**Selenium** automates web browsers. It is most famous for enabling rapid, repeatable web-app testing, which allows developers to ship new releases faster and with confidence.

Selenium WebDriver:

WebDriver drives a browser natively, as a user would, either locally or on a remote machine using the Selenium server, marks a leap forward in terms of browser automation.

Selenium WebDriver refers to both the language bindings and the implementations of the individual browser controlling code. This is commonly referred to as just WebDriver.

* WebDriver is designed as a simple and more concise programming interface.
* WebDriver is a compact object-oriented API.
* It drives the browser effectively.

**Selenium IDE** records multiple locators for each element it interacts with. If one locator fails during playback, the others will be tried until one is successful.

* Simple, turn-key solution to quickly author reliable end-to-end tests. Works out of the box for any web app.
* Enjoy easier test debugging with rich IDE features like setting breakpoints and pausing on exceptions.
* Run your tests on any browser/OS combination in parallel using the [Command-line Runner for Selenium IDE](https://www.selenium.dev/selenium-ide/docs/en/introduction/command-line-runner).

Selenium Grid 4 Selenium Grid allows the execution of WebDriver scripts on remote machines (virtual or real) by routing commands sent by the client to remote browser instances. It aims to provide an easy way to run tests in parallel on multiple machines.

Selenium Grid allows us to run tests in parallel on multiple machines, and to manage different browser versions and browser configurations centrally (instead of in each individual test). Central entry point for all tests

Main functionalities

* Management and control of the nodes / environment where the browsers run
* Scaling
* Running tests in parallel
* Cross-platform testing
* Load balancing