Introduction to Test Automation & Tools

What Will We Cover Today?

- What is automation?
- Why automation – Why do we automate?
- What is Selenium?
- Why do we use Selenium?
- How Selenium works?
- Create project and add Selenium
- Do some basic navigations using Selenium

What is The Goal Of Today's Class?

- After today's class you should be able to:
- Understand how Selenium Works
- Create projects with Selenium
- Open a browser using Java & Selenium
- Do some basic navigations with selenium

What is Automation? (By Definition)

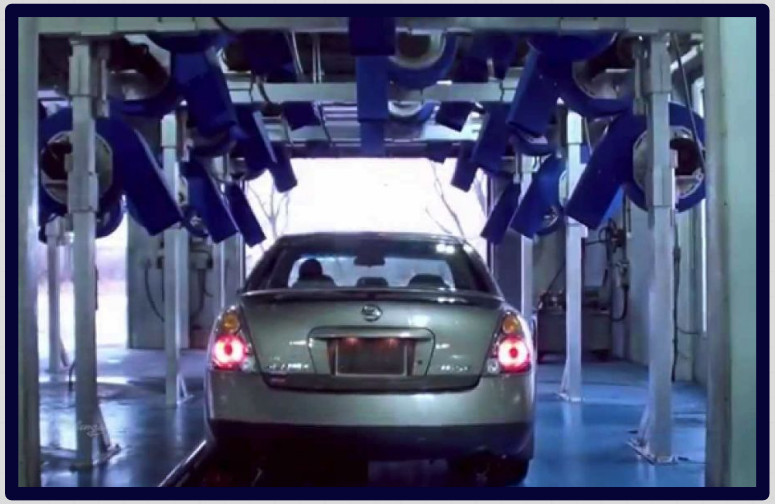
- Definition#1: The use of machines and technology to make processes run on their own without manpower.
- Definition #2: Automation is the technology by which a process or procedure is performed with minimal human assistance.
- Basically if there is a simple process that is just repeating itself, and you take out human interruption, you automate it.

Any Examples You Can Think Of?

AUTOMATION

Simple Example:

- Carwash:
  - Pay
  - Get receipt
  - Drive on rails
  - Shampoo
  - Brushing
  - Rinse
  - Dry



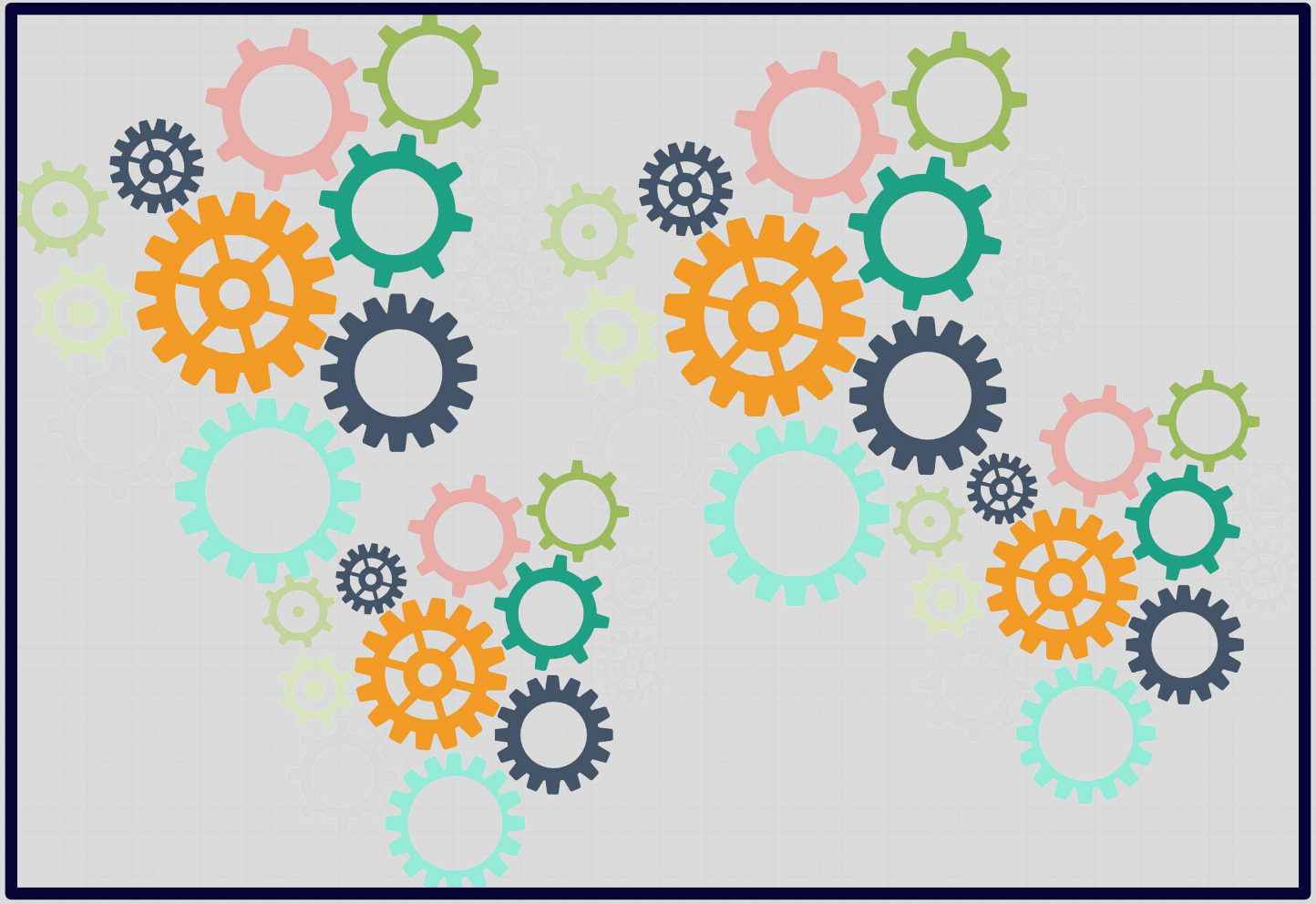
Simple Example:

