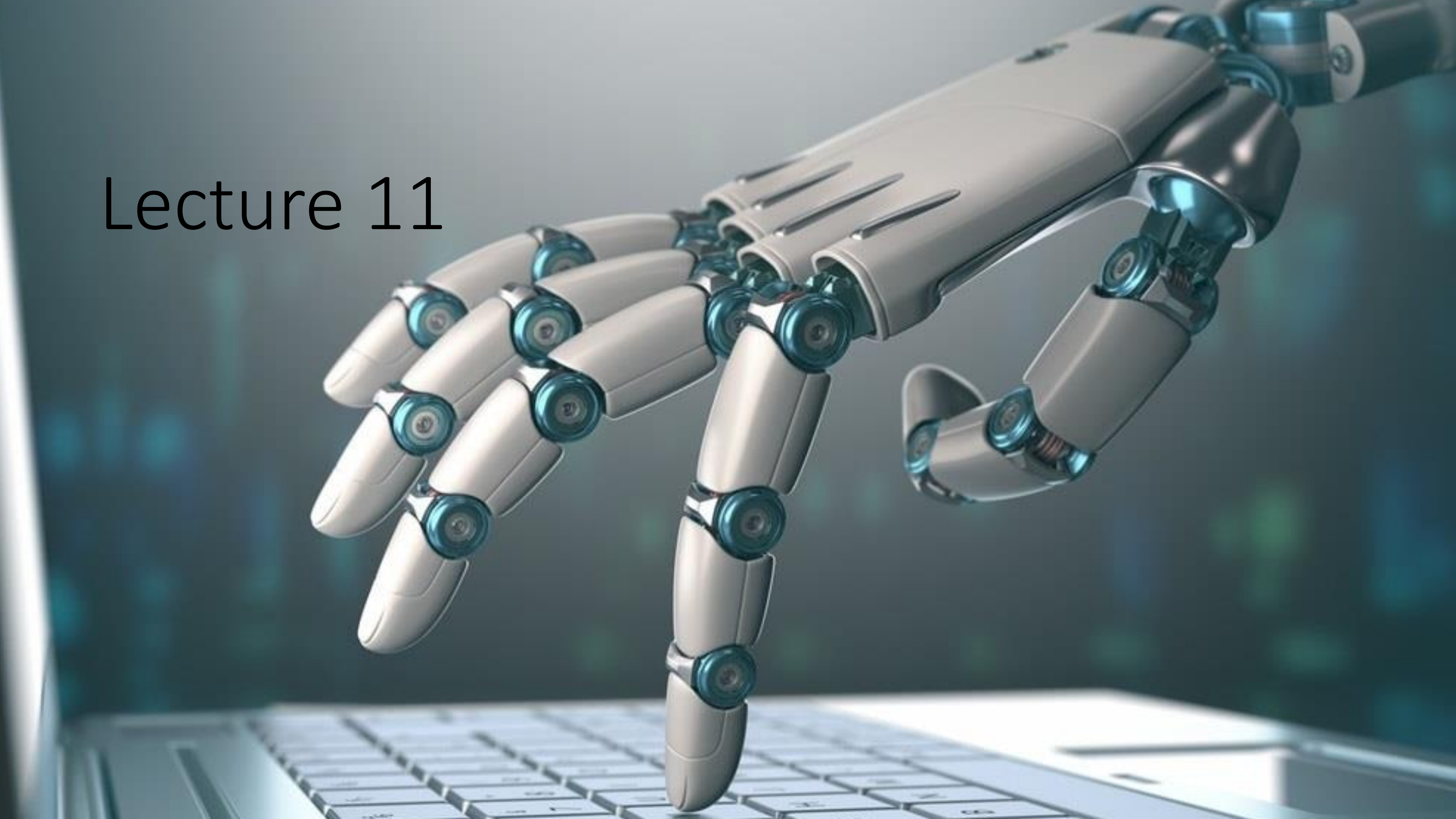
Lecture 11



Agenda

- **What is Selenide**
- Selenide usage and examples
- Final project
- Practice

Selenide

- Selenide is a framework for test automation powered by Selenium WebDriver that brings the following advantages:
- Concise fluent API for tests
- Ajax support for stable tests
- Powerful selectors
- Simple configuration
- You don't need to think how to shutdown browser, handle timeouts and StaleElement Exceptions or search for relevant log lines, debugging your tests.
  Just focus on your business logic and let Selenide do the rest!

