

# Selenium



## Agenda

- Who and What
- The Basics
- Real life Examples





## Oliver & Tom

lets over ons

**Zoek je een  
vakantiebaan  
in de IT met een  
uitdaging?**

Bouw mee aan een applicatie  
voor een van de goede doelen!



**InfoSupport**  
Solid Innovator

Afgelopen jaar is voor de Vlaamse Scholierenkoepel (VSK) een interactieve map gebouwd. In 2019 is een app voor VZW Prinses Harte ontwikkeld waarop kinderen met kanker en hun ouders hun verhaal kunnen delen. Ook in 2021 bieden we een goed doel de mogelijkheid gebruik te maken van onze kennis en kunde. Ben je IT-student? Dan kan je mee-ontwikkelen! Meer informatie of solliciteren?

[>> carriere.infosupport.com/vakantiejob](http://carriere.infosupport.com/vakantiejob)

**Jouw profiel**

- > 2de/3de jaar student
- > Toegepaste Informatica
- > Passie voor IT
- > Eerste ervaring in .NET/Java en Angular/React/...

**Wij bieden**

- > Interessante IT-vakantiejob in juli
- > in Mechelen, Generaal de Wittelaan 17
- > Goed loon
- > Begeleiding van onze consultants
- > Optie op stage/job

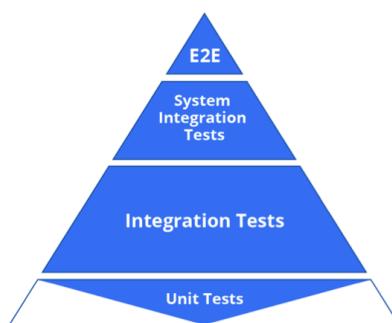
**Start in juli**  
**Solliciteer voor 1 mei**



**InfoSupport**  
Solid Innovator

# Selenium?

## The Pyramid



## Example Tools

Unit test



## Integration & System Test



## E2E Test

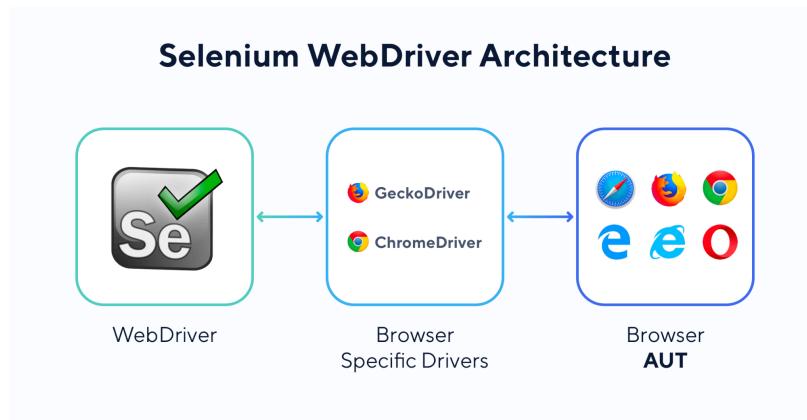


## Languages

- Java
- Python
- C#
- Ruby
- Javascript
- Perl, PHP
- GO
- Kotlin\*

## Basics

# WebDriver



## Driver Methods

```
driver.quit()  
driver.navigate()  
driver.get()  
driver.wait()  
driver.findElement() / driver.findElements()  
...
```

## Flavours

- ChromeDriver
- FirefoxDriver
- EdgeDriver
- OperaDriver
- SafariDriver
- InternetExplorerDriver

## Selectors

## Query Selector

```
By.id()  
By.name()  
By.className()  
By.tagName()  
By.link()  
By.partialLinkText()  
By.cssSelector()  
By.xpath()
```

### Select by ID

```
<button id="submitButton"></button>  
  
WebElement element = driver.findElement(By.id("submitButton"));
```

## Select by className

```
<button class="greenButton"></button>
```

```
WebElement element = driver.findElement(By.className("greenButton"));
```

