Unit Testing - Recap

with JUnit

Unit Testing

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently analysed for proper operation.

Types of Testing

- Unit Testing
- Integration Testing
- Regression Testing
- ...

https://www.softwaretestinghelp.com/types-of-software-testing/

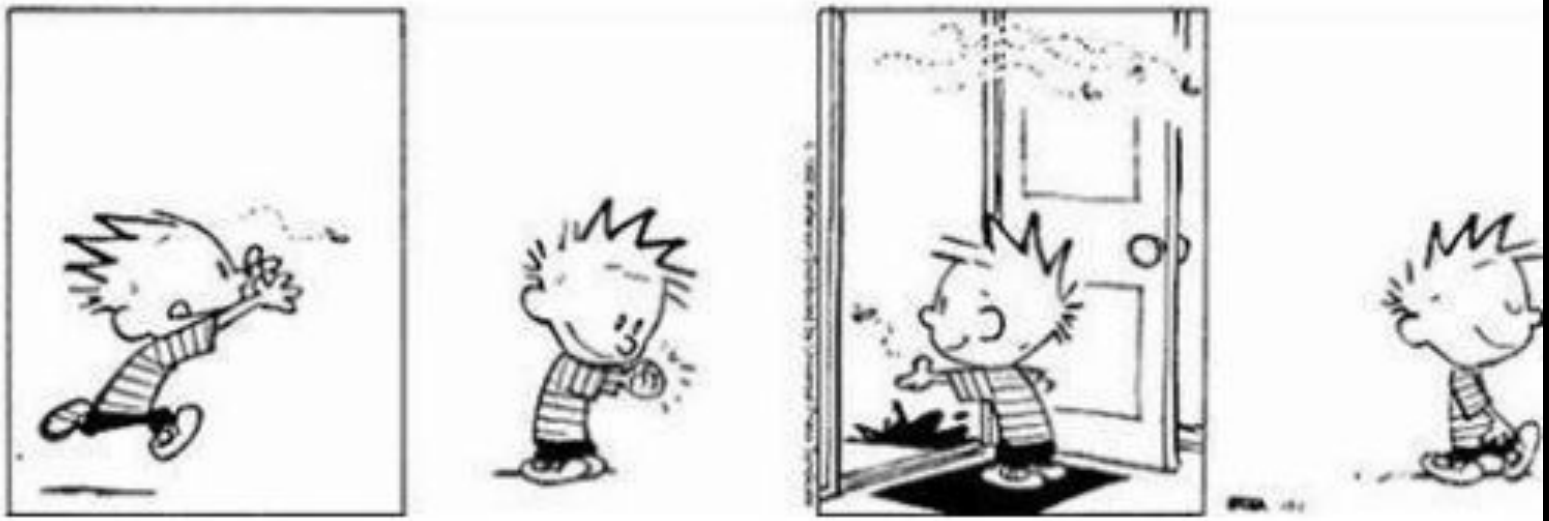
Integration Testing

Individual modules are combined and tested as a group.
Data transfer between the modules is tested as well.

Regression Testing

Regression means retesting the unchanged parts of the application.

Regression:
"when you fix one bug, you introduce several newer bugs."



Regression Testing

Test cases are re-executed in order to check whether the previous functionality of the application is working fine and the new changes have not introduced any new bugs.

This test can be performed on a new build when there is a significant change in the original functionality that too even in a single bug fix.

Manual Testing

The oldest type of software testing.

It requires a tester to perform manual test operations on the test software without automation scripts.

The tester choose which tests to run, when to run them, and how many times.

Do you want to run tests manually?

Automated Testing

Automated testing

To automatically verify main functionality, ensure new version does not cause new defects, provide regression testing and help the teams to run a large number of tests in a short period of time.

Companies having great number of projects are looking for specialists in the field of automated testing.