- Automation requires the use of machines/tools to carry out manufacturing processes with levels of;
  - speed,
  - consistency,
  - stamina (durability),
  - and precision beyond the capacity of a human worker.

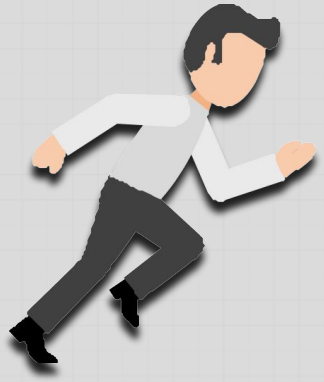
Simple Example:

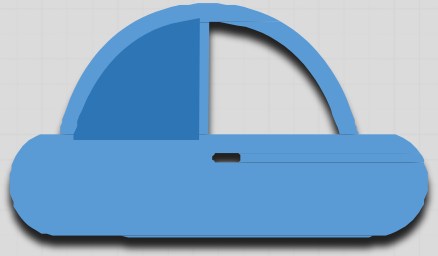
- Most of the things that we are using on a WebApp is automated.
- When you login to Facebook…
- There isn’t someone that is checking your username and password manually.

Why Test Automation



Walking vs Driving





Walking can take hours/days.

Not efficient.

Doable maybe once in the while.

While Driving will only take minutes/hours.

Fast.

Doable all the time.

Manual vs Automation

Manual Testing is time consuming.

Ex: In our current project manual testers can execute 4 test cases a day per recourse in average.

Automation can do probably 100 times more.

If the test cases are automated, we can execute 100 test cases a day per machine.

Manual vs Automation

Manual testing is repetitive effort.
Every time testers have to repeat the same time and effort to execute the same test cases.

Once the test cases are automated we can run the same code as much as we need.
Automation is one time effort. Only minor maintenance is needed.

Manual vs Automation

Manual needs more human resources.
Accenture had more than 70 manual testers for healthcare.gov project.

Automation needs more machines.
They were 8 automation testers and 60 virtual machines for the same project.

Manual vs Automation

Manual testing can be not accurate.

Since manual testers have to repeat the same task again and again in the same project it can become boring and testers might miss the defect.

Automation is highly accurate.

Even close to 100%.

It can catch the difference between AutOmation and Automation.

Manual vs Automation

In most cases production deployment will be performed at midnight. Manual testing can be run only when testers are present.

Automated tests can run anytime without human intervention. It can be run day and night. You can kick off your tests at midnight and entire team can get the test run report at 7 AM in the morning.

Manual vs Automation

Test data creation can take a lot of time and effort.

In some cases the testers might spend 15 minutes to create one test data and they can only use it one time.

Test data creation can be automated.

Test data can be created very efficiently using automation. It can save the company a lot of time and the money.

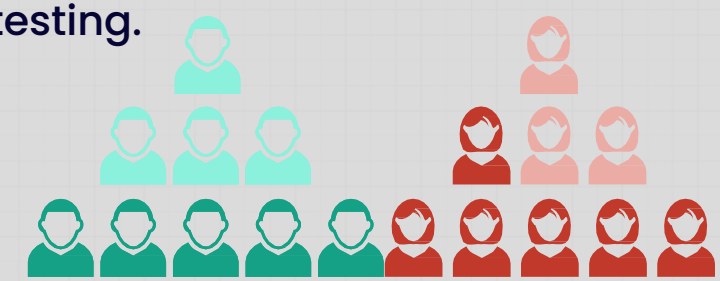
Manual vs Automation

Manual testing will not go away.
It is necessity.
Some test scenarios cannot be automated.

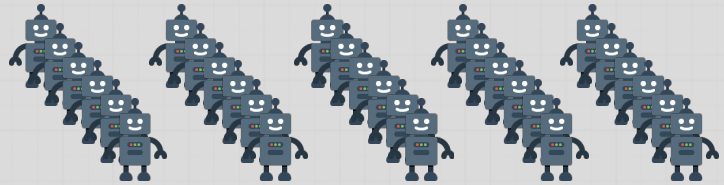
100% automation is not possible.
Normally it is recommended to automate after the functional or manual testing is done. You can't automate broken software.

