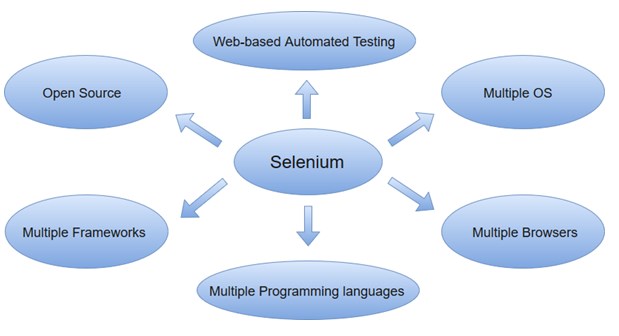
What is Selenium

Selenium is one of the most widely used open source Web UI (User Interface) automation testing suite.It was originally developed by Jason Huggins in 2004 as an internal tool at Thought Works. Selenium supports automation across different browsers, platforms and programming languages.

Selenium can be easily deployed on platforms such as Windows, Linux, Solaris and Macintosh. Moreover, it supports OS (Operating System) for mobile applications like iOS, windows mobile and android.

Selenium supports a variety of programming languages through the use of drivers specific to each language.Languages supported by Selenium include C#, Java, Perl, PHP, Python and Ruby.Currently, Selenium Web driver is most popular with Java and C#. Selenium test scripts can be coded in any of the supported programming languages and can be run directly in most modern web browsers. Browsers supported by Selenium include Internet Explorer, Mozilla Firefox, Google Chrome and Safari.



Selenium can be used to automate functional tests and can be integrated with automation test tools such as **Maven**, **Jenkins**, **& Docker** to achieve continuous testing. It can also be integrated with tools such as **TestNG**, & **JUnit** for managing test cases and generating reports.

## Prerequisites

Before learning the concepts of Selenium, you should have a basic understanding of java or any other object-oriented programming language.

Languages supported by Selenium include C#, Java, Perl, PHP, Python and Ruby.Currently, Selenium Webdriver is most popular with Java and C#. So, if you know any of the languages then it won't be tough to understand the concepts of Selenium. In addition, you should have prior knowledge of software testing techniques like automation testing, functional testing, etc.

Functional testing :

In software development, functional testing is a form of software system testing that verifies whether software matches its design.

Generally, functional testing is black-box meaning the internal program structure is ignored (unlike for white-box testing).

Functional testing can evaluate compliance to functional requirements.

Sometimes, functional testing is a quality assurance (QA) process.

Functional testing differs from acceptance testing. Functional testing verifies a program by checking it against design document(s) or specification(s), while acceptance testing validates a program by checking it against the published user or system requirements.

As a form of system testing, functional testing tests slices of functionality of the whole system. Despite similar naming, functional testing is not testing the code of a single function.

The concept of incorporating testing earlier in the delivery cycle is not restricted to functional testing.

Functional testing typically involves six steps :

1. The identification of functions that the software is expected to perform
2. The creation of input data based on the function's specifications
3. The determination of output based on the function's specifications
4. The execution of the test case
5. The comparison of actual and expected outputs
6. To check whether the application works as per the customer need

# 

# Selenium Basic Terminology

Before proceeding with this tutorial, let us first understand some of the key concepts associated with Automation testing of an application.

With the growing need for efficient software products, every software development group need to carry out a series of tests before launching the final product into the market. Test engineers strive to catch the faults or bugs before the software product is released, yet delivered software always has defects. Even with the best manual testing processes, there's always a possibility that the final software product is left with a defect or is unable to meet the end user requirement. Automation testing is the best way to increase the effectiveness, efficiency and coverage of your software testing.

## Automation Testing

Automation testing uses the specialized tools to automate the execution of manually designed test cases without any human intervention. Automation testing tools can access the test data, controls the execution of tests and compares the actual result against the expected result. Consequently, generating detailed test reports of the system under test.

Automation testing covers both functional and performance test on an application.