## Select by name

```
<button name="greenButton"></button>
```

```
WebElement element = driver.findElement(By.name("greenButton"));
```

## Select by cssSelector

Select an element using the CSS selection syntax. See:  
[https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

```
<textarea id="textArea"></textarea>
<button data-testid="submit-button"></button>
```

```
WebElement textarea = driver.findElement(By.cssSelector("#textArea"));
WebElement button = driver.findElement(By.cssSelector("button[data-testid='subm"));
```

## Select by XPath

Select an element using Xpath. See:  
[https://www.w3schools.com/xml/xpath\\_syntax.asp](https://www.w3schools.com/xml/xpath_syntax.asp)

```
<textarea id="textArea"></textarea>
<button data-testid="submit-button"></button>
```

```
WebElement textarea = driver.findElement(By.xpath("//textarea[@id='textArea']"))
WebElement button = driver.findElement(By.xpath("//button[@data-testid='submit-"));
```

# WebElements

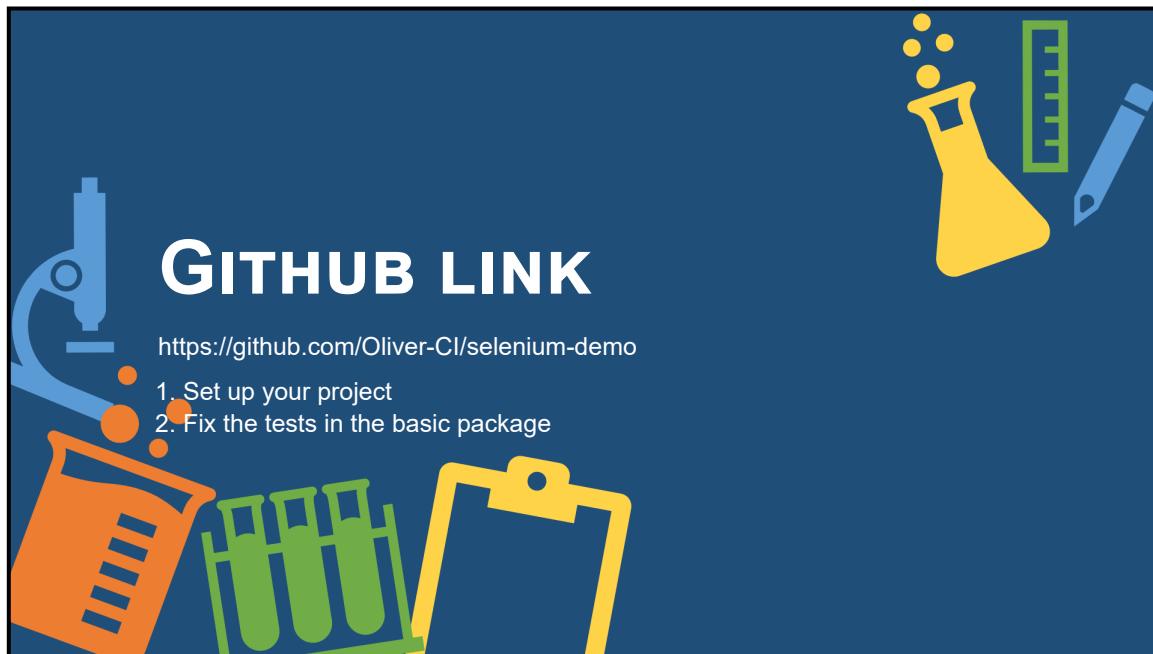
## Finding elements

driver.findElement vs driver.findElements

## Element methods

```
WebElement element = new WebElement();
element.click()
element.getText()
element.isDisplayed()
element.getCssValue()
element.sendKeys()
...
```

