

Go course

Petr Shevtsov

2019-05-08

Contents

Preface	5
1 Basic Concepts	7
1.1 What is Go?	7
1.2 Hello, world!	7
1.3 The Go CLI	7
1.4 Value types	7
1.5 Variables	7
1.6 Operators	7
1.7 Constants	7
1.8 Comments	7
1.9 Packages and imports	7
Module project	7
2 Conditionals and Loops	9
2.1 The <code>if</code> statement	9
2.2 The <code>else</code> statement	9
2.3 <code>if/else</code> chains	9
2.4 The <code>if</code> statement with expression	9
2.5 The <code>switch</code> statement	9
2.6 The <code>switch</code> without condition	9
2.7 The <code>for</code> statement	9
2.8 The <code>defer</code> statement	9
3 Composite Data Types	11
3.1 Arrays	11
3.2 Loops and arrays	11
3.3 Slices	11
3.4 Appending items to slices	11
3.5 Range	11
3.6 Maps	11
3.7 Arrays vs maps	11
3.8 Structs	11
3.9 Struct literals	11
3.10 Operations with structs	11
4 Functions and pointers	13
4.1 Function declaration	13
4.2 Functions parameters	13
4.3 Return values	13
4.4 Error handling	13
4.5 Variadic functions	13

4.6	Iteration and recursion	13
4.7	Anonymous functions	13
4.8	Panic	13
4.9	Pointers	13
4.10	Functions and pointers	13
5	Methods	15
5.1	Method declarations	15
5.2	Methods with a pointer receiver	15
5.3	Composing types with structs	15
5.4	Working with struct methods	15
5.5	Method values	15
5.6	Method expressions	15
5.7	Encapsulation	15
6	Interfaces	17
6.1	Introduction	17
6.2	Interface types	17
6.3	Satisfaction	17
6.4	<code>flag.Value</code>	17
6.5	Interface values	17
6.6	Sorting with <code>sort.Interface</code>	17
6.7	The <code>error</code> interface	17
6.8	Type assertions	17
6.9	Type switches	17
7	Goroutines and channels	19
7.1	What is goroutine	19
7.2	Introduction to concurrency	19
7.3	Channels	19
7.4	Types of channels	19
7.5	Pipelines	19
7.6	Looping in parallel	19
7.7	<code>time.Tick</code>	19
7.8	The <code>select</code> statement	19
7.9	Cancellation	19

Preface

The course is targeted at a beginner level student new to Go but might be familiar with 1-2 other languages (e.g. Python, HTML).

Module 1

Basic Concepts

1.1 What is Go?

1.2 Hello, world!

1.3 The Go CLI

1.4 Value types

1.5 Variables

1.6 Operators

1.7 Constants

1.8 Comments

1.9 Packages and imports

Module project

```
// This is a comment.  
// Every Go file must be a part of some package.  
// This file is a part of package main.  
package main  
  
// We import package "fmt" from the standard Go library.  
import "fmt"  
  
// `who` is a constant.
```

```
const who = "world"

// Function main() is the main entry point of any application written in Go.
func main() {
    // We declare `greeting` as a variable of type string and assign the value.
    var greeting string = "Hello"
    // We declare `message` variable using the shorthand syntax. The type of
    // the variable is determined by the assigned value. In our case it is
    // string type.
    message := greeting + ", " + who
    // Let's print the value of the variable `message` using the function from
    // the package "fmt".
    fmt.Println(message)
}

## Hello, world
```


Module 2

Conditionals and Loops

2.1 The `if` statement

2.2 The `else` statement

2.3 `if/else` chains

2.4 The `if` statement with expression

2.5 The `switch` statement

2.6 The `switch` without condition

2.7 The `for` statement

2.8 The `defer` statement

Module 3

Composite Data Types

3.1 Arrays

3.2 Loops and arrays

3.3 Slices

3.4 Appending items to slices

3.5 Range

3.6 Maps

3.7 Arrays vs maps

3.8 Structs

3.9 Struct literals

3.10 Operations with structs

Module 4

Functions and pointers

4.1 Function declaration

4.2 Functions parameters

4.3 Return values

4.4 Error handling

4.5 Variadic functions

4.6 Iteration and recursion

4.7 Anonymous functions

4.8 Panic

4.9 Pointers

4.10 Functions and pointers

Module 5

Methods

- 5.1 Method declarations
- 5.2 Methods with a pointer receiver
- 5.3 Composing types with structs
- 5.4 Working with struct methods
- 5.5 Method values
- 5.6 Method expressions
- 5.7 Encapsulation

Module 6

Interfaces

6.1 Introduction

6.2 Interface types

6.3 Satisfaction

6.4 `flag.Value`

6.5 Interface values

6.6 Sorting with `sort.Interface`

6.7 The `error` interface

6.8 Type assertions

6.9 Type switches

Module 7

Goroutines and channels

7.1 What is goroutine

7.2 Introduction to concurrency

7.3 Channels

7.4 Types of channels

7.5 Pipelines

7.6 Looping in parallel

7.7 `time.Tick`

7.8 The `select` statement

7.9 Cancellation