* Functional automation is used for automation of functional test cases. For example, regression tests, which are repetitive in nature, are automated.
* Performance automation is used for automation of non-functional performance test cases. For example, measuring the response time of the application under considerable (say 100 users) load.

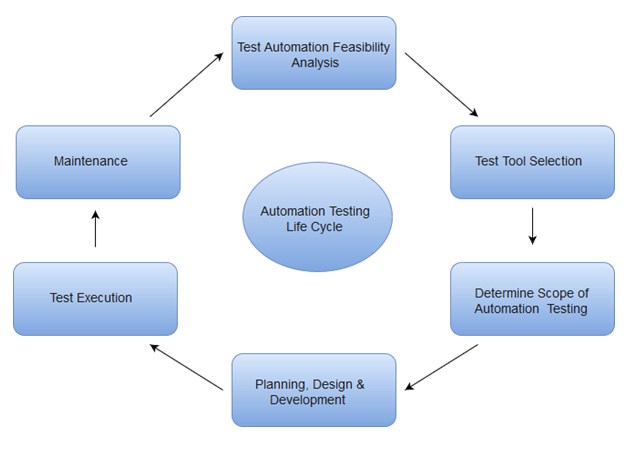
Automation Testing tools which are used for functional automation:

* Quick Test Professional, provided by HP.
* Rational Robot, provided by IBM.
* Coded UI, provided by Microsoft.
* Selenium, open source.
* Auto It, open Source.

Automation Testing tools which are used for non-functional automation:

* Load Runner, provided by HP.
* JMeter, provided by Apache.
* Burp Suite, provided by PortSwigger.
* Acunetix, provided by Acunetix.

## Automation Testing Life Cycle



Why Automated Testing

Automation testing has specific advantages for improving long-term efficiency of any software. The key benefits of test automation are:

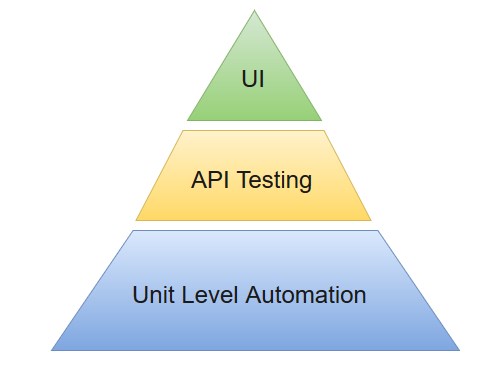
* Automated testing has long been considered beneficial for big software organizations. Although, it is often thought to be too expensive or difficult for smaller companies to implement.
* Automated testing tools can be programmed to build and execute test scripts at a specific time without involving any human intervention.For instance, automated test can be automatically kicked off overnight, and the testers can analyse the results of the automated the next morning.
* Automated testing tools are able to playback pre-recorded and pre-defined actions.
* Automation testing supports frequent regression testing.
* It provides rapid feedback to developers.
* It provides unlimited iterations of test case execution.
* It provides disciplined documentation of test cases.
* Automated test generates customized defect reports.
* Less error prone as compared to manual testing.

Test Automation for Web Applications

If we take a look at the type of software applications prevailing in current market scenario, most of the software applications are written as web-based applications to be run in an internet browser. The testing strategy for web-based applications varies widely among companies and organizations.In an era of highly interactive and responsive software processes where many organizations are using some form of agile methodology, test automation is frequently becoming a requirement for software projects.

The most effective manner to carry out test automation for web application is to adopt a pyramid testing strategy.This pyramid testing strategy includes automation tests at three different levels. Unit testing represents the base and biggest percentage of this test automation pyramid. Next comes, service layer, or API testing. And finally, GUI tests sit at the top. The pyramid looks something like this:

Test Automation Pyramid:



