Welcome to the 10th Go devroom!

Do not leave empty spaces in between!
Welcome to the 10th Go devroom!

Please defragment to this side →
THE STATE OF GO
What's new since Go 1.20
Who am I

Maartje Eyskens
Sr. Software Engineer

Lives on: all European trains
Works on: SIG-Service Mesh

@maartje@blahaj.social

Photo copyright RTBF
Agenda

- Changes to:
  - the language
  - the standard library
  - the tooling

- Updates on design drafts

- Community update
What's new since Go 1.20

- Go 1.21
  - Released August 8th 2023

- Go 1.22
  - To be released in February 2024
CHANGES TO THE LANGUAGE
New built in functions

- \texttt{min()} and \texttt{max()} functions

\begin{align*}
m & := \texttt{min}(2, 10) \quad \text{// } m = 2 \\
m & := \texttt{max}(2, 10) \quad \text{// } m = 10 \\
f & := \texttt{max}(0, \texttt{float32}(x)) \quad \text{// type of } f \text{ is } \texttt{float32} \text{ (generics!)}
\end{align*}
New built in functions

- `clear()` function
  - for slices and maps
  - sets all elements to the zero value

```go
s := []int{1, 2, 3, 4, 5}
clear(s)

len(s) // 5
cap(s) // 5
```
New built in functions

- `clear()` function
  - for slices and maps
  - sets all elements to the zero value

```go
s := []int{1, 2, 3, 4, 5}
clear(s)
for _, v := range s {
    println(v) // 0
}
```
Fixes non well defined behaviour for non explicit imports
The new algorithm is:
  - sort by import path
  - repeat until all:
    - find the first package which all imports are already initialized of
    - initialize that package
Nil defer panic

- funcions
1 REVOLUTIONARY CHANGE TO THE LANGUAGE
```go
for i := 0; i < 10; i++ {
    fmt.Println(i)
}
```
for i := 0; i < 10; i++ {
    fmt.Println(i)
}
for i := range 10 {
    fmt.Println(i)
}
range over an integer in for loops

- Just like in other languages like Python
- Part of the Rangefunc Experiment

```go
for i := range 10 {
    fmt.Println(i)
}
```

0
1
2
3
4
5
6
7
8
9
for loops

- Variables get created for each loop iteration
- Prevents value reuse bugs
Go 1.22

for loops

- Variables get created for each loop iteration
- Prevents value reuse bugs

var zero *int

for i := 0; i < 100; i++ {
    if i == 0 {
        zero = &i
    }
}

fmt.Println(*zero)
for loops

- Variables get created for each loop iteration
- Prevents value reuse bugs

```go
var zero *int

for i := 0; i < 100; i++ {
    if i == 0 {
        zero = &i
    }
}

fmt.Println(*zero) // 100
```
for loops

- Variables get created for each loop iteration
- Prevents value reuse bugs

```go
var zero *int
for i := range 100 {
    if i == 0 {
        zero = &i
    }
}
fmt.Println(*zero)
```
for loops

- Variables get created for each loop iteration
- Prevents value reuse bugs

```go
var zero *int
for i := range 100 {
    if i == 0 {
        zero = &i
    }
}
fmt.Println(*zero) // 0
```
This page is intentionally left void
go vet
go vet: new warnings

- missing values after `append` (classic mistake!)

```go
numbers := []int{}
for i := 0; i < 100; i++ {
    numbers = append(numbers)
}
```
go vet: new warnings

- Reports call to `time.Since` within a `defer` statement

```go
    t := time.Now()
    defer log.Println(time.Since(t)) // prints time of this line, not deferred time

    defer func() {
        log.Println(time.Since(t)) // a correctly deferred call to time.Since
    }()
```
go vet: new warnings

- Reports call to `time.Since` within a `defer` statement

```go
// Reports call to time.Since within a defer statement

// Prints time of this line, not deferred time

// A correctly deferred call to time.Since