Why to know Selenide

- Popular in Java World
- Simple to use
- Concise API inspired by jQuery
- Automatic handling of most problems with Ajax, waiting and timeouts
- Automatic handling of browser lifecycle
- Automatic screenshots on test failures

Selenide showstoppers

- Only for Java
- Some workaround for C#
- Currently no support for virtual DOM
- Can be addictive for simplicity ☺

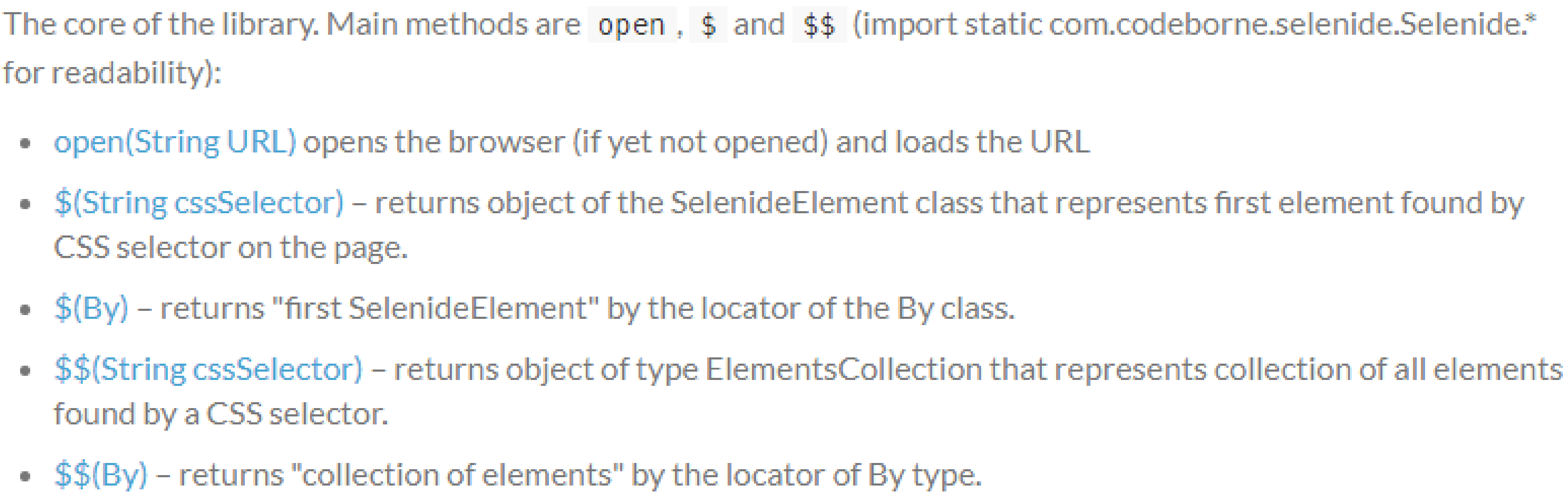
Agenda

- What is Selenide
- **Selenide usage and examples**
- Final project
- Practice

How to use in gradle project?

```
dependencies {
testCompile 'com.codeborne:selenide:5.2.4'
}
```

