

A Tour of Go

Workshop @ Golang Brno meetup #4
24 January 2017

Rodolfo Carvalho
Red Hat

A Tour of Go workshop

Objectives:

- Fun & enjoyment
- First contact with Go
- Close gaps
- Introduction to automated testing
- Networking

Note: the name of the language is **Go**, the website is golang.org

Introduction

Let's get to know each other:

- Name
- Occupation
- Programming experience (including or not Go)

Agenda

- A Tour of Go
- The go tool
- The testing package
- Coding Dojo
- Retrospective

Rules

- Ask questions
- Bring questions to the group
- Play with examples (run, modify, run, ...)

And...

- Work in pairs
- Pair with someone you met for the first time
- Alternate who controls the keyboard

A Tour of Go

- Online

tour.golang.org

- Offline

```
go tool tour
```

- Keyboard shortcuts

```
PageUp / PageDown : previous / next  
Ctrl+Enter       : format  
Shift+Enter      : run
```

- Skip the exercises, we'll go back to them later
- We will pause after the "Basics"

The go tool

- a tool for managing Go source code

```
$ go
```

```
...
```

The commands are:

build	compile packages and dependencies
doc	show documentation for package or symbol
fmt	run gofmt on package sources
run	compile and run Go program
test	test packages
...	

go test

- automates testing packages
- recompiles each package along with any files with names matching the pattern `"*_test.go"`
- `go test` with no arguments compiles and tests the package with source in the current directory

The testing package

```
$ go doc testing
```

```
... provides support for automated testing of Go packages. (...) used in  
concert with the ``go test'' command, which automates execution of any  
function of the form
```

```
func TestXxx(*testing.T)
```

```
...
```

Some methods of testing.T

```
$ go doc testing.T
```

```
func (c *T) Fail()  
func (c *T) FailNow()
```

```
func (c *T) Log(args ...interface{})  
func (c *T) Logf(format string, args ...interface{})
```

```
func (c *T) Error(args ...interface{})  
func (c *T) Errorf(format string, args ...interface{})
```

```
func (c *T) Fatal(args ...interface{})  
func (c *T) Fatalf(format string, args ...interface{})
```

A minimal test file template

foo_test.go

```
package foo

import "testing"

func TestFoo(t *testing.T) {
    got := Foo()
    want := "bar"
    if got != want {
        t.Errorf("Foo() = %v, want %v", got, want)
    }
}
```

Table testing

github.com/golang/go/wiki/TableDrivenTests

```
func TestSame(t *testing.T) {
    tests := []struct {
        trees [2]*tree.Tree
        want  bool
    }{
        {
            trees: [2]*tree.Tree{tree.New(1), tree.New(1)},
            want:  true,
        },
        {
            trees: [2]*tree.Tree{tree.New(1), tree.New(2)},
            want:  false,
        },
    }
    for _, tt := range tests {
        if got := Same(tt.trees[0], tt.trees[1]); got != tt.want {
            t.Errorf("Same(%v, %v) = %v, want %v", tt.trees[0], tt.trees[1], got, tt.want)
        }
    }
}
```

Run

Coding Dojo

- Change pairs
- Work on exercises using TDD
- Discuss different implementations with your partner

TDD rules:

- Write a complete test case
- See it FAIL
- Make it PASS
- Refactor

Exercises

Basics:

- [Loops and Functions: Sqrt](#)
- [Slices: Pic](#)
- [Maps: WordCount](#)
- [Fibonacci closure](#)

What if I don't have Go installed?

You can code online using [The Go Playground](#). Running tests there can be accomplished with a little trick, as in this example:

play.golang.org/p/UhY6SqAdCu

In the future, it may become easier to run tests in the playground:

github.com/golang/go/issues/6511

Exercises

Methods and interfaces:

- [Stringers: IPAddr](#)
- [Errors: Sqrt](#)
- [Readers: infinite stream of 'A'](#)
- [rot13Reader](#)
- [Images](#)

Exercises

Concurrency:

- [Equivalent Binary Trees](#)
- [Web Crawler](#)

Coding Dojo Brno

If you want to practice more TDD, we have weekly meetings:

facebook.com/CodingDojoBrno



Retrospective

Code retreat style...

- What, if anything, did you learn today?
- What, if anything, surprised you today?
- What, if anything, would you like to do with Go in the future?

Thank you

Rodolfo Carvalho

Red Hat

rhcarvalho@gmail.com

<http://rodolfocarvalho.net>

<https://dojo-brno.github.io>