Test Automation & Selenium

GTS EDUTECH

Shreyata Sugandhi

Corporate Trainer & Consultant

shreyata@gtsedutech.com

https://www.linkedin.com/in/shreyatasugandhi/

Agenda

GTS EDUTECH

- Introduction to Automation
  - Process of Test automation
  - Tool Selection
  - Frameworks
- Introduction to Selenium
- Components of Selenium
- Pros & Cons Selenium Components
- Use Cases of Selenium Components

Introduction To Automation

GTS EDUTECH

What is Automation Testing?

- Converting manual tests into a program that can be executed multiple times without any manual interventions.
- Test automation tool can be used for developing such programs.

Why Automation Testing?

GTS EDUTECH

Automated testing is important because:

- Manual Testing of all work flows, all fields , all negative scenarios is time and cost consuming
- It is difficult to test for multi lingual sites manually
- Automation does not require Human intervention. You can run automated test unattended (overnight)
- Automation increases speed of test execution
- Automation helps increase Test Coverage
- Manual Testing can become boring and hence error prone.

What to And not to Automate?

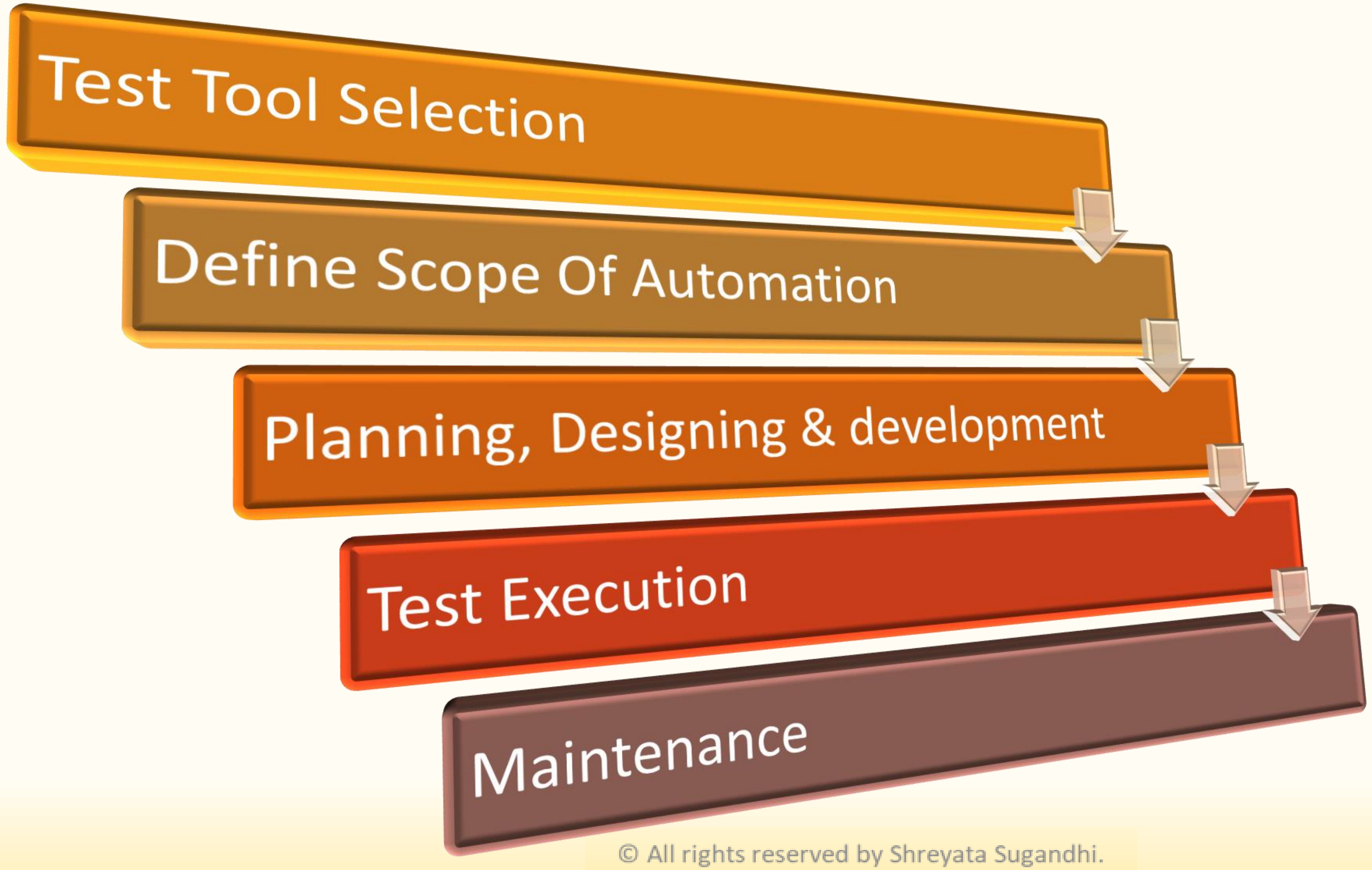
Test cases for automation can be selected using the following criteria to increase the automation ROI