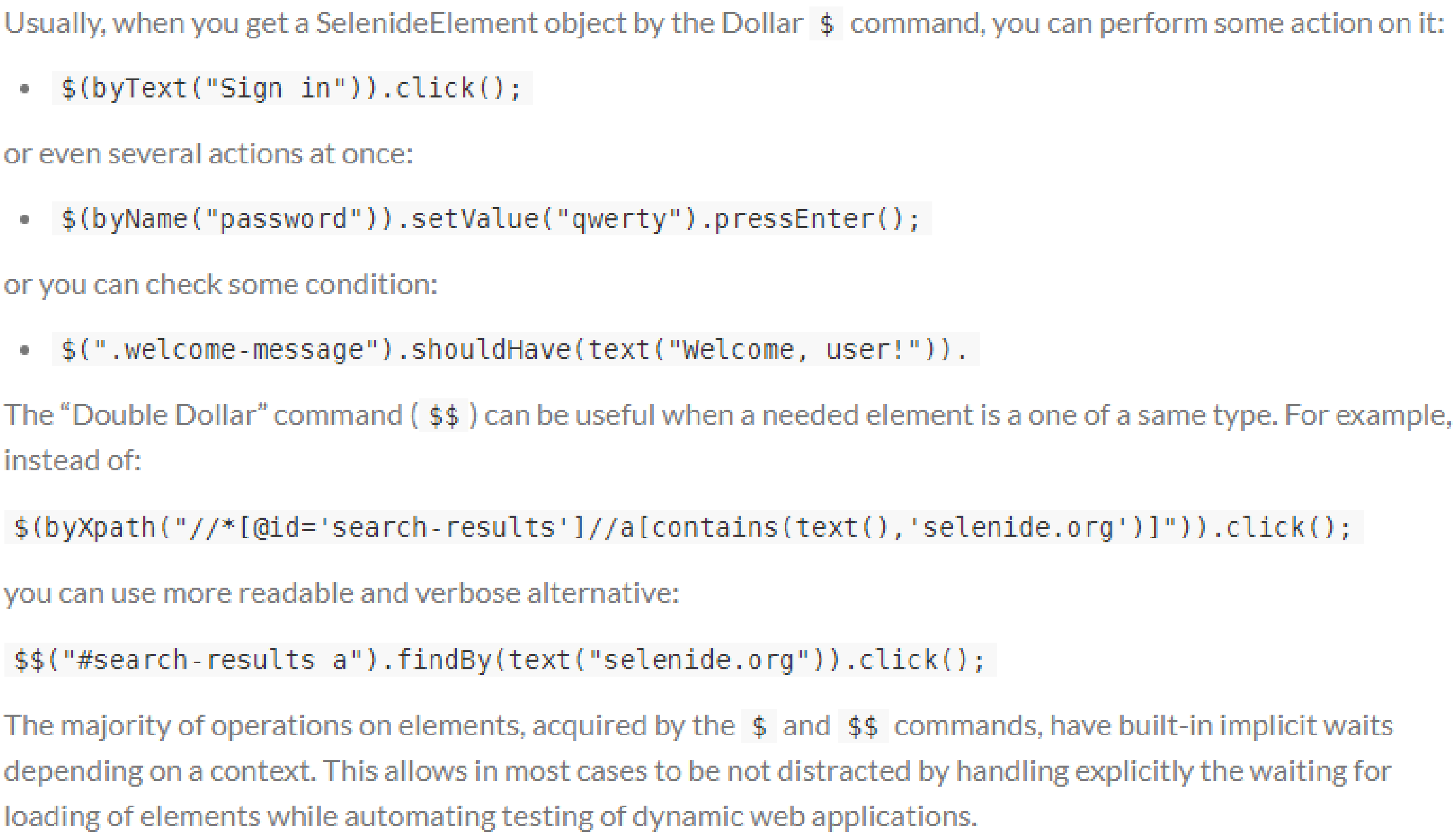
Selenide basics



The core of the library. Main methods are `open` , `$` and `$$` (import static com.codeborne.selenide.Selenide.* for readability):

- open(String URL) opens the browser (if yet not opened) and loads the URL
- $(String cssSelector) – returns object of the SelenideElement class that represents first element found by CSS selector on the page.
- $(By) – returns "first SelenideElement" by the locator of the By class.
- $$(String cssSelector) – returns object of type ElementsCollection that represents collection of all elements found by a CSS selector.
- $$(By) – returns "collection of elements" by the locator of By type.

Selenide basics



Usually, when you get a SelenideElement object by the Dollar `$` command, you can perform some action on it:

- `$(byText("Sign in")).click();`

or even several actions at once:

- `$(byName("password")).setValue("qwerty").pressEnter();`

or you can check some condition:

- `$(".welcome-message").shouldHave(text("Welcome, user!")).`

The "Double Dollar" command (`$$`) can be useful when a needed element is a one of a same type. For example, instead of:

```
$(byXpath("//*[@id='search-results']//a[contains(text(),'selenide.org')]")).click();
```

you can use more readable and verbose alternative:

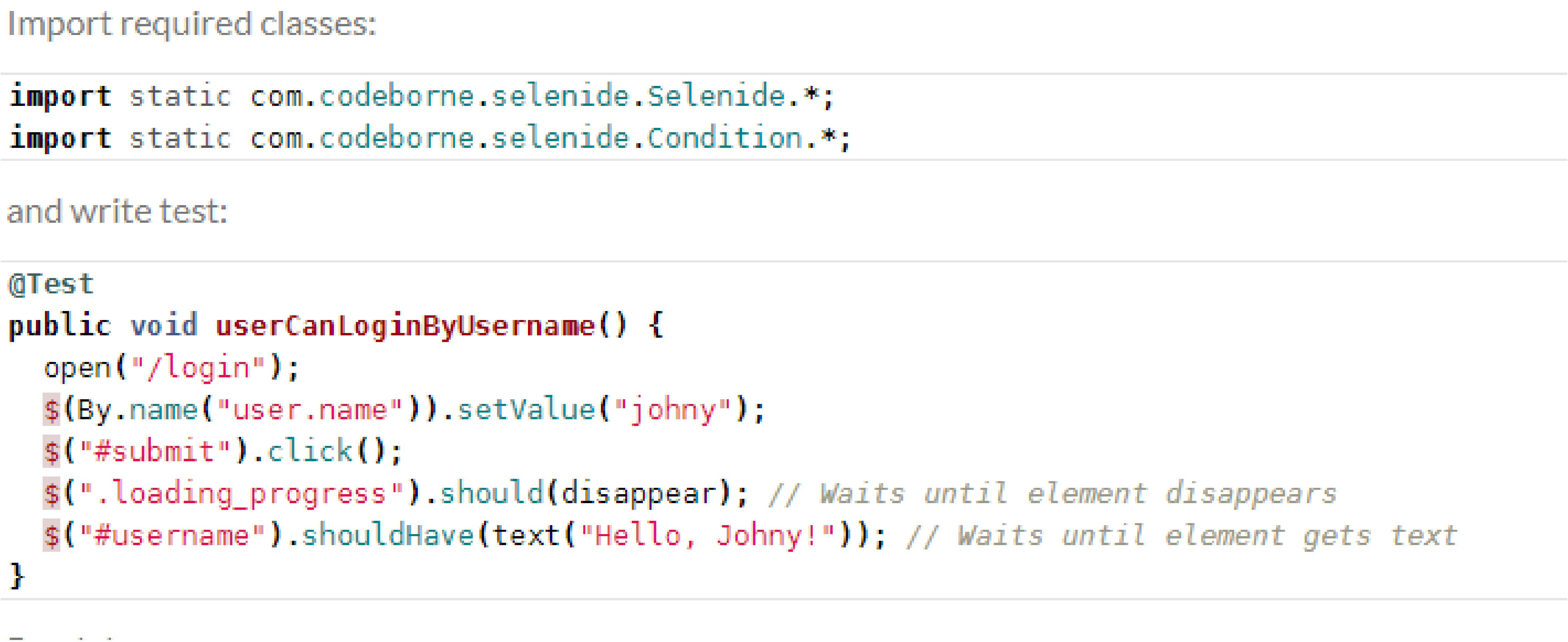
```
$$("#search-results a").findBy(text("selenide.org")).click();
```

