



# GO

## FUNDAMENTALS

MARK BATES

CORY LANOU

GOPHER GUIDES



Foreword by STEVE FRANCIA



# Go Fundamentals

---

# Go Fundamentals: Gopher Guides

## Table of Contents

Cover

Half Title

Title Page

Copyright Page

Contents

Foreword

Preface

Acknowledgments

About the Authors

1 Modules, Packages, and Dependencies

    Modules

        Toolchain

        Initializing a Module

        VCS and Modules

    Packages

        Naming Packages

    Folders, Files, and Organization

        Multiple Packages in a Folder

        File Names

        Package Organization

    Importing Packages and Modules

        Import Path

        Using Imports

# Table of Contents

Resolving Import Name Clashes

## Dependencies

Using a Dependency

Requiring with Go Get

The Go Sum File

Updating Dependencies

Semantic Versioning

Multiple Versions

Cyclical Imports

## Summary

## 2 Go Language Basics

### Go Language Overview

Static Typing

Garbage Collection

Compilation

Keywords, Operators, and Delimiters

### Numbers

Picking the Correct Numeric Type

Overflow versus Wraparound

Saturation

### Strings

Interpreted String Literals

Raw String Literals

### UTF-8

Runes

Iterating over UTF-8 Characters

### Variables

Variable Declaration

Variable Assignment

# Table of Contents

Zero Values

Nil

Zero Values Cheat Sheet

Variable Declaration and Initialization

Assigning Multiple Values

Unused Variables

## Constants

Typed Constants

Untyped Constants (Inferred Typing)

Type Inference

## Naming Identifiers

Naming Style

Conflicting with Package Names

Exporting through Capitalization

## Printing and Formatting

Formatting Functions

New Lines

Multiple Arguments with Println

Using Formatting Functions

Escape Sequences

Formatting Strings

Formatting Integers

Formatting Floats

Printing a Values Type

Printing a Value

Printing a Value with More Detail

Printing a Values Go-Syntax Representation

Using Formatting Verbs Incorrectly

Explicit Argument Indexes

# Table of Contents

Converting Strings to and from Numbers

Summary

## 3 Arrays, Slices, and Iteration

List Types: Arrays and Slices

Differences between Arrays and Slices

Spotting the Difference

Initializing Arrays and Slices

Array and Slice Zero Values

Indexing Arrays and Slices

Array and Slice Types

Array and Slice Type Definitions

Setting Array and Slice Values

Appending to Slices

Appending Slices to Slices

How Slices Work

Length and Capacity

Growing a Slice

Making a Slice

Make with Length and Capacity

Make and Append

What Happens When a Slice Grows

Slice Subsets

Mutating Slice Subsets

Copying Slices

Converting an Array to a Slice

Iteration

The for Loop

Iterating over Arrays and Slices

The range Keyword

# Table of Contents

Controlling Loops

Do While Loop

Summary

## 4 Maps and Control Structures

Maps

Length and Capacity

Initializing Maps

Uninitialized Maps

Map Keys

Structs as Keys

Iterating Maps

Deleting Keys from a Map

Nonexistent Map Keys

Checking Map Key Existence

Exploiting Zero Value

Testing Presence Only

Maps and Complex Values

Copy on Insert

Listing Keys in a Map

If Statements

The else Statement

The else if Statement

Assignment Scope

Logic and Math Operators

Switch Statements

Default

Fallthrough

Summary

## 5 Functions

# Table of Contents

## Function Definitions

- Arguments
- Arguments of the Same Type
- Return Arguments
- Multiple Return Arguments
- Named Returns
- First-Class Functions
- Functions as Arguments
- Closures
- Anonymous Functions
- Functions Accepting Arguments from Functions

## Variadic Arguments

- Variadic Argument Position
- Expanding Slices
- When to Use a Variadic Argument

## Deferring Function Calls

- Deferring with Multiple Returns
- Defer Call Order
- Deferred Calls and Panic
- Defers and Exit/Fatal
- Defer and Anonymous Functions
- Defer and Scope

## Init

- Multiple Init Statements
- Init Order
- Using init Functions for Side Effects

## Summary

## 6 Structs, Methods, and Pointers

### Structs



# Table of Contents

Declaring New Types in Go

Defining a Struct

Initializing a Struct

Initialization without Field Names

Accessing Struct Fields

Struct Tags

Struct Tags for Encoding

Using Struct Tags

## Methods

Differences Between Methods and Functions

Methods Expressions

Methods on Third-Party Types

Functions as Types

Methods on Functions

No Inheritance

## Pointers

Pass by Value

Receiving a Pointer

Getting a Pointer

Passing with Pointers

Using Pointers

Value versus Pointer Receivers

New

Performance

## Nil Receivers

Nil Receivers Checks

## Summary

## 7 Testing

Testing Basics

# Table of Contents

Naming

The \* testing.T Type

Marking Test Failures

Using t. Error

Using t. Fatal (Recommended)

