

# Programmeren om een website te laten bewegen

codeToMove

## Onze "eerste" Selenium code van de XAB!

```
ExampleFirstSeleniumTest logInSuccessFull()  
1 package examples;  
2  
3 import io.github.bonigarcia.wdm.ChromeDriverManager;  
4 import org.openqa.selenium.By;  
5 import org.openqa.selenium.WebDriver;  
6 import org.openqa.selenium.chrome.ChromeDriver;  
7 import org.testng.annotations.Test;  
8  
9 public class ExampleFirstSeleniumTest {  
10  
11     @Test  
12     public void logInSuccessFull() {  
13         ChromeDriverManager.getInstance().setup();  
14         WebDriver driver = new ChromeDriver();  
15  
16         driver.get("https://techblog.polteq.com/testshop/index.php");  
17  
18         driver.findElement(By.className("login")).click();  
19  
20         driver.findElement(By.id("email")).sendKeys("bootcamper@polteq.com");  
21         driver.findElement(By.id("SubmitLogin")).click();  
22  
23         driver.quit();  
24     }  
25 }  
26
```

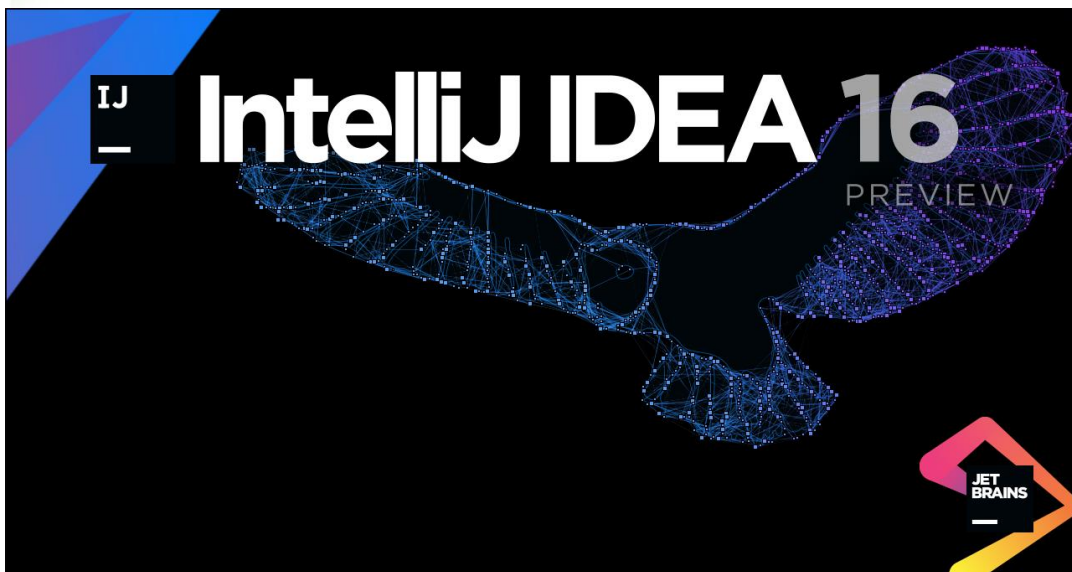
## Benodigdheden

- IDE geïnstalleerd
- Programmeertaal dependencies geïnstalleerd
  - Bijv Java SDK
  - Bijv Python 3.2 release
- Build and dependency management tool
- Test Framework
- Selenium

3



## Java IDE



4



## Java (8) zelf



<https://www.java.com/nl/download/>

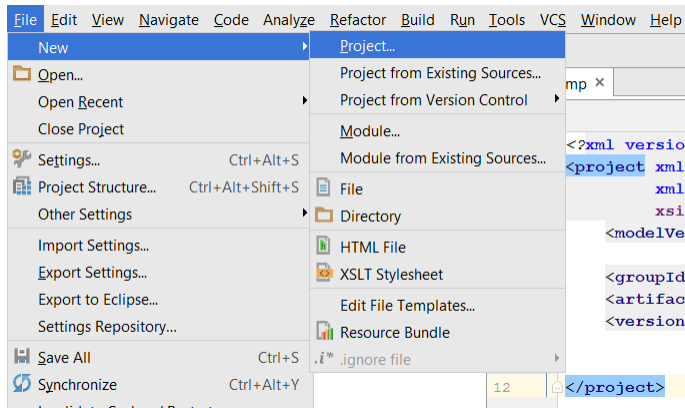


## Build and dependency management tool



## Maven project aanmaken

- Start IntelliJ op
- Vervolgens:

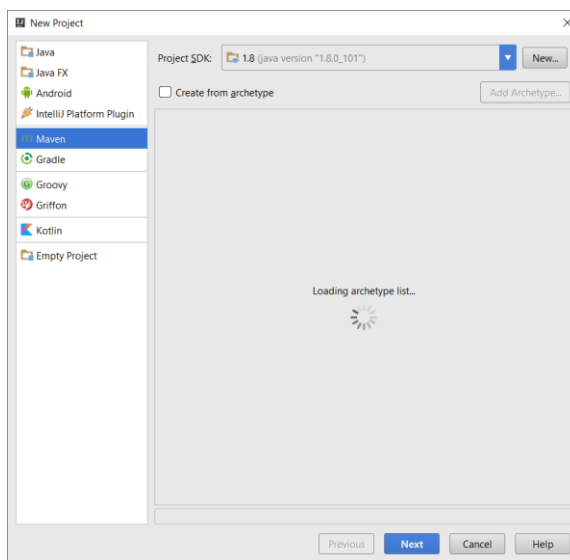


7



## Maven project aanmaken

- Selecteer Maven
- Click Next



8



## Maven project aanmaken

- Vul in de gegevens:
- Click Next

New Project

GroupId: com.polteq.training ☐ Inherit

ArtifactId: bootcamp ☐ Inherit

Version: 1.0-SNAPSHOT

Previous Next Cancel Help



## Maven project aanmaken

- Vul in de naam
- Vul in de locatie
- Click Finish

New Project

Project name: bootcamp

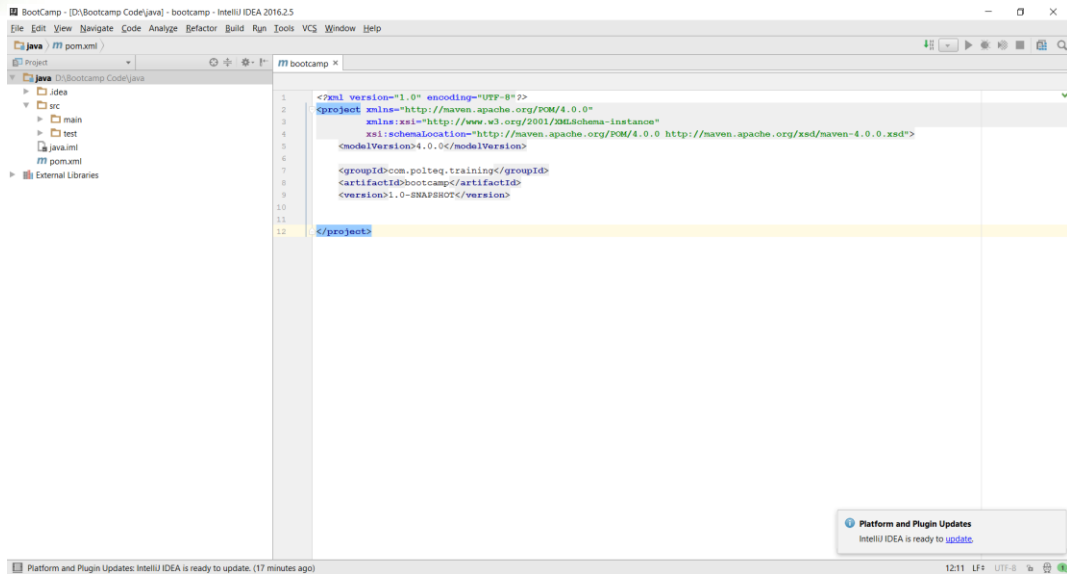
Project location: D:\BootCampCode\java

More Settings

Previous Finish Cancel Help



## Maven project aanmaken



11

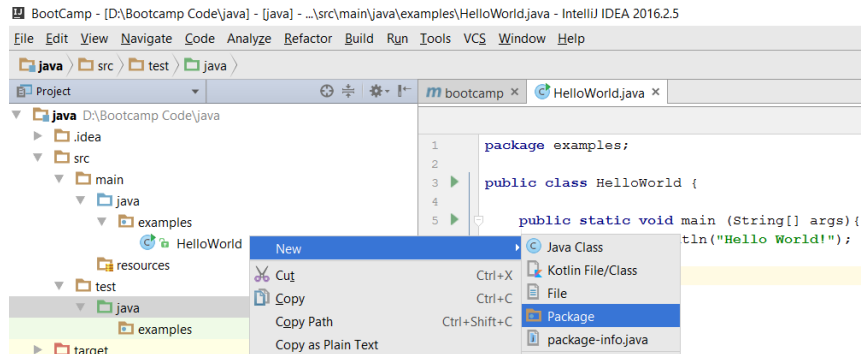
## Wat hebben we nodig om een test te runnen?