The majority of operations on elements, acquired by the `$` and `$$` commands, have built-in implicit waits depending on a context. This allows in most cases to be not distracted by handling explicitly the waiting for loading of elements while automating testing of dynamic web applications.

Selenide cheat sheet

- https://gist.github.com/mkpythonanywhereblog/947633ba1bf0bc239639

Sample



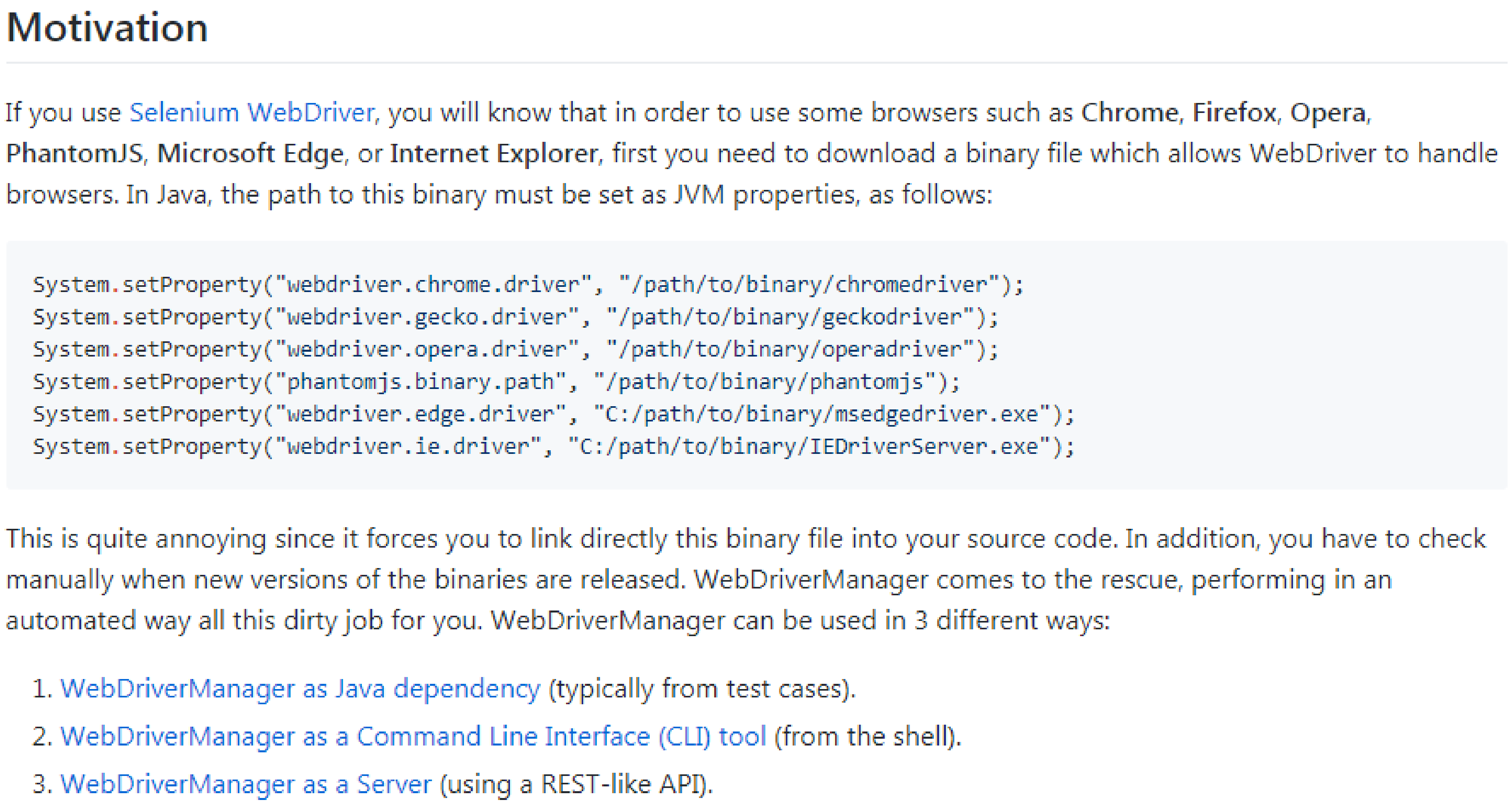
Import required classes:

```
import static com.codeborne.selenide.Selenide.*;
import static com.codeborne.selenide.Condition.*;
```

and write test:

```
@Test
public void userCanLoginByUsername() {
  open("/login");
  $(By.name("user.name")).setValue("johny");
  $("#submit").click();
  $(".loading_progress").should(disappear); // Waits until element disappears
  $("#username").shouldHave(text("Hello, Johny!")); // Waits until element gets text
}
```

