What is WebDriver?

- WebDriver is a web automation framework that allows you to execute your tests against different browsers, not just Firefox, Chrome (unlike Selenium IDE) but also browsers like Safari, Opera browser, Internet Explorer.

- WebDriver also enables you to **use a programming language** in creating your test scripts (not possible in Selenium IDE). Useable languages: JAVA, Python, Perl, .NET, PHP, Ruby

- You can now use **conditional operations** like if-then-else or switch-case. You can also perform looping like do-while.

Definition of 'Selenium Web Driver'

- **Definition:** Selenium WebDriver is a collection of open source APIs which are used to automate the testing of a web application.
- **Description:** Selenium WebDriver tool is used to automate web application testing to verify that it works as expected. It supports many browsers such as Firefox, Chrome, IE, and Safari. However, using the Selenium WebDriver, we can automate testing for web applications only. It does not qualify for window-based applications. It also supports different programming languages such as C#, Java, Perl, PHP and Ruby for writing test scripts. Selenium WebDriver is platform-independent since the same code can be used on different Operating Systems like Microsoft Windows, Apple OS and Linux. It is one of the components of the selenium family, which also includes Selenium IDE, Selenium Client API, Selenium Remote Control and Selenium Grid.

Why Selenium

- – Selenium is an open source tool with Corporate backing.
- – The tests can then be run against most modern web browsers.
- – Selenium deploys on Windows, Linux, and Macintosh platforms.
- – It allows recording, editing, and debugging tests.
- – Recorded tests can be exported in most language e.g. html, Java, .net, perl, ruby etc.
- – Selenium has the support of some of the largest browser vendors who have taken (or are taking) steps to make Selenium a native part of their browser.

Difference between Selenium RC and Webdriver

Before the advent of WebDriver in 2006, there was another, **automation tool called Selenium Remote Control.** Both WebDriver and Selenium RC have following features:

- They both allow you to **use a programming language** in designing your test scripts.
- They both allow you to **run your tests against different browsers.**

So how do they differ? Let us discuss the answers.

1. Architecture

WebDriver's architecture is simpler than Selenium RC's.

- It controls the browser from the OS level
- All you need are your programming language's IDE (which contains your Selenium commands) and a browser.
  Selenium RC's architecture is way more complicated.

- You first need to launch **a separate application called Selenium Remote Control (RC) Server** before you can start testing
- The Selenium RC Server **acts as a "middleman" between your Selenium commands and your browser**
- When you begin testing, Selenium RC Server "injects" a **Javascript program called Selenium Core** into the browser.
- Once injected, Selenium Core will start receiving instructions relayed by the RC Server from your test program.
- When the instructions are received, **Selenium Core will execute them as Javascript commands.**
- The browser will obey the instructions of Selenium Core and will relay its response to the RC Server.
- The RC Server will receive the response of the browser and then display the results to you.
- RC Server will fetch the next instruction from your test script to repeat the whole cycle.

2. Speed

- **WebDriver is faster than Selenium RC since it** speaks directly to the browser uses the browser's own engine to control it.

- **Selenium RC is slower since it uses a Javascript program called Selenium Core.** This Selenium Core is the one that directly controls the browser, not you.

3. Real-life Interaction

WebDriver interacts with page elements in a more realistic way. For example, if you have a disabled text box on a page you were testing, WebDriver really cannot enter any value in it just as how a real person cannot.

This is what happens in Selenium RC

Selenium Core, just like other JavaScript codes, can access disabled elements. In the past, Selenium testers complain that Selenium Core was able to enter values to a disabled text box in their tests. Differences in API

4. API

Selenium RC's API is more matured but contains redundancies and often confusing commands. For example, most of the time, testers are confused whether to use type or typeKeys; or whether to use click, mouseDown, or mouseDownAt.
Worse, **different browsers interpret each of these commands in different ways too!**
WebDriver's API is simpler than Selenium RC's. It does not contain redundant and confusing commands.

5. **Browser Support** WebDriver can support the headless HtmlUnit browser

HtmlUnit is termed as "headless" because it is an invisible browser - it is GUI-less.

It is a very fast browser because no time is spent in waiting for page elements to load. This accelerates your test execution cycles.

Since it is invisible to the user, it can only be controlled through automated means.

Selenium RC cannot support the headless HtmlUnit browser. It needs a real, visible browser to operate on.

