[Different versions of Selenium](https://yizeng.me/2014/04/25/relationships-between-different-versions-of-selenium/" \l "heading-different-versions-of-selenium)

* [Selenium IDE](https://yizeng.me/2014/04/25/relationships-between-different-versions-of-selenium/#heading-selenium-ide)
* [Selenium 1 / Selenium RC](https://yizeng.me/2014/04/25/relationships-between-different-versions-of-selenium/#selenium-rc)
* [Selenium 2 / Selenium WebDriver](https://yizeng.me/2014/04/25/relationships-between-different-versions-of-selenium/#selenium-webdriver)
* [Selenium 3](https://yizeng.me/2014/04/25/relationships-between-different-versions-of-selenium/#selenium-3)
* [WebDriver](https://yizeng.me/2014/04/25/relationships-between-different-versions-of-selenium/#heading-webdriver)
* [Selenium Grid](https://yizeng.me/2014/04/25/relationships-between-different-versions-of-selenium/#heading-selenium-grid)
* [Selenium](https://yizeng.me/2014/04/25/relationships-between-different-versions-of-selenium/#heading-selenium)

**Selenium IDE**

Firefox add-on [Selenium IDE](http://docs.seleniumhq.org/projects/ide/) allows users to record and re-play user actions in Firefox. It supports exporting the recorded scripts into Selenium RC or Selenium WebDriver code. But it is a failure due to several reasons:

Selenium IDE does not directly support to

1. Condition & iteration statements
2. Logging & reporting of test results
3. Error Handling, Particularly unexpected errors
4. Database Testing
5. Capture Screenshots on test failures

**Selenium 1 / Selenium RC**

Also known as Selenium 1 incorrectly[1](https://yizeng.me/2014/04/25/relationships-between-different-versions-of-selenium/#fn:1), [Selenium Remote Control](http://docs.seleniumhq.org/projects/remote-control/) is the first version of Selenium API, which was generally known as "Selenium" without any version suffixes at the time. After the release of second generation of Selenium, it started to be called with version number or name in order to be distinguishable from the new API. It is now officially deprecated but still released within [Selenium WebDriver](https://yizeng.me/2014/04/25/relationships-between-different-versions-of-selenium/#selenium-webdriver) library for backward compatibility purpose.

**Selenium 2 / Selenium WebDriver**

Selenium 2, a.k.a. [Selenium WebDriver](http://docs.seleniumhq.org/projects/webdriver/), is the latest API in Selenium project, which replaces Selenium RC with fundamentally different mechanisms and dominates web UI automation market right now.

**Selenium 3**

The next release of Selenium project, which is only in staging at the time of writing. One possible major change would be breaking the backward compatibility, i.e. Selenium RC will be no longer a part of Selenium release. More details can be followed and discussed in [this post](https://groups.google.com/d/topic/selenium-developers/EdkY-QN5uqU/discussion) on Selenium Developers' forum.

**WebDriver**

The term "WebDriver" might have different meanings in various contexts.

* Synonym for Selenium WebDriver / Selenium 2.
* A tool called "WebDriver" which was created independently then got merged into Selenium.
  + According to [Selenium History](http://docs.seleniumhq.org/about/history.jsp), before the era of Selenium 2, WebDriver was a different web testing tool coded by Simon Stewart separately, while probably at the same time, developers within Google were making changes to Selenium RC. After the meeting at GTAC, the decision was made to merge these two projects together into the next generation of Selenium, which is commonly referred as [Selenium 2 / Selenium WebDriver](https://yizeng.me/2014/04/25/relationships-between-different-versions-of-selenium/#selenium-webdriver). At the time, some people used the word "WebDriver" alone to address this new project, so that users wouldn't get confused with then dominating project - Selenium RC.
* WebDriver Wire Protocol (JSON-over-HTTP wire protocol)
  + [JsonWireProtocol](https://github.com/SeleniumHQ/selenium/wiki/JsonWireProtocol) is the underlying technology used by client side drivers to communicate with server side RemoteWebDriver.
* WebDriver W3C Specification
  + With the rapid growth of Selenium WebDriver API, the maintainers made it a W3C specification, which is currently released as [W3C working draft](http://www.w3.org/TR/webdriver/) called "WebDriver" again. This is the specification defines the WebDriver standard API, which is what Selenium WebDriver implements.