Automated testing tools

Automated web testing

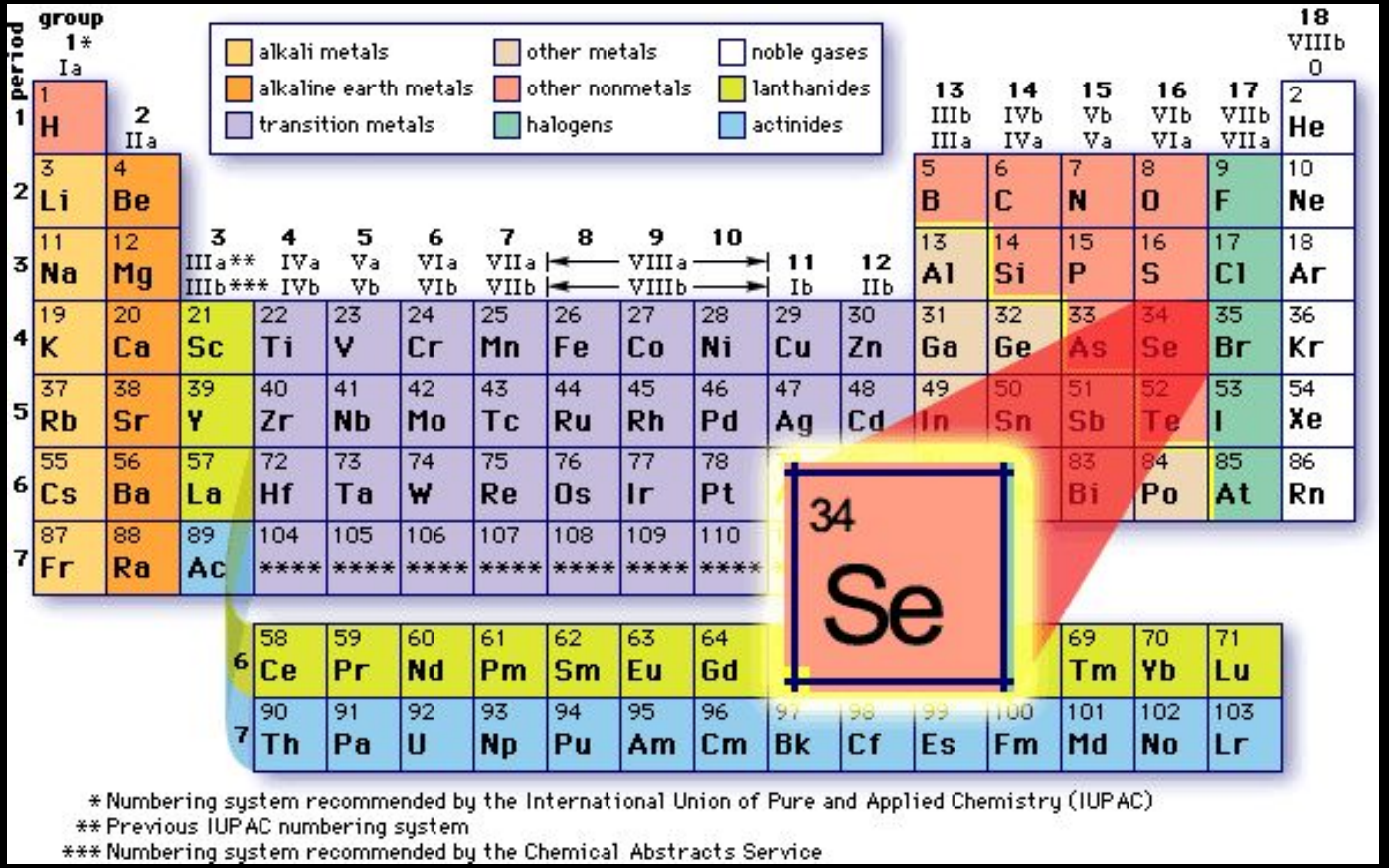
with Selenium

What is Web Testing?

What is Selenium?

Selenium

is a chemical element with symbol **Se** and atomic number **34**



What is Selenium?

Selenium is a suite of tools to automate web browsers across many platforms.

We will use it for automating tests of web applications

Why is it called Selenium then?

Why is it called Selenium then?