- Packages voor de overzichtelijkheid
- Class en methoden (komt nog uitgebreid aan bod 😊)
- Test framework met Annotaties

12



## Package aanmaken

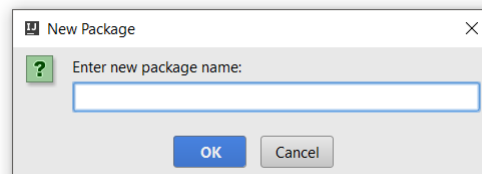


Packages zijn o.a. folders waarmee we files kunnen ordenen en verdelen

13



## Package aanmaken



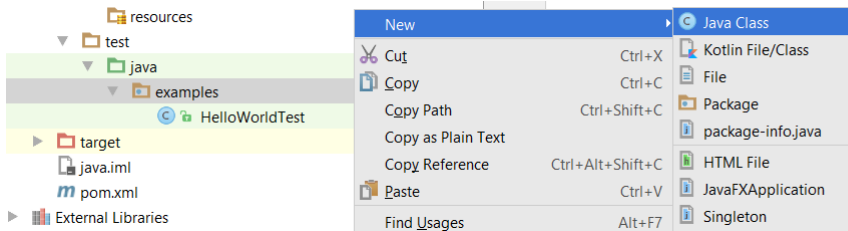
Kleine letters!

14



## Class aanmaken

- Java Class
- Click op de package, rechter muis knop, selecteer Java class bij add new

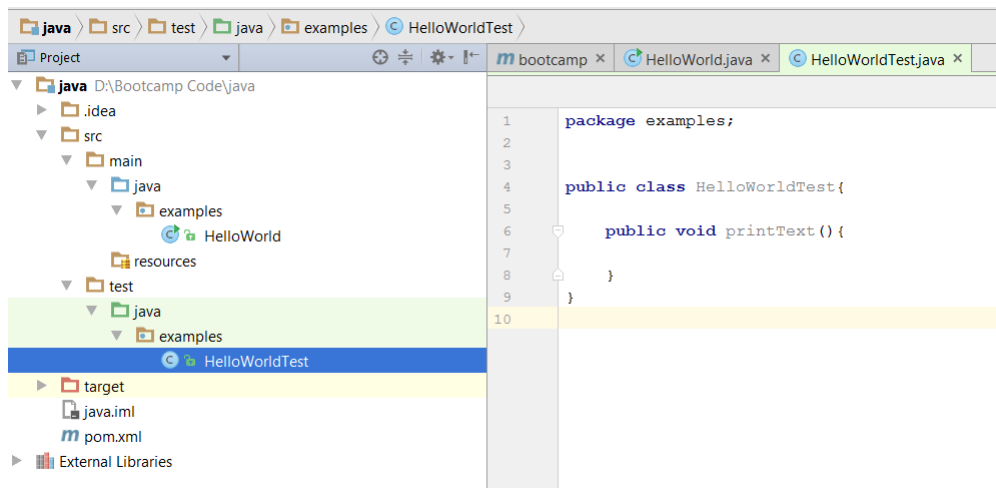


Hoofdletters en camelcasing!

15



## Methode aanmaken

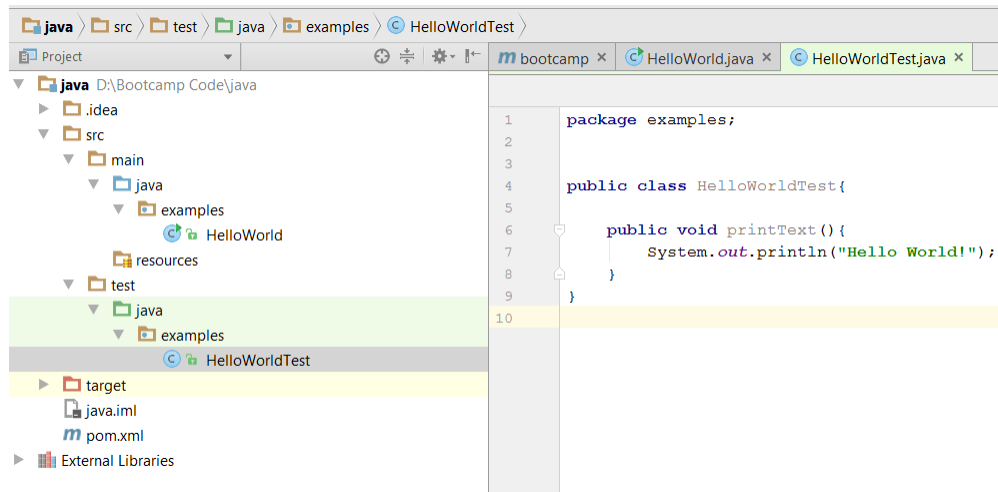


16





## Methode aanmaken



17



## Projectstructuur

- Project bevat source folders (zoals test en java)
- Source folders bevatten packages
- Packages bevatten gerelateerde classes
- Classes bevatten methoden en **variabelen** <- hierover strak zeker meer!

