Selenium is an open-source suite of tools and libraries that is used for browser automation.

Selenium **can only test web applications**, desktop and mobile apps can't be tested.

Selenium allows the testing of web applications across **different platforms**.

It allows users to test their websites functionally **on different browsers**.

It provides a single interface that lets you write test scripts in programming languages like Ruby, Java, NodeJS, PHP, Perl, Python, JavaScript, and C#, among others. Selenium is very extensible and can be integrated with other tools and frameworks like TestNG, JUnit, Cucumber, etc.

1. **Language Support:** Selenium allows you to create test scripts in different languages like Ruby, Java, PHP, Perl, Python, JavaScript, and C#, among others.
2. **Browser Support:** Selenium enables you to test your website on different browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, Internet Explorer (IE), etc.
3. **Scalability:** Automated testing with Selenium can easily scale to cover a wide range of test cases, scenarios, and user interactions. This scalability ensures maximum test coverage of the application’s functionality.
4. **Reusable Test Scripts:** Selenium allows testers to create reusable test scripts that can be used across different test cases and projects. This reusability saves time and effort in test script creation and maintenance.
5. **Parallel Testing:** Selenium supports parallel test execution, allowing multiple tests to run concurrently. This helps reduce the overall testing time, making the development process more efficient.
6. **Documentation and Reporting:** Selenium provides detailed test execution logs and reports, making it easier to track test results and identify areas that require attention.
7. **User Experience Testing:** Selenium can simulate user interactions and behavior, allowing testers to assess the user experience and ensure that the application is intuitive and user-friendly.
8. **Continuous Integration and Continuous Deployment (CI/CD):** Selenium can be integrated into CI/CD pipelines to automate the testing of each code change. This integration helps identify and address issues earlier in the development cycle, allowing for faster and more reliable releases.

Selenium Suite has 4 components namely:

1. Selenium IDE
2. Selenium RC
3. Selenium WebDriver
4. Selenium Grid

### Selenium IDE

Selenium IDE is a Chrome and Firefox plugin. The primary use of a Selenium IDE is to record user interactions such as clicks, selections etc in the browser and play them back as automated tests .

It then generates the test script (of the automated tests) in programming languages like C#, Java, Python, and Ruby and Selenese (Selenium’s own scripting language).

Selenium IDE helps in:

* Creating automated test scripts and validating them at speed
* Identifying and highlighting errors during the replay of interactions

### Limitations of Selenium IDE

* Not suitable for testing extensive data
* Connections with the database can not be tested
* Cannot handle the dynamic part of web-based applications
* Does not support capturing of screenshots on test failures
* No feature available for generating result reports

### Selenium RC

Selenium RC was **built to automate the testing of web applications by simulating user interactions** across different browsers and platforms. It provided a way to browser automation remotely and execute test scripts written in various programming languages.

Limitations of Selenium RC:

* **Browser Limitations:** Selenium RC had to work with browsers using a JavaScript-based “proxy” mechanism, which introduced potential instability and limitations, especially when working with modern web applications.
* **Speed and Performance:** The use of a JavaScript proxy added overhead and affected the speed and performance of test execution.
* **Maintenance and Compatibility:** Selenium RC required separate “drivers” for each browser, making maintenance and compatibility challenging as browsers continued to update and evolve.
* **Synchronization Issues:** Selenium RC often faced synchronization problems, where test scripts had to wait for the browser to respond before proceeding to the next step.
* **Complex Setup:** Setting up Selenium RC involved multiple components, which could be complex and difficult to configure correctly.

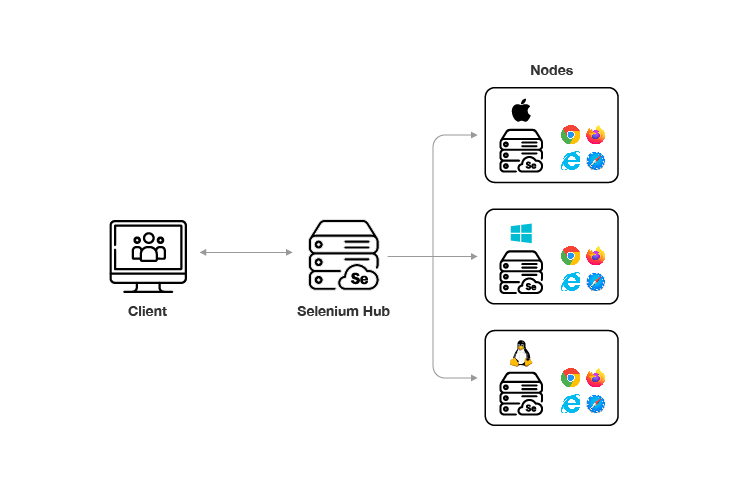
### Selenium WebDriver

Selenium WebDriver is a **powerful and enhanced version of Selenium RC** which was developed to overcome the limitations of Selenium RC. WebDriver communicates with browsers directly with the help of browser-specific native methods, thereby completely eliminating the need of Selenium RC.

WebDriver works closely with Selenium IDE and Selenium Grid resulting in reliable test execution at speed and scale.

### Selenium Grid

Selenium Grid is a smart proxy server that allows QAs to run tests in parallel on multiple machines and manages different browser versions and browser configurations centrally (instead of separately, in individual tests). This is done by routing commands to remote web browser instances, where one server acts as the hub. This hub routes test commands that are in JSON format to multiple registered Grid nodes.



Pros:

* Open source/ No licensing costs
* Language-independent
* Third-party integrations
* Parallel testing
* Direct browser communication
* More realistic browser interaction

Cons:

* Limited support for testing desktop applications
* No built-in reporting mechanism for test results
* Difficulty in handling dynamic web elements, pop-ups, and dialogs
* Limited support for touch-enabled devices and gestures
* Performance limitations for large-scale testing