During the development of selenium core 2004 there was another competitive product developed by a company called **Mercury Interactive**.

A joke by Jason Huggins saying *that* ***"mercury poisoning can be cured by taking selenium supplements".***

Selenium is used to remove the toxic content mercury from the human body, so Jason coined the term Selenium for their creative open-source project.

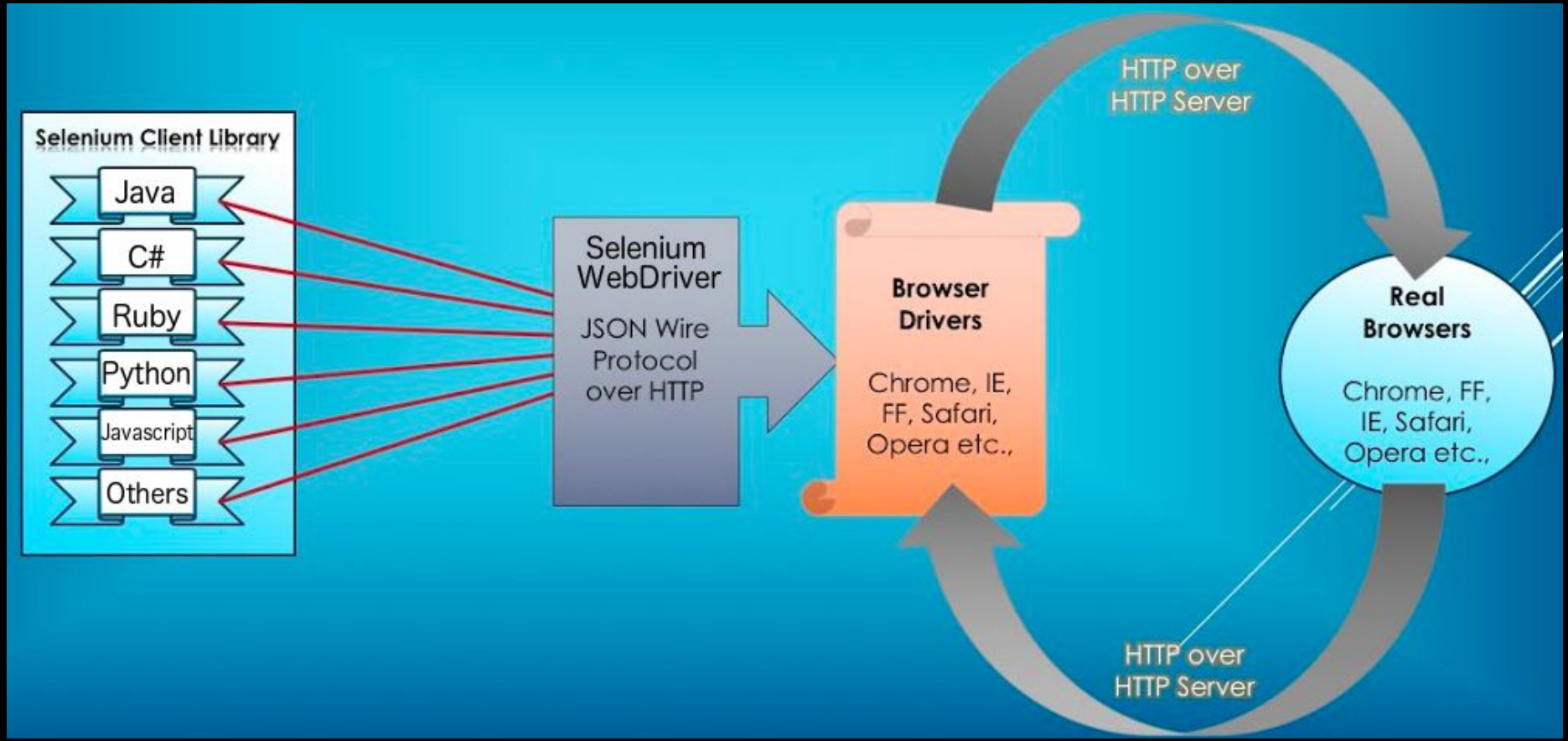
Which is the idea?

to remotely control browsers so that we can do things like write automated tests for the content they run or tests for the browser UI itself.

