У термина довольно много разных значений, и много всего под ним подразумевают в разных контекстах.

Давайте разберемся и разложим все по полочкам!

Для начала: Selenium — это проект, который предназначен главным образом для автоматизации тестирования веб-приложений. Проект кстати, open-source, то есть с открытым исходным [кодом](https://github.com/SeleniumHQ/selenium). В любой момент можно заглянуть внутрь кода и посмотреть, как работает тот или иной метод, а также можно написать свои предложения разработчикам проекта, и, более того, сделать pull request с исправлениями и улучшениями кода.

Это собирательный, зонтичный проект*,* то есть он состоит из множества самых разных компонентов, например:

* Selenium WebDriver
* Selenium Grid
* Selenium Server
* Selenium IDE
* и не только...

А также проект содержит ряд конкретных драйверов браузера и клиентских библиотек для разных языков программирования.

Обычно из контекста понятно, о какой из частей Selenium идет речь. В данном курсе речь пойдет о Selenium WebDriver

**Что такое Selenium WebDriver?**

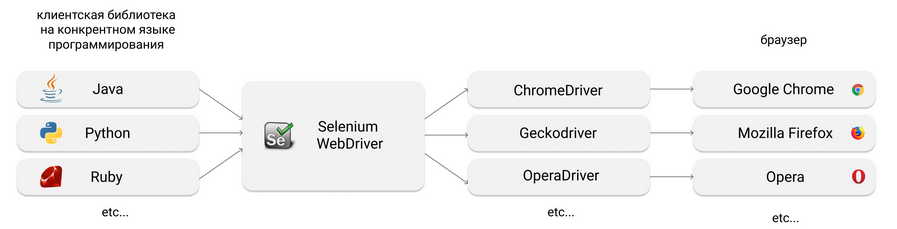
Автотесты бывают очень разными, проверяют разные вещи и работают для разных уровней приложения. Наверняка вы не раз видели так называемую "пирамиду тестирования". В нашем курсе мы в основном будем говорить про end-to-end тесты, то есть такие, которые проверяют полную работу системы и имитируют действия пользователя.

Что нам нужно для того, чтобы написать такой тест для веб-приложения?

1. Очевидно, код на языке программирования. Там будет содержаться какая-то логика действий (например: залогинить пользователя, добавить товар в корзину) и проверки.
2. Браузер, чтобы имитировать действия пользователя. Желательно, какой-нибудь популярный. А еще лучше — несколько.

Эти две части нужно как-то связать воедино. И как раз для этого нужен Selenium WebDriver. По сути своей это — универсальный интерфейс, который позволяет манипулировать разными браузерами напрямую из кода на языке программирования. Его крутость как раз в его универсальности: за некоторыми исключениями, все наши тесты, написанные один раз, можно запускать для разных браузеров.

Верна универсальность и в другую сторону: браузер будет выполнять одинаковые действия, не важно из библиотеки какого именно языка программирования эти методы будут вызваны. Вообще-то, Selenium WebDriver поддерживает довольно много [языков](https://github.com/SeleniumHQ/selenium#documentation) программирования, хотя наибольшей популярностью в индустрии пользуются Java и Python.



**WebDriver не имеет прямого отношения к тестированию**. Он всего лишь предоставляет автотестам доступ к браузеру. На этом его функции заканчиваются.  
  
Структурирование, группировку и запуск тестов, а также генерацию отчётов о тестировании, обеспечивает фреймворк тестирования, такой как [JUnit](http://www.junit.org/) или [TestNG](http://testng.org/) для Java, [NUnit](http://www.nunit.org/) или [Gallio](http://www.gallio.org/) для .Net, [RSpec](http://rspec.info/) или [Cucumber](http://cukes.info/) для Ruby и так далее. Разработка тестов ведётся в среде [Eclipse](http://www.eclipse.org/), [Intellij IDEA](http://www.jetbrains.com/idea/), [Visual Studio](http://www.microsoft.com/visualstudio/rus/products/visual-studio-overview), [RubyMine](http://www.jetbrains.com/ruby/) и так далее. Сборка осуществляется посредством [Maven](http://maven.apache.org/), [Gradle](http://www.gradle.org/), [Ant](http://ant.apache.org/), [NAnt](http://nant.sourceforge.net/),[Rake](http://rake.rubyforge.org/) и так далее. Запуск тестов по расписанию и публикацию отчётов выполняет сервер непрерывной интеграции – [Jenkins](http://jenkins-ci.org/), [CruiseControl](http://cruisecontrol.sourceforge.net/), [Bamboo](http://www.atlassian.com/software/bamboo/), [TeamCity](http://www.jetbrains.com/teamcity/) и так далее. И всё это – самостоятельные инструменты, не имеющие отношения к проекту Selenium.