18



## Test framework?

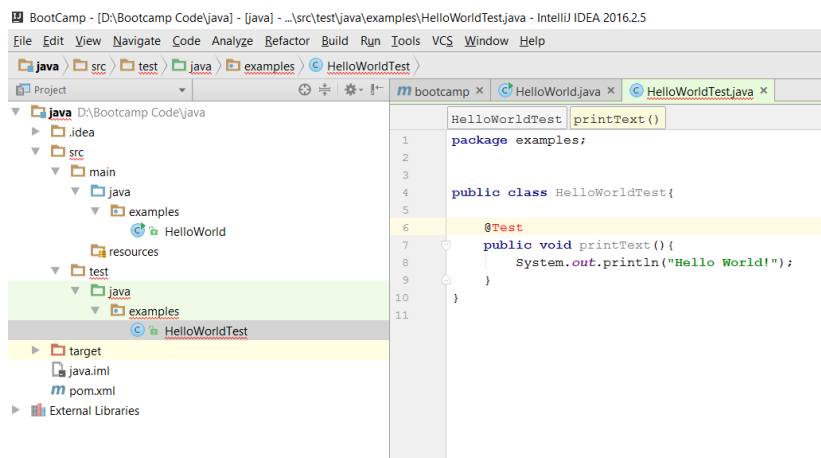
- Annotatie gebruik zoals @Test
- Annotaties voor volgorde zoals @BeforeMethod
- Rapportage



19



## TestNG toevoegen



Voeg toe @Test en zie hoe alles rood wordt

20



## TestNG toevoegen

```

1 package examples;
2
3
4
5
6
7
8
9
10
11

```

? org.testng.annotations.Test? Alt+Enter

```

6 @Test
7 public void printText(){
8     System.out.println("Hello World!");
9 }

```

Of click op het rode lampje en selecteer testng to classpath

21



## TestNG toevoegen

```

HelloWorldTest printText()
1 package examples;
2
3
4 import org.testng.annotations.Test;
5
6 public class HelloWorldTest{
7
8     @Test
9     public void printText(){
10         System.out.println("Hello World!");
11     }
12 }
13

```

Import is toegevoegd en we hebben nu play buttons!

22



## TestNG rapportage

- TestNG Rapport
  - {project} → test-output → emailable-report.html
  - {project} → test-output → index.html
  - Rechtermuisknop → Open With → Web Browser

Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Command line suite						
Command line test	1	0	0	22		

Class	Method	Start	Time (ms)
Command line suite			
Command line test — passed			
oefeningen.MyTest	myFirstTest	1438915372982	11

### Command line test

oefeningen.MyTest#myFirstTest

Messages
Tester

```
public class MyFirstTest{

    @Test
    public void myFirstTest() {
        Reporter.Log("Hello World");
    }

}
```

23



Toch wel belangrijk om goed te weten!

# WE HADDEN HET NET OVER METHODEN & VARIABLEN

24



## Variabelen en datatypes

Bouwstenen van programmeren, stukje Java voor Testers!

## Variabelen en datatypes

- Variabele
  - Verwijzing naar een waarde. Deze waarde is aan te passen zonder de verwijzing zelf aan te passen. (leeft dus centraal).
- Datatype
  - Het formaat van een variabele, bijv: tekst, heel getal, decimal getal, enz
- [Datatype] [naam variabele] = [waarde];
  - `int myVariable = 1;`
- [Datatype] [naam van de variabele];
  - `int myVariableWithoutValue;`

Reserveren een plek in het geheugen met default waarde

## Naamgevingsconventie

- Een naam van een variabele begint met een kleine letter. Elk nieuw woord in die naam begint met een hoofdletter om de leesbaarheid te vergroten.
  - `String welcomeMessage;`
  - `int numberOfProducts;`

27



## Datatypes

- Twee hoofdcategorieën
  - Primitive type
  - Reference type
- Primitive type
  - `double percentage = 3.1;`
- Reference type (hier over later meer)
  - `Calculator calculator = new Calculator();`

28



## Primitive types

- Dit zijn alle primitive types en de waarde als deze niet expliciet is toegekend

Datatype	Standaard waarde
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
String	null
boolean	false

Technisch gezien geen primitive, maar een reference type. Een soort hybride.