- **High Risk** - Business Critical **test cases**
- Test cases that are **executed repeatedly**
- Test Cases that are very **tedious** or difficult to perform manually
- Test Cases which are **time consuming**

The following category of test cases are not suitable for automation:

- **Test Cases that are newly designed** and not executed manually at least once
- **Test Cases** for which the **requirements are changing frequently**
- **Test cases** which are executed on **ad-hoc basis**

Automation Process



Test Tool Selection Criteria

GTS EDUTECH

- Identify the requirements
- Know your Budget & Duration of project
- Identify requirement matching tools available in the market
- Create a Comparison Matrix based on
  - Tool capabilities
  - Ease of adoption
  - Ease of scripting and reporting capabilities
  - Tool usage
  - Ease of maintenance
  - POC

Test Automation Framework

GTS EDUTECH

What is Test Automation Framework?

- Test Automation Framework is a defined and extensible support structure within which the test automation suites are developed and implemented using the selected tool.
- It includes physical structures used for test creation and implementation as well as the logical interaction between the components like
  - Set of standards/guidelines
  - Directory structure
  - Test data management
  - Object repository management
  - Environments
  - Common functions or methods to run the scripts
  - Error handling
  - Results

Types of Test Automation Framework

GTS EDUTECH

- Data Driven
- Key Driven
- Modularized
- Page Object Model
- Hybrid

Introduction to Selenium

GTS EDUTECH

- Selenium is an automation tool for web application
- Open source
- Languages support
- browser support
- OS
- Web & mobile application
- Flexible
- Extendable

Who Developed Selenium?

