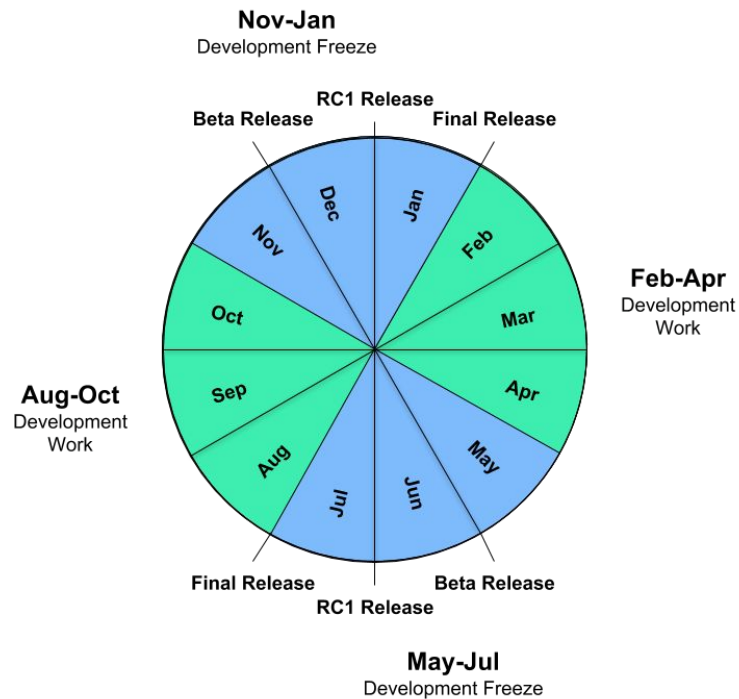


What's new in Go

November 2022

Go release cycle



<https://github.com/golang/go/wiki/Go-Release-Cycle>

Go development freeze for Go 1.20

- Effective Nov 21 (yesterday)
- Release target: early Feb 2023

“We’ll be **increasing the development window by 3 weeks**, keeping the final release date the same.”

<https://groups.google.com/g/golang-dev/c/uKe8zWxuyOI/m/xTVvkYQ3BgAJ>

Long-term discussion: error handling

Recent: simplify zero values when returning errors

- return mystruct{}, package.PackageStruct{}, err
+ return ..., err

Or:

- return mystruct{}, package.PackageStruct{}, err
+ return err

Status: Active discussion

<https://github.com/golang/go/issues/21182>

<https://github.com/golang/go/issues/56628>

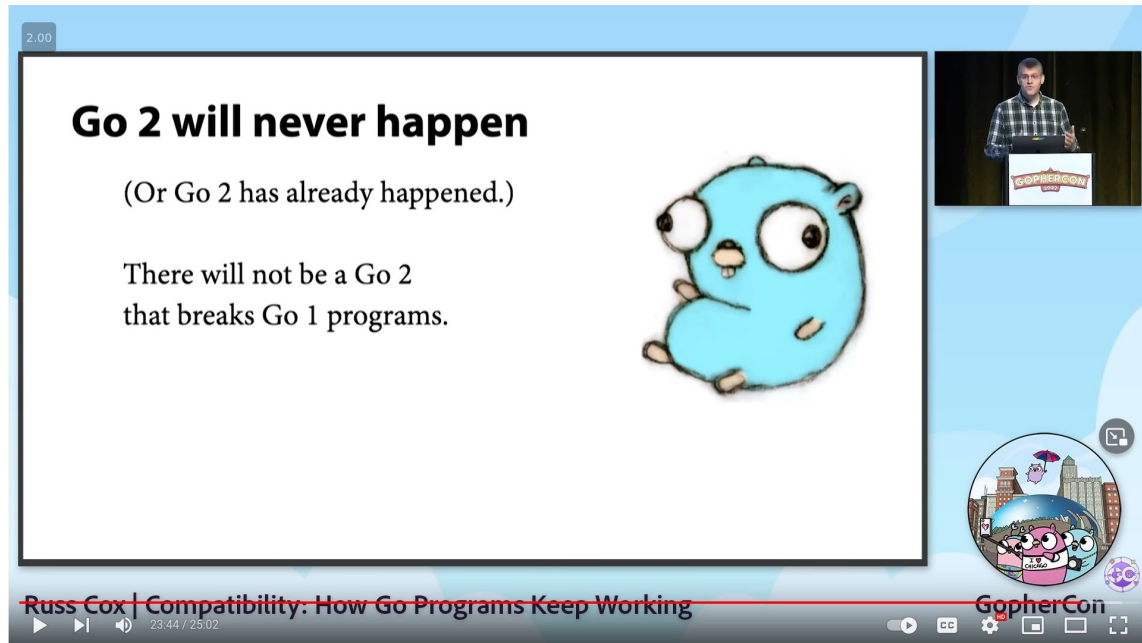
Active proposal: slog for structured, leveled logging