Should I write a test in a different way per each browser that is out there? No, to this end, a group of people from several organizations is working on the WebDriver Specification.

https://w3c.github.io/webdriver/

Selenium Architecture



Selenium WebDriver

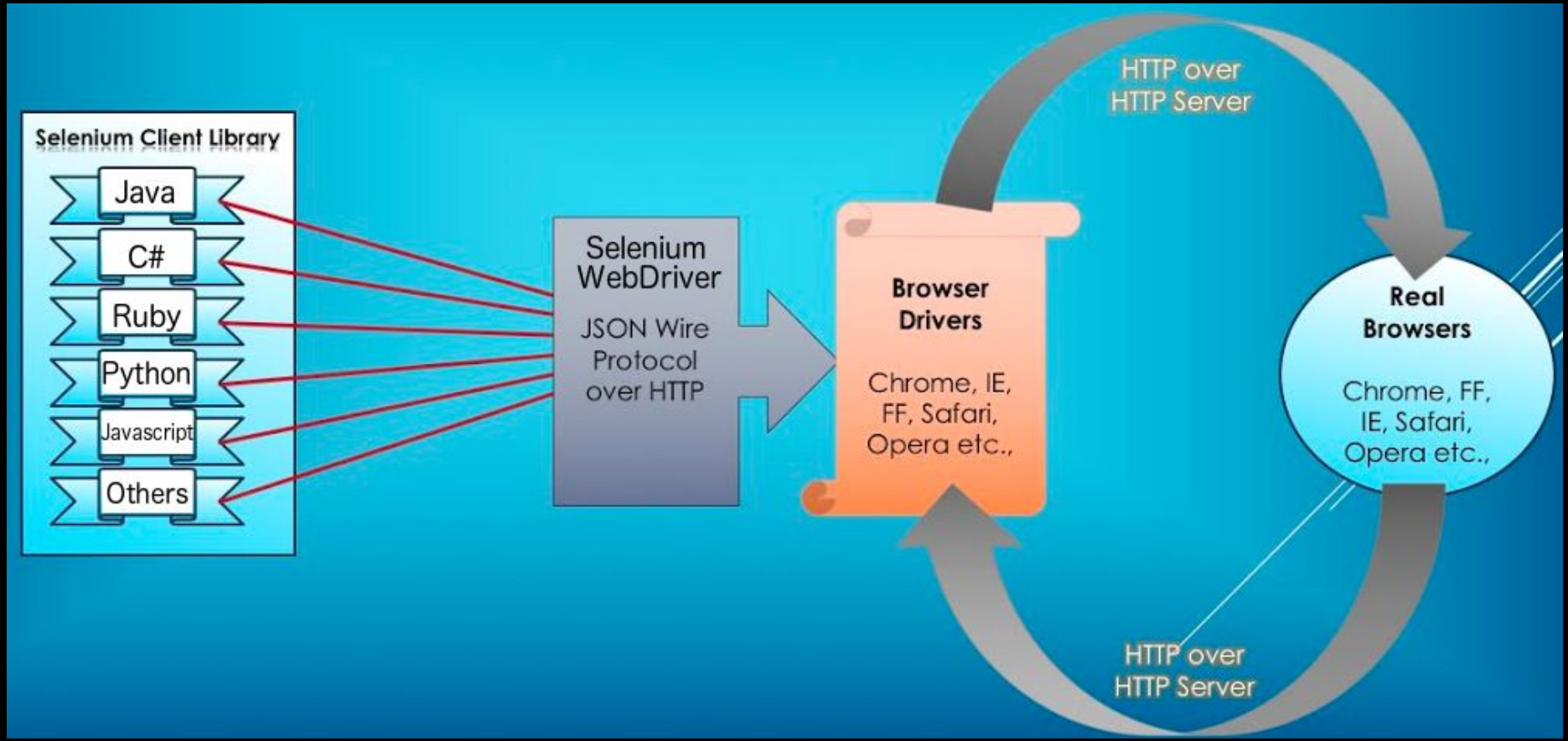
Selenium WebDriver provides APIs so that you can write code in your favourite language to simulate user actions like this:

```
client.get("http://pros.unicam.it/")
link = client.find_element_by_id("participate")
link.click()
```

Underneath that API, commands are transmitted via JSON over HTTP.

