VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF INFORMATION TECHNOLOGY AND ENGINEERING

SOFTWARE QUALITY ASSURANCE/TESTING

FALL SEMESTER 2022-2023

DIGITAL ASSIGNMENT-01

NAME : MUDASSIR KHAN.G

REGISTER NUMBER : 20BCS0130

COURSENAME:SOFTWARE QUALITY ASSURANCE/TESTING

COURSE CODE : CSC4001

SLOT : D1

FACULTY NAME : DR IYAPPARAJA.M

DATE: 09.09.2022

TESTING TOOL : SELENIUM

(OPEN SOURCE)

What is Selenium?

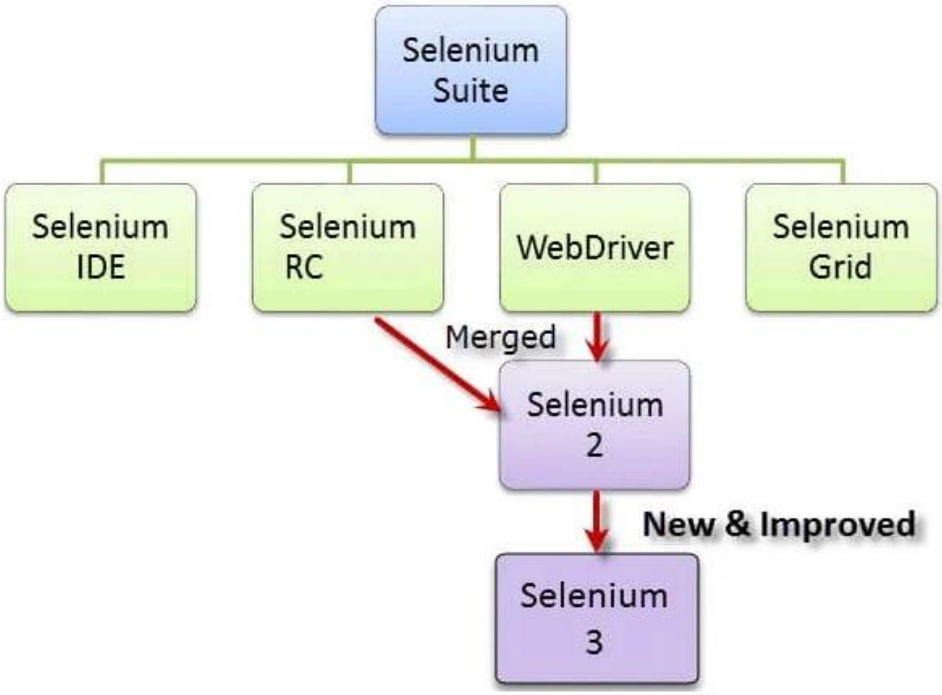
Selenium is a free (open-source) automated testing framework used to validate web applications across different browsers and platforms. You can use multiple programming languages like Java, C#, Python etc to create Selenium Test Scripts. Testing done using the Selenium testing tool is usually referred to as Selenium Testing.

Selenium Software is not just a single tool but a suite of software, each piece catering to different Selenium QA testing needs of an organization. Here is the list of tools

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

Introduction to Selenium:

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.



- What is Selenium?
- Who developed Selenium?
- The Same Origin Policy Issue
- Birth of Selenium Remote Control (Selenium RC)
- Birth of Selenium Grid
- Birth of Selenium IDE
- Birth of WebDriver
- Birth of Selenium 2
- So, Why the Name Selenium?
- Brief Introduction Selenium IDE
- Brief Introduction Selenium Remote Control (Selenium RC)
- Brief Introduction WebDriver
- Selenium Grid
- Note on Browser and Environment Support
- How to Choose the Right Selenium Tool for Your Need
- A Comparison between Selenium and QTP(now UFT)
- Advantages of QTP over Selenium