**Selenium Grid**

[Selenium Grid](https://github.com/SeleniumHQ/selenium/wiki/Grid2) is a tool uses Selenium Server to execute either Selenium RC or Selenium WebDriver tests in parallel on different machines.

**Selenium**

Everything above can be referred as "Selenium" by different people, which in a way confuses the public. All those terminologies are either a part of the current Selenium project, or were once inside Selenium project.

**Comparison of Selenium versions**

| **Version** | **Version** | **Comparison** |
| --- | --- | --- |
| Selenium 1 | Selenium RC | Essentially the same thing. Selenium 1 has never been an official name, but is commonly used in order to distinguish between versions. |
| Selenium 2 | Selenium WebDriver | Essentially the same thing. The term "Selenium WebDriver" is now more commonly used. |
| Selenium RC | Selenium WebDriver | Selenium RC is the predecessor of Selenium WebDriver. It has been deprecated and now released inside Selenium WebDriver for backward compatibility. |
| Selenium IDE | Selenium RC/WebDriver | Selenium IDE is a recording tool for automating Firefox, with the ability to generate simple RC/WebDriver code. Selenium RC/WebDriver are frameworks to automate browsers programmatically. |
| Selenium Grid | Selenium WebDriver | Selenium Grid is a tool to execute Selenium tests in parallel on different machines. Selenium WebDriver is the core library to drive web browsers on a single machine. |

Selenium IDE

Selenium IDE stands for Selenium Integrated Development Environment. It is a Firefox plugin which helps testers to recorded their actions and then export them as a reusable script in one of many programming languages. Selenium IDE has a recording feature, that provides an easy-to-use interface for developing automated tests.

Selenium IDE was developed by Shinya Kasatani. Selenium IDE became a part of Selenium Package in the year 2006. Then tool becomes a great value and potential to the community.

Selenium IDE is the most simplest and easiest of all the tools within the Selenium Package. The record and playback feature makes it exceptionally easy to learn with minimal knowledge of any programming language. The disadvantages of IDE are in reality not disadvantages, but just limitations to what the IDE could achieve. These limitations can be overcome by using Selenium RC or WebDriver.

**Advantages**

* Selenium IDE is very easy to use.
* It has the capability to convert the test to different programming languages such as html, java etc
* Programming language experience is not required for Selenium IDE
* Selenium IDE provides Logging capabilities using file login plug-in.
* In Selenium IDE, user can debug and set breakpoints
* Selenium IDE is flexible for the users.

**Disadvantage**

* Selenium IDE is Firefox plugin, thus its support is limited to Firefox only
* It will not support iteration and conditional statement
* Selenium IDE doesn't support error handling
* Doesn't support test script grouping
* Selenium IDE do not support Database testing

Selenium RC

The Selenium IDE is basically something having record & playback options which present in the every automation tool like QTP, Sliktest etc. & also has very good user interface. The core part of Selenium IDE is based on JavaScript & also supports different extension in it. Along with record & playback, you can use Selenium IDE for multiple dynamic stuffs. The main limitation of Selenium IDE is that, it supported in only Firefox browser. If you want to execute your scripts on different browsers, then you can use Selenium RC (Selenium Remote Control). The Selenium RC supports multiple browsers like IE, Firefox, Chrome, Safari, Opera etc.

It also supports multiple languages like Java, Ruby, C#, Perl, Python etc. You have to get expertise in one language (preferred Java language) & code in selenium RC. The application under test in developed in C# & it does not matter the to create your script in the Java or C# or in any language. It’s totally independent on which your testing is carried out. Similar to language independent it is also platform independent, same code will work on Windows OS, Linux, Mac & Solaris. Most common extension used in the selenium RC is the Java Extension, because Java is platform independent language. Similar to Selenium IDE, the RC is also has its limitations. Before start testing, we have to start & stop the server to execute you test.

