**Blog on Given Questions :**

1. **Difference between Selenium IDE, Selenium Webdriver and Selenium Grid.**

**Selenium Introduction:**

Selenium Introduction – It is an open source (free) automated testing suite to test web applications. It supports different platforms and browsers. It has gained a lot of popularity in terms of web-based automated testing and giving a great competition to the famous commercial tool HP QTP (Quick Test Professional) AKA HP UFT (Unified Functional Testing).

Selenium is a set of different software tools. Each tool has a different approach in supporting web based automation testing.

It has four components namely,  
i. **Selenium IDE (Integrated Development Environment)  
ii. Selenium RC (Remote Control)  
iii. Selenium WebDriver  
iv. Selenium Grid**

Here we see the difference between Selenium IDE, Selenium Webdriver and Selenium Grid.

**What is Selenium IDE?**

Selenium IDE (Integrated Development Environment) is a Firefox plugin. It is the simplest framework in the Selenium Suite. It allows us to record and playback the scripts. Even though we can create scripts using Selenium IDE, we need to use Selenium RC or Selenium WebDriver to write more advanced and robust test cases.

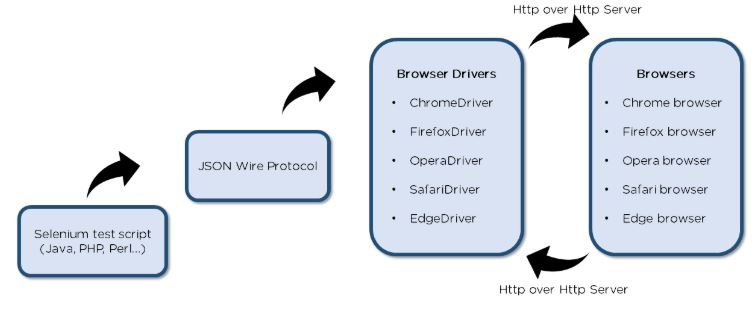
Selenium IDE has a few shortcomings:

1. It does not support data-driven testing
2. It can’t perform database testing
3. It can’t provide a detailed test report
4. It can’t export to WebDriver scripts

**What is Selenium WebDriver?**

Selenium WebDriver AKA Selenium 2 is a browser automation framework that accepts commands and sends them to a browser. It is implemented through a browser-specific driver. It controls the browser by directly communicating with it. Selenium WebDriver supports Java, C#, PHP, Python, Perl, Ruby.

The architecture of Selenium WebDriver is simple and easy to understand:



* Selenium test script - Selenium test script is the test code written in any programming language be it Java, Perl, PHP, or Python that can be interpreted by the driver.
* JSON Wire Protocol - JSON Wire Protocol provides a transport mechanism to transfer data between a server and a client. JSON Wire Protocol serves as an industry standard for various web services.
* Browser drivers - Selenium uses drivers specific to each browser to establish a secure connection with the browser.
* Browsers - Selenium WebDriver supports various web browsers to test and run applications on.