Manual vs Automation

It is almost not possible to perform performance testing manually. No company can hire 10000 people to perform load testing.



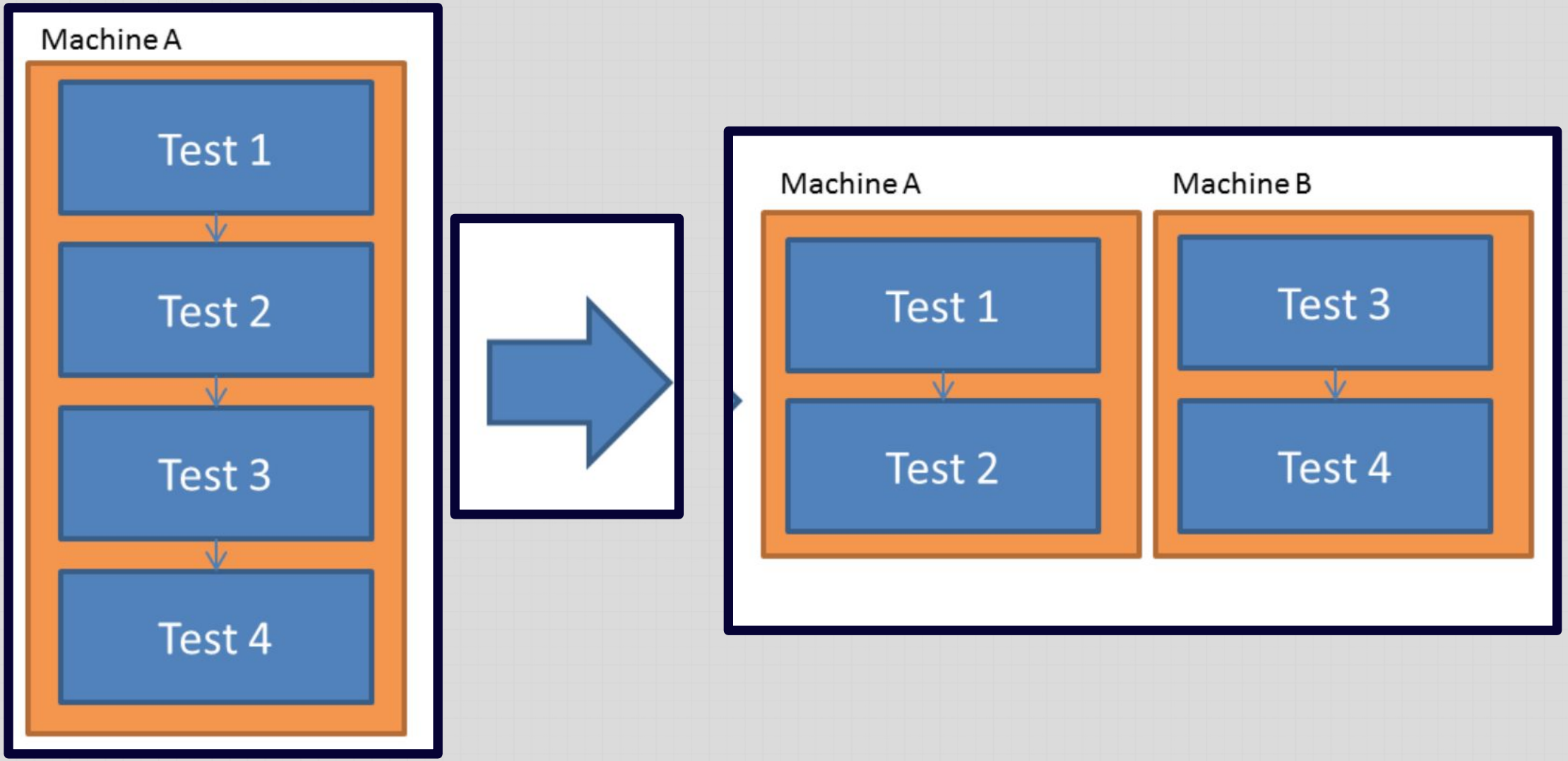
Performance testing can be done by creating many virtual users.



Parallel Execution

- You can create different virtual machines and run parallel executions to shorten the testing duration.

Parallel Testing



Automation Process

1. Test tool selection
   - Test tool selection largely depends on the technology the Application Under Test is built on.
   - Application Under Test (AUT): The application you are working on to test.
2. Define the scope of automation.
   - Scope of automation is the area of your application (AUT), which will be automated.
3. Planning, design, and development.
   - During this phase, you create automation strategy & plan.
   - What type of tool should be used,
   - What kind of framework should be used
   - To what extend we are planning to automate

Automation Process

4. Test Execution

- Automation scripts are executed during this phase.

5. Maintenance

- As new functionalities are added to the Application, automation scripts needs to be added, reviewed, and maintained for each release cycle.
- Maintenance becomes necessary to improve effectiveness of Automation Scripts.