For example,
to tell the browser to navigate to a url, a client sends a POST request to the endpoint */session/{session id of the browser instance you're talking to}/url* with body *{"url": "http://pros.unicam.it/"}*.

Selenium Architecture



Browser Driver

At the beginning Selenium had to develop drivers for some browser they wanted to interact with.

Then, browser vendors started implementing the Selenium JSON Wire Protocol themselves!

This makes a lot of sense: they're in the best position to maintain the server side and they can build the necessary behaviour directly into the browser.

It started with OperaDriver in 2009-2011, and then others followed such as ChromeDriver and Mozilla's geckodriver with Marionette. This is where the motivation for a WebDriver standard comes from.

Selenium

GO TO: https://www.seleniumhq.org/

Hands on!

Step 0 Open Eclipse
Step 1 Create a new Maven Project
Step 2 Add Selenium Maven Dependency
Step 3 Download Third Party Browser Drivers
Step 4 Follow Me! :D

Selenium Locators

Preferred selector order : id > name > css > xpath

HTML

- Html is a standard markup language for creating Web pages. Html elements are the building blocks of HTML pages. HTML tags label pieces of content, such as “heading”, “paragraph”, “table”, and so on.

- HTML elements usually consists of start tag and end tag with content inserted between them.

- For example:
  - <h1> An example for an HTML title </h1>
  - An example for an HTML paragraph

CSS

- CSS is a language that describes the style of an HTML document.
- CSS describes how HTML elements should be displayed.
- A CSS rule-set consists of a selector and a declaration block. For example: selector - h1, declaration – {color:blue;}.

```
<style>

body {
  background-color: lightblue;
}

</style>
```

CSS

- CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.

- Some Selectors:

    - Element selector : The element selector selects elements based on the element name. for example: p, h1 etc.

    - ID selector: The id selector uses the id attribute of an HTML element to select a specific element. For example: #id1, #para2 etc.

- A CSS rule-set consists of a selector and a declaration block. For example: selector - h1, declaration – {color:blue;}.

XPath

- XPath stands for XML Path Language, it can be used to navigate through elements and attributes in an XML document.

- XPath uses path expressions to select nodes or node-sets in an XML document

Syntax:

- // - Selects nodes in the document from the current node that match the selection no matter where they are.
- @ - Selects attributes

XPath

- Example

```
<!DOCTYPE html>
<html>
<head>
<title> Page Title</title>
</head>
<body>
<h1 id=”h1_id”> This is part of the presentation title</h1>
<p>This is the paragraph.</p>
</body>
</html>
```

The query: //h1[@id=’h1_id’] will get the h1 element

Selenium WebDriver

- A Selenium Web driver must be created
- For using Chrome:
  - System.setProperty("webdriver.chrome.driver", projectPath+"/drivers/chromedriver");*
  - WebDriver driver = new ChromeDriver();
- Interaction with the Chrome instance will be made in the code on the driver.

*Note: you need to specify, before instantiating the BrowserDriver, the path to the actual driver that you downloaded following instructions from the selenium website https://www.seleniumhq.org/download/.

Selenium WebDriver

- Navigation using a Selenium WebDriver is very simple, given a defined URL. It can be done in two ways, driver.get(…) or driver.navigate().to(…)

  - *driver.get("https://www.google.com/");*
  - *driver.navigate().to("https://www.google.com/");*

- *The driver.get(…) and driver.navigate().to(…) do exactly the same thing. driver.navigate() supports also driver.navigate().forward() and driver.navigate().backward()*

Finding Web Elements

- Web elements can be defined as each opening and closing tags in the web page. For example: <button>Click Me</button> - a web element.

