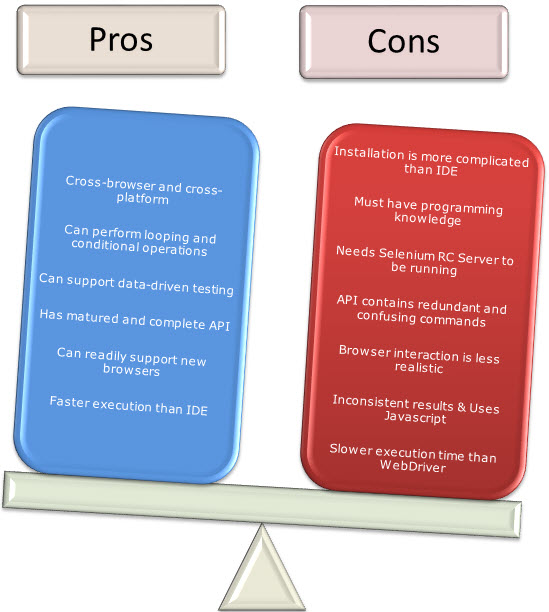
## ****1****.Selenium Grid

**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.**

## Intoduction 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**.
* RC can support the following programming languages:
* Java
* C#
* PHP
* Python
* Perl
* Ruby

[](https://www.guru99.com/images/RCProCon.jpg)

**Selenium IDE**

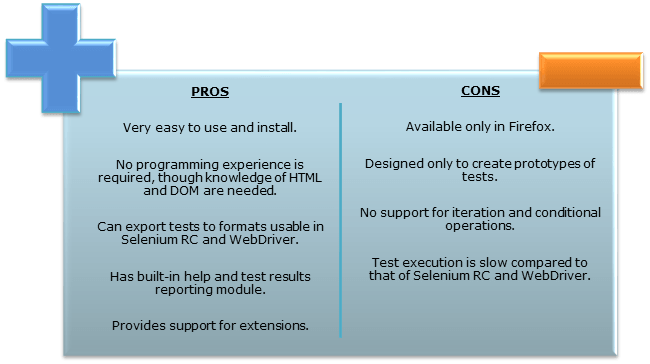
**Selenium IDE**

**Shinya Kasatani**

* It is 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**.

**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

[](https://www.guru99.com/images/SeleniumIDEProCon.png)

**WebDriver**

**WebDriver**

**Simon Stewart** created WebDriver circa **2006**

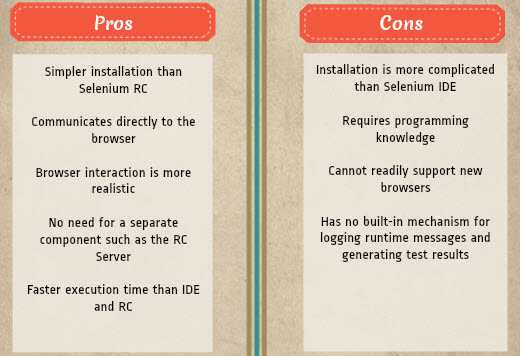
* When browsers and web applications were becoming more powerful and more restrictive with JavaScript programs like Selenium Core.
* **It was the first cross-platform testing framework**
* **It could control the browser from the OS level**

**Introduction WebDriv**

* 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

[](https://www.guru99.com/images/pros_cons_-_webdriver.jpg)

**Selenium 2**

**Selenium 2**

**2008**

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

**Selenium Grid**

**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.

**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** |
| --- | --- | --- |
| **Browser** **Support** | 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. |

## 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 have a native IDE and only third party IDE can be used for 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 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 |
| 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 |
| Parameterization Support is built | Parameterization can be done via programming but is difficult to implement. |
| Test Reports are generated automatically | No native support to generate test /bug reports. |

Though clearly, [QTP](https://www.guru99.com/quick-test-professional-qtp-tutorial.html) 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)