**Link: 1.**[**https://www.youtube.com/watch?v=UoMAEIVAfAM&list=PLsGOlyTzNH6f\_pZuGZDFjI5IeCWGQqiwF&index=1**](https://www.youtube.com/watch?v=UoMAEIVAfAM&list=PLsGOlyTzNH6f_pZuGZDFjI5IeCWGQqiwF&index=1)

**2.**[**https://www.automationtestinginsider.com/p/selenium-vi.html**](https://www.automationtestinginsider.com/p/selenium-vi.html)

**Selenium Basics**

[**Selenium and it’s History**](https://www.automationtestinginsider.com/2019/07/selenium-webdriver-part1-selenium-and.html)

**What is Selenium?**  
  
Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms.  
  
Selenium is not just a single tool but a suite of software's, each catering to different testing needs of an organization. It has four components.  
  
**Selenium Integrated Development Environment (IDE)**  
**Selenium Remote Control (RC)**  
**WebDriver**  
**Selenium Grid**  
  
**Selenium Core**  
The story starts in 2004 at ThoughtWorks in Chicago, with Jason Huggins building the Core mode as "JavaScriptTestRunner“. Its JavaScript program that would automatically control the browser's actions.

|  |
| --- |
| [A person with his hand on his chin  Description automatically generated with medium confidence](https://1.bp.blogspot.com/-LQKw4rLY02Q/XkwJaQ2fI3I/AAAAAAAAATE/uG-Ta5UvVhEslHZFXWdXke1goZvwApaHwCLcBGAsYHQ/s1600/James-Huggins.webp) |
| **Jason Huggins** |

“JavaScriptTestRunner” was later named as “Selenium Core” and released into the market as an Open Source tool.  
  
This Open Source tool started gaining demand in the market and people started using it for automating the repeated tasks in their Web Applications.  
  
**Selenium Remote Control**  
  
Unfortunately; testers using Selenium Core had to install the whole application under test and the web server on their own local computers because of the restrictions imposed by the same origin policy. To resolve this another ThoughtWork's engineer, Paul Hammant created system (in 2007) known as the Selenium Remote Control or Selenium 1.  
  
Selenium 1 = Selenium IDE + Selenium RC + Selenium Grid

|  |
| --- |
| [A person smiling for the camera  Description automatically generated with medium confidence](https://1.bp.blogspot.com/-r3ZHCqJ5Gxw/XkwNa92c9BI/AAAAAAAAATU/UtogXTefkXsnPBGcn3W2JKpjyEPpvqhQACLcBGAsYHQ/s1600/Paul-Hammant.webp) |
| **Paul Hammant** |

**Selenium Grid**

Patrick Lightbody to address the need of minimizing test execution times as much as possible, So he created Selenium Grid. Basically grid is for parallel execution and execute your test scripts on multiple environments.

|  |
| --- |
| [A person wearing glasses  Description automatically generated with low confidence](https://1.bp.blogspot.com/-U3y5fWIbOns/XkwOXkeYJeI/AAAAAAAAATg/oxCUGIMpEIAmw6AApZKbXpvk1S_sYbTfQCLcBGAsYHQ/s1600/Patric-Lightbody.webp) |
| **Patrick Lightbody** |

Using “Selenium Grid”, testers were able to distribute the tests across multiple machines and get them executed them on different machines over their network to reduce or minimize the time taken for overall execution of tests.  
  
**Selenium IDE**  
  
“Shinya Kasatani”, who developed a Firefox extension named as “Selenium IDE”.  
  
“Selenium IDE” using its record and playback feature, records the automation tests like recording a video and executes the recorded tests like playing the recorded videos.

|  |
| --- |
|  |
| **Shinya Kasatani** |

**Selenium WebDriver**  
  
Earlier Selenium 1 used to be the major project of Selenium.  
  
