**Selenium 1.0**

**Selenium IDE**

**Selenium RC**

**Selenium 2.0**

**Selenium WebDriver**

**Selenium RemoteWebDriver**

**Selenium 3.0**

**Appium**

**Winium**

**Selenium IDE:**

We can record-play our scripts. IDE has a record button which records manual interaction which tester performs on an application, in the form of Selenese commands:

sendKeys(typing)

click(clicking)

**Advantages:**

1. Record-play facility

**Disadvantages:**

1. We can not write our scripts in any programming language like Java, C# etc. hence we can not use programming capabilities like for loop, if loop etc. for writing scripts
2. IDE used to support only Firefox browser initially. Not all browsers are still supported by IDE.

**RC(Remote-Control)**

Script

Script can be written in any language using libraries in which sendkeys,click functions are implemented. Command called SendKeys

**Remote Control:**  It will translate script statements in JS

**Browser**

**Application**

**Selenium Core:**

Selenium core is JS program which accepts translated

JS from translator and runs them inside browser

Selenium core runs inside a Sandbox environment and

Can not execute any script statement requiring an access of anything other than application for example it can not change some browser setting.

I have a test case in which I want to change setting of a

Browser like proxy setting and run login test steps.

**Selenium WebDriver**

Script can be written in any language like Java,C# etc.

**ChromeDriver**

**GeckoDriver EdgeDriver**

Chrome

Edge

FireFox

**In WebDriver we have separate translators written for each web browser. These translators or drivers translates Selenium APIs into respective browser’s Native UI Automation API. Native UI Automation APIs means every browser exposes functions/APIs to get programmatically controlled.**

**Remote WebDriver**

Running Selenium scripts on a browser running on a remote machine.

Running a Selenium Script

Machine A

Script will instantiate a communication session(Http) with standalone server running on Machine B. And every Selenium script statement will be sent to Standalone server.From Standalone server they would be sent driver.Driver wll translate them into Native UI automation API of browser and browser as well application opened inside it will be automated. Script mentions which driver Selenium standalone server should start by mentioning browserName desiredcapability.

Machine B

ChromeDriver

Chrome Browser

**Appium:**

Appium script establishes a session with Appium server to start Android driver.Android driver

Translates Appium APIs like SendKeys,Click etc. in UiAutomator API calls. These translated UiAutomator API calls then are sent to bootstrap agent running on device which then executes these translated command on application running on mobile device

Appium Script

**Android Driver**

IOSDriver

WindowsDriver

AndroidDriver

Android Device

BootStrap Agent

Application

Google provides UI automation API called UiAutomator to automate UI tests on Android mobile application

1. We need to write tests in Java language
2. We need to learn different API function made available to find elements, to perform different interactions like typing/clicking etc.
3. If we want this test to be executed on other platform apps like IOS,Windows mobiles we can not run same tests there as scripts are written in Android specific UI automation tool.

Apple provides UI automation API called XCUITest to automate UI tests on IOS mobile application

1. We need to write tests in ObjectiveC /Swift language
2. We need to learn different API function made available to find elements, to perform different interactions like typing/clicking etc.
3. If we want this test to be executed on other platform apps like Android,Windows mobiles we can not run same tests there as scripts are written in IOS specific UI automation tool.

Microsoft provides UI automation API called MicrosoftUI automation to automate UI tests on Windows mobile application

1. We need to write tests in C# language
2. We need to learn different API function made available to find elements, to perform different interactions like typing/clicking etc.
3. If we want this test to be executed on other platform apps like Android,IOS mobiles we can not run same tests there as scripts are written in WIndows specific UI automation tool.