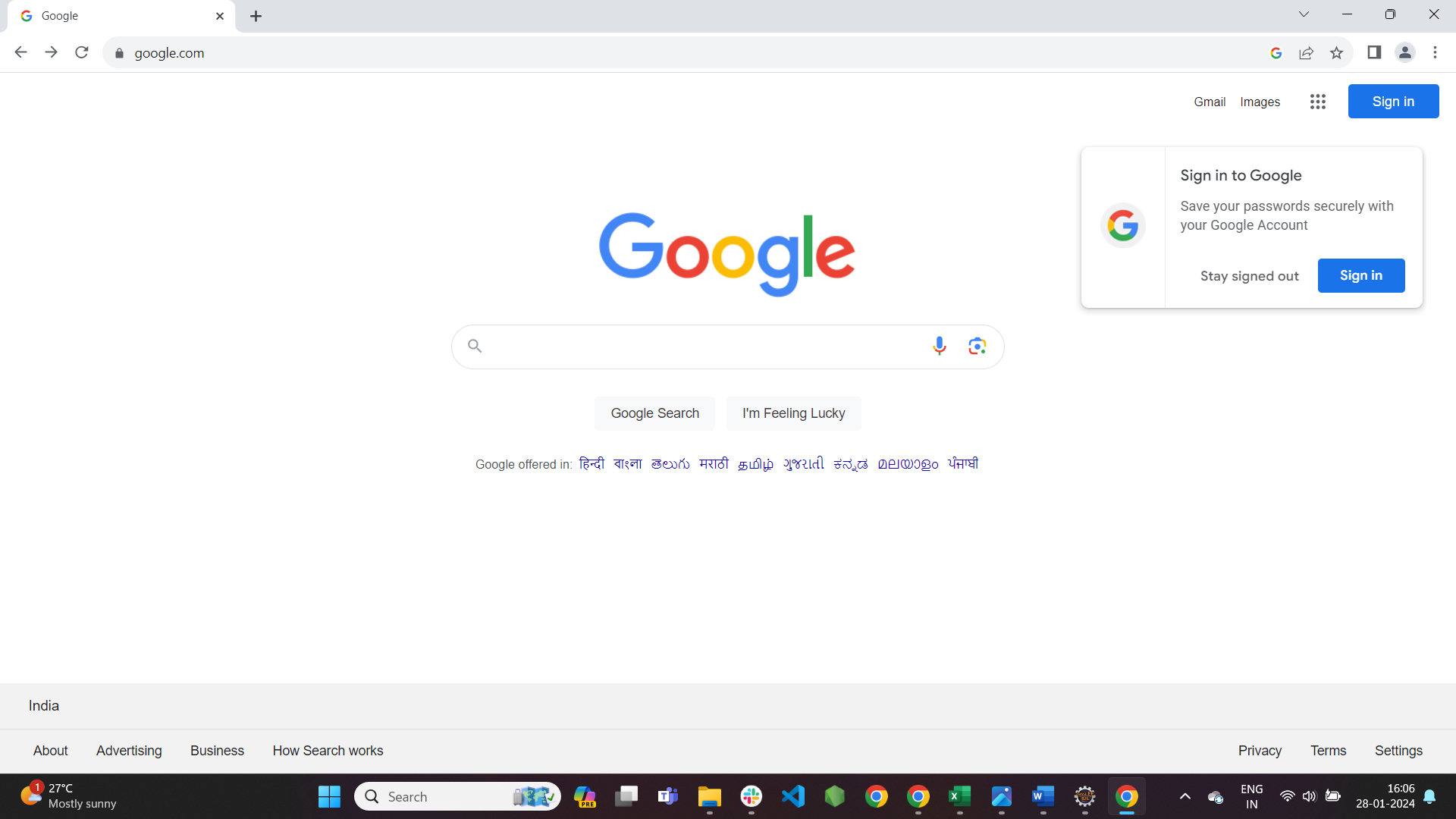
**What is Selenium Grid?**

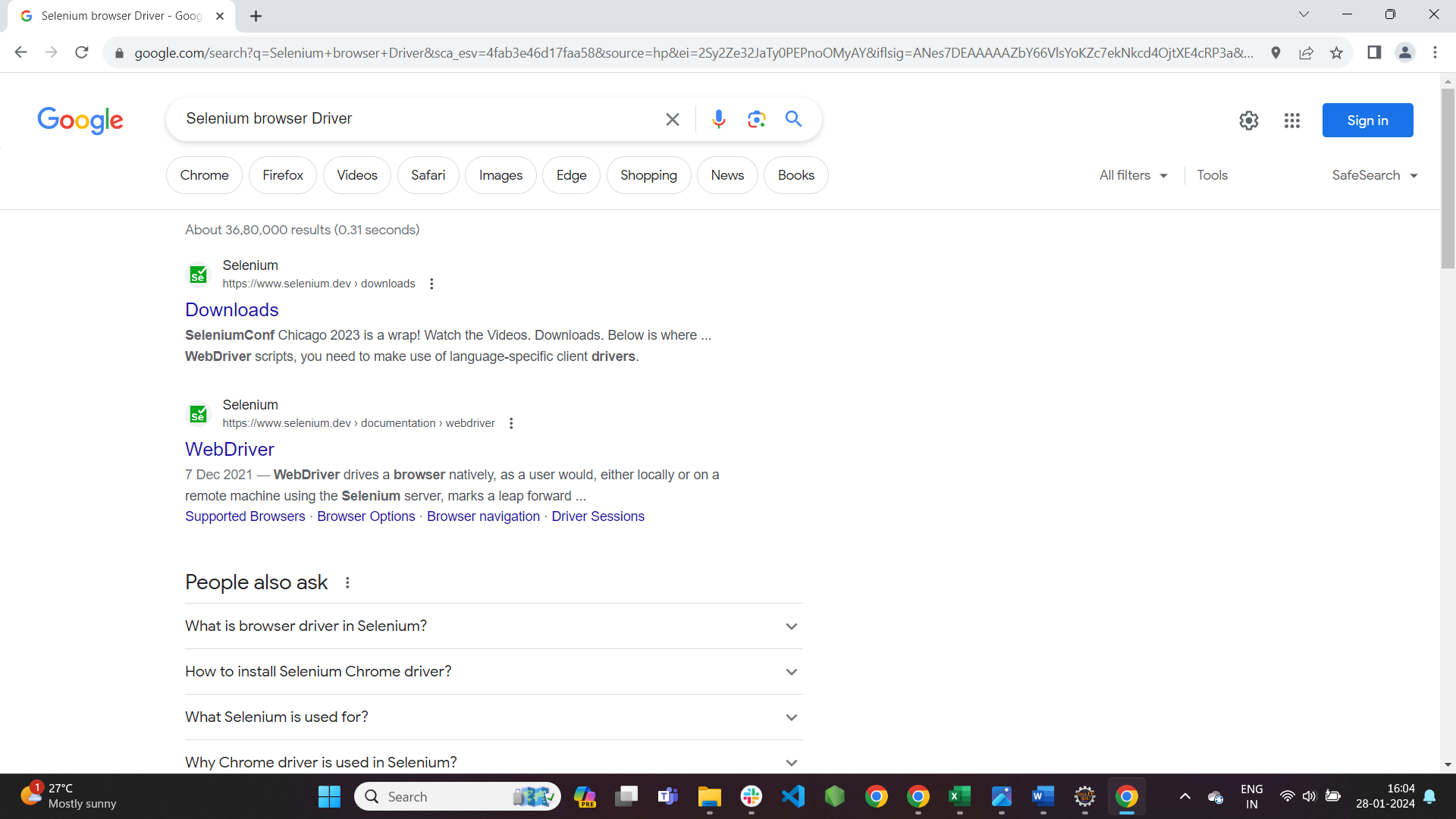
Selenium Grid is a tool used together with Selenium RC to run tests on different machines against different browsers in parallel. That is, running multiple tests at the same time against different machines running different browsers and operating systems.

