**Question #1**

**What is the test automation framework? What is selenium? How does it work? and why do you need it?**

**Answer:**

A test automation framework is a platform that is a combination of programs, compilers, features, tools, etc. It provides an environment where you can execute automated test scripts. A test automation framework is a platform that is a combination of programs, compilers, features, tools, etc. It provides an environment where you can execute automated test scripts. A test automation framework is a set of components that facilitate executing tests and comprehensive reporting of test results. The major components that implement a test automation framework successfully are equipment, testing tools, scripts, procedures, and most importantly, test automation engineers.

A **Test Automation Framework** is a set of guidelines like coding standards, test-data handling, object repository treatment etc.… which when followed during automation scripting produces beneficial outcomes like increased code re-usage, higher portability, reduced script maintenance cost etc. These are just guidelines and not rules; they are not mandatory and you can still script without following the guidelines.

**Why do you need a Framework?**

Let’s consider an example to understand why you need a Framework.I am sure you have attended a seminar/lecture/conference where the participants were asked to observe the following guidelines –

* Participants should occupy their seat 5 minutes before the start of a lecture
* Bring along a notebook and pen for note taking.
* Read the abstract so you have an idea of what the presentation will be about.
* Mobile Phones should be set on silent
* Use the exit gates at opposite end to the speaker should you require to leave in the middle of the lecture.
* Questions will be taken at the end of the session

**What is Selenium?**

Selenium is an open-source tool that automates web browsers. It provides a single interface that lets you write test scripts in programming languages like Ruby, Java, Node JS, PHP, Perl, Python, and C#, among others.

[Selenium](https://www.browserstack.com/selenium) is the biggest open source automation testing suite that has been making testing seamless. It is widely used for testing among the bigwigs like Google, Netflix, Fitbit, etc. to the emerging startups such as likewise. Here’s a graph that shows how Selenium framework is extensively used among various companies of different scale and sizes.

Selenium Framework is a suite of automation testing tools that is based on the JavaScript framework. It could run the tests directly on the target browser, drive the interactions on the required web page and rerun them without any manual input.

Selenium conforms with the idea of Agile, and DevOps, which endorse the continuous delivery workflow. Thus, Selenium has been one of the favorite tools for testing as it meets the requirement of quick and reliable testing, which helps enterprises save time and money on testing.

[Selenium WebDriver](https://www.browserstack.com/guide/selenium-webdriver-tutorial) which is also known as Selenium 2.0, is a collection of language-specific bindings that drive a browser in the way it has to be driven. It is the successor of Selenium Remote Control and consists of API, Library, Driver, and Framework that work together to enable test execution.

The API ports the scripts that are written in different scripting languages to Selenium’s in-built language: Selenese, with the help of bindings. These language-specific bindings such as [Selenium Java](https://www.browserstack.com/guide/selenium-with-java-for-automated-test), [Selenium Ruby](https://www.browserstack.com/automate/ruby), Selenium dotnet, etc. and the API are stored in the Library.

While the driver, which is an executable module opens up the browser instance and runs the test script. These drivers could be browser-specific such as Chrome driver for Google Chrome. Frameworks such as [Cucumber](https://www.browserstack.com/guide/learn-about-cucumber-testing-tool) and [TestNG](https://www.browserstack.com/guide/testng-framework-with-selenium-automation) are used for testing on the connection of WebDriver’s client and the browser/driver. These tests can also be conducted on multiple drivers with the help of a [Selenium Grid](https://www.browserstack.com/guide/selenium-grid-tutorial).

**How it Works:**The WebDriver protocol has a local end (‘client’) which sends the commands (test script) to a browser-specific driver. The driver executes these commands on its browser-instance. So, if the test script calls for execution on Chrome and Firefox, the Chrome Driver will execute the test on Chrome; the Gecko Driver will do the same on Firefox.