Automation Tools

- UFT : Formerly known as QTP. It was the market leader. Costly.
- Rational Robot : It is an IBM tool used to automate tests. Costly.
- Worksoft...
- TestComplete...
- Protractor...
- Watir...
- Selenium : It is an open source WebApp Automation Tool.

Automation Tools

We Will Learn More Than Selenium!

How To Choose An Automation Tool

- Tool selection is one of the biggest challenges to be tackled before going for automation
- First identify the requirements, explore various tools and its capabilities
- Set the expectation from the tool and go for **proof of concept***

*Proof of concept: Basically you create a pilot to test if it is working well or not.

How To Choose An Automation Tool

Product	Selenium	Katalon Studio	Unified Functional Testing	TestComplete	watir
Available since	2004	2015	1998	1999	2008
Application Under Test	Web apps	Web (UI & API), Mobile apps	Web (UI &API), Mobile, Desktop, Packaged apps	Web (UI &API), Mobile, Desktop apps	Web apps
Pricing	Free	Free	$$$$	$$	Free
Supported Platforms	Windows Linux OS X	Windows Linux OS X	Windows	Windows	Windows Linux OS X
Scripting languages	Java, C#, Perl, Python, JavaScript, Ruby, PHP	Java/Groovy	VBScript	JavaScript, Python, VBScript, JScript, Delphi, C++ and C#	Ruby
Programming skills	Advanced skills needed to integrate various tools	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts	Advanced skills needed to integrate various tools
Ease of Installation and Use	Require advanced skills to install and use	Easy to setup and use	Complex in installation. Need training to properly use the tool	Easy to setup. Need training to properly use the tool	Advanced skills needed to integrate various tools

What is Selenium?

- A tool that automates browser
- Lets us control browser using code
- Mostly used for testing
- It is actually just a library (classes and methods)

More About Selenium...

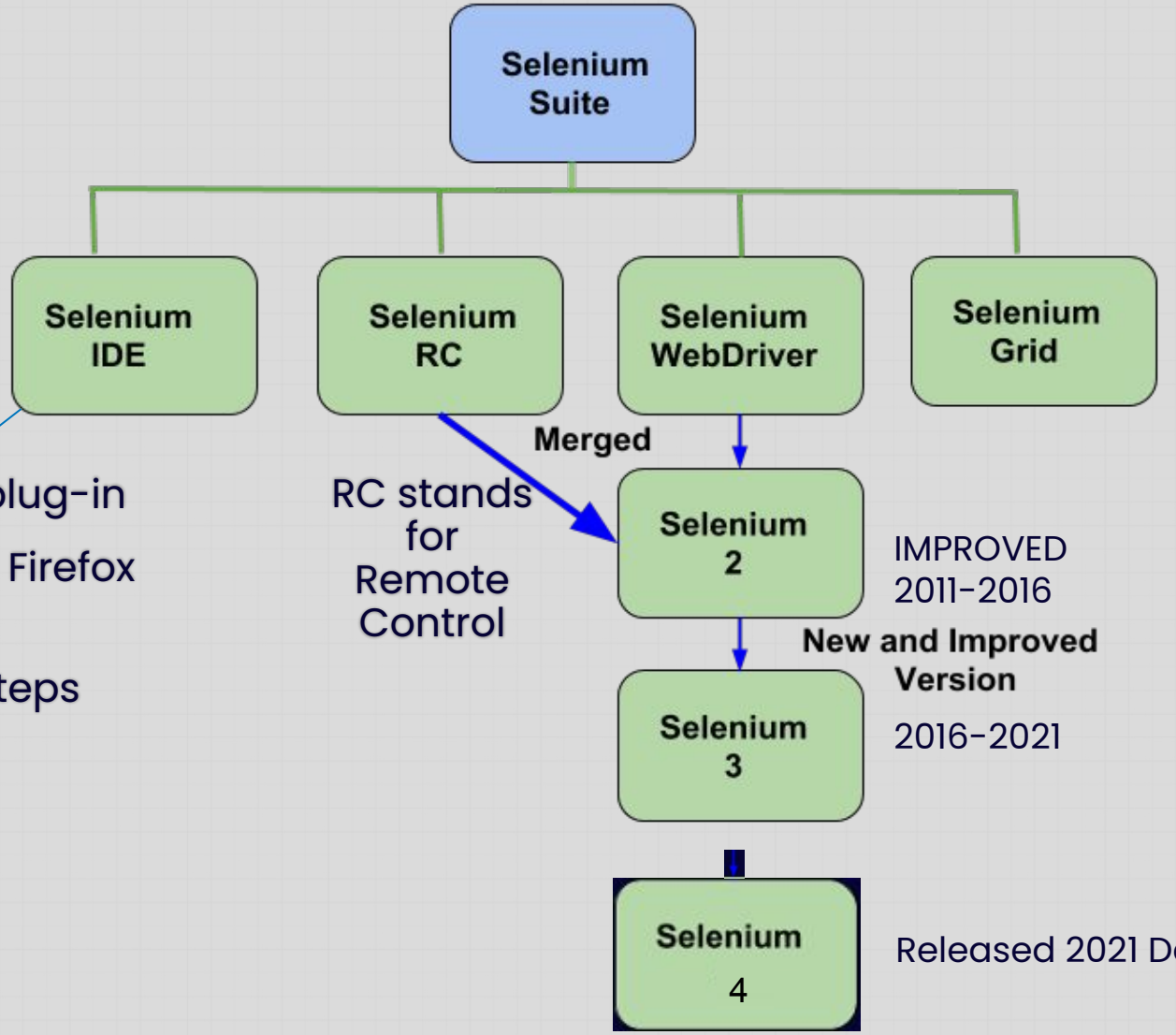
- Selenium is an open source automation testing tool
- It is used exclusively for web based applications
- You can work on multiple operating systems using Selenium
- Platforms supported by Selenium
  - Windows
  - OS X
  - Linux
- Languages are used with Selenium
  - Java
  - C#
  -
- Browsers are supported by Selenium
  - ~~Internet Explorer~~
  - Firefox
  - Chrome
  - Safari
  - Edge