# Selenium Features

* Selenium is an open source and portable Web testing Framework.
* Selenium IDE provides a playback and record feature for authoring tests without the need to learn a test scripting language.
* It can be considered as the leading cloud-based testing platform which helps testers to record their actions and export them as a reusable script with a simple-to-understand and easy-to-use interface.
* Selenium supports various operating systems, browsers and programming languages. Following is the list:
  + Programming Languages: C#, Java, Python, PHP, Ruby, Perl, and JavaScript
  + Operating Systems: Android, iOS, Windows, Linux, Mac, Solaris.
  + Browsers: Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera, Safari, etc.
* It also supports parallel test execution which reduces time and increases the efficiency of tests.
* Selenium can be integrated with frameworks like Ant and Maven for source code compilation.
* Selenium can also be integrated with testing frameworks like TestNG for application testing and generating reports.
* Selenium requires fewer resources as compared to other automation test tools.
* WebDriver API has been indulged in selenium whichis one of the most important modifications done to selenium.
* Selenium web driver does not require server installation, test scripts interact directly with the browser.
* Selenium commands are categorized in terms of different classes which make it easier to understand and implement.
* Selenium Remote Control (RC) in conjunction with WebDriver API is known as Selenium 2.0. This version was built to support the vibrant web pages and Ajax.

# Selenium Limitations

* Selenium does not support automation testing for desktop applications.
* Selenium requires high skill sets in order to automate tests more effectively.
* Since Selenium is open source software, you have to rely on community forums to get your technical issues resolved.
* We can't perform automation tests on web services like SOAP or REST using Selenium.
* We should know at least one of the supported programming languages to create tests scripts in Selenium WebDriver.
* It does not have built-in Object Repository like UTF/QTP to maintain objects/elements in centralized location. However, we can overcome this limitation using Page Object Model.
* Selenium does not have any inbuilt reportingcapability; you have to rely on plug-ins like **JUnit** and **TestNG** for test reports.
* It is not possible to perform testing on images. We need to integrate Selenium with **Sikuli** for image based testing.
* Creating test environment in Selenium takes more time as compared to vendor tools like UFT, RFT, Silk test, etc.
* No one is responsible for new features usage; they may or may not work properly.
* Selenium does not provide any test tool integration for Test Management.

Selenium vs HP - QTP

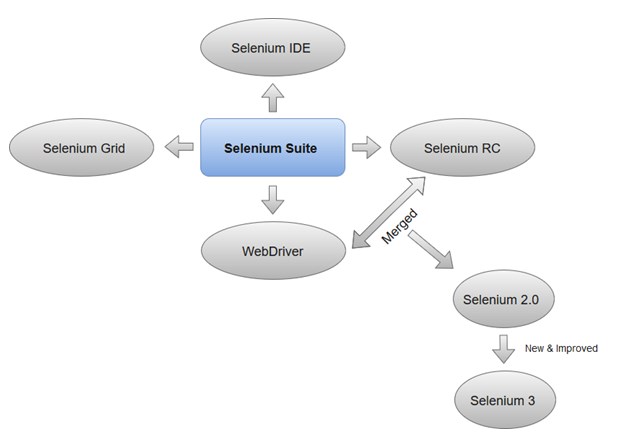
Selenium and QTP are the most frequent used automation test tools in the market. Hence, we have compared some of the features of Selenium over QTP.

|  |  |  |
| --- | --- | --- |
| **Features** | **Selenium** | **HP QTP** |
| License | Open source tool | Required |
| Customer support | Dedicated HP support | Selenium community forums |
| Test Support | Supportsautomation only for web-based applications. | Support tests on both web and desktop based applications. |
| Resource consumption during test scripts execution | Low resource consumption | High resource consumption |
| Supported programming languages | Java, C#, Ruby, Python, Perl, PHP and JavaScript | VB Script. |
| Supported Environments | Android, iOS, Windows, Linux, Mac, Solaris. | Only for Windows |
| Supported Browsers | Google Chrome, Mozilla Firefox, Internet Explorer, Edge, Opera, Safari, etc. | Specific versions of Google Chrome, Mozilla Firefox and Internet Explorer. |
| Object Repository/Recovery Scenario | Absent | Built-in object repository and recovery scenario. |
| Browser Controls | None | Controls like favourites bar, backward and forward buttons can be accessed within the browser. |
| Test Report Generation | It relies on external tool for generating test reports. | Built-in test report generation within the tool. |
| Parameterization | You have to rely on any one of the supported programming language for parameterization. | Built-in tools are available for parameterization. |