**Selenium 1** = Selenium IDE + Selenium RC + Selenium Grid  
  
Later Selenium Team has decided to merge both Selenium WebDriver and Selenium RC to form a more powerful Selenium tool.  
  
They both got merged to form “Selenium 2”  
  
“Selenium WebDriver” was the core of “Selenium 2” and “Selenium RC” used to run in maintenance mode.  
  
Hence **Selenium 2** = Selenium IDE + Selenium WebDriver 2.x + Selenium Grid.  
  
“Selenium 2” released on July 8, 2011.  
  
Selenium team has decided to completely remove the dependency for Selenium RC.  
  
After 5 years, “Selenium 3" was released on October 13, 2016 with a major change, which is the original Selenium Core implementation and replacing it with one backed by WebDriver and lot more improvements.

Hence **Selenium 3** = Selenium IDE + Selenium WebDriver 3.x + Selenium Grid.  
  
After 3 years from it’s a major release, now Selenium has put out its first alpha version of Selenium 4 on Apr 24, 2019. Still, there is no official announcement about the release date of Selenium 4, but we are expecting it around October 2019. Till that there can be several alpha or beta versions released time to time with stabilization.  
  
[**Download and Install Java and Eclipse**](https://www.automationtestinginsider.com/2019/07/selenium-webdriver-download-and-install.html)

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented. It is intended to let application developers "write once, run anywhere.  
  
Let's understand what is JDK, JRE and JVN in brief.  
  
JDK  
Java Development Kit is the core component of Java Environment and provides all the tools, executables and binaries required to compile, debug and execute a Java Program. JDK is a platform-specific software and that’s why we have separate installers for Windows, Mac, and Unix systems. We can say that JDK is the superset of JRE since it contains JRE with Java compiler, debugger, and core classes.  
  
JRE  
JRE is the implementation of JVM, it provides a platform to execute java programs. JRE consists of JVM and java binaries and other classes to execute any program successfully. JRE doesn’t contain any development tools like java compiler, debugger etc.  
  
JVM  
JVM is the heart of Java programming language. When we run a program, JVM is responsible for converting Byte code to the machine specific code. JVM is also platform dependent and provides core java functions like memory management, garbage collection, security etc.

[**First Selenium Program**](https://www.automationtestinginsider.com/2019/07/selenium-webdriver-first-selenium.html)

Prerequisites to run first simple WebDriver Program are:  
  
1. Java IDE (Eclipse/ IntelliJ) - Environment that you will use to write your code in.  
2. Selenium Webdriver libraries - Libraries which allows you use all the Selenium functions and classes.  
3. Browser drivers (Chrome/ Firefox/IE) - The driver with with Selenium WebDriver communicates.

[**Selenium WebDriver Architecture**](https://www.automationtestinginsider.com/2019/07/selenium-webdriver-selenium-webdriver.html)

There are four components of **Selenium Architecture**:  
  
Selenium Client Library  
JSON Wire Protocol over HTTP  
Browser Drivers  
Browsers  
  
**Selenium Client Library:**  
  
Selenium supports multiple libraries such as Java, Ruby, Python, etc., Selenium Developers have developed language bindings to allow Selenium to support multiple languages.  
Below is the link to download Selenium Client Language Bindings:  
  
<https://www.seleniumhq.org/download/#client-drivers>  
  
**JSON Wire Protocol over HTTP:**  
  
JSON stands for JavaScript Object Notation. It is used to transfer data between a server and a client on the web. JSON Wire Protocol is a REST API that transfers the information between HTTP server. Each BrowserDriver (such as FirefoxDriver, ChromeDriver etc.,)  has its own HTTP server.  
  
**Browser Drivers:**  
  
Each browser contains separate browser driver. Browser drivers communicate with respective browser without revealing the internal logic of browser’s functionality. When a browser driver is  received any command then that command will be executed on the respective browser and the response will go back in the form of HTTP response.  
  
