

# **Golang Workshop**

Introduction

# About me

- Wojciech Barczynski
- Tech lead
- Golang Developer with many years of experience

# What I like

- Simplicity
- Standard library
- Composition
- Pragmatism
- Readability

# Culture

Golang is simple.

So we dive directly into talking about problems to solve.

# Birth

- Born out of frustration.
- Efficient compilation, efficient execution, or ease of programming?
- First sketches in September 2007 (Robert Griesemer, Rob Pike, Ken Thompson).
- Plenty of siblings (Rust, Elixir, Swift)

# Language goals

- Easy: compiled, statically typed, but with a dynamic feel.
- Modern: support for networked and multicore computing.
- Fast: at most a few seconds to build a large executable on a single computer.

# Language goals

"Go was intended as a language for writing server programs that would be easy to maintain over time." (golang.org)

"Clean procedural language designed for scalable cloud software."  
(Rob Pike)

# Core principles

- Reduce clutter and complexity.
- Everything is declared exactly once.
- Syntax is clean and light on keywords.
- No hierarchy: types just are, they don't have to announce their relationships.
- Orthogonality of concepts.
- Orthogonality of features.



# Where is my feature?

- Go does not compete on features.
- As of Go  $\pm 2$ , the language is fixed.
- Features add complexity and hurt readability.
- Simple features that interact in simple ways.
- Simple means readable and predictable.
- Simple means easy to maintain.

# Simple things in Go

- garbage collection
- goroutines
- constants
- interfaces
- packages

Each hides complexity behind a simple facade.

# Proverbs

- <https://go-proverbs.github.io/>

# Idiomatic Golang

- Simplicity
- Pragmatic
- Readability

# Idiomatic Golang

Examples:

- Test against real db
- Dislike mocks
- Dislike ORMs
- Avoid unnecessary abstractions

# Idiomatic Golang

- Talks of Rob Pike
- Peter Bourgon's talks and blog (e.g., [peter.bourgon.org/go-for-industrial-programming](https://peter.bourgon.org/go-for-industrial-programming))
- Dave Cheney

# Idiomatic Golang

- Join Golang meetups in your city
- Join the slack [invite.slack.golangbridge.org](https://invite.slack.golangbridge.org)

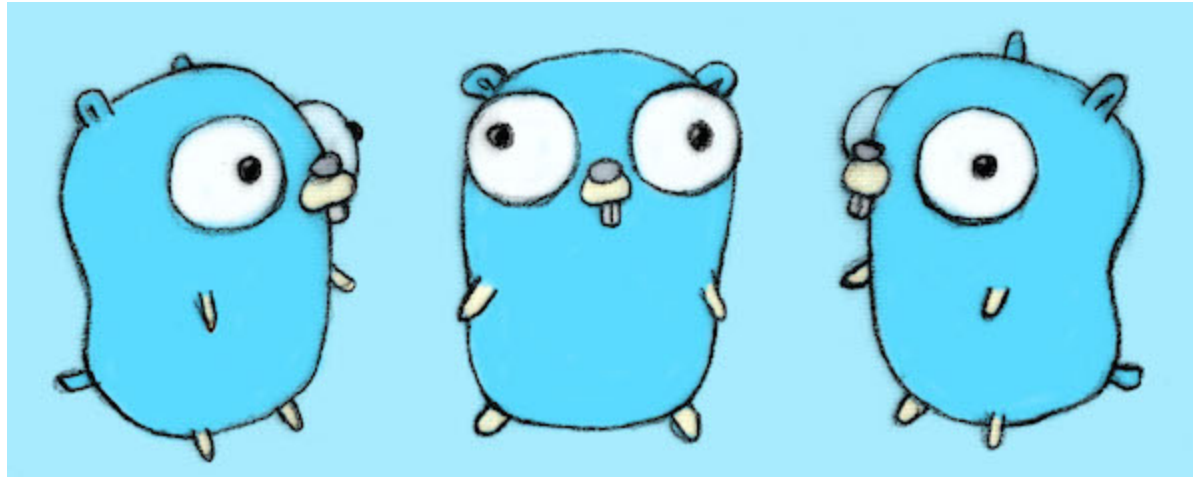
# Where Golang shines

- macro/micro-services, especially on K8S
- Kubernetes/CloudNative ecosystem
- Serverless



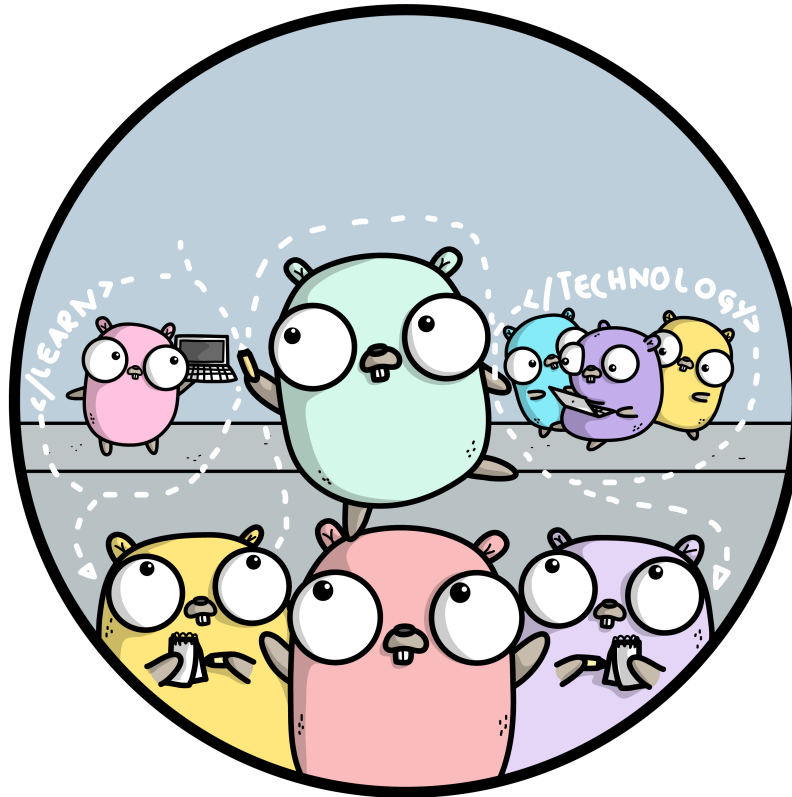
# The most important

From <https://go.dev/blog/gopher>:



# The most important

From <https://github.com/ashleymcnamara/gophers>:



# Basics

Packages:

- collection of files in the same dir;
- all functions,...,variables are visible within the package.

# Basics

Repository:

- one or more modules;
- modules is a collection of packages;
- each module path indicates where to download it.

# References

- [Rob Pike - Simplicity is complicated](#)
- [Tomasz Grodzki - Why Golang looks like that](#)