-> Java
-> C#
-> JavaScript
-> Perl
-> Python
-> Ruby
-> PhP

Where Did Selenium Come From?

- Selenium.dev
- Selenium RC was created in 2004
- Two projects merged in 2009 and Selenium WebDriver was created in 2009.

Where Did Selenium Come From?



It is just a Firefox plug-in

You add it to your Firefox browser

You record your steps

And re-play

RC stands for Remote Control

IMPROVED
2011-2016

New and Improved Version

2016-2021

Released 2021 December

Selenium WebDriver: Pros

- Free
- Multi platform
- Multi browser
- Multi language
- Mobile testing
- Scalable and parallel testing
- Integration with different tools and frameworks
- Community support

Selenium WebDriver: Cons

- Requires knowledge of a programming language
- No built-in reporting
- Only works on browsers
- No image-based testing
- No professional support

Question

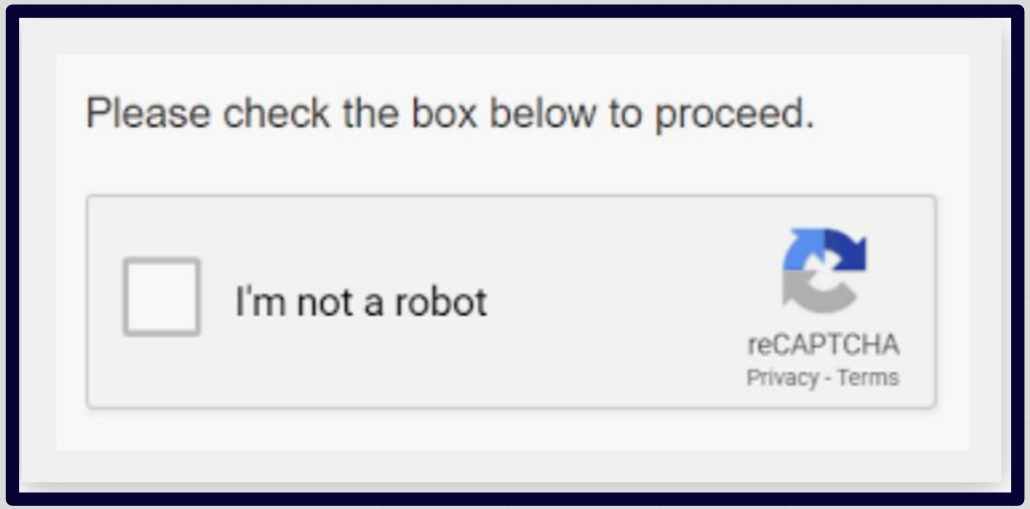
- My application is developed with C#. Can I use selenium with Java to automate my application?

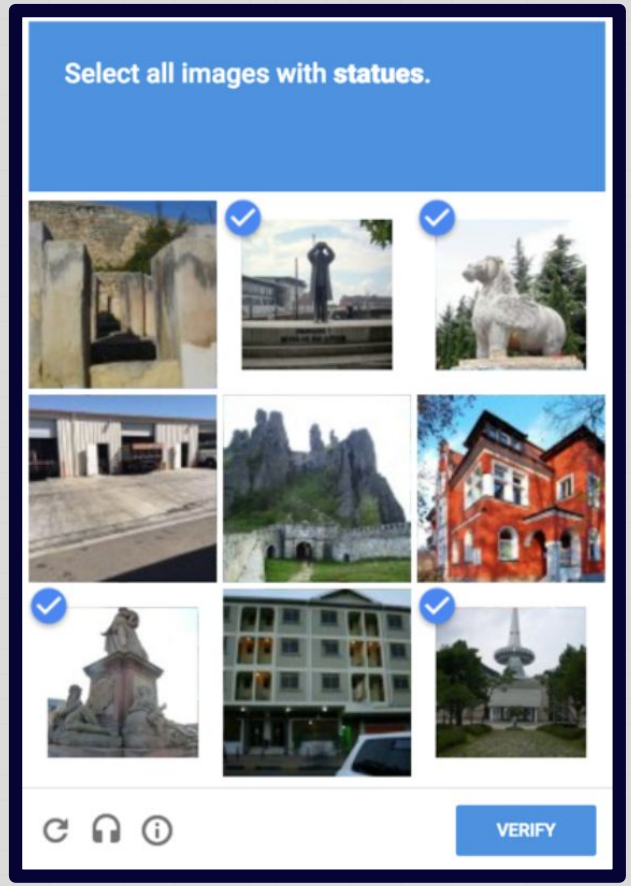
What Can Be Automated With Selenium?