**Real Browsers:**  
  
Selenium supports multipe browsers such as Firefox, Chrome, IE, Safari etc.  
  
**How Selenium Works Internally:**  
  
Lets say you have written below couple of lines of code:  
  
WebDriver driver  = new ChromeDriver();  
driver.get(https://www.automationtestininsider.com)  
  
Based on the above statements, Chrome browser will be launched and it will navigates to automationtestinginsider website.  
  
Once you Run the program, every statement in your script will be converted as a URL with the help of JSON Wire Protocol over HTTP. The URL’s will be passed to the Browser Drivers. (In this case, ChromeDriver). Here in our case the client library (java) will convert the statements of the script to JSON format and communicates with the ChromeDriver. URL looks as shown below.  
  
http://localhost:8080/{"url":"https://www.automationtestinginsider.com"}  
  
Every Browser Driver uses a HTTP server to receive HTTP requests.  Once the URL reaches the Browser Driver, then the Browser Driver will pass that request to the real browser over HTTP. Then the commands in your selenium script will be executed on the browser.  
  
If the request is POST request then there will be an action on browser  
  
If the request is a GET request then the corresponding response will be generated at the browser end and it will be sent over HTTP to the browser driver and the Browser Driver over JSON Wire Protocol and sends it to the UI (Eclipse IDE).  
  
This is how Selenium Commands works internally.

[**Launch Chrome, FireFox and IE browsers with Selenium WebDriver**](https://www.automationtestinginsider.com/2019/09/launch-chrome-firefox-and-ie-browsers.html)

Prerequisites to run Selenium Scripts on Different Browsers:  
 **Programming Language** – Java  
https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html  
  
**IDE (Code Editor)** -  Eclipse  
https://www.eclipse.org/downloads/  
  
**Browser Drivers** – In order to communicate with actual browsers, selenium need something called driver. Every browser has its own driver.  
https://www.seleniumhq.org/download/  
  
**Selenium WebDriver Libraries** – Libraries which allow you to use all the selenium classes and functions  
https://www.seleniumhq.org/download/  
  
**Browser** – Chrome. Firefox, IE  
Application Under Test – Orange HRM (https://opensource-demo.orangehrmlive.com/)

**Basic Script:**

package launchBrowser;  
import org.openqa.selenium.By;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.chrome.ChromeDriver;  
  
public class FirstSeleniumScriptChrome {  
   
 WebDriver driver;  
   
 public void launchBrowser() {  
 System.setProperty("webdriver.chrome.driver", "D://chromedriver.exe");  
 driver=new ChromeDriver();  
 System.out.println("Browser is Launched");  
 driver.manage().window().maximize();  
 driver.get("https://opensource-demo.orangehrmlive.com/");  
 System.out.println("Orange HRM url is opened");  
 }  
   
 public void login() {  
 driver.findElement(By.name("txtUsername")).sendKeys("Admin");  
 driver.findElement(By.id("txtPassword")).sendKeys("admin123");  
 driver.findElement(By.id("btnLogin")).click();  
 System.out.println("User is Logged in");  
 }  
   
 public void title() {  
 String title=driver.getTitle();  
 System.out.println("Title of the Page is:"+title);  
 }  
   
 public void logout() throws InterruptedException {  
 driver.findElement(By.id("welcome")).click();  
 Thread.sleep(2000);  
 driver.findElement(By.xpath("//\*[@id='welcome-menu']//\*[text()='Logout']")).click();  
 System.out.println("Logged out from Oranage HRM Website");  
 }  
   
 public void closeBrowser() {  
 driver.close();  
 System.out.println("Browser is Closed");  
 }  
  
 public static void main(String[] args) throws InterruptedException {  
   
 FirstSeleniumScriptChrome obj= new FirstSeleniumScriptChrome();  
 obj.launchBrowser();  
 obj.login();  
 obj.title();  
 obj.logout();  
 obj.closeBrowser();  
   
 }  
  
}