29



## Integer

- Hele getallen (32 bit)
  - Minimale waarde  $-2.147.483.648$  ( $-2^{31}$ )
  - Maximale waarde  $2.147.483.647$  ( $2^{31}-1$ )
- Nuttig voor checks op aantallen, bijvoorbeeld:
  - Aantal zoekresultaten
  - Aantal producten in winkelmandje
- `int numberOfProducts = 5;`

30



## Double

- Decimale getallen (64 bit)
- Nuttig voor de meeste\* van onze berekeningen en checks bij het testen, bijvoorbeeld:
  - Percentage van premies berekenen
  - Kortingspercentage of BTW toepassen
- `double discountPercentage = 10.15D;`
  - De D is optioneel dus het onderstaande volstaat
- `double discountPercentage = 10.15;`

31



## Boolean

- De (on)waarheid van een conditie
  - `true` (waar)
  - `false` (onwaar)
- Nuttig voor condities, bijvoorbeeld:
  - Leeftijd is hoger dan of gelijk aan 21
  - Gebruiker is ingelogd
- `boolean isLoggedIn = false;`
- `boolean isLoggedIn = true;`

32





## String

- Tekst in UTF-16 format
- Nuttig voor bewerkingen met tekst, bijvoorbeeld:
  - Gebruikersnaam en wachtwoord invullen
  - Teksten van meldingen extraheren / controleren
- `String welcomeMessage = "Welcome to our new online store!";`

33



## Methoden

Logisch code ordenen

## Nut van methoden

- Een resultaat bereiken dat door één of meerdere statements tot stand komt, met als voordelen:
  - Leesbaarheid
  - Onderhoudbaarheid
- Voorbeelden van methoden in de praktijk:
  - het aantal producten tellen in een shopping cart;
  - inloggen;
  - berekeningen doen;
  - ...

35



## Methoden

- Een methode:
  - bevat een of meerdere statements;
  - kan parameters hebben;
  - kan iets retourneren;
  - kan andere methode(n) aanroepen;
  - kan geen definities van methoden of classes bevatten.

```
private void someMethod() {
    private void calculatePrice() {
    }
    public class SomeClass {
    }
}
```

36



## Methoden

- Naamgevingsconventie
  - Een naam van een methode begint met een kleine letter. Elk nieuw woord in die naam begint met een hoofdletter om de leesbaarheid te vergroten.

```
private void showPrice() {
    System.out.println("€50");
    System.out.println(greeting());
}
```

```
private String greeting() {
    return "Hi";
}
```

37



## Methode zonder return waarde

- Gekenmerkt door het keyword **void**
  - voert statement(s) uit

```
private void printBeatles() {
    System.out.println("Ringo Starr");
    System.out.println("Paul McCartney");
    System.out.println("George Harrison");
    System.out.println("John Lennon");
}
```

- Vanuit een @Test methode aanroepen  
`printBeatles();`

38



## Methode met return waarde

- Gekenmerkt door het datatype en het keyword **return** met daarachter een waarde van dat datatype
  - voert eventueel statement(s) uit en geeft een waarde terug

```
private String theUltimateAnswer() {
    return "The answer to life, the universe and
    everything is 42";
}
```

- Vanuit een @Test methode aanroepen

```
System.out.println(theUltimateAnswer());
```

39



## Methode met parameters

- Een of meerdere parameters tussen haakjes

```
private String fullName(String
    firstname, String lastName) {
    return firstname + " " + lastName;
}
```

- Vanuit een @Test methode aanroepen

```
System.out.println(fullName("Max", "Verstappen"));
```

40



## Wat wij gaan doen met een methode

- Een berekening schrijven die twee willekeurige getallen kan vermenigvuldigen
- Hoe kun je in Java rekenen?

41



## Rekenkundige operators

- Optellen: +
  - Ook voor het concateneren van tekst

```
String a = "d" + "e";
```
- Aftrekken: -
- Vermenigvuldigen: \*
- Delen: /
- Modulo: %

42



## Voorbeeld gebruik operators

- `int sum = 2 + 1;`
- `double quotient = a / b;`
  - (double a = 3.6 , double b = 1.7 → quotient 2.1176470588235294)

43



## Opdracht met gebruik van methoden

- Doel: leren werken met methoden, return waarden, parameters en rekenkundige operators
- Maak een nieuwe TestNG class **AboutMethods**
  - Noem de @Test methode **printProduct**
- Maak een private methode **multiply** die twee parameters (getallen) heeft en het product van deze twee getallen retourneert
- Roep **multiply** vanuit de **printProduct** aan met twee argumenten en print de uitkomst naar de console

44



## The road so far...

- IntelliJ is up and running
- Maven project aangemaakt (voor onder andere het makkelijker binnen halen van dependencies, gaan we zo doen!)
- Maken nu gebruik van TestNg om tests te runnen
- Variabelen en datatypes besproken
- Methoden zonder return waarden, met parameters en return waarden geoefend