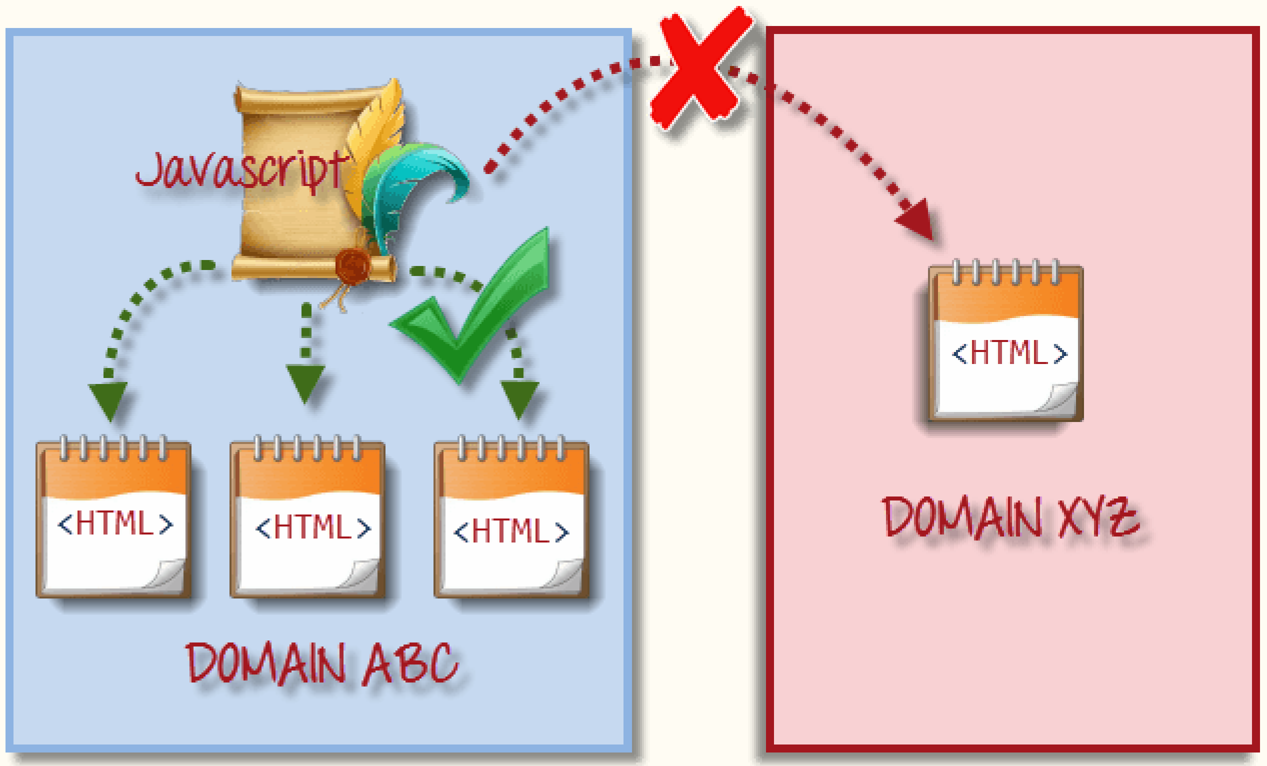
GTS EDUTECH

- Selenium was created by Jason Huggins in 2004 in ThoughtWorks.
- He was working on a web application that required frequent testing. He observed that the repetitive manual testing of their application was becoming more and more inefficient, he created a JavaScript program that would automatically control the browser's actions. He named this program as the "**JavaScriptTestRunner**.“
- Seeing potential in this idea to help automate other web applications, he made JavaScriptTestRunner open-source which was later re-named as **Selenium Core**.

Why the Name Selenium?

- Selenium is a well-known antidote for Mercury.
- Another automated testing framework was popular during Selenium's development, and it was by the company called **Mercury Interactive.** (QTP)

The Same Origin Policy Issue



Under Same Origin Policy, a Javascript program can only access pages on the same domain where it belongs. It cannot access pages from different domains

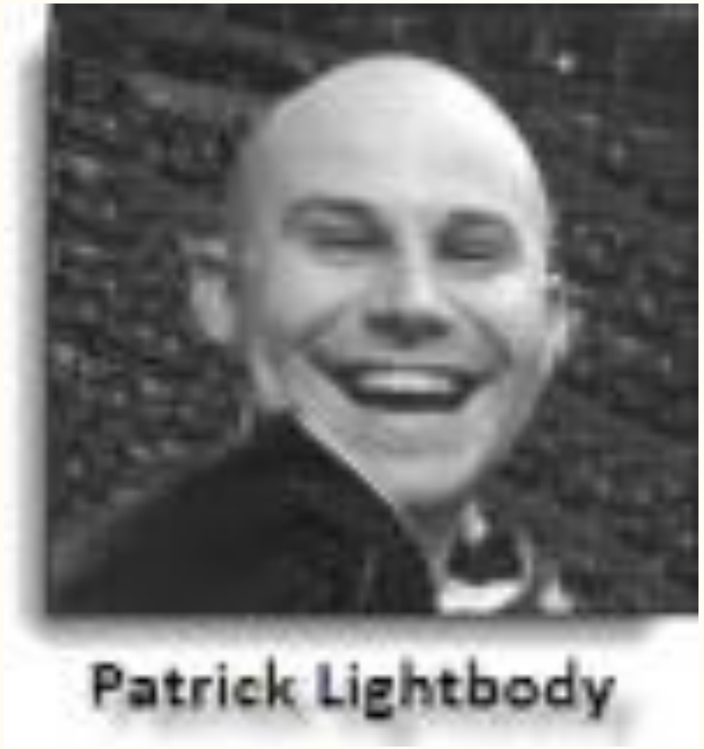
- Same Origin policy prohibits JavaScript code from accessing elements from a domain that is different from where it was launched.
- This is the reason why prior to Selenium RC, testers needed to install local copies of both Selenium Core (a JavaScript program) and the web server containing the web application being tested so they would belong to the same domain.