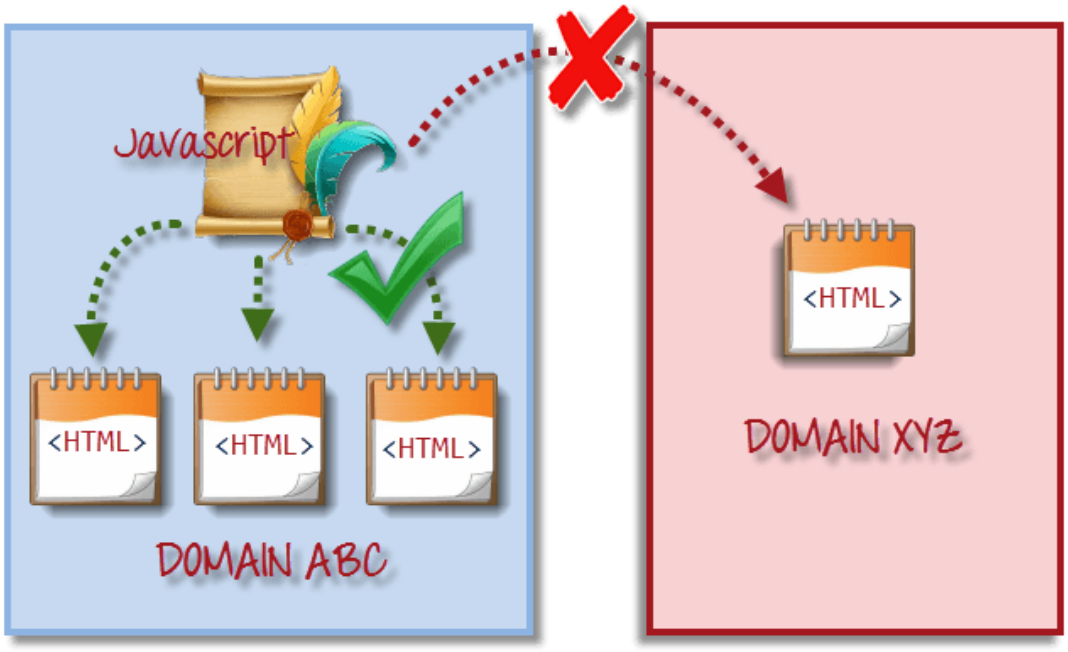
Who developed Selenium?

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 Jason Huggins in 2004. An engineer at ThoughtWorks, 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 JavaScriptRunner 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

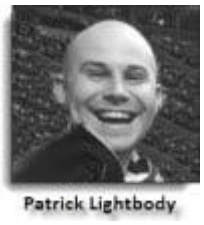
This is the reason why prior to Selenium RC, testers needed to install local copies of both Selenium Core (a JavaScript program) and the web server containing the web application being tested so they would belong to the same domain

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**.

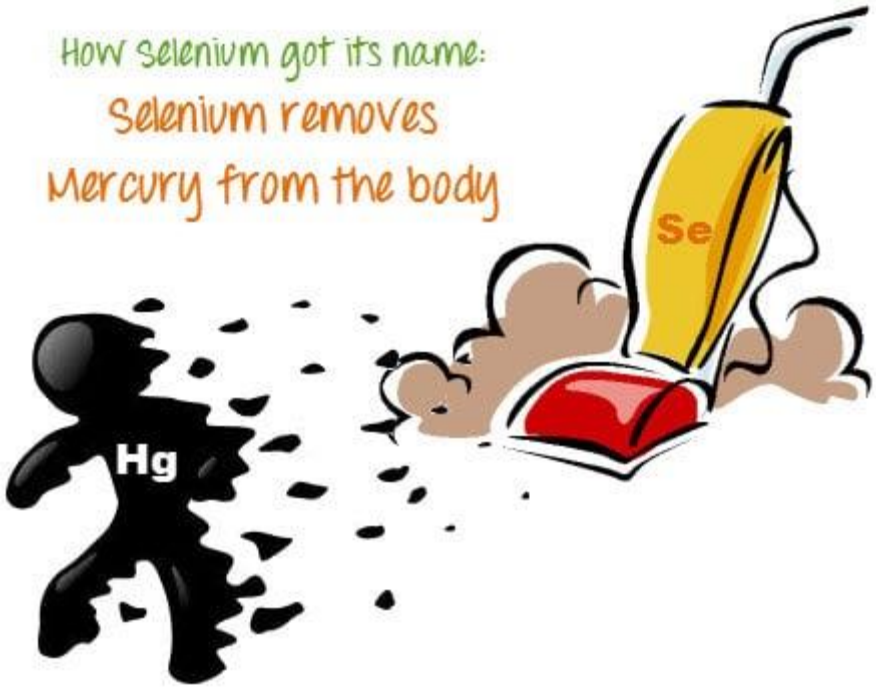
Birth of Selenium 2

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?