- What is the difference between a WebApp and Desktop app?
- WebApp is any website you open in a browser
- Desktop app, is any app only your computer from calculator to sublime, to anything else.
- Selenium can only automate Web based applications.

What Can’t Be Automated By Selenium?

Captcha
Completely Automated Public Turing test to tell Computers and Humans Apart .





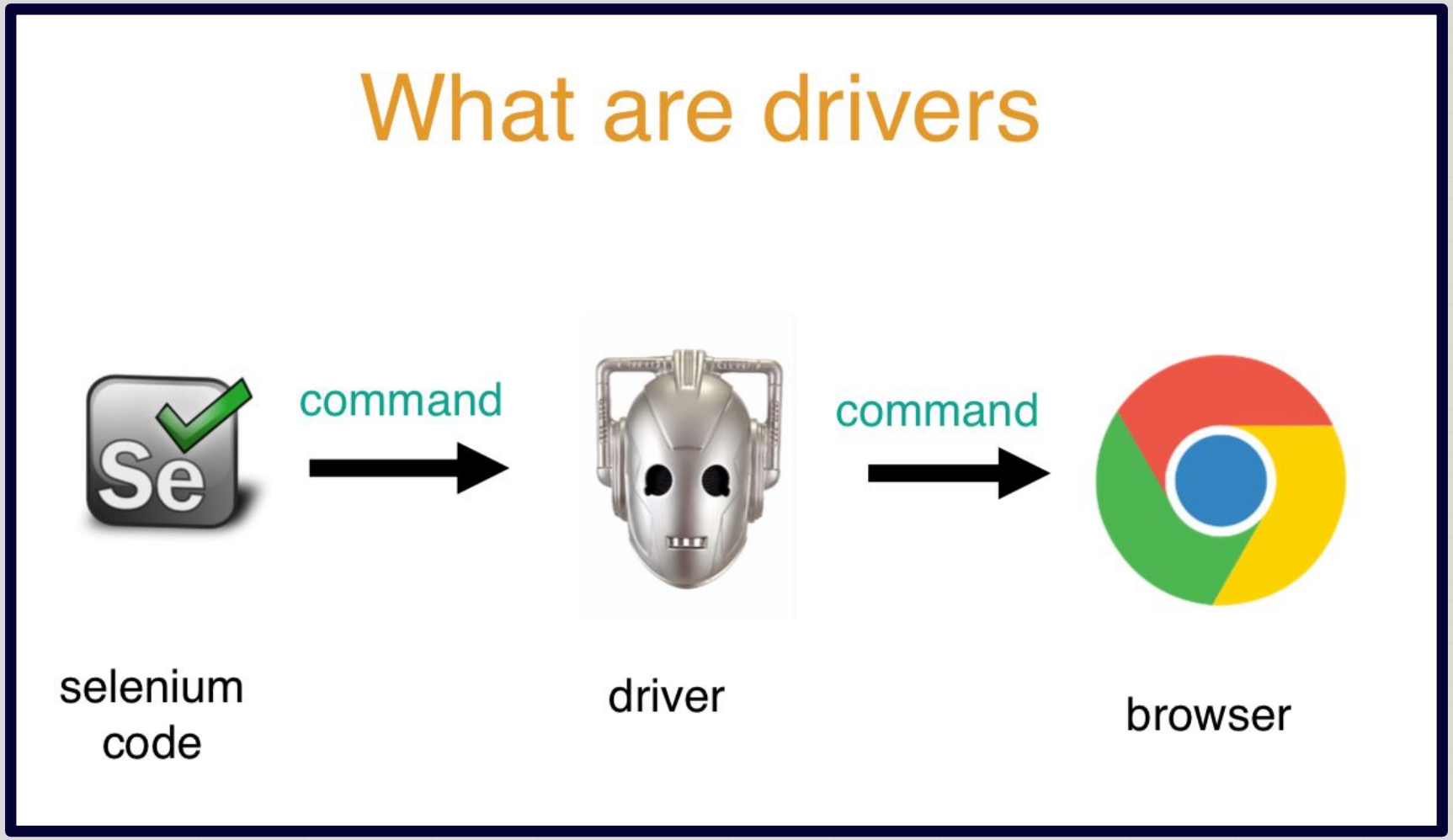
How Selenium WebDriver Works?

- Let’s think taxi driving
- We have 3 actors
  1. the client; he tells the taxi driver where he wants to go and how to get there
  2. the taxi driver; he executes the client's requests; the taxi driver sends his own requests to the car
  3. the car; the car executes the taxi driver's requests

How Selenium WebDriver Works?

- Test automation engineer is like a taxi client
- The browser driver is like a taxi driver
- The browser is like a taxi

What Happens In Reality?