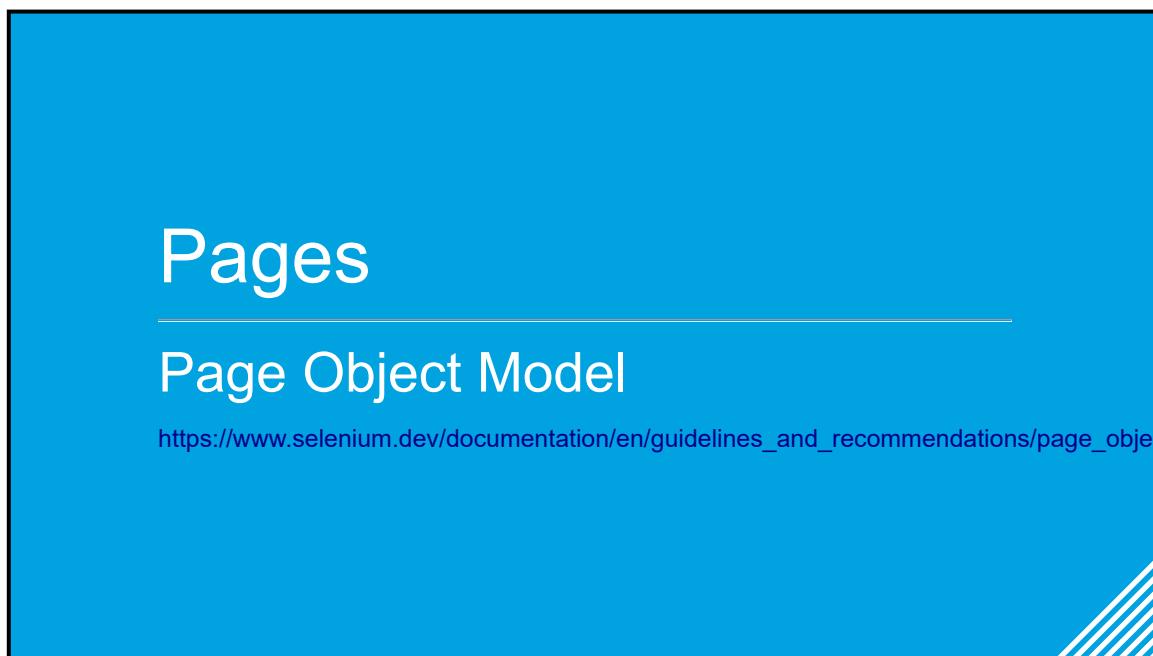
Understand the basics



**GITHUB LINK**

<https://github.com/Oliver-Cl/selenium-demo>

1. Set up your project
2. Fix the tests in the basic package



## Pages

---

### Page Object Model

[https://www.selenium.dev/documentation/en/guidelines\\_and\\_recommendations/page\\_object\\_model/](https://www.selenium.dev/documentation/en/guidelines_and_recommendations/page_object_model/)

# Why use POM?

- Encapsulate logic to pages
- Reusability of a page
- The assertions are only present in the tests
- Clean separation between test and page services

```
public class TestLogin {  
  
    @Test  
    public void testLogin() {  
        // fill login data on sign-in page  
        driver.findElement(By.name("user_name")).sendKeys("testUser");  
        driver.findElement(By.name("password")).sendKeys("my supersecret passwo  
        driver.findElement(By.name("sign-in")).click();  
  
        // verify h1 tag is "Hello userNmae" after login  
        driver.findElement(By.tagName("h1")).isDisplayed();  
        assertThat(driver.findElement(By.tagName("h1")).getText(), is("Hello us  
    }  
}
```

## SignInPage

```
public class SignInPage {  
    private final WebDriver driver;  
  
    public SignInPage(WebDriver driver) {  
        this.driver = driver;  
    }  
  
    public void signInPage(final String userName, final String password){  
        // page logic  
    }  
}
```

## Page Factory

```
public class SignInPage {  
    @FindBy(css = ".page-title")  
    private WebElement title;  
  
    public SignInPage(WebDriver driver) {  
        this.driver = driver;  
        PageFactory.initElements(driver, this);  
    }  
}
```