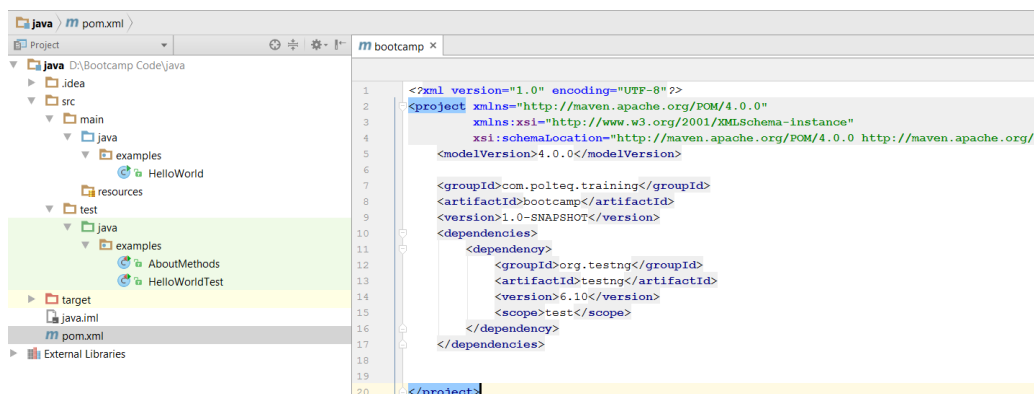
En dan nu Selenium!

45



## Selenium "binnen halen"

- De pom.xml



46



## Selenium “binnen halen”

- De pom.xml
- Typ tussen `</version>` en `</project>` het onderstaande
 

```
<dependencies>

</dependencies>
```
- Ga naar <http://mvnrepository.com>
- Zoek **selenium-java** en klik op het eerste resultaat
- Klik op het meest recente versienummer
- Kopieer het `<dependency>...</dependency>` gedeelte en plak het tussen `<dependencies>` `</dependencies>`

47



## Selenium binnenhalen gelukt?

- IntelliJ gaat Selenium binnen halen:

```
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>3.4.0</version>
</dependency>
```

Indexing...

- Daarna is POM niet meer rood:


```
<dependency>
  <groupId>org.seleniumhq.selenium</groupId>
  <artifactId>selenium-java</artifactId>
  <version>3.4.0</version>
</dependency>
```

48





## WebDriver code: Browser starten



```
@Test
public void logInSuccesFull() {
    ChromeDriverManager.getInstance().setup();
    WebDriver driver = new ChromeDriver();

    driver.get("https://techblog.polteq.com/testshop/index.php");
}
```

- Starten Chrome op door een object te maken van de WebDriver class (hierover later meer)
- Navigeren naar de URL met behulp van .get()
- Belangrijk: ChromeDriverManager!

49



## ChromeDriverManager

- Ga naar <http://mvnrepository.com>
- Zoek op webdrivermanager
- Voeg de code snippet toe aan de pom.xml

50



## WebDriver code: Elementen benaderen

```
driver.findElement(By.className("login")).click();

driver.findElement(By.id("email")).sendKeys("bootcamper@polteq.com");
driver.findElement(By.id("SubmitLogin")).click();
```

- Benaderen een DOM element met By.xxxx

51



## Wat is de DOM?

- Code van de website die wij niet zien in een normale browser..
- ..maar er wel degelijk is!

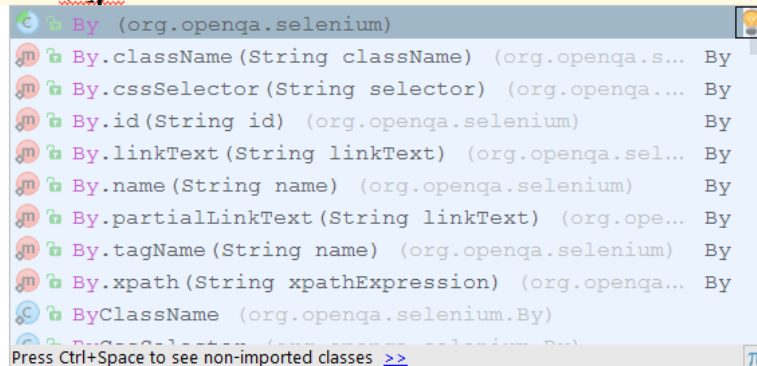
```
<input id="inSlowConfig" type="hidden" value="false">
<script type="text/javascript">
<div id="header" class="global-nav guest">
<hr>
<div id="body">
  <div class="wrapper">
    <div id="global-error"> </div>
    <div id="main">
      <div id="content">
        <div>Over 100 million professionals use LinkedIn to exchange inf
          <ul>
            <li id="reconnect">Stay informed about your contacts and inc
            <li id="answers">Find the people [camp] knowledge you need t
            <li id="power">Control your professional identity online</li>
          </ul>
        </div>
      </div>
      <div id="module-id3" class="leo-module mod-feat jointoday">
        <div class="header">
          <h2>Join LinkedIn Today</h2>
        </div>
        <div class="content">
          <form id="yul-gen2" class="feature" name="coldRegistrationFo
            <fieldset>
              <input id="webmailImport-coldRegistrationForm" type="hid
              <input id="csrfToken-coldRegistrationForm" type="hidden"
              <input id="sourceAlias-coldRegistrationForm" type="hidden"
            </form>
          <div id="control-3" class="li-control" type="linkedin/con
            1
            2
```

