

Литература

Заповеди Go

- [Harvey Kandola: Go Proverbs](#)
- [YouTube: The Go Programming Language: Go Proverbs - Rob Pike - Gopherfest - November 18, 2015:](#)
 - [Errors are values](#)
 - [Don't just check errors, handle them gracefully](#)

Интерфейсы в Go

- [A Tour of Go: Interfaces](#)
- [Digital Ocean: Gopher Guides: Использование интерфейсов в Go](#)
- [Effective Go: Interfaces and other types](#)
- [Effective Go: Interface names](#)
- [The Go Programming Language Specification: Interface types](#)
- [The Go Programming Language Specification: Switch Statements](#)
- [The Go Programming Language Specification: Type Assertions](#)
- [Хабр: Mail.ru Group: AloneCoder: Разбираемся с интерфейсами в Go](#)

errors

- [A Tour of Go: Errors](#)
- [Effective Go: Errors](#)
- [Frequently Asked Questions \(FAQ\): Why does Go not have exceptions?](#)
- [Go talks: Rob Pike: Go at Google: Language Design: 16. Errors](#)

- [Golang: Package errors](#)
- [Golang Tutorial: Return and handle an error](#)
- [The Go Blog: Andrew Gerrand: Error handling and Go](#)
- [The Go Blog: Rob Pike: Errors are values](#)
- [The Go Blog: Russ Cox: Experiment, Simplify, Ship: Errors](#)
- [The Go Programming Language Specification: Errors](#)
- [Wiki: Go - Обработка ошибок и исключительных ситуаций](#)

Работа с ошибками в Go 1.13

- [Earthly: Brandon Schurman: Effective Error Handling in Golang](#)
- [Medium: Gett Engineering: Alon Abadi: Error Handling in Go](#)
- [Medium: Felipe Dutra Tine e Silva: Golang: How to handle Errors in v1.13](#)
- [Medium: XING Engineering: Jens Hausherr: Designing errors with Go 1.13](#)
- [The Go Blog: Damien Neil and Jonathan Amsterdam: Working with Errors in Go 1.13](#)
- [Блог о языке программирования Go: Работа с ошибками в Go 1.13](#)
- [Хабр: Mail.Ru Group: AloneCoder: Работа с ошибками в Go 1.13](#)
- [Хабр: OTUS: MaxRokatansky: Обработка ошибок в Go](#)

Стиль

- [golang/go: Wiki: CodeReviewComments: Error Strings](#)
- [golang/go: Wiki: CodeReviewComments: Handle errors](#)
- [golang/go: Wiki: CodeReviewComments: In-Band Errors](#)
- [golang/go: Wiki: CodeReviewComments: Indent Error Flow](#)
- [Uber Go Style Guide: Error Types](#)
- [Uber Go Style Guide: Error Wrapping](#)

- [Uber Go Style Guide: Linting](#)

Мнение

- [golang/go: Wiki: ExperienceReports: Error Handling](#)
- [golang/go: Wiki: ExperienceReports: Error Values](#)

YouTube

- [Donald Feury: Will error wrapping change everything!? Go / Golang 1.13 Error Wrapping | #feurious](#)
- [dotconferences: dotGo 2019 - Marcel van Lohuizen - Go 2 error values today](#)
- [Sphere Software: Helpful Golang Practices: Error Handling](#)
- [The Art of Developing: Изучаем Golang. Урок 7. Основы. Работа с ошибками](#)
- [The Go Programming Language: Go Proverbs - Rob Pike - Gopherfest - November 18, 2015](#)

В книгах

- **Go Cookbook: Build modular, readable, and testable applications in Go** (Aaron Torres)
 - Chapter 4: Error Handling in Go
- **Go in 24 Hours** (George Ornbo)
 - Hour 10. Handling Errors
- **Go in Action** (William Kennedy, Brian Ketelsen, Erik St. Martin)
 - 2 Go quick-start
- **Go in Practice** (Matt Butcher, Matt Farina)
 - 4.1 Error handling