Limitations of WebDriver

WebDriver Cannot Readily Support New Browsers

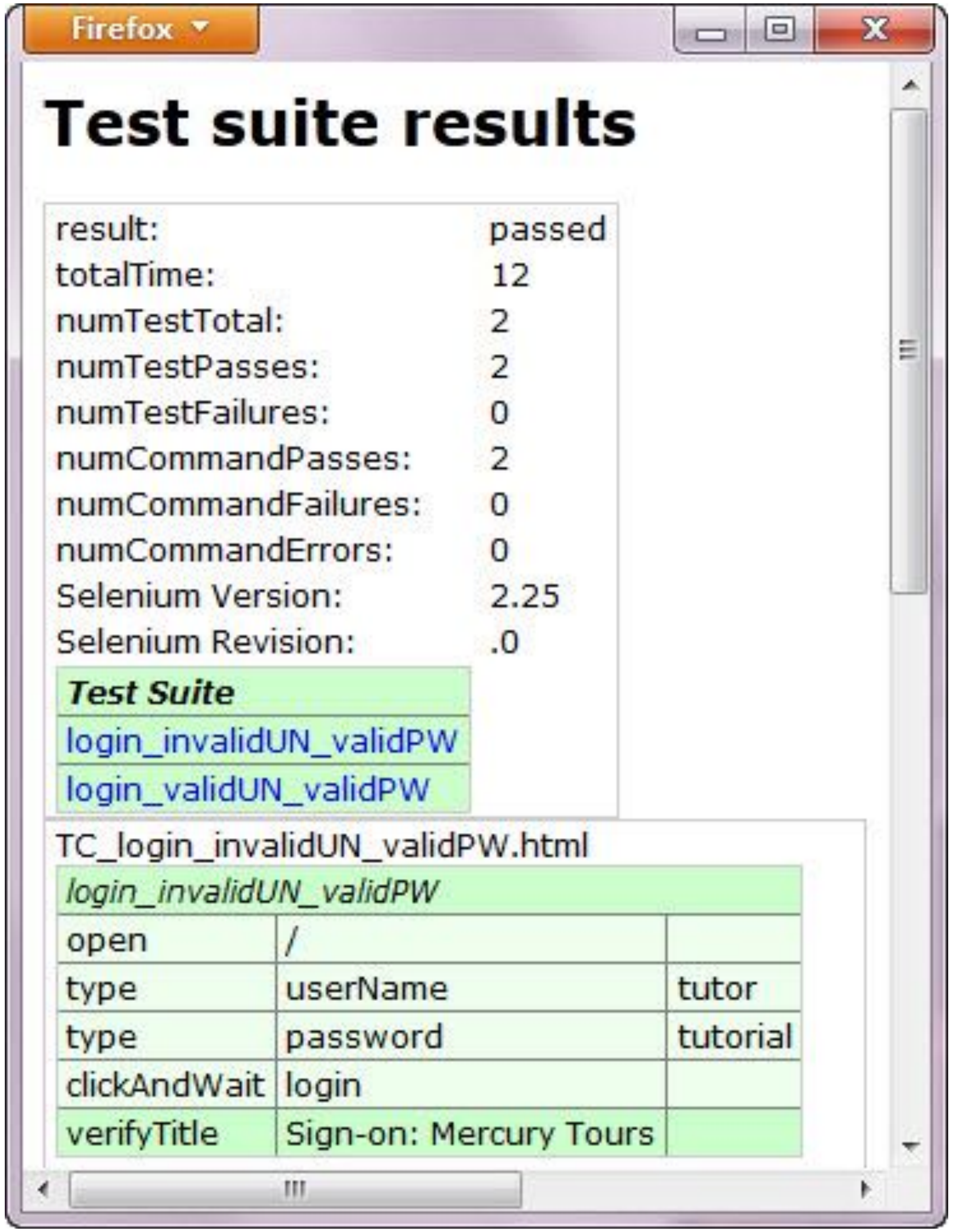
Remember that WebDriver operates on the OS level. Also, remember that different browsers communicate with the OS in different ways. If a new browser comes out, it may have a different process of communicating with the OS as compared to other browsers. So, **you have to give the WebDriver team quite some time to figure that new process out**before they can implement it on the next WebDriver release.

However, it is up to the WebDriver's team of developers to decide if they should support the new browser or not.

