# Lets TDD in GO

2017/10 GoBxl

# Thanks to our Host

**Le Wagon Brussels**

Change your life: learn to code

**https://www.lewagon.com**

**@LeWagonBrussels**

# Objectives

- Fun with **GO** :)
- Introduction to TDD (short)
- Try TDD and apply it to **GO**

# TDD principles

- Write tests BEFORE the code

- Automate Tests that run quickly and often

# TDD Advantages

- You are your first user
- Your code becomes the documentation
- It protects from regression (Changing Tests is NOT Refactoring)

# TDD Workflow

- Write the test
- Make it fails (Why?)
- You write the **smallest** amount of code to make it pass
- Refactor

# TDD Tooling for GO

- go test….
  - -v for verbose
  - -timeout 2s
  - -short ,testing.Short(), t.Skip()
  - t.Parallel()

# Kata and Fun

- A **code kata** is an exercise in **programming** which helps programmers hone their skills through practice and repetition. The **term** was probably first coined by Dave Thomas, co-author of the book The Pragmatic Programmer, in a bow to the Japanese concept of **kata** in the martial arts.

**https://en.wikipedia.org/wiki/Kata_(programming)**

# Kata 1 : String Calculator

By Roy Osherove

- Rules
- Try not to read ahead
- Do one thing at the time. Learn incrementally…
- Test for correct inputs. No error management here but…

- Create a simple string calculator that can take 0, 1,2 or any numbers and return their sums

    **Add(numbers string) (int,error)**

    - “”,”1”,”x”,”1,2”, “x,y’
    - Support any number
    - Support new line “\n” instead of “,”
    - Support any separator: should be specified at beginning **“//;\n1;2”**
        - The separator  declaration is optional all test should still pass
    - Calling Add with <0 should return error “No negative numbers”
    - Numbers bigger than 1000 should be ignored
    - Delimiters can be of any length **“//[***]\n1***2”**
    - Handle multiple delimiters **“//[;][|]\n1|2;3”**
    - Handle multiple delimiters of any length