Webdriver manager



Motivation

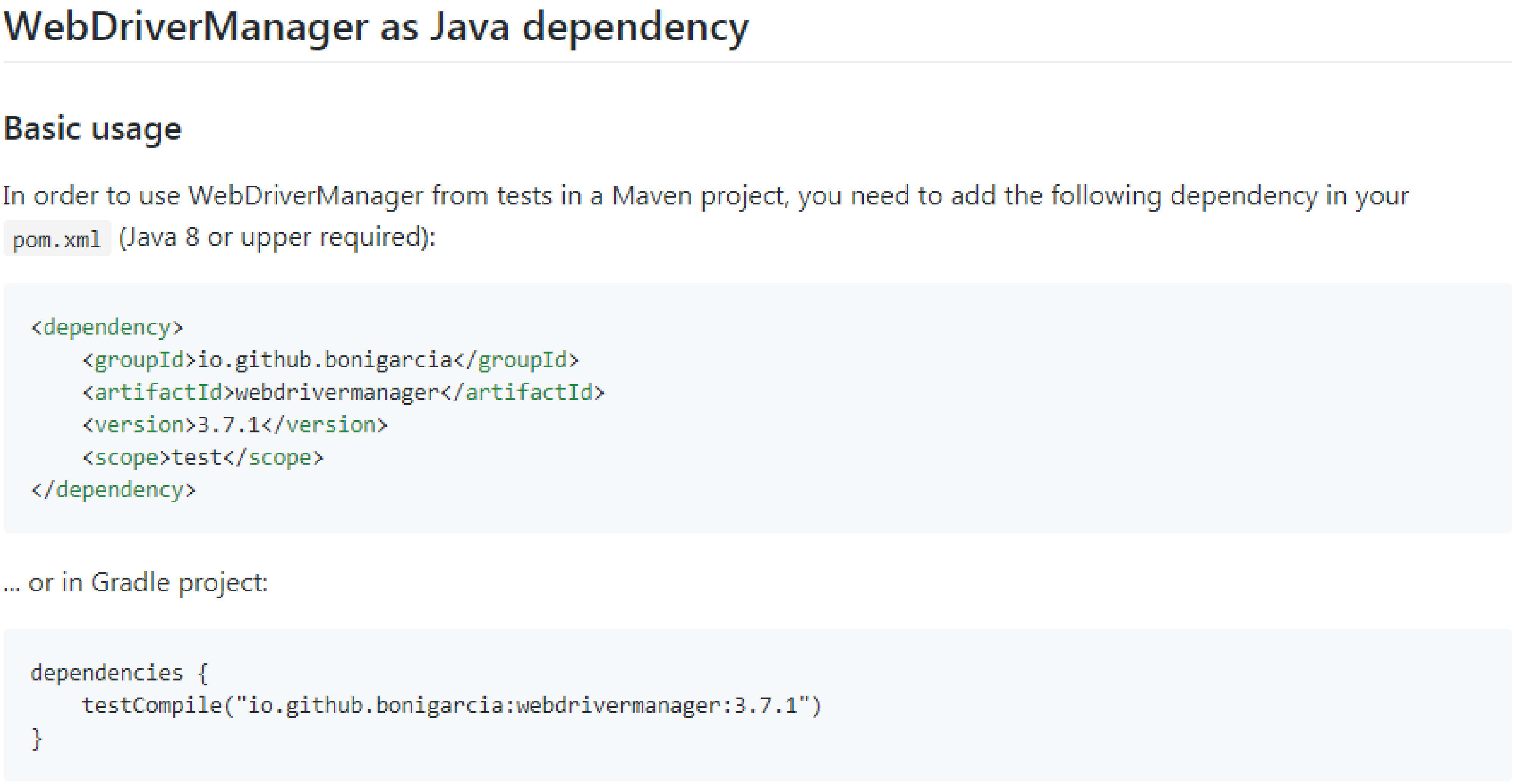
If you use Selenium WebDriver, you will know that in order to use some browsers such as **Chrome**, **Firefox**, **Opera**, **PhantomJS**, **Microsoft Edge**, or **Internet Explorer**, first you need to download a binary file which allows WebDriver to handle browsers. In Java, the path to this binary must be set as JVM properties, as follows:

```
System.setProperty("webdriver.chrome.driver", "/path/to/binary/chromedriver");
System.setProperty("webdriver.gecko.driver", "/path/to/binary/geckodriver");
System.setProperty("webdriver.opera.driver", "/path/to/binary/operadriver");
System.setProperty("phantomjs.binary.path", "/path/to/binary/phantomjs");
System.setProperty("webdriver.edge.driver", "C:/path/to/binary/msedgedriver.exe");
System.setProperty("webdriver.ie.driver", "C:/path/to/binary/IEDriverServer.exe");
```