The Grid consists of a hub connected to several nodes. It receives the test to be executed along with information about the operating system and browser to be run on and picks a node that conforms to the requirements (browser and platform), passing the test to that node. The node now runs the browser and executes the selenium commands within it.



**2. OUTPUT FOR SELENIUM SCRIPT IN JAVA TO OPEN GOOGLE AND SEARCH FOR “SELENIUM BROWSER DRIVER”.**





**3. What is Selenium?**

Selenium is an open-source, automated testing tool used to test web applications across various browsers. Selenium can only test web applications, unfortunately, so desktop and mobile apps can’t be tested. However, other tools like Appium and HP’s QTP can be used to test software and mobile applications.

## What makes Selenium Such a Widely Used Testing Tool?

1. Selenium is easy to use since it’s primarily developed in JavaScript
2. Selenium can test web applications against various browsers like Firefox, Chrome, Opera, and Safari
3. Tests can be coded in several programming languages like Java, Python, Perl, PHP, and Ruby
4. Selenium is platform-independent, meaning it can deploy on Windows, Linux, and Macintosh
5. Selenium can be integrated with tools like JUnit and TestNG for test management

## **Advantages of Using Selenium for Automated Testing**

## **1. Language and Framework Support**

When someone chooses a tool the first thing that comes to mind is: "Does my tool supports the language that I know?"

Well, this is not the case with Selenium as it supports all major languages like Java, Python, JavaScript, C#, Ruby, and Perl programming languages for software test automation.

You can write your scripts in any of these programming languages and Selenium converts it into Selenium compatible codes in no time. So, there is no need for knowing Selenium only languages. Also, every Selenium supported language has dedicated frameworks which help in writing test script for Selenium test automation. So, when you go for Selenium as a tool for performing automation testing, you don’t have to worry about language and framework support as Selenium does that for you!

## **2. Open Source Availability**

One of the many things that adds to the advantages of Selenium is its open source availability. So, being an open source tool, Selenium is a publicly accessible automation framework and is free, with no upfront costs. So, you can save bucks here and use them for other good causes.

The Selenium community is continuously helping developers and software engineers in automating the web browser features and functionalities. Selenium being open source also helps you customize the code for better code management and enhance the functionality of predefined functions and classes. Selenium has become the most reliable web automation tool because of the ease of generating test scripts to validate functionality.

## **3. Multi-Browser Support**

“One Selenium script for all browsers” is what the Selenium community has been working on and improvising every day. As per [StatCounter](http://gs.statcounter.com/browser-market-share), Chrome, Firefox, Safari, Internet Explorer, Opera, and Edge browsers are the most used browsers worldwide and Selenium script is compatible with all the mentioned browsers. You don’t need to rewrite scripts for every browser, just one script for all browsers.

## **4. Support Across Various Operating Systems**

Different people use different operating systems and it is necessary that your automation tool supports all of them. Selenium is yet a highly portable tool that supports and can work across different operating systems like Windows, Linux, Mac OS, UNIX, etc.

You can create Selenium test suites over any platform like Windows and can execute the same test suite on another platform, for example, Mac or Linux. This enables developers and software testers to easily write test automation scripts without laying much emphasis on the platform on which it will run.