Crafting Good Test Failure Messages

## Code Coverage

Basic Code Coverage

Generating a Coverage Profile

The go tool cover Command

Generating an HTML Coverage Report

Editor Support

## Table Driven Testing

Anatomy of a Table Driven Test

Writing a Table Driven Test

Subtests

Anatomy of a Subtest

Writing Subtests

## Running Tests

Running Package Tests

Running Tests with Subpackages

Verbose Test Output

Logging in Tests

Short Tests

Running Package Tests in Parallel

Running Tests in Parallel

Running Specific Tests

Timing Out Tests

Failing Fast

# Table of Contents

Disabling Test Caching

Instantiating Generic Functions

## Test Helpers

Defining Test Helpers

Marking a Function as a Helper

Cleaning Up a Helper

## Summary

## 8 Interfaces

Concrete Types versus Interfaces

Explicit Interface Implementation

Implicit Interface Implementation

Before Interfaces

Using Interfaces

Implementing `io.Writer`

Multiple Interfaces

Asserting Interface Implementation

The Empty Interface

The `any` Keyword

The Problem with Empty Interfaces

Using an Empty Interface

Defining Interfaces

Defining a Model Interface

Implementing the Interface

Embedding Interfaces

Defining an `Validatable` Interface

Type Assertion

Asserting Assertion

Asserting Concrete Types

# Table of Contents

## Assertions through Switch

- Capturing Switch Type

- Beware of Case Order

## Using Assertions

- Defining the Callback Interfaces

- Breaking It Down

## Summary

## 9 Errors

### Errors as Values

- The error Interface

- Handling Errors

- Using Errors

### Panic

- Raising a Panic

- Recovering from a Panic

- Capturing and Returning Panic Values

### Dont Panic

- Checking for Nil

- Maps

- Pointers

- Interfaces

- Functions

- Type Assertions

- Array/Slice Indexes

### Custom Errors

- Standard Errors

- Defining Custom Errors

### Wrapping and Unwrapping Errors

# Table of Contents

Wrapping Errors

Unwrapping Errors

Unwrapping Custom Errors

To Wrap or Not To Wrap

Errors As/Is

As

Is

Stack Traces

Summary

## 10 Generics

What Are Generics?

The Problem with Interfaces

Type Constraints

Multiple Generic Types

Instantiating Generic Functions

Defining Constraints

Multiple Type Constraints

Underlying Type Constraints

The Constraints Package

Type Assertions

Mixing Method and Type Constraints

Generic Types

Summary

## 11 Channels

Concurrency and Parallelism

Concurrency Is Not Parallelism

Understanding Concurrency

Gos Concurrency Model



# **Table of Contents**

Goroutines

Goroutine Memory

The Go Scheduler

Work Sharing and Stealing

Dont Worry about the Scheduler

Goroutine Example

## **Communicating with Channels**

What Are Channels?

Understanding Channel Blocking/Unblocking

Creating Channels

Sending and Receiving Values

A Simple Channel Example

Ranging over a Channel

Listening to Channels with select

Using select Statements

Channels Are Not Message Queues

## **Unidirectional Channels**

Understanding Unidirectional Channels

## **Closing Channels**

Detecting Closed Channels on Read

Zero Value on Closed Read

Closing an Already-Closed Channel

Writing to a Closed Channel

## **Buffered Channels**

Basic Buffered Channel Example

Reading from Closed Buffered Channels

## **Capturing System Signals with Channels**

The os/signals Package

Implementing Graceful Shutdown

# **Table of Contents**

Listening for System Signals

Listening for Shutdown Confirmation

Timing Out a Nonresponsive Shutdown

Summary

## **12 Context**

The Context Interface

Context#Deadline

Context#Done

Context#Err

Context#Value

Helper Functions

The Background Context

Default Implementations

Context Rules

Context Nodal Hierarchy

Understanding the Nodal Hierarchy

Wrapping with Context Values

Following the Context Nodes

Context Values

Understanding Context Values

Key Resolution

Problems with String Keys

Key Collisions

Custom String Key Types

Securing Context Keys and Values

Securing by Not Exporting

Cancellation Propagation with Contexts

Creating a Cancelable Context

# **Table of Contents**

Canceling a Context

Listening for Cancellation Confirmation

## **Timeouts and Deadlines**

Canceling at a Specific Time

Canceling after a Duration

## **Context Errors**

Context Canceled Error

Context Deadline Exceeded Error

## **Listening for System Signals with Context**

Testing Signals

## **Summary**

# **13 Synchronization**

## **Waiting for Goroutines with a WaitGroup**

The Problem

Using a WaitGroup

The Wait Method

The Add Method

The Done Method

Wrapping Up Wait Groups

## **Error Management with Error Groups**

The Problem

The Error Group

Listening for Error Group Cancellation

Wrapping Up Error Groups

## **Data Races**

The Race Detector

Most, but Not All

Wrapping Up the Race Detector

## **Synchronizing Access with a Mutex**



# Table of Contents

Locker

Using a Mutex

RWMutex

Improper Usage

Wrapping Up Read/Write Mutexes

## Performing Tasks Only Once

The Problem

Implementing Once

Closing Channels with Once

## Summary

## 14 Working with Files

### Directory Entries and File Information

Reading a Directory

The FileInfo Interface

Stating a File

### Walking Directories

### Skipping Directories and Files

Skipping Directories

### Creating Directories and Subdirectories

### File Path Helpers

Getting a Files Extension

Getting a Files Directory

Getting a File/Directories Name

Using File Path Helpers

Checking the Error

### Creating Multiple Directories

### Creating Files

Truncation

# **Table of Contents**

Fixing the Walk Tests

    Creating the Files

Appending to Files

Reading Files

Beware of Windows

The FS Package

    The FS Interface

    The File Interface

Using the FS Interface

    File Paths

Mocking a File System

    Using MapFS

Embedding Files

    Using Embedded Files

    Embedded Files in a Binary

    Modifying Embedded Files

    Embedding Files as a String or Byte Slice

Summary

Index