- **Go: Design Patterns for Real-World Projects** (Vladimir Vivien, Mario Castro, Contreras Mat Ryer)
 - Chapter 5: Functions in Go: Error signaling and handling (взято из **Learning Go Programming** (Vladimir Vivien))
- **Head First Go: A Brain-Friendly Guide** (Jay McGavren)
- **How To Code in Go** (Mark Bates, Cory LaNou, Tim Raymond)
 - Handling Errors in Go
 - Creating Custom Errors in Go
- **Introducing Go: Build Reliable, Scalable Programs** (Caleb Doxsey)
 - Errors
- **Mastering Go** (Mihalis Tsoukalos)
 - Error handling
- **The Go Programming Language** (Alan Donovan, Brian Kernighan)
 - 7.8. The error Interface
- **The Go Workshop: A New, Interactive Approach to Learning Go** (Delio D'Anna, Andrew Hayes, Sam Hennessy, Jeremy Leason, Gobin Sougrakpam, and Dániel Szabó)
 - Chapter 6: Errors
- **The Little Go Book** (Karl Seguin)

Прочее

- [Bacancy Technology: Archita Nayak: Golang Error Handling: Go's Way of Handling Errors](#)
- [Build Web Application with Golang: Error handling](#)
- [Carl M. Johnson: Creating Domain Specific Error Helpers in Go With errors.As](#)
- [Dave Cheney: Don't just check errors, handle them gracefully](#)

- [DEV Community: Lane Wagner: Wrapping Errors in Go – How to Handle Nested Errors](#)
- [Develop Paper: Breakout of golang error](#)
- [Digital Ocean: Gopher Guides: Обработка ошибок в Go](#)
- [Digital Ocean: Gopher Guides: Создание настраиваемых ошибок в Go](#)
- [Donald Feury: Error Chains in Go 1.13](#)
- [Educative: Edpresso Team: How to handle errors in Golang](#)
- [Gabriel Tanner: Error handling in Golang](#)
- [Go by Example: Errors](#)
- [Golang By Example: Error in Go \(Golang\) - Advanced](#)
- [Golang Tutorial: Example to create custom error](#)
- [GoLang Docs: Errors and Exception Handling in GoLang](#)
- [GopherCon: Jonathan Amsterdam: Working with Error Wrapping](#)
- [golangbot.com: Naveen Ramanathan: Custom Errors](#)
- [golangbot.com: Naveen Ramanathan: Error Handling](#)
- [golangify.com: Vasile Buldumac: Обработка ошибок в Golang](#)
- [Hackernoon: Stupid Gopher: Golang – handling errors gracefully](#)
- [Ilija Eftimov: Deep Dive in the Upcoming Go Error Inspection Changes](#)
- [inanzzz.com: Creating custom errors with Golang](#)
- [Learn Go with tests: Error types](#)
- [LogRocket Blog: Raphael Ugwu: Error handling in Golang](#)
- [Medium: ITNEXT: Che Dan: Golang Error Handling – Best Practice in 2020](#)
- [Medium: NOP::Nuances of Programming: Olga Sayfudinova: Golang – изящная обработка ошибок](#)
- [Medium: Sebastian Dahlgren: Go Best Practices – Error handling](#)
- [Medium: YAKKA: Luke Freeman: Better Go error handling](#)
- [Peter Bourgon: Programming with errors](#)
- [Practical Go Lessons: Maximilien Andile: Chapter 23: Errors](#)

- yourbasic.org/golang: 3 simple ways to create an error
- [Антон Жиянов: Красавица и чудовище. Обработка ошибок в Go](#)

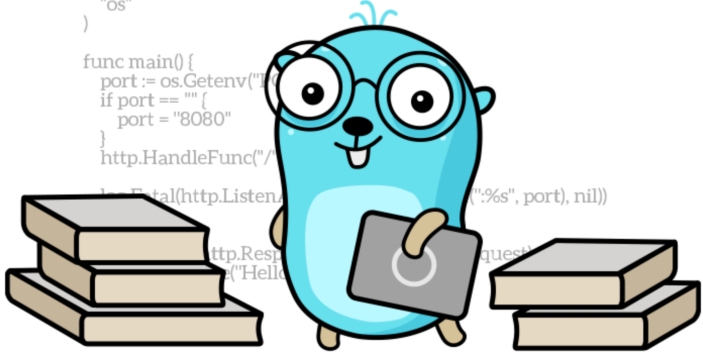
Делитесь своими ссылками в комментариях :)



```
// This server can run on App Engine.
package main

import (
    "fmt"
    "log"
    "net/http"
    "os"
)

func main() {
    port := os.Getenv("PORT")
    if port == "" {
        port = "8080"
    }
    http.HandleFunc("/hello", func(w http.ResponseWriter, r *http.Request) {
        fmt.Fprintln(w, "Hello, world!")
    })
    log.Fatal(http.ListenAndServe(":"+port, nil))
}
```



go learn()