**What Is Selenium?**

Selenium is an open-source tool that is used for automating the tests carried out on web browsers (Web applications are tested using any web browser).

The WebDriver is the core component of Selenium, acting as a bridge between your code and the web browser, allowing you to control the browser and interact with web elements. Selenium works with a variety of browsers, including Chrome, Firefox, Safari, and Edge.Selenium is widely used for quality assurance in the software industry, particularly in web application testing. It has become a standard tool for web developers and testers working on projects.

Wait, before you get carried away, let me re-iterate that, only testing of web applications is possible with Selenium. We can neither test any desktop (software) application nor test any mobile application using Selenium.

It’ a bummer, right? I can feel your pain. But don’t worry, there are many tools for testing software and mobile applications like: IBM’s RFT, HP’s QPT, Appium and many more. But, the focus of this blog is, testing dynamic web applications and why Selenium is the best for that purpose.

**What are the advantages of Selenium?**

Since Selenium is open-source, there is no licensing cost involved, which is a major advantage over other testing tools. Other reasons behind Selenium’s ever growing popularity are:



* + Test scripts can be written in any of these programming languages: [**Java**](https://www.edureka.co/blog/java-tutorial/), [**Python**](https://www.edureka.co/blog/videos/python-tutorial/), [**C#**](https://www.edureka.co/blog/c-sharp-tutorial/), [**PHP**](https://www.edureka.co/blog/php-tutorial-for-beginners/), [**Ruby**](https://www.edureka.co/blog/ruby-on-rails-tutorial/), [**Perl**](https://www.edureka.co/blog/videos/learn-perl-the-jewel-of-scripting-languages/) & **.Net**
  + Tests can be carried out in any of these OS**:** **Windows**, **Mac** or **Linux**
  + Tests can be carried out using any browser: **Mozilla Firefox**, **Internet Explorer**, **Google Chrome**, **Safari** or **Opera**
  + It can be integrated with tools such as [**TestNG**](https://www.edureka.co/blog/dataprovider-in-testng/) & [**JUnit**](https://www.edureka.co/blog/junit-tutorial/) for managing test cases and generating reports
* It can be integrated with [**Maven**](https://www.edureka.co/blog/create-selenium-maven-project), [**Jenkins**](https://www.edureka.co/blog/jenkins-tutorial/) & **Docker** to achieve Continuous Testing

**Limitations of Selenium Testing**

Selenium has a few flaws, which can include:

* Because Selenium is open-source, it lacks a developer community and, as a result, reliable technical support.
* Selenium does not support testing mobile or desktop applications.
* Selenium provides only limited support for image testing.
* Selenium provides only limited support for test management. Selenium is frequently used in conjunction with tools such as JUnit and
* TestNG for this purpose.
* To use Selenium, you may need to be familiar with programming languages.

But there surely has to be shortcomings right?

* We can use Selenium only to test web applications. We cannot test desktop applications or any other software
* There is no guaranteed support available for Selenium. We need to leverage on the available customer communities
* It is not possible to perform testing on images. We need to integrate Selenium with Sikuli for image based testing
* There is no native reporting facility. But we can overcome that issue by integrating it with frameworks like TestNG or JUnit

Before going any further in this *what is Selenium* blog, you ought to know the story behind how Selenium came into being what it is today. So, let’s understand the scenario in the below order:

* 1. [Need for software testing](https://www.edureka.co/blog/what-is-selenium/?utm_source=youtube&utm_medium=youtube-description&utm_campaign=selenium-tutorial-video#NeedForSoftwareTesting)
  2. [Challenges with manual testing](https://www.edureka.co/blog/what-is-selenium/?utm_source=youtube&utm_medium=youtube-description&utm_campaign=selenium-tutorial-video#ChallengesManualTesting)
  3. [How automation testing beats manual testing?](https://www.edureka.co/blog/what-is-selenium/?utm_source=youtube&utm_medium=youtube-description&utm_campaign=selenium-tutorial-video#AutomationTestingBeatsManualTesting)
  4. [Selenium vs. other testing tools?](https://www.edureka.co/blog/what-is-selenium/?utm_source=youtube&utm_medium=youtube-description&utm_campaign=selenium-tutorial-video#SeleniumQTPRFT)
  5. [Selenium suite of tools](https://www.edureka.co/blog/what-is-selenium/?utm_source=youtube&utm_medium=youtube-description&utm_campaign=selenium-tutorial-video#SeleniumSuiteOfTools)

**Need For Software Testing**

[Software testing](https://www.edureka.co/blog/what-is-software-testing/) is where it all boils down to. Today’s world of technology is completely dominated by machines, and their behavior is controlled by the software powering it. Will the machines behave exactly as we want them to? Everytime? Everywhere? The answer to these questions lie in software testing.