This is quite annoying since it forces you to link directly this binary file into your source code. In addition, you have to check manually when new versions of the binaries are released. WebDriverManager comes to the rescue, performing in an automated way all this dirty job for you. WebDriverManager can be used in 3 different ways:

1. WebDriverManager as Java dependency (typically from test cases).
2. WebDriverManager as a Command Line Interface (CLI) tool (from the shell).
3. WebDriverManager as a Server (using a REST-like API).

Webdriver Dependency



WebDriverManager as Java dependency

Basic usage

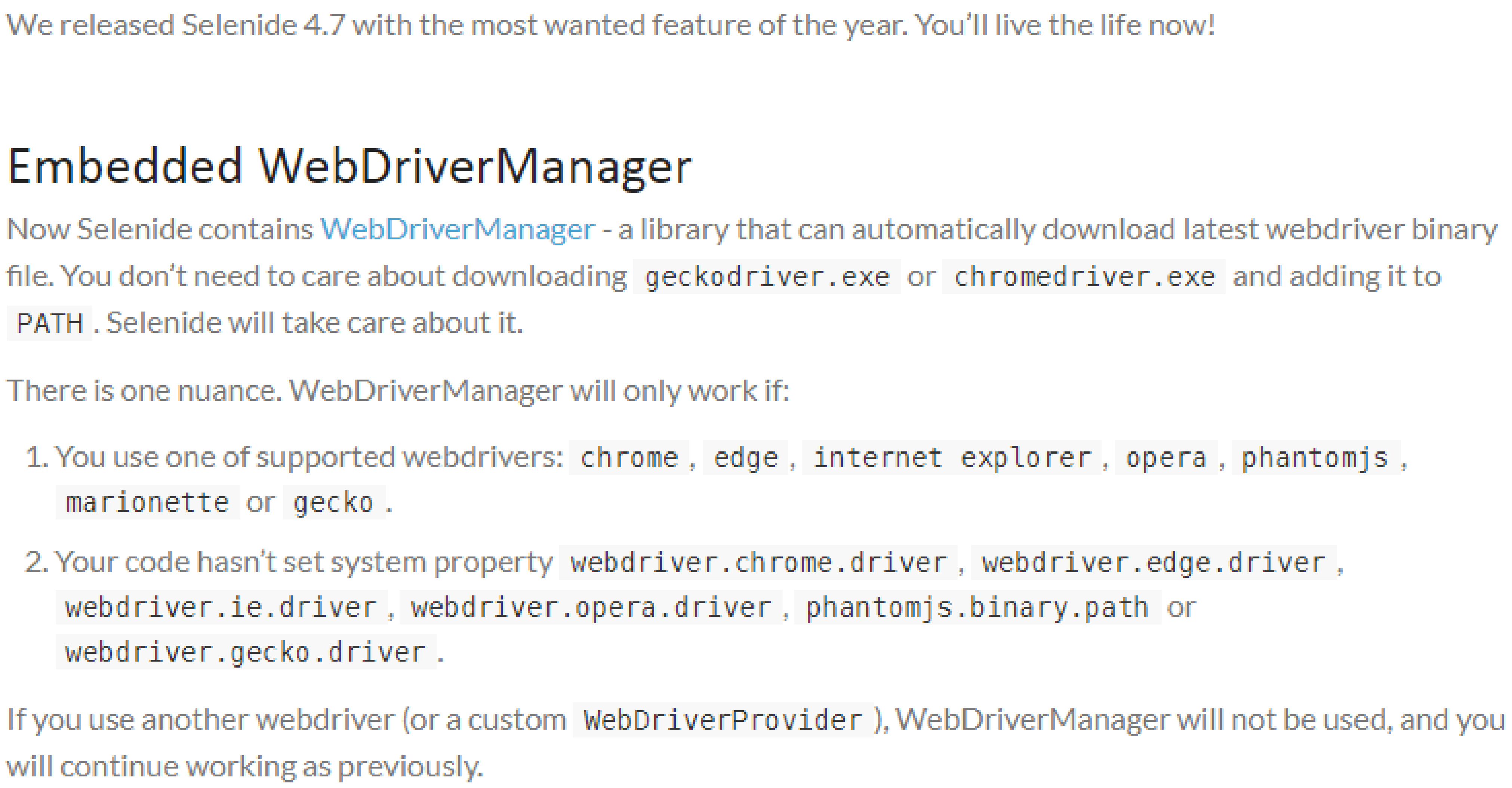
In order to use WebDriverManager from tests in a Maven project, you need to add the following dependency in your `pom.xml` (Java 8 or upper required):

```
<dependency>
    <groupId>io.github.bonigarcia</groupId>
    <artifactId>webdrivermanager</artifactId>
    <version>3.7.1</version>
    <scope>test</scope>
</dependency>
```

... or in Gradle project:

```
dependencies {
    testCompile("io.github.bonigarcia:webdrivermanager:3.7.1")
}
```

Webdriver manager



We released Selenide 4.7 with the most wanted feature of the year. You'll live the life now!