Birth of Selenium Remote Control

GTS EDUTECH

- **Paul Hammant**, another engineer at ThoughtWorks, decided to create a server that will act as an HTTP proxy to "trick" the browser into believing that Selenium Core and the web application being tested come from the same domain.
- This system became known as the **Selenium Remote Control** or **Selenium 1**.

Birth of Selenium Grid



- Selenium Grid was developed by **Patrick Lightbody** to address the need of minimizing test execution times as much as possible.
- He initially called the system "**Hosted QA**."
- It was capable of capturing browser screenshots during significant stages, and also of **sending out Selenium commands to different machines simultaneously.**

Birth of Selenium IDE

GTS EDUTECH

- **Shinya Kasatani** of Japan created **Selenium IDE**, a Firefox extension that can automate the browser through a record-and-playback feature.
- He came up with this idea to further increase the speed in creating test cases. He donated Selenium IDE to the Selenium Project in **2006**.

GTS EDUTECH

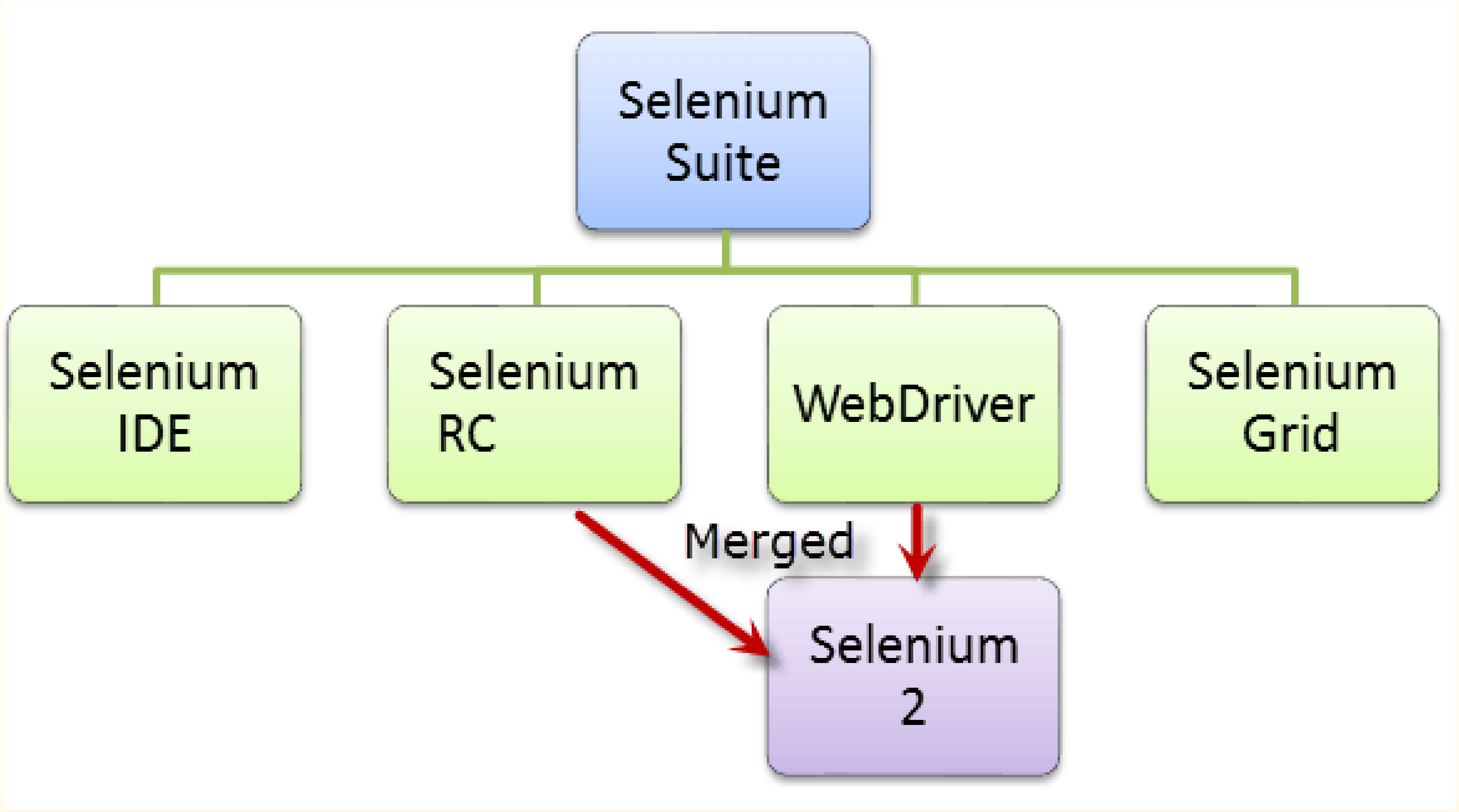
Birth of WebDriver

- **Simon Stewart** created WebDriver circa **2006** when browsers and web applications were becoming more powerful and more restrictive with JavaScript programs like Selenium Core.
- It was the first cross-platform testing framework that could control the browser from the OS level.

Birth of Selenium 2

- In **2008**, the whole Selenium Team decided to merge WebDriver and Selenium RC to form a more powerful tool called **Selenium 2**, with **WebDriver being the core**.
- Currently, Selenium RC is still being developed but only in maintenance mode.
- Most of the Selenium Project's efforts are now focused on Selenium 2.

Components Of Selenium



Selenium IDE

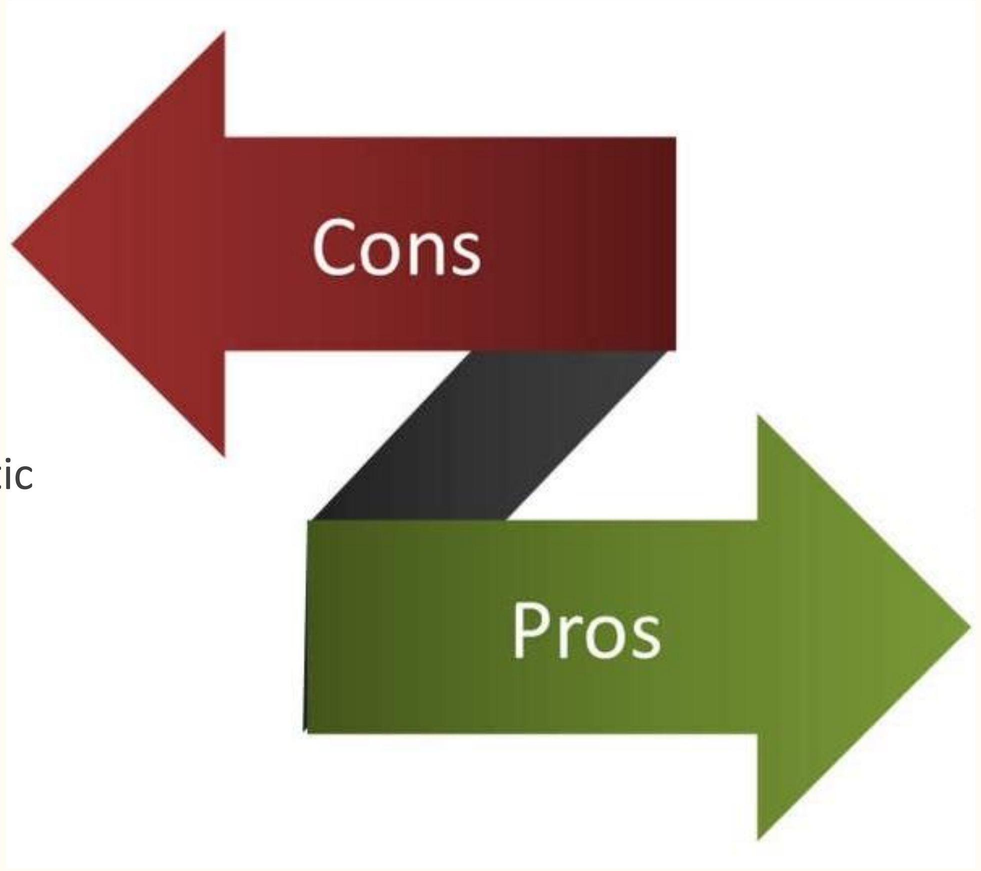
GTS EDUTECH

- Easy
- No need of programming knowledge
- Allows to export tests into different format (RC/WebDriver) and language
- Built-in reporting and help
- Provides support for extensions