At the end of the day, it is the software application’s success rate which is going to control your business growth. The same thing can be said even for web applications because most businesses today are completely reliant on the internet.

Take for example, any e-commerce company. Be it Amazon or E-Bay or Flipkart, they rely on the customer traffic on their web sites and traffic on their web based mobile applications for business.

Imagine, if something catastrophic happens like the prices of a number of products being capped off at 10$, all because of a small bug in a “not so easily readable” part of the code. Then what can be done, and how can we prevent it the next time?

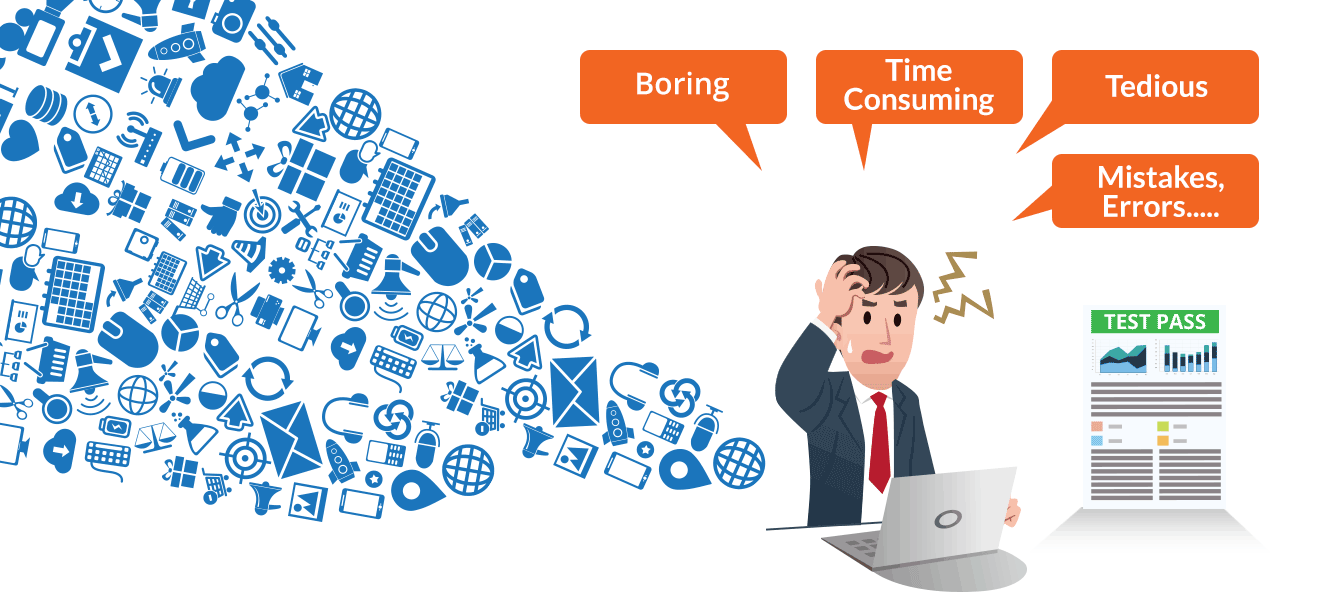
By testing the code before deployment right? So, that is the need for software testing.  But, what is Selenium? Is it a software testing tool? Well, Selenium is an automation testing tool!

**Related Article:** [Software Testing Interview Questions](https://www.edureka.co/blog/interview-questions/software-testing-interview-questions/)

Before I go any further, let me clear out that, Software testing is of two types: Manual Testing & Automation Testing. Selenium was founded as an automation testing tool to overcome the drawbacks/ limitations of Manual testing. So, in the next section of this what is selenium blog, let’s understand the challenges with manual testing.

**Challenges With Manual Testing**

Manual testing means the (web) application is tested manually by QA testers. Tests need to be performed manually in every environment, using a different data set and the success/ failure rate of every transaction should be recorded.

Look at the above image of a poor chap, who manually verifies the transactions recorded. The challenges he is facing cause fatigue, boredom, delay in work, mistakes and errors because of manual effort. This lead to the invention of  Selenium (automation testing tool).

**Automation Testing Beats Manual Testing**

Automation testing beats manual testing every time. Why? Because it is faster, needs less investment in human resource, it is not prone to errors, frequent execution of tests is possible, supports lights out execution, supports regression testing and also functional testing.