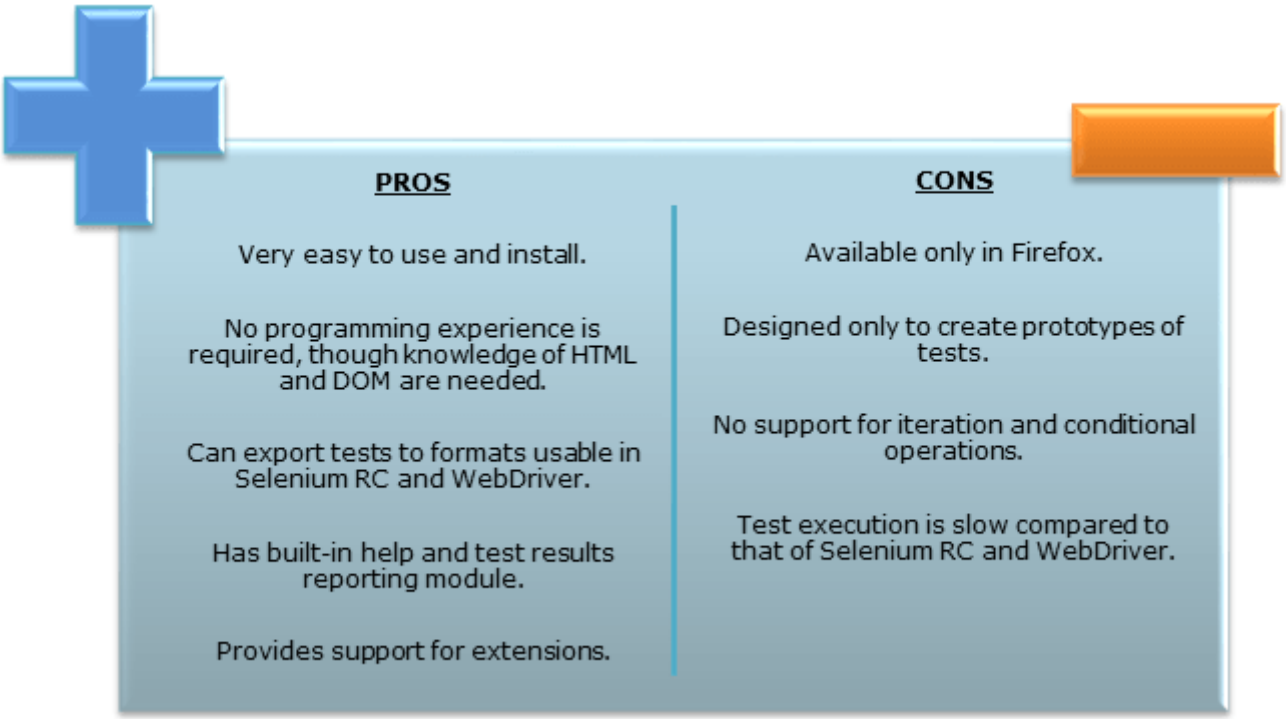
The Name Selenium came from a joke which Jason cracked once to his team. During Selenium's

development, another automated testing framework was popular made 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 and his teammates took it. So that is how we got to call this framework up to the present.



Brief Introduction Selenium IDE

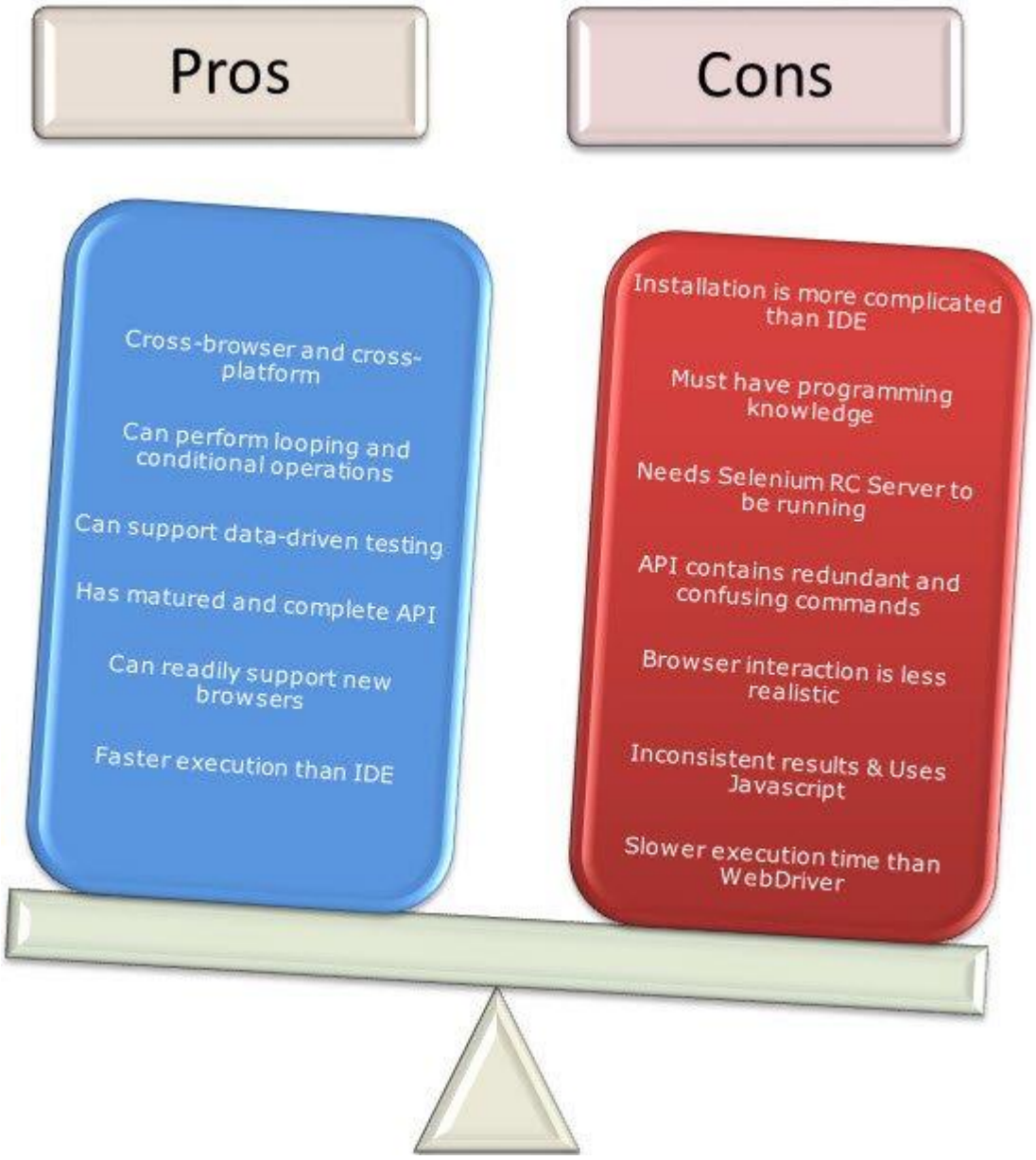
Selenium Integrated Development Environment (IDE) is the **simplest framework** in the Selenium suite and is **the easiest one to learn**. It is a **Firefox plugin** that you can install as easily as you can with other plugins. However, because of its simplicity, Selenium IDE should only be used as a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