Selenium RC Has Built-In Test Result Generator

Selenium RC automatically generates an HTML file of test results. The format of the report was pre-set by RC itself. Take a look at an example of this report below.

WebDriver has no built-in command that automatically generates a Test Results File. You would have to rely on your IDE's output window, or design the report yourself using the capabilities of your programming language and store it as text, HTML, etc.



Test suite results

result:	passed
totalTime:	12
numTestTotal:	2
numTestPasses:	2
numTestFailures:	0
numCommandPasses:	2
numCommandFailures:	0
numCommandErrors:	0
Selenium Version:	2.25
Selenium Revision:	.0

Test Suite
login_invalidUN_validPW
login_validUN_validPW

TC_login_invalidUN_validPW.html

login_invalidUN_validPW		
open	/	
type	userName	tutor
type	password	tutorial
clickAndWait	login	
verifyTitle	Sign-on: Mercury Tours	

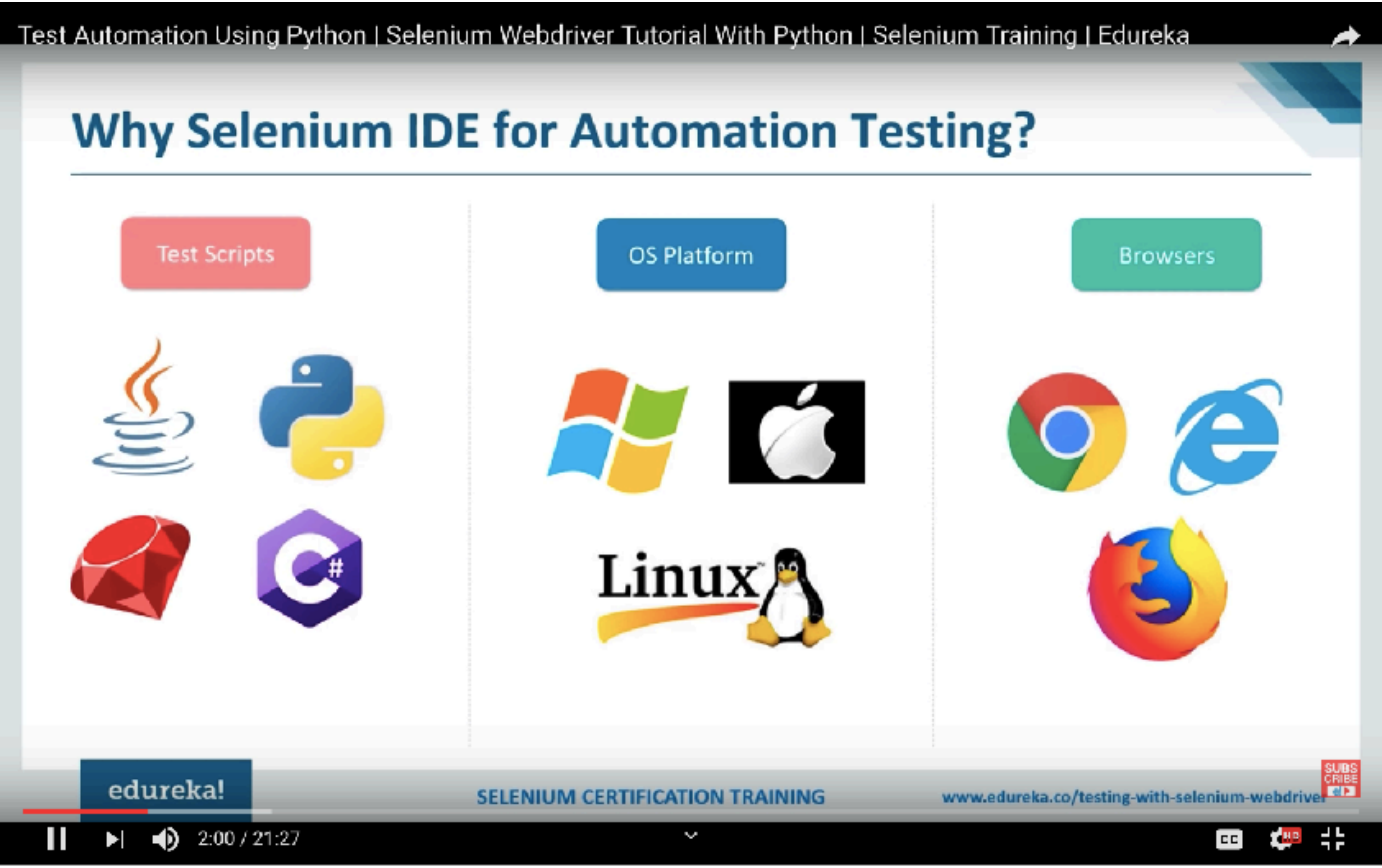
Summary

- WebDriver is a tool for testing web applications **across different browsers** using different programming languages.
- You are now able to make powerful tests because WebDriver **allows you to use a programming language** of your choice in designing your tests.
- WebDriver is **faster than Selenium RC** because of its simpler architecture.
- WebDriver **directly talks to the browser** while Selenium RC needs the help of the RC Server in order to do so.
- WebDriver's API is more **concise** than Selenium RC's.