- Available only for Firefox
- Designed to create only prototypes of tests
- No support for conditional or iteration operations
- Test execution is slow

Selenium RC

- Installation and execution is complicated than IDE
- Programming knowledge is required
- API contents redundant and confusing command
- Browser interaction is less realistic
- Inconsistent results
- Execution is slower than Web Driver



- Cross-browser & cross-platform support
- Can use full of programming features like conditional & iterating operations
- Supports Data-driven testing
- Faster than IDE

Selenium Web Driver

GTS EDUTECH

- Direct communication with browser
- Browser interaction is more realistic
- No need of any separate components such as RC Server
- Faster than RC & IDE

- Installation is much complicated
- Required programming language
- Can’t readily support new browser
- No-built in logger for messages or results

Environment Support

	Browser Support	OS Support
IDE	Mozilla Firefox	Windows Mac OS X Linux
RC	Mozilla Firefox, IE, Google Chrome, Safari, Opera, Konqueror, Others	Windows Mac OS X Linux Solaris
Web Driver	IE 6 and above, Firefox 3.0 to current version, Google Chrome 12.0.712.0 and above, Opera 11.5 and above, Android - 2.3 and above iOS 3+ for phones & 3.2+ for tablets HtmlUnit 2.9 and above	All operating systems where these browsers can run

Use Case of IDE

- To learn about concepts on automated testing and Selenium:
  - Selenese commands such as type, open, clickAndWait, assert, verify, etc.
  - Locators such as id, name, xpath, css selector, etc.
  - Executing customized JavaScript code using runScript
  - Exporting test cases in various formats.
- To create tests with little or no prior knowledge in programming.
- To create simple test cases and test suites that you can export later to RC or WebDriver.
- To test a web application against Firefox only.

Use Case of RC

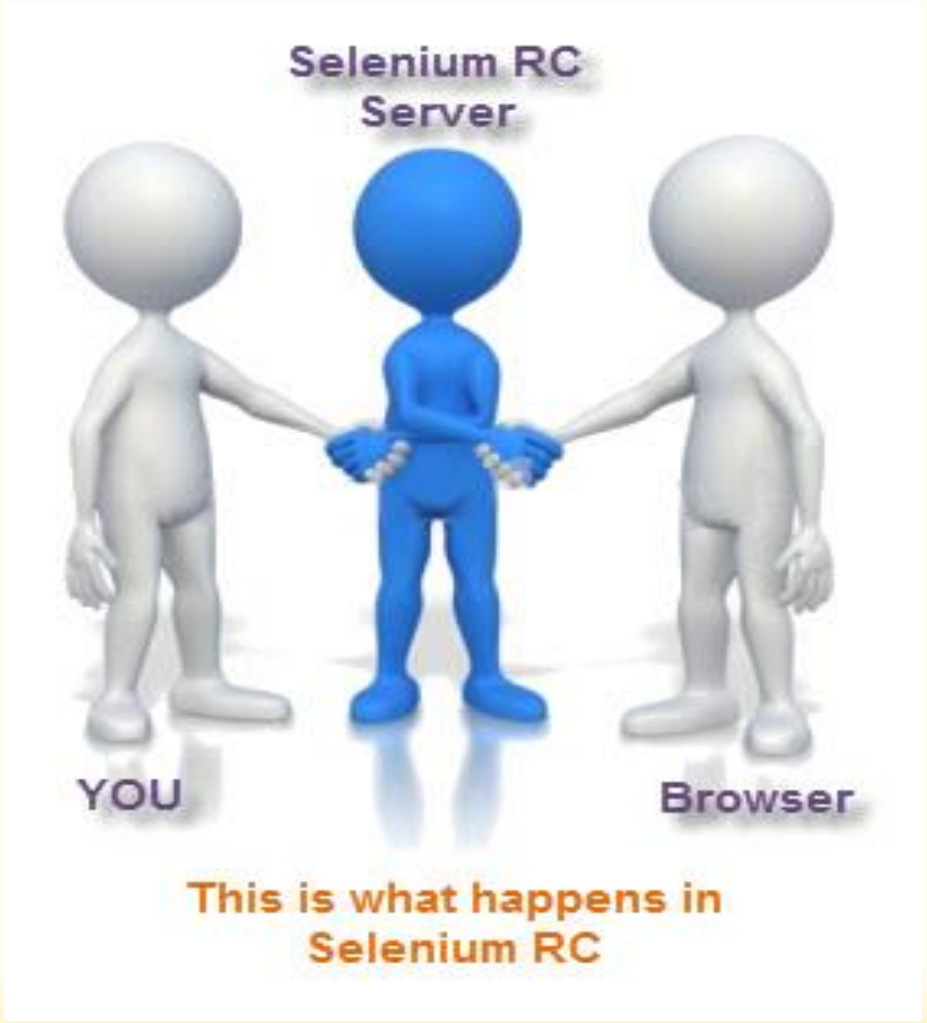
- To design a test using a more expressive language than Selenese
- To run your test against different browsers (except HtmlUnit) on different operating systems.
- To deploy your tests across multiple environments using Selenium Grid.
- To test your application against a new browser that supports JavaScript.
- To test web applications with complex AJAX-based scenarios.