```

go vet: **new** removed warnings

- no longer reports references to loop variables outside the loop

```go
var zero *int

for i := 0; i < 100; i++ {
    if i == 0 {
        zero = &i
    }
}
fmt.Println(*zero)
```
go command
Go test

- `go test -fullpath`
  prints full path names in test log messages
Go test

- `go test -cover` prints coverage summaries for packages without tests

? mymod/mypack [no test files]

mymod/mypack coverage: 0.0% of statements
Go Modules Vendoring

- Workspaces can now use a vendor directory containing the dependencies of the workspace.
go mod

- `go mod init` no longer attempts to import module requirements from configuration files for other vending tools (such as Gopkg.lock).
Other vendor tools??

- Sit down my child... story time...
- When I was young we had a package manager battle!

$GOPATH → $GOPATH/.../vendor

- go dep
- glide
- Maartje’s dep tool
- go mod
- godep
STANDARD LIBRARY
- **Buffer type has two new methods**
  - `Available() int`
  - `AvailableBuffer() []byte`
  - useful with the `Write` method to append directly
context

- `WithoutCancel()`
  - returns a copy of a context that is not canceled when the original context is
- `WithDeadlineCause()` and `WithTimeoutCause()`
  - Provides `Cause()` then timeout hit
- `AfterFunc()`
  - registers a function to run after a cancel
crypto/ecdsa & crypto/rsa

- PublicKey.Equal & PrivateKey.Equal now execute in constant time
You can now control the content of session tickets.

VersionName() returns TLS version

Better error messages using RFC 5246 and RFC 8446

TLS1.2 is default minimum
errors

- `ErrUnsupported` a new standard error
  - HTTP 405 unsupported method
  - `os.Link()` on non hard-link filesystem
A flag definition
- Bool, BoolVar, Int, IntVar, etc.
- now panic if `Set()` was called on the same name

intended to detect cases with unintended initialisation order
Once Func()
Once Value() & Once Values()
capture a common use of Once() to lazily initialize a value on first use
Writer.AddFS adds all of the files from an fs.FS to the archive
Enhanced HTTP routing

- `net/http.ServeMux` accept methods and wildcards

```go
mux := http.NewServeMux()
mux.HandleFunc("/", func(w, r) {
    fmt.Fprintf(w, "Welcome to the Go Devroom")
})
```
Enhanced HTTP routing

- `net/http.ServeMux` accept methods and wildcards

```go
mux := http.NewServeMux();
mux.HandleFunc("POST /talk", func(w,r) {
    fmt.Fprintf(w, "CFP Received")
})
```
Enhanced HTTP routing

- `net/http.ServeMux` accept methods and wildcards

```go
mux := http.NewServeMux()

mux.HandleFunc("GET /talk/{id}", func(w, r) {
    if r.PathValue("id") == "0" {
        fmt.Fprintf(w, "State of Go")
    }
})
```
Enhanced HTTP routing

- `net/http.ServeMux` accept methods and wildcards

```go
mux := http.NewServeMux()
mux.HandleFunc("GET /recordings/{file...}", func() {
    os.Open("recordings/" + r.PathValue("file"))
    [...]}
})
```
Enhanced HTTP routing

- `net/http.ServeMux` accept methods and wildcards

```go
mux.HandleFunc("GET /home/", ...) // matches /home/*
mux.HandleFunc("GET /home/{$}", ...) // matches /home/
```
Enhanced HTTP routing

- `net/http.ServeMux` accept methods and wildcards

- Patterns with "{" and "}" behave differently
net/http

- Support for fs.FS
  - ServeFileFS
  - FileServerFS
  - NewFileTransportFS

- server & client reject an invalid empty Content-Length header
NEW PACKAGES
log/slog