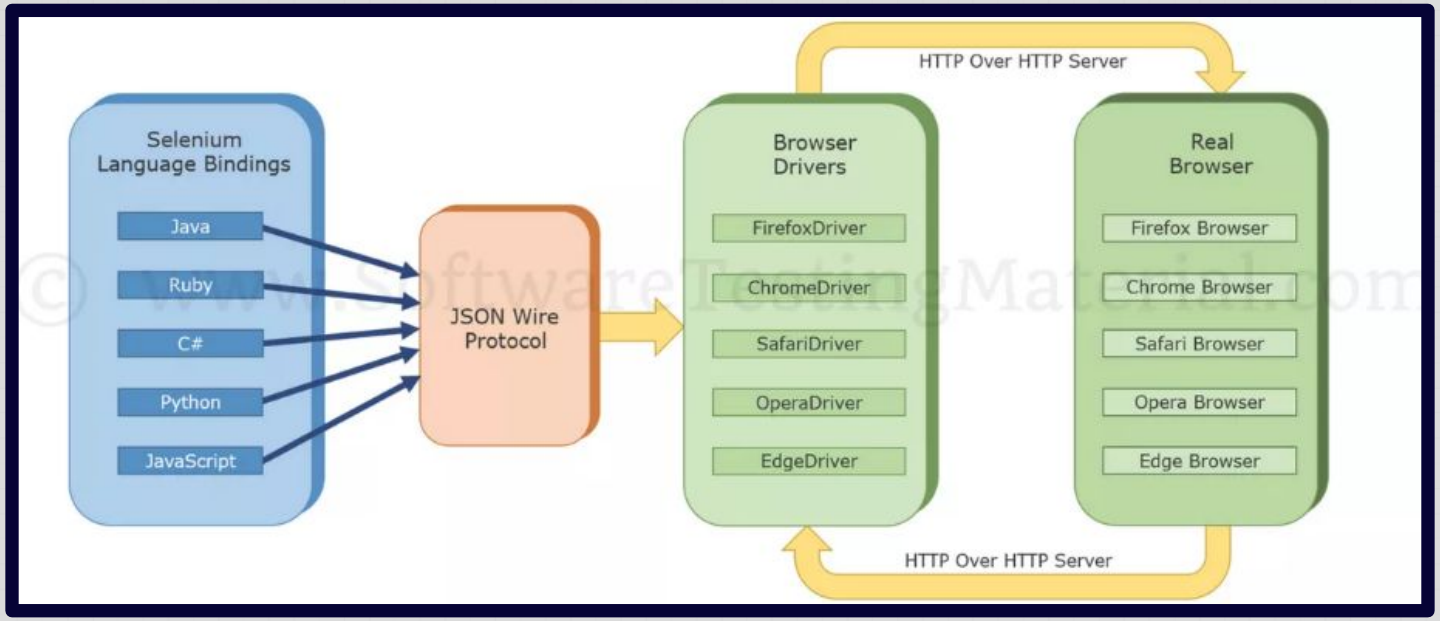


How Selenium WebDriver Works?

- When the automation script is executed, the following steps happen:
  - **for each Selenium command, a HTTP request is created and sent to the browser driver**
  - the browser driver **uses a HTTP server for getting the HTTP requests**
  - **the HTTP server** determines the **steps needed for implementing the Selenium command**
  - the implementation steps are executed on the browser
  - the execution status is sent back to the HTTP server
  - the **HTTP server sends the status back to the automation script**

How Selenium WebDriver Works?

- **WebDriver** is a web automation framework that allows you to execute your tests against different browsers