- Standard library: `log.Printf` and `log.Fatalf/log.Panicf`.
- Alternatives: `logrus`, `zap`, ...
- Leveled logging: `Debug`, `Info`, `Warning`, `Error`, etc.
- Structured logging, i.e. with named fields, as `JSON` or `logfmt` or ...
- Low overhead (limit allocations).
- Single common log package is helpful in applications with many dependencies.

Status: Proposal

<https://github.com/golang/go/issues/56345>

Go compatibility



The screenshot shows a video player interface. The main content is a presentation slide with a white background and a black border. The slide text reads: **Go 2 will never happen**, followed by *(Or Go 2 has already happened.)*, and then **There will not be a Go 2 that breaks Go 1 programs.** To the right of the text is a cartoon illustration of a blue gopher. The video player's top bar is light blue and shows a '2:00' timer. The bottom bar is dark grey and contains a red progress bar, a play button, a volume icon, and a timestamp '23:44 / 25:02'. On the right side of the player, there is a small inset video of a man speaking, a circular logo for 'GopherCon' featuring a city skyline and a gopher, and a 'GopherCon' text logo.

2:00

Go 2 will never happen

(Or Go 2 has already happened.)

**There will not be a Go 2
that breaks Go 1 programs.**

Russ Cox | Compatibility: How Go Programs Keep Working

23:44 / 25:02

GopherCon

GopherCon 2022: Russ Cox - Compatibility: How Go Programs Keep Working

<https://www.youtube.com/watch?v=v24wrd3RwGo>

Go compatibility

- Extending Go backward compatibility
 - Compiling with newer compiler than specified in go.mod
 - GODEBUG=... and //go:debug x509sha1=1 and go version in go.mod
- Extending Go forward compatibility
 - Compiling code that specifies new compiler in go.mod
 - Simply download new compiler?

Status: Discussion

<https://github.com/golang/go/discussions/55090>

<https://github.com/golang/go/discussions/55092>

Profile-guided optimization (PGO)

- Compile binary with profile for better compiler optimization decisions.
 - Initially: inlining.
- Goals: easy to use, profiles (pprof) from production workload.
- Originated at Uber.

Status: Proposal accepted, in development, preview in Go 1.20?

<https://go.googlesource.com/proposal/+master/design/55022-pgo.md>

clear(x) builtin

- To efficiently clear a map, slice, pointer to array
- Discussion also involved floating point NaN as keys. NaN != NaN.

Status: Proposal, likely accept

<https://github.com/golang/go/issues/56351>

Wrap multiple errors in a single error

- `fmt.Errorf("%w: %w", err1, err2)`
- `func (e Error) Unwrap() []error`

Status: In Go 1.20

<https://github.com/golang/go/issues/53435>

Context cancellation with cause/error

- `func WithCancelCause(Context) (Context, CancelCauseFunc)`
- `type CancelCauseFunc func(error)`
- `func Cause(Context) error`

Status: In Go 1.20

<https://github.com/golang/go/issues/51365>

Fewer .a files in Go distributions and less need for cgo

"Compressed as a zip file with default settings, the archive is 135M. Without .a files, it's 86M (63%)."

- Packages net and os/user will use a pure Go (or macOS libc shim) implementation: less cgo needed.
- No C compiler needed for race detector on macOS.

Status: In Go 1.20

<https://github.com/golang/go/issues/47257>

golang.org/x/ repo's have version tags

E.g. your go.mod files:

- `golang.org/x/net v0.0.0-20220927171203-f486391704dc // indirect`
- `golang.org/x/sys v0.0.0-20220928140112-f11e5e49a4ec // indirect`
- + `golang.org/x/net v0.1.0 // indirect`
- + `golang.org/x/sys v0.1.0 // indirect`

Status: Already happened, not tied to Go release.

<https://groups.google.com/g/golang-nuts/c/rfAsWvH1zbs/m/EmA5A1juBwAJ>

How to stay up to date with Go development

Tip of the month:

- Subscribe the Golang weekly newsletter, <https://golangweekly.com/>

More tips:

- whats-new.md in <https://github.com/goamsterdam/meetups>

You can do the next “What’s new in Go”!