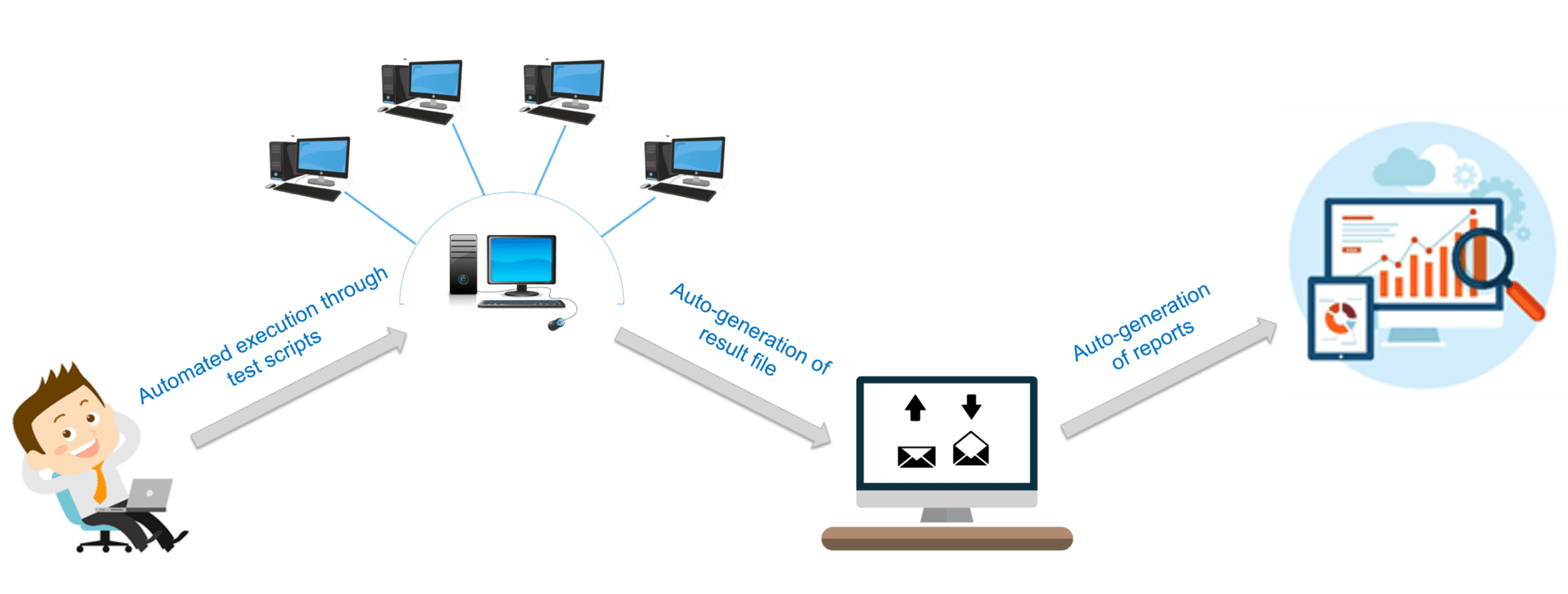
Let’s take a similar example to the one mentioned earlier. Suppose there is a login page and we need to verify if all the login attempts are successful, then it will be really easy to write a piece of code which will validate if all the transaction/ login attempts are a success or not (automated test case execution).

Moreover, these tests can be configured in such a way that they are tested in different environments and web browsers. What else can be done? You can automate the generation of result file, by scheduling it for a particular time during the day. Then you can also automate the generation of reports based on those results and what not.

**Tips for Selenium Automation Testing:**

* For better test organization and maintenance, use the Page Object Model (POM).
* To make scripts more robust, use proper exception handling.
* Consider parallel testing to achieve faster results.
* Update browser drivers on a regular basis to ensure compatibility.
* Use testing frameworks to implement advanced features such as parameterization, grouping, and reporting.
* Remember that Selenium is primarily intended for web applications, and that for other types of applications (e.g., desktop or mobile),
* additional tools such as Appium may be required.

The key point is that automation testing makes a tester’s job a whole lot simpler. Check out the image below which shows a more relaxed environment in which the same tester is working.



Now, let me talk about Selenium in particular.

Now let us see where Selenium stands in the market.

**Selenium vs QTP vs RFT**

I have compared its performance with two other popular tools: QTP and RFT in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Features** | **HP QTP** | **IBM RFT** | **Selenium** |
| License | Required | Required | Open-source |
| Cost | High | High | Open-source software |
| Customer Support | Dedicated HP support | Dedicated IBM support | Open-source Community |
| Hardware consumption during script execution | High | High | Low |
| Coding experience | Not required | Required | Ample amount of coding skills and experience needed |
| Environment support | Only for Windows | Only for Windows | Windows, Linux, Solaris OS X (If browser & JVM or JavaScript support exists) |
| Language Support | VB Script | Java and C# | Java, C#, Ruby, Python, Perl, PHP and JavaScript |

It is pretty clear from the above table why Selenium is the most preferred tool. But there are many flavors in Selenium and you should know which is the appropriate right Selenium tool for your need.

**Selenium Suite Of Tools**

* Selenium RC (Now depreciated)
* Selenium IDE
* Selenium Grid
* Selenium WebDriver