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

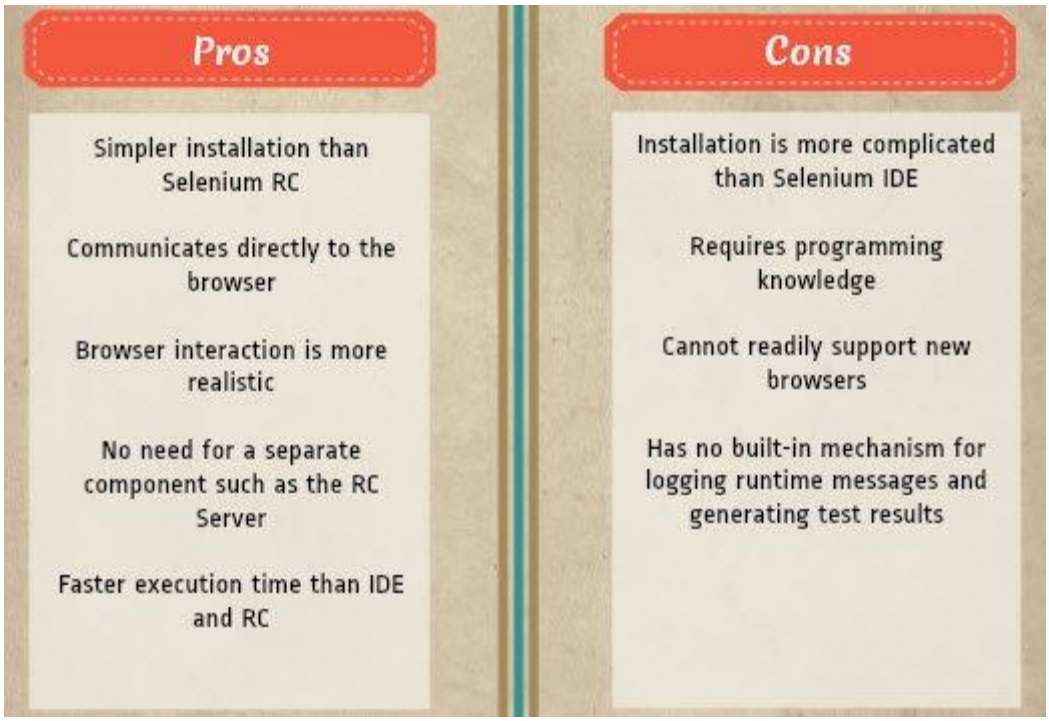


Brief Introduction WebDriver

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 Selenium Automation Testing. **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
-



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
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.

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 and Chrome only.

Tool	Why Choose?
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 designng 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 and Benefits 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

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 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 Software Testing 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. you to run tests in parallel, unlike in QTP where you are only allowed to run tests sequentially.