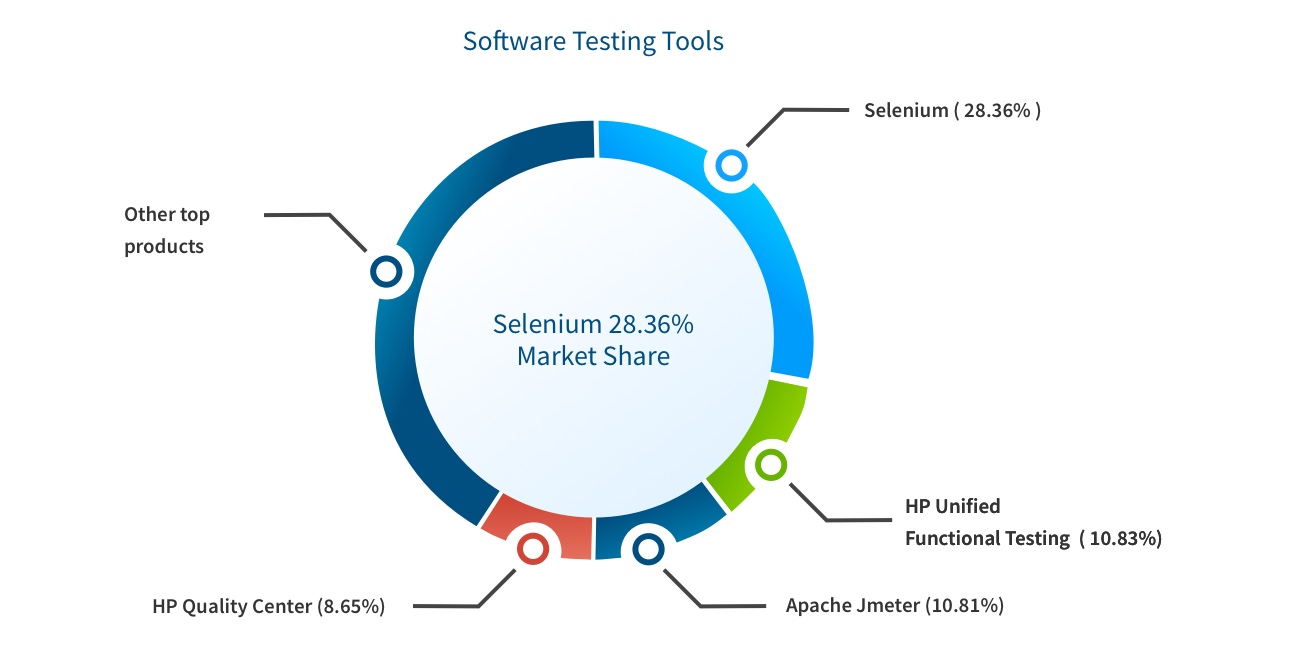
Framework Support libraries for integration with natural or programming language test frameworks, like Selenium with Cucumber or Selenium with TestNG.

[Selenium IDE](https://www.browserstack.com/guide/what-is-selenium-ide) is a Chrome and Firefox plugin, which is capable of logging natural interactions in the browser and generate its code in different programming languages. These programming languages include Java, Python, Ruby, C#, and Selenese.

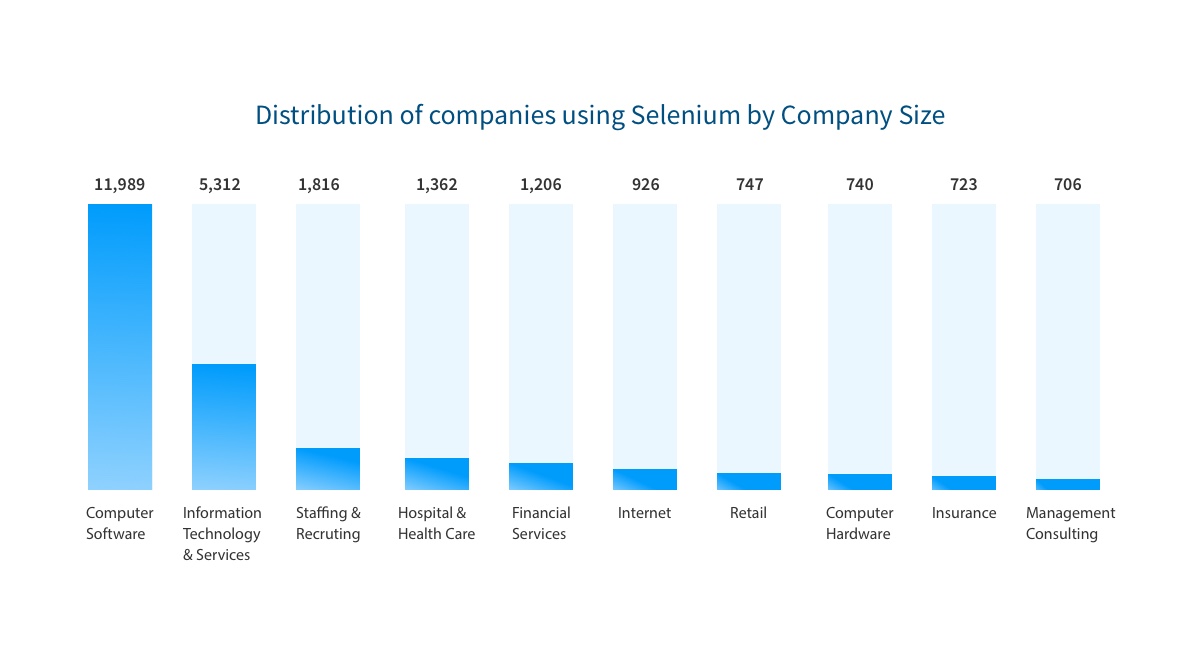
It enables the testers to record within the IDE and play out the test scenario on the browser, where the IDE can replay them and highlight the errors in code with Red color. This makes Selenium IDE useful for conducting Regression testing, where it can be used for prototyping the errors as the code might not be very legible.