Embedded WebDriverManager

Now Selenide contains WebDriverManager - a library that can automatically download latest webdriver binary file. You don't need to care about downloading `geckodriver.exe` or `chromedriver.exe` and adding it to `PATH`. Selenide will take care about it.

There is one nuance. WebDriverManager will only work if:

1. You use one of supported webdrivers: `chrome`, `edge`, `internet explorer`, `opera`, `phantomjs`, `marionette` or `gecko`.
2. Your code hasn't set system property `webdriver.chrome.driver`, `webdriver.edge.driver`, `webdriver.ie.driver`, `webdriver.opera.driver`, `phantomjs.binary.path` or `webdriver.gecko.driver`.

If you use another webdriver (or a custom `WebDriverProvider`), WebDriverManager will not be used, and you will continue working as previously.

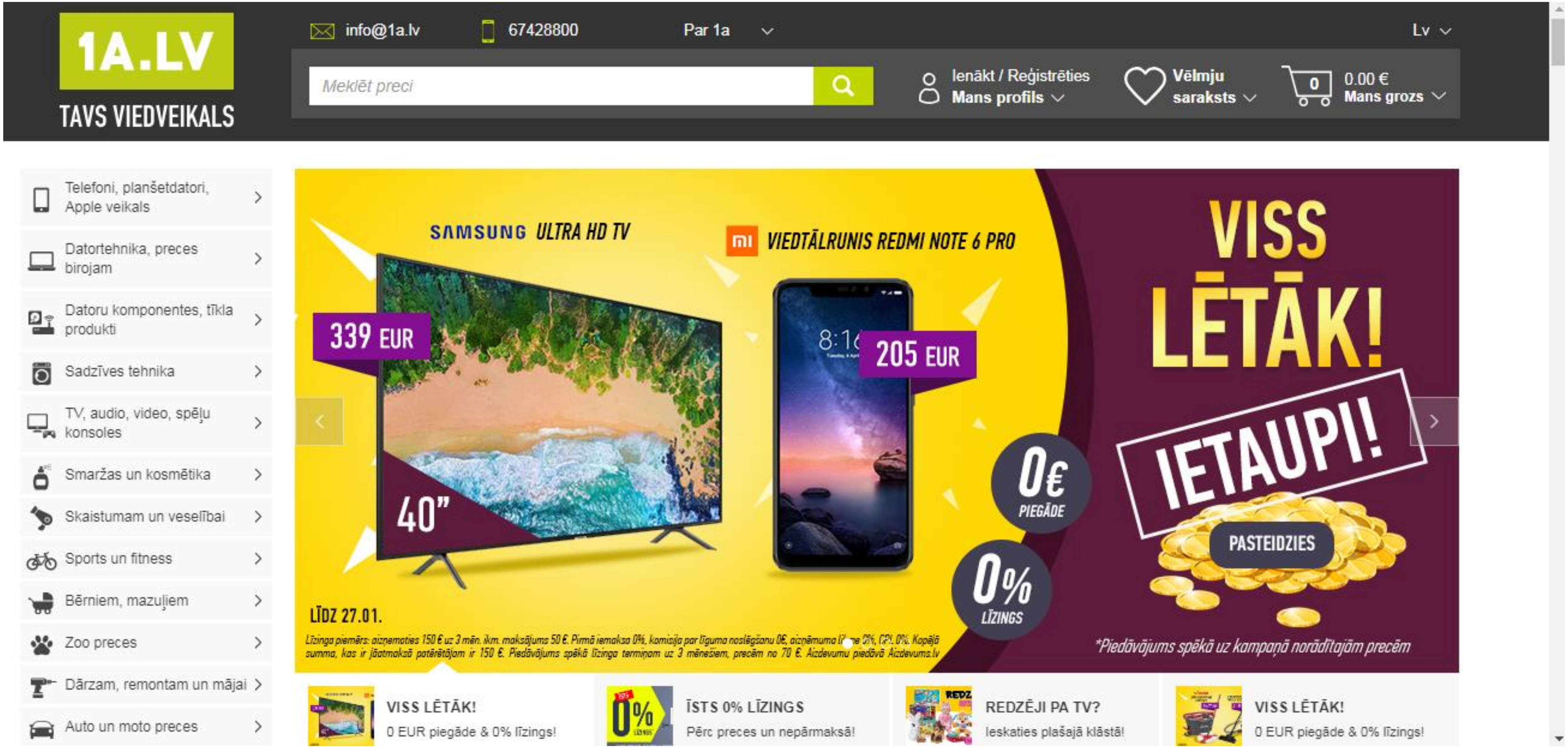
Agenda

- What is Selenide
- Selenide usage and examples
- **Final project**
- Practice

Web page

- https://www.1a.lv/
- It is a webshop - ecommerce
- Most commonly automated website
- More details in practice part
- 3 levels of difficulty is available for Final project

Web shop – what we see?