No more

```
public class SignInPage {  
    ...  
  
    public void signInPage(final String userName, final String password) {  
        driver.findElement(By.cssSelector(".page-title"));  
    }  
}
```

# Conditions

## Wait

```
// 1. implicit
driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

// 2. explicit
WebDriverWait wait= new WebDriverWait(driver,30);
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("myId")));

// 3. fluent
Wait<WebDriver> wait = new FluentWait<>(driver)
.withTimeout(Duration.ofSeconds(30))
.pollingEvery(Duration.ofSeconds(5))
.ignoring(NoSuchElementException.class);

WebElement foo = wait.until(driver -> driver.findElement(By.id("foo")));
```

## ExpectedConditions

```
alertIsPresent()
elementSelectionStateToBe()
elementToBeClickable()
elementToBeSelected()
frameToBeAvailableAndSwitchToIt()
invisibilityOfTheElementLocated()
invisibilityOfElementWithText()
presenceOfAllElementsLocatedBy()
presenceOfElementLocated()
textToBePresentInElement()
textToBePresentInElementLocated()
textToBePresentInElementValue()
titleIs()
titleContains()
visibilityOf()
```

Real life automation

**GITHUB LINK**

<https://github.com/Oliver-Cl/selenium-demo>

1. Try out Twitter with POM

**Advanced Topics**

# Webdriver Decorators & Interface

Used to unlock more complex functionality to the Webdriver

- Actions (Decorator)
- EventListener (Decorator)
- Screenshots (Interface)

## Actions Decorator

```
final Actions actions = new Actions(driver);
actions.moveToElement(driver.findElement(By.id("home")));
actions.moveByOffset(-10,-20);
```

Allows the use of more complex functionality such as:

- Drag/Drop
- Click&Hold
- Move Mouse

## Event Decorator

```
final EventFiringWebDriver eventFiringDriver = new EventFiringWebDriver(driver)
// Requires a WebDriverEventListener instance
eventFiringDriver.register(new MyListener());

public class MyListener extends AbstractWebDriverEventListener {
    // Override any methods you want to react on.
    @Override
    public void beforeClickOn(WebElement element, WebDriver driver) {
        // Do something with the element or driver :-)
    }
}
```

Allows you to intercept whenever an action is happening.

## Taking Screenshots

Most Webdrivers implement the TakesScreenshot-interface. This means we could cast them to that interface and use it to create a screenshot.

```
File screenshot = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);

// Read it from default location
String absolutePath = screenshot.getAbsolutePath();

// Copy it to somewhere else.
Path destination = Path.of("c:/temp/screenshot" + System.currentTimeMillis() +
Files.copy(screenshot.toPath(), destination);
```

# Browser Interaction

- Cookies
- Accessing Local Storage
- Executing Javascript

## Cookies

```
Cookie cookie = new Cookie("foo", "bar");
driver.manage().addCookie(cookie);

Cookie driverCookie = driver.manage().getCookieNamed("foo");
```

## Accessing LocalStorage/SessionStorage

```
WebStorage storage = (WebStorage) new Augmenter().augment(driver);

// Select a type of storage
LocalStorage localStorage = storage.getLocalStorage();
SessionStorage sessionStorage = storage.getSessionStorage();

// Use the storage
String item = localStorage.getItem("KEY");
localStorage.setItem("KEY", "VALUE");
```

## Execute Javascript from Selenium

```
//Creating the JavascriptExecutor interface object by Type casting
JavascriptExecutor js = (JavascriptExecutor)driver;

js.executeScript("alert('Hello from Selenium :-')");
```

# Wrap-up

## Useful links

<https://testautomationu.applitools.com/selenium-webdriver-tutorial-java/>

<https://www.selenium.dev/documentation/en/>

<https://www.baeldung.com/java-selenium-with-junit-and-testng>

<https://www.baeldung.com/selenium webdriver-page-object>

## Useful next steps

Selenide: Selenium and Assertions

## Questions?

Last chance!