**Why is Selenium Framework so popular for Automation Testing?**

Selenium Framework, which has been used by over 41 thousand companies across the world, enjoys the highest market share among the other contemporary software testing tools. A [study from Enlyft (formerly iDatalabs)](https://enlyft.com/tech/products/selenium), suggests that Selenium has a market share of 28.36%.



Although it is mainly used by the Computer Software and IT services companies, Selenium has not just confined itself to them and serves other industries like Staffing & Recruiting, Healthcare, Financial Services, etc.



**When to use these Selenium frameworks?**

When there are a large number of data sets to be tested for the web application. Then you must opt for a data-driven framework, which separates them from the actual code.

For instance, when there are more functionalities to be tested for the web application. It is suggested that the team goes for a keyword driven framework, where the operations are stored in a separate table in the form of keywords. These keywords are called to use a specific functional operation.

On occasions when data sets and functionalities both are high in number, use a hybrid driven framework to avoid complexity.

**Question #2**

**The most common tools that are used for configuration management are packer and ansible. You need to concisely compare both of them.**

**Answer:**

**Ansible vs Packer: What are the differences?**

**What is Ansible?** Radically simple configuration-management, application deployment, task-execution, and multi-node orchestration engine. Ansible is an IT automation tool. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or zero downtime rolling updates. Ansible’s goals are foremost those of simplicity and maximum ease of use.

**What is Packer?**

Create identical machine images for multiple platforms from a single source configuration. Packer automates the creation of any type of machine image. It embraces modern configuration management by encouraging you to use automated scripts to install and configure the software within your Packer-made images.

Ansible belongs to **"Server Configuration and Automation"** category of the tech stack, while Packer can be primarily classified under **"Infrastructure Build Tools"**.

Some of the features offered by Ansible are:

* Ansible's natural automation language allows sysadmins, developers, and IT managers to complete automation projects in hours, not weeks.
* Ansible uses SSH by default instead of requiring agents everywhere. Avoid extra open ports, improve security, eliminate "managing the management", and reclaim CPU cycles.
* Ansible automates app deployment, configuration management, workflow orchestration, and even cloud provisioning all from one system.

Packer provides the following key features:

Super fast infrastructure deployment. Packer images allow you to launch completely provisioned and configured machines in seconds, rather than several minutes or hours.

Multi-provider portability. Because Packer creates identical images for multiple platforms, you can run production in AWS, staging/QA in a private cloud like OpenStack, and development in desktop virtualization solutions such as VMware or Virtual Box.

Improved stability. Packer installs and configures all the software for a machine at the time the image is built. If there are bugs in these scripts, they'll be caught early, rather than several minutes after a machine is launched.

**Link Addresses:**

<https://www.testim.io/blog/test-automation-frameworks/>

<https://www.guru99.com/test-automation-framework.html>

<https://www.browserstack.com/selenium>

<https://www.browserstack.com/guide/selenium-framework#:~:text=Selenium%20Framework%20is%20a%20suite,them%20without%20any%20manual%20input>

<https://stackshare.io/stackups/ansible-vs-packer>