1A.LV
TAVS VIEDVEIKALS

info@1a.lv
67428800
Par 1a
Lv

Meklēt preci

Ienākt / Reģistrēties
Mans profils

Vēlmju saraksts

0
0.00 €
Mans grozs

- Telefoni, planšetdatori, Apple veikals
- Datortehnika, preces birojam
- Datoru komponentes, tīkla produkti
- Sadzīves tehnika
- TV, audio, video, spēļu konsoles
- Smaržas un kosmētika
- Skaistumam un veselībai
- Sports un fitness
- Bērniem, mazuļiem
- Zoo preces
- Dārzam, remontam un mājai
- Auto un moto preces

Līzinga piemērs: aizņemoties 150 € uz 3 mēn. ikm. maksājums 50 €. Pirmā iemaksa 0%, komisija par līguma noslēgšanu 0€, aizņēmuma likme 0%, GPL 0%. Kopējā summa, kas ir jāatmaksā patērētājam ir 150 €. Piedāvājums spēkā līzinga termiņam uz 3 mēnešiem, precēm no 70 €. Aizdevumu piedāvā Aizdevums.lv

*Piedāvājums spēkā uz kampaņā norādītajām precēm

VISS LĒTĀK!
0 EUR piegāde & 0% līzings!

ĪSTS 0% LĪZINGS
Pērc preces un nepārmaksā!

REDZĒJI PA TV?
Ieskaties plašajā klāstā!

VISS LĒTĀK!
0 EUR piegāde & 0% līzings!

Final project Requirements – Steps on Automation

1. Search product Any Product
2. Choose Brand
3. Choose filter with most stars/most popular - optional
4. Press on any product
5. Add to cart
6. Go to Cart
7. Proceed to check out without user
8. Choose Get in office

Final project Requirements – Steps on Automation

1. Userform
2. Choose pay with cash
3. Validate final price
4. Do not press button buy ☺

Additional Requirements

- Validate that page has been loaded
- Price on product page = Final price
- Product name = product name in the end
- Proper architecture
- Product and Customer model
- Should work with headless and normal webdriver
- You will show me your demo project on last lecture
- **Only who will do final project will receive Certificate**

Level 1 Basic: Requirements

- Models
- POM
- Selenium
- Junit @Test
- @Before and @After use to setup webdriver

Level 2 Advanced: Requirements

- Models
- POM
- Selenide
- Cucumber

Level 3 Professional: Requirements

- Create a web scrapper – which will scrape specific product and send email to you
- Architecture: Selenium+POMs+Junit4+Models
- Print results in console: Email content – Subject – Product name, Body – Name of product, price and link attached
- To send by email – you have to read additional guide - https://www.tutorialspoint.com/java/java_sending_email.htm
- To save to file all your results – you have to read additional guide - https://www.w3schools.com/java/java_files_create.asp

Agenda

- What is Selenide
- Selenide usage and examples
- Final project
- **Practice**

Steps for Selenide usage

- Add library to project
- Have fun ☺

Demo

Practice 1

- Go to https://www.ss.com/lv/
- Open cars
- Enter price 6000 - 10000
- year from 2001
- engine max 3.0
- colour – Balta
- Click submit

Practice 2

- Go to https://www.1a.lv/
- Press **Datortehnika, preces birojam**
- Press **Portatīvie datori un aksesuāri**
- Press **Lenovo**

Practice 3

- Go to https://www.aliexpress.com/
- In search write: tattoo
- Press search
- Set min price 10
- Set max price 20
- Press ok

Bonus – Same, but use Selenide

- Register some user for Forum Cinemas
- Based on POM Create Login page
- Profile POM page
- Write a test to login to ForumCinemas.lv
- Go To profile and Edit information – Edit all the fields on this page
- Profile information as Models
- Add cucumber steps on above level
- **Bonus** – Select any random movie and open seat page

Bonus 2 – Same, but use Selenide

- Open https://www.janisroze.lv
- In search enter bvs superman
- Click on first found search (assert figure name is Figūra POP! DC: BvS: Superman)
- Get item to basket and go to Basket
- Create a Order model
- Open Order form and fill it
- Enter Coupon name WhyNot?
- DO NOT PRESS order button
- Add cucumber steps on above level

Homework

- Continue working on your project
- Additional task – works with any category
- On final lecture we will check the results and compare what we have done
- Each of you will show small demo of your final project and show the code

Reference

- Selenide Examples - https://github.com/selenide/selenide
- Selenide - https://selenide.org/
- Webdriver Manager - https://github.com/bonigarcia/webdrivermanager

Reference

- Best Practices for Automation - https://www.blazemeter.com/blog/top-15-ui-test-automation-best-practices-you-should-follow

THANK YOU FOR YOUR ATTENTION