# Selenium Tool Suite

Selenium is not just a single tool but a suite of software, each with a different approach to support automation testing. It comprises of four major components which include:

1. Selenium Integrated Development Environment (IDE)
2. Selenium Remote Control (Now Deprecated)
3. WebDriver
4. Selenium Grid



### 1.Selenium Integrated Development Environment (IDE)

Selenium IDE is implemented as Firefox extension which provides record and playback functionality on test scripts. It allows testers to export recorded scripts in many languages like HTML, Java, Ruby, RSpec, Python, C#, JUnit and TestNG. You can use these exported script in Selenium RC or Webdriver.

>Selenium IDE has limited scope and the generated test scripts are not very robust and portable.

### 2. Selenium Remote Control

Selenium RC (officially deprecated by selenium)allows testers to write automated web application UI test in any of the supported programming languages. It also involves an HTTP proxy server which enables the browser to believe that the web application being tested comes from the domain provided by proxy server.

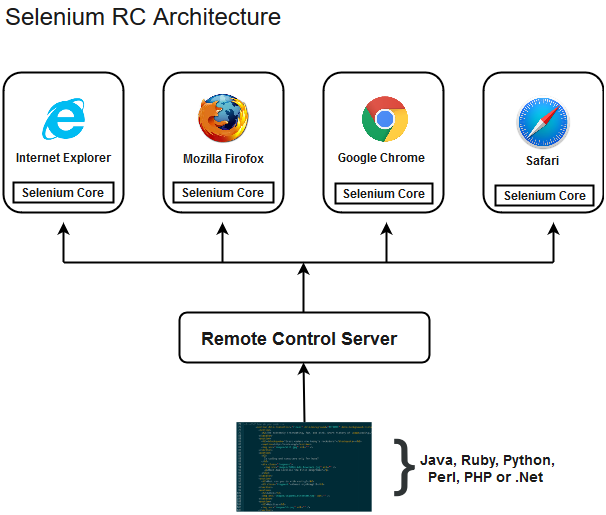
Selenium RC comes with two components.

1. Selenium RC Server (acts as a HTTP proxy for web requests).

>

1. Selenium RC Client (library containing your programming language code).

The figure given below shows the architectural representation of Selenium RC.



Selenium RC had been considered quite effective for testing complex AJAX-based web user interfaces under a Continuous Integration System.

### 3. Selenium WebDriver

Selenium WebDriver (Selenium 2) is the successor to Selenium RC and is by far the most important component of Selenium Suite. SeleniumWebDriverprovides a programming interface to create and execute test cases. Test scripts are written in order to identify web elements on web pages and then desired actions are performed on those elements.

Selenium WebDriver performs much faster as compared to Selenium RC because it makes direct calls to the web browsers. RC on the other hand needs an RC server to interact with the web browser.

Since, WebDriver directly calls the methods of different browsers hence we have separate driver for each browser. Some of the most widely used web drivers include:

* Mozilla Firefox Driver (Gecko Driver)
* Google Chrome Driver
* Internet Explorer Driver
* Opera Driver
* Safari Driver
* HTML Unit Driver (a special headless driver)

#### Note:Selenium version 2 merged the best features of Selenium RC and Selenium WebDriver into Selenium WebDriver. The latest release Selenium 3 has new added features and functionalities

### 4. Selenium Grid

Selenium Grid is also an important component of Selenium Suite which allows us to run our tests on different machines against different browsers in parallel. In simple words, we can run our tests simultaneously on different machines running different browsers and operating systems.

Selenium Grid follows the **Hub-Node Architecture** to achieve parallel execution of test scripts. The Hub is considered as master of the network and the other will be the nodes. Hub controls the execution of test scripts on various nodes of the network.