- WebDriver **can support HtmlUnit** while Selenium RC cannot.
- The only drawbacks of WebDriver are:
    - It **cannot readily support new browsers**, but Selenium RC can.
    - It **does not have a built-in command** for automatic generation of test results.

NOTE:

Gecko Driver is an executable file that you need to have in one of the system path before starting your tests.

Why python for automation testing?
- Python is extremely simple and easy to learn and it closely resembles the english language.
- Python is free and open source. It is high level, interpreted and blessed with a large community and python has many built- in testing frameworks that covers debugging and fastest workflows.
- There are lots of tools and modules to make things easier such as selenium and splinter and it also supports testing with cross platform and cross browser with frameworks such as fighters, and robot frameworks.



Test Automation Using Python | Selenium Webdriver Tutorial With Python | Selenium Training | Edureka

Why Selenium IDE for Automation Testing?

edureka!

SELENIUM CERTIFICATION TRAINING

www.edureka.co/testing-with-selenium-webdriver

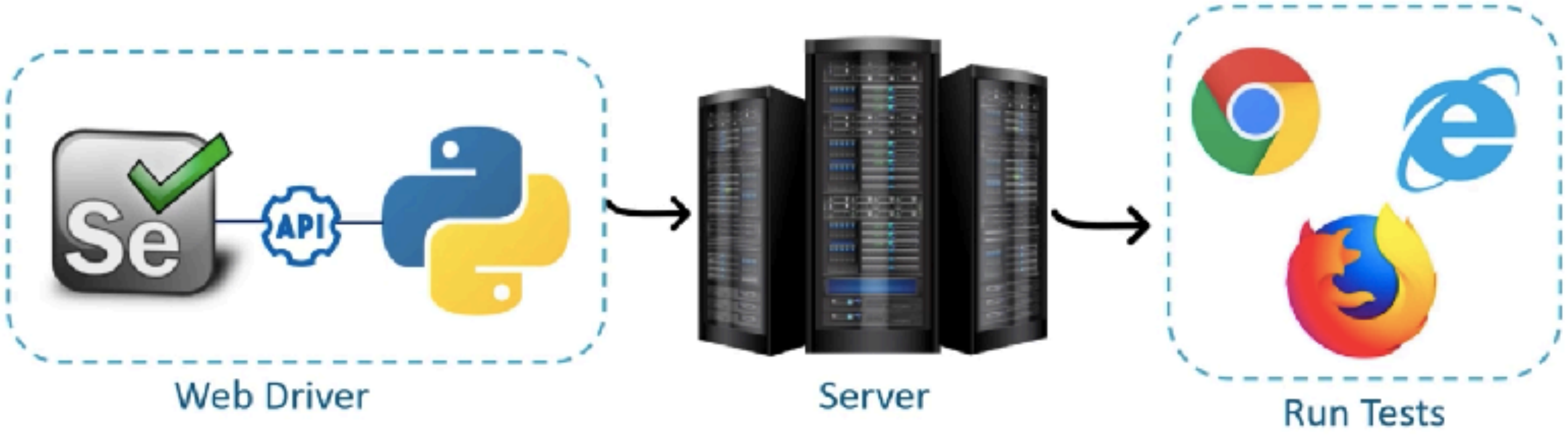
Why Python for Automation Testing?

```
a=3
b=5
Sum=a+b
```

- Open Source
- High-level
- Interpreted
- Large community

Selenium Python Binding

Selenium Python bindings provides a simple API to write functional tests using Selenium WebDriver



to selenium server and selenium server
then runs the test cases on