- Finding a web element and interacting with it can be done in several ways: - ID. - Class. - Name. - Xpath. - Css Selector, etc.

Finding Web Elements

- An example:
  - Assuming that we have the following web page:

```
<html>
    <body>
        <button id= “my_button”> Click Me</button>
    </body>
</html>
```

  - The following lines of code will be used for clicking the button:

```
WebElement button = driver.findElement(By.id(" my_button "));
button.click();
```

Selenium IDE

Download it from:

https://www.seleniumhq.org/selenium-ide/

and let us see what we can do with it...

Katalon Recorder

Download the extension for the browser you want to use

Try to record the same test

Try to find any difference

If you missed last lecture...

You can find here a nice introduction on Selenium (not done by me):

https://www.youtube.com/watch?v=SQkyo1k7c8A

Selenium

...still something to say about it

Explicit Waits

An explicit wait is code you define to wait for a certain condition to occur before proceeding further in the code.

Avoid using Thread.sleep(ms);

```
WebDriver driver = new FirefoxDriver();
driver.get("http://somedomain/url_that_delays_loading");

WebElement myDynamicElement = (new WebDriverWait(driver, 10))
.until(ExpectedConditions.presenceOfElementLocated(By.id("myDynamicElement")));
```

Screenshot

```
public void takeSnapShot(WebDriver webdriver,String fileWithPath){

    TakesScreenshot scrShot =((TakesScreenshot)webdriver);

    //Call getScreenshotAs method to create image file
    File SrcFile=scrShot.getScreenshotAs(OutputType.FILE);

    //Move image file to new destination
    File DestFile=new File(fileWithPath);
    FileUtils.copyFile(SrcFile, DestFile);

}
```

Headless Browser...

Headless Browser…

- It is a browser without graphical interface
- What is it for?

Headless Browser...

It is a browser without graphical interface

Headless browsers are commonly used for:

- Website and application testing
- JavaScript library testing
- JavaScript simulation and interactions
- Running one or more automated UI tests in the background

Headless Browser...

In a headless testing environment, you can write and execute scripts to:

- Test basic and alternative flows
- Simulate clicks on links and buttons
- Automate form filling and submission
- Test SSL performance
- Experiment with various server loads
- Get reports on page response times
- Scrape useful website code
- Take screenshots of results

Testing these use cases provides you with a solid overview of how a site's UI performs and gives you essential information for making changes before deployment.

Which Headless Browser…?

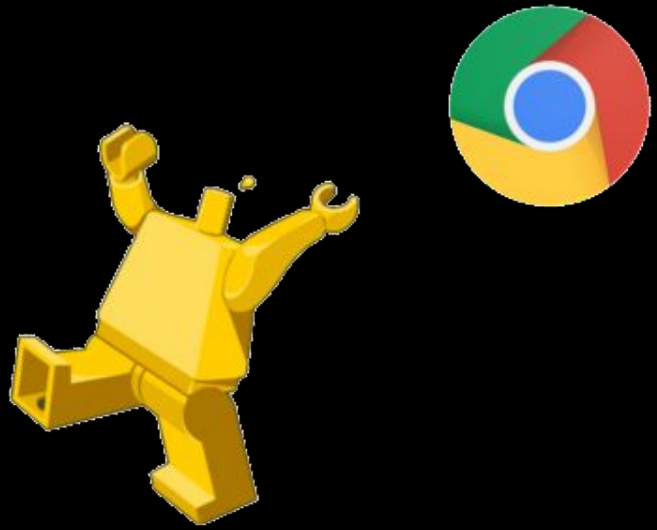
Can you name one Headless Browser?

Which Headless Browser…?

- Firefox Headless Mode
- Headless Chrome
- PhantomJS
- Zombie JS
- HtmlUnit
- Splash

Hands on!

1.Headless Chrome



2.Html Unit

3.PhantomJS

Keep on practicing, take a look at:

https://github.com/FabrizioFornari/PracticeWithSelenium.git