## **5. Ease Of Implementation**

Selenium automation framework is very easy-to-use tool. Selenium provides a user-friendly interface that helps create and execute test scripts easily and effectively. You can also watch while tests are running. You can analyze detailed reports of Selenium tests and take follow-up actions.

And finally, you will never feel alone. A huge Selenium community is always available to help you in case of need. You can ask your queries and perform brainstorming in the community.

## **6. Reusability and Integrations**

As mentioned earlier, Selenium automation test suites are reusable and can be tested across multiple browsers and operating systems. However, the twist is if that Selenium is not an all-inclusive web automation testing tool. Hence, it needs third-party frameworks and add-ons to broaden the scope of testing.

For example, you need to integrate Selenium with [TestNG](https://testng.org/) and [JUnit](https://junit.org/junit5/) for managing test cases and generating reports. For achieving continuous testing, you’ll need to integrate it with some CI/CD tools like [Jenkins](https://jenkins.io/), Maven, and Docker. Also, for performing image-based testing, you need to integrate Selenium with tools like Sikuli, and for performing cross-browser testing with cloud-grid. You can integrate Selenium with almost all management tools.

## **7. Flexibility**

Test management is what which is very important in testing lifecycle. It becomes easier and more efficient with Selenium features like regrouping and refactoring of test cases. This helps developers and testers in quick changes to the code, reducing duplication, minimizing complications and improving maintainability. These features make Selenium more flexible and usable as compared to other automation testing tools and hence helps Selenium to keep an edge.

## **8. Parallel Test Execution and Faster Go-to-Market**

The main aim of automated testing is to save time and efforts. With the help of Selenium Grid, we can execute multiple tests in parallel, hence reducing the test execution time. With the help of cloud-grids for cross-browser testing you can test across as many as hundreds of browsers in parallel using Selenium hence saving you time in multiples of hundreds.

## **9. Less Hardware Usage**

If you compare Selenium with other vendor focused automation tools like QTP, UFT, SilkTest, you will find that Selenium requires less hardware as compared to other testing tools.

## **10. Easy to Learn and Use**

Selenium scripts are not something like writing hundred-page complex algorithm. Writing Selenium scripts is not more than writing a few pieces of codes to automate functionalities of your website. Also, documentation on the [Selenium website](http://www.seleniumhq.org/) is very helpful for developer and testers to start with Selenium automation testing. With the radically growing community, Selenium tutorials, testing, and development support is just a Google search away.

Also with Selenium IDE extension on Firefox browser, you can use record and play functionality to generate Selenium scripts for future reference.

## **11. Constant Updates**

As Selenium is supported by a community and we all know that an active community doesn’t like to stay stagnant, the Selenium community is also constantly releasing constant updates and upgrades. The best part about having a community is that these upgrades are readily available and easy to understand hence you do not need any specific training. This makes Selenium resourceful as compared to other tools and cost-effective as well.

**4**.**All Browser driver used in Selenium?**

Selenium WebDriver interacts with web browsers through browser-specific drivers. Each browser has its own driver, and Selenium supports a variety of browsers. Here are some of the commonly used browser drivers in Selenium:

1. ***ChromeDriver:*** Used for interacting with the Google Chrome browser. You need to download the ChromeDriver executable and set its path in your Selenium script.
2. ***GeckoDriver (Firefox):*** Used for interacting with the Mozilla Firefox browser. It is required for automating Firefox with Selenium. Similar to ChromeDriver, you need to download GeckoDriver and set its path in your script.
3. ***Microsoft WebDriver (Edge):*** Used for interacting with the Microsoft Edge browser. It is necessary for automating Edge with Selenium. The EdgeDriver is available as a separate download.
4. ***InternetExplorerDriver (IE):*** Used for interacting with the Internet Explorer browser. It is required for automating Internet Explorer with Selenium. Note that Internet Explorer is being phased out, and Microsoft recommends using Microsoft Edge instead.
5. ***SafariDriver (Safari):*** Used for interacting with the Safari browser. SafariDriver comes bundled with Safari, and you need to enable the “Remote Automation” option in Safari’s Develop menu to use it.
6. ***OperaDriver (Opera):*** Used for interacting with the Opera browser. Similar to Chrome and Firefox, you need to download the OperaDriver executable and set its path in your script.

5.**STEPS TO CREATE A SIMPLE WEB DRIVER SCRIPT:**

Set up a Java Integrated Development Environment (IDE) like EclipsE.

Copy the dependency of webdriver manager and Selenium .

Add to the pom.xml file and the dependencies will be added in Maven Dependencies.

**Create a New Maven Project:**

· Open your IDE and create a new Java project.

Create a package inside the java project.

Create a class with a name and write the script for it.

· Create a new Java class in a package.

· Open the browser and launch the Url.

· Set the driver and add the Url.

· Write the script for Selenium WebDriver and Chrome driver to interact with a web page.