52



## WebDriver code: Elementen benaderen

```
driver.findElement(By)
```

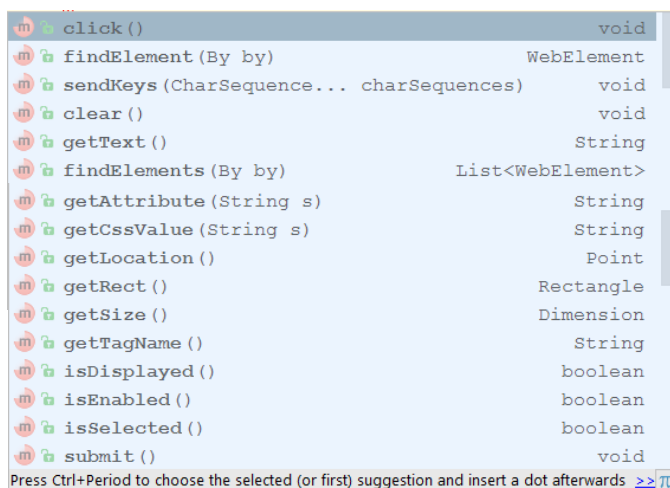


- Gebruik je IDE vooral ook omdat elke taal net iets anders kan zijn wat betreft syntax!

53













## WebDriver code: Elementen gebruiken



54



## WebDriver code: Elementen gebruiken

	<code>click()</code>	<code>void</code>
	<code>findElement(By by)</code>	<code>WebElement</code>
	<code>sendKeys(CharSequence... charSequences)</code>	<code>void</code>
	<code>clear()</code>	<code>void</code>
	<code>getText()</code>	<code>String</code>
	<code>findElements(By by)</code>	<code>List&lt;WebElement&gt;</code>
	<code>getAttribute(String s)</code>	<code>String</code>
	<code>getCssValue(String s)</code>	<code>String</code>
	<code>getLocation()</code>	<code>Point</code>
	<code>getRect()</code>	<code>Rectangle</code>

```
driver.findElement(By.className("login")).click();
```

```
driver.findElement(By.id("email")).sendKeys("bootcamper@polteq.com");
driver.findElement(By.id("SubmitLogin")).click();
```

55



## Tools om locaters te vinden

- Firefox
  - Firebug (<http://getfirebug.com>) -> Deprecated ☹ [or is it...](#)
  - FirePath (Firefox add-on)
- Internet explorer
  - F12
- Edge
  - F12
- Chrome
  - F12
  - SelectorGadget
- Opera
  - <http://dev.opera.com/articles/view/opera-developer-tools/>
- Safari
  - Preferences -> Advanced -> Show Develop menu in menu bar

56



## Locator?

Een manier om Selenium te laten weten met welk element we *iets* willen doen

### Verschillende manieren:

Xpath, CSS, id, name, link, text

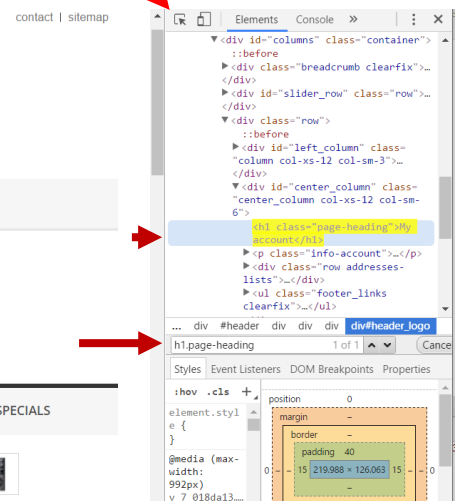
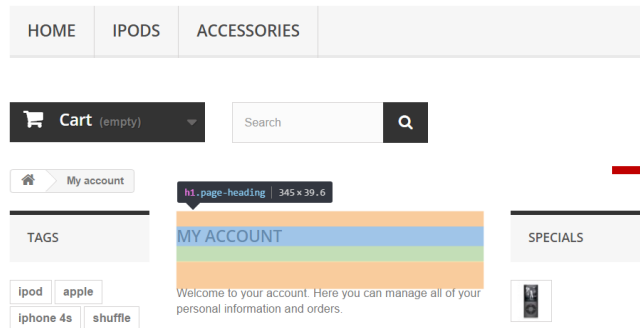
57