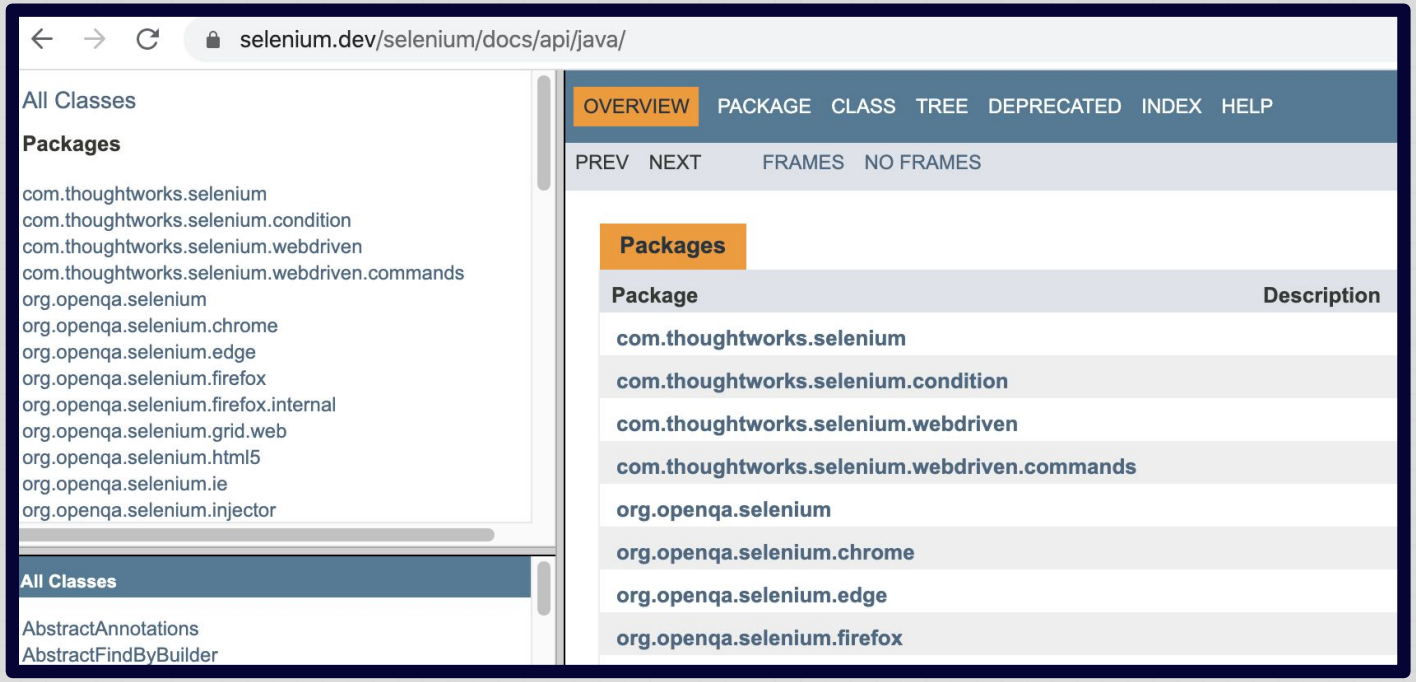
Selenium Architecture

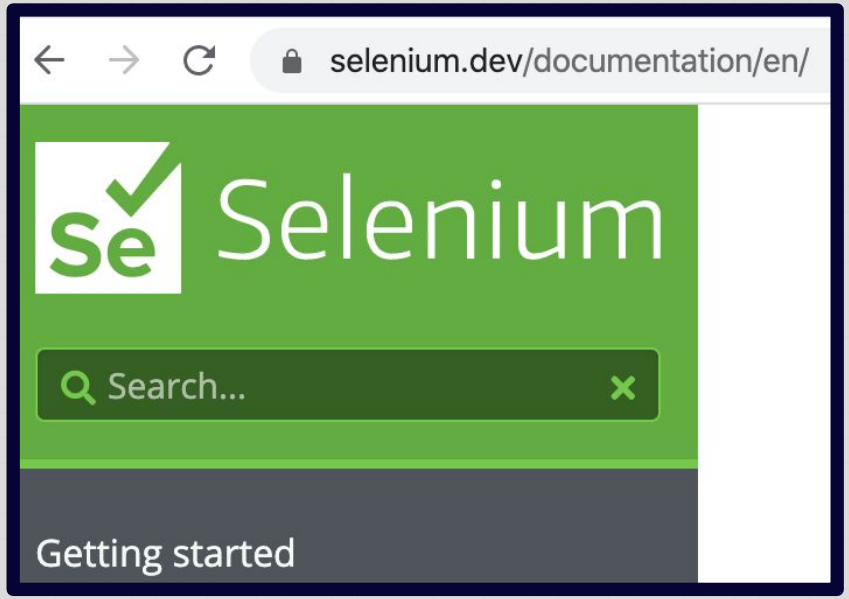
- 4 components of Selenium Architecture
- Selenium Client Library (Language Bindings) (Taxi Driver speaks only Chinese?, Italian?, we need to speak proper language)
- Browser Drivers (Different vehicles, different drivers like a truck, airplane, bulldozer etc.
- Browsers
- JSON Wire Protocol over HTTP

What Do I Need To Able To Run Selenium Program?

- JAVA installed
- IDE installed
  - Eclipse
  - Netbeans
  - IntelliJ IDEA
- Selenium jars or dependencies downloaded
- Configure selenium jars in project build path
- Create driver object based on browser chosen
- Set system property of the browser (not required in maven project using WebDriverManager)

Documentation Of Selenium





Let’s Create Our First Selenium Project!

- If you already created from the video and everything is working, please do not re-create!

.get(“url”);

- * get("url")--> this method launches the browser and goes to the given url.

SELENIUM NAVIGATIONS

- back -> takes to previous page
- forward -> takes us to forwarded page
- refresh -> refreshes the page
- to -> takes user to another given url.
- Syntax:

  driver.navigate().back();
- driver.navigate().forward;
- driver.navigate().refresh();
- driver.navigate().to();

.getTitle();

- Gets the title and returns it in a String
- Good for confirming that you are on the intended page or not.
- Syntax:

  driver.getTitle();

.getTitle();

- Thread: line of execution. One java program is running on one Thread.
- Thread.sleep(2000); -> pauses the program execution for given milliseconds.

.getCurrentUrl();

- Returns string that is current URL on browser.
- Good for confirming that you are on the intended page or not.

driver.manage().window().maximize();

- maximizes the current window

* driver.close();

- closes the current window