So to overcome the all issues & increase the scope of Selenium RC, introduced new version of SE called Selenium WebDriver. WebDirver is come up with the some cool features. Also supports the multiple languages. Main feature over the Selenium RC is that we don’t have to start the server in the Selenium WebDriver. One of the cool feature is that it supports the Android Testing & iPhone testing as well.

The code of WebDriver look different than RC & IDE, it allows you to convert the IDE code to WD & RC code. As IDE supports with the user interface but WebDirver & RC does not have UI, we have to use core programming language in it.

# What are difference between Selenium IDE, RC and WebDriver

|  |  |  |
| --- | --- | --- |
| **Selenium IDE** | **Selenium RC** | **Selenium WebDriver** |
| It only works in Mozilla browser. | It supports with all browsers like Firefox, IE, Chrome, Safari, Opera etc. | It supports with all browsers like Firefox, IE, Chrome, Safari, Opera etc. |
| It supports Record and playback | It doesn’t supports Record and playback | It doesn’t supports Record and playback |
| Doesn’t required to start server before executing the test script. | Required to start server before executing the test script. | Doesn’t required to start server before executing the test script. |
| It is a GUI Plug-in | It is standalone java program which allow you to run Html test suites. | It actual core API which has binding in a range of languages. |
| Core engine is Javascript based | Core engine is Javascript based | Interacts natively with browser application |
| Very simple to use as it is record & playback. | It is easy and small API | As compared to RC, it is bit complex and large API. |
| It is not object oriented | API’s are less Object oriented | API’s are entirely Object oriented |
| It doesn’t supports of moving mouse cursors. | It doesn’t supports of moving mouse cursors. | It supports of moving mouse cursors. |
| Need to append full xpath with ‘xpath=\\’ syntax | Need to append full xpath with ‘xpath=\\’ syntax | No need to append full xpath with ‘xpath=\\’ syntax |
| It does not supports listeners | It does not supports listeners | It supports the implementation of listeners |
| It does not support to test iphone/Android applications | It does not support to test iphone/Android applications | It support to test iphone/Android applications |

# What all different testing Frameworks can be used along with Selenium?

When we use selenium, then we make scripts like Script1, Script2, Script3… etc. & execute the script. Sometimes to execute script we have to get the test data from XLS file or user. To read the data from XLS file, the process of reading the data from XLS file is called Parameterization. Along with this you have to generate the test reports, we need to know what happened after executing the script, is script Passed or Failed? Also along with the reports you need to add logging as well. If your script is taking more time to execute script then you need to know what happened at each and every minute, you need to log each & everything to get idea what script is doing & also at what point script is failed & why.

So we need a centralized controller which will read the test data, execute the test cases, generate reports & do the logging as well. In the market two frameworks are available for testing. So that the centralize testing controller are **TestNG** OR**JUnit framework**. You can use selenium with TestNG or JUnit framework. These frameworks will execute the test scripts. They will read the data from XLS file generate the reports & also do the logging while executing the scripts. How to started with these testing frameworks can be seen in next couple of articles to get hands on it.

Differences between selenium 2 & 3

The major difference between Selenium 2 and Selenium 3 is, Selenium 3 has bug fixes from selenium 2. Also, Selenium 3 is more mobile automation focused.

Here is the summary of the change from Official Selenium Blog.

* For WebDriver users, it's more of bug fixes and drop-in replacement for 2.x
* Selenium Grid bug fixes are done as well.
* Selenium project will not actively support only the WebDriver API.
* The Selenium RC APIs have been moved to a “legacy” package.
* The original code powering Selenium RC has been replaced with something backed by WebDriver, which is also contained in the "legacy" package.
* Mozilla has made internals of Firefox browser more stable. So community provided Firefox Driver will no longer work. That means that from Firefox 48 you must use their geckodriver to use that browser, regardless of whether you're using Selenium 2 or 3.