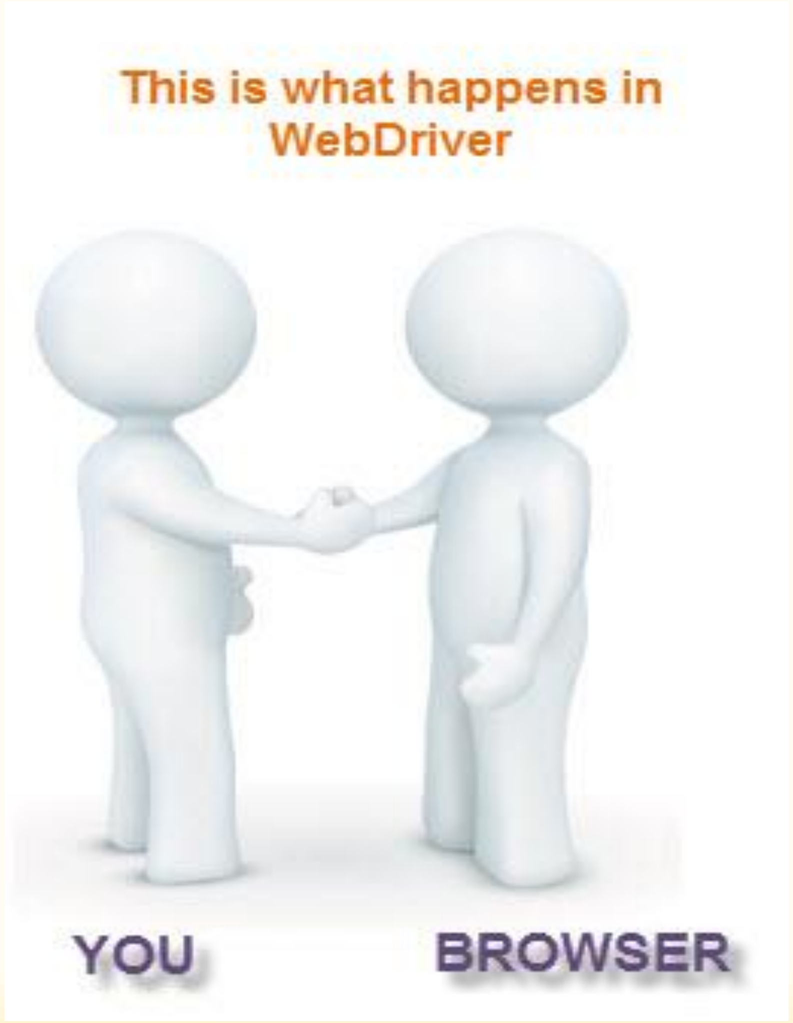
Use Case of Web Driver

- To use a certain programming language in designing your test case.
- To test applications that are rich in AJAX-based functionalities.
- To execute tests on the HtmlUnit browser.
- To create customized test results.

RC vs WebDriver

RC vs WebDriver





RC vs WebDriver

GTS EDUTECH

type or typeKeys?!?
clickAt or mouseAt!??!
click or mouseDown?! ?!
click or
click or
?!?
Selenium RC
For me, they're just click() and sendKeys()
WebDriver

Use Case of Grid

- To run your Selenium RC scripts in multiple browsers and operating systems simultaneously.
- To run a huge test suite, that need to complete in soonest time possible.