## Chrome

- Druk op F12

- Druk Ctrl+F



58



## Op basis van attribuut – deze gaan we veel gebruiken!!

- Voorbeeld:
- `<input type="text" class="firstField" name="first" id="firstTextfield">`
- `input#firstTextfield`
- `input[id='firstTextfield']`
- `input.firstField`
- `input[class='firstField']`
- `input[name='first']`

59



## Op basis van tekst

- Voorbeeld:
- `<a href="#">This link does not have an identifier</a>`
- `//a[text()='This link does not have an identifier']`
- `//*[text()='This link does not have an identifier']`
- `//a[contains(text(), 'This link does not')]`

Alleen Xpath

60



## Browser sluiten

```
}  
    driver.quit();  
}
```

- Kan heel belangrijk zijn. Waarom denken jullie?

61



## Opdracht: Maak een Selenium Testcase om in te loggen

- Maak een nieuwe TestNG class genaamd FirstSeleniumTest
- Noem de @Test methode **logInSuccesFull**
- Gebruik vervolgens in de bovengenoemde methode Selenium voor het succesvol inloggen op de Polteq webshop.

62





Testen is....?

**MAAR WAT ZIJN WIJ VERGETEN?**

63



**Betrouwbare validatie!**

Groen vinkje in de IDE is echt niet genoeg



## Daarvoor gebruiken we Assertions

- Boolean values
- Integer values
- String values
- ...

65



## Assertions – boolean

```
import org.assertj.core.api.Assertions;
```

```
@Test
public void assertBoolean() {
    boolean testBoolean = true;

    Assertions.assertThat(testBoolean).as("Selection box is
        disabled").assertFalse();
}
```

Actuele waarde

Verwachte uitkomst

Melding indien check  
faalt

[Selection box is disabled]  
Expecting:  
<true>  
to be equal to:  
<false>  
but was not.

66



## Assertions – String Contains

```
@Test
public void assertStringContains() {
    String text = "ThisStringStartsWithSomeTextAndEndWithSomeOtherText";

    Assertions.assertThat(text).as("Username in header").contains("tekst");
}
```

[Username in header]

Expecting:

<"ThisStringStartsWithSomeTextAndEndWithSomeOtherText">

to contain:

<"tekst">

67



## Assertions – Integer greaterThan

```
@Test
public void assertIntegerGreaterThan() {
    int testInt = 42;
    Assertions.assertThat(testInt).as("Number of
        participants").isGreaterThan(42);
}

@Test
public void assertIntegerGreaterThanOrEqualTo() {
    int testInt = 42;

    Assertions.assertThat(testInt).as("Number of
        participants").isGreaterThanOrEqualTo(42);
}
```

[Number of participants]

Expecting:

<42>

to be greater than:

<42>

68



## AssertJ

- Ga naar <http://mvnrepository.com>
- Zoek op AssertJ
- Voeg de code snippet toe aan de pom.xml

69



## Oefening

- Maak in je **examples** package een TestNG class **Controles**
- Pas een aantal assertions toe voor het valideren van:
  - Teksten
  - Getallen
  - Boolean waarden
- Maar gebruik een Assert type die niet op de sheets staan! (gebruik je IDE voor de mogelijkheden)

70



## Soft Assertions

```
public class SomeTest {

    @Test
    public void multipleChecks() {
        SoftAssertions softly = new SoftAssertions();

        softly.assertThat("enddesignstart").as("Welcome message")
            .startsWith("start")
            .endsWith("end")
            .contains("test");

        softly.assertAll();
    }
}
```

71



## Soft Assertions

The following 3 assertions failed:

1) [Welcome message]

Expecting:

<"enddesignstart">

to start with:

<"start">

at SomeTest.multipleChecks(SomeTest.java:14)

2) [Welcome message]

Expecting:

<"enddesignstart">

to end with:

<"end">

at SomeTest.multipleChecks(SomeTest.java:15)

3) [Welcome message]

Expecting:

<"enddesignstart">

to contain:

<"test">

72 at SomeTest.multipleChecks(SomeTest.java:16)



## Opdracht: Voeg een Assert toe aan de Selenium Testcase om in te loggen

- Gebruik de TestNG class genaamd FirstSeleniumTest
- Analyseer welk element je wilt gebruiken voor validatie aan het einde van de testcase (tip: dat is een element op de My Account pagina)
- Benader dit element met Selenium code, store het element in een variable
- Maak een Assert met de bovengenoemde variable om te controleren of de test de inlog pagina heeft bereikt.

73



# Vragen?