- meow

This slide is intentionally left *unfinished*
Structured logging

- log/slog
- Machine readable logging
- Hopes to replace logrus, zap, zerolog, logr, glog, hclog, klog

https://go.googlesource.com/proposal/+master/design/56345-structured-logging.md
import "log/slog"

func main() {
    slog.Info("hello", "name", "Al")
    slog.Error("oops", net.ErrClosed, "status", 500)
    slog.LogAttrs(slog.LevelError, "oops",
}
Structured logging

```go
import "log/slog"

func main() {
    slog.Info("hello", "name", "Al")
    slog.Error("oops", net.ErrClosed, "status", 500)
    slog.LogAttrs(slog.LevelError, "oops",
                  slog.Int("status", 500),
                  slog.Any("err", net.ErrClosed))
}
```

time=2022-10-24T16:05:48.054-04:00 level=INFO msg=hello name=Al
time=2022-10-24T16:05:48.054-04:00 level=ERROR msg=oops status=500 err="use of closed network connection"
time=2022-10-24T16:05:48.054-04:00 level=ERROR msg=oops status=500 err="use of closed network connection"
Structured logging

Info()
Error()
Warn()
LogAttrs()
Structured logging

- Implementations and Interfaces for:
  - Log levels
    - Debug
    - Info
    - Warn
    - Error
  - Passing data in arguments
  - Outputting to text, JSON, ...

https://go.googlesource.com/proposal/+master/design/56345-structured-logging.md
Structured logging

- [https://pkg.go.dev/testing/slogtest](https://pkg.go.dev/testing/slogtest)

[https://go.googlesource.com/proposal/+/master/design/56345-structured-logging.md](https://go.googlesource.com/proposal/+/master/design/56345-structured-logging.md)
maps package

- Clone()
- Copy(dst, src)
- DeleteFunc(m, func(K,V) bool)
- Equal(m1,m2) bool
- EqualFunc(m1,m2,func() bool) bool

https://pkg.go.dev/maps
slices package

- Clone(), Copy(), DeleteFunc(), Equal(), EqualFunc() bool
- But also... BinarySearch(), Grow(), Index(), Insert(), Reverse(), Sort()
- too much for 1 slide, see https://pkg.go.dev/slices
cmp package

- `Compare(x, y) int`
  - -1 less, 0 equal, +1 more
- `Less(x, y) bool`

- `Or(x, y)` returns the first non zero value.

- [https://pkg.go.dev/cmp](https://pkg.go.dev/cmp)
math/rand/v2

- First “v2” package in the standard library!
- Low level `Read()` is removed, use crypto/math
- Global generator is unconditionally randomly seeded
- More idiomatical function spelling
- Generic `rand.N(5*time.Minute)`
- ChaCha8 algorithm
RUNTIME
runtime/trace

- Collecting traces 10x faster*
- Traces contain all explicit stop-the-world events

*on amd64 and arm64
PGO

- The -pgo build flag now defaults to -pgo=auto
  - Used if default.pgo is found

- Build speed by up to 6%, largely thanks to building the compiler itself with PGO.
Performance improvements

- The runtime now keeps type-based garbage collection metadata nearer to each heap object, improving the CPU performance (latency or throughput) of Go programs by 1–3%. This change also reduces the memory overhead of the majority Go programs by approximately 1% by deduplicating redundant metadata. Some programs may see a smaller improvement because this change adjusts the size class boundaries of the memory allocator, so some objects may be moved up a size class.

- A consequence of this change is that some objects' addresses that were previously always aligned to a 16 byte (or higher) boundary will now only be aligned to an 8 byte boundary. Some programs that use assembly instructions that require memory addresses to be more than 8-byte aligned and rely on the memory allocator's previous alignment behavior may break, but we expect such programs to be rare. Such programs may be built with GOEXPERIMENT=noallocheaders to revert to the old metadata layout and restore the previous alignment behavior, but package owners should update their assembly code to avoid the alignment assumption, as this workaround will be removed in a future release.

- On the windows/amd64 port, programs linking or loading Go libraries built with -buildmode=c-archive or -buildmode=c-shared can now use the SetUnhandledExceptionFilter Win32 function to catch exceptions not handled by the Go runtime. Note that this was already supported on the windows/386 port.
PORTS
PORTS

- Go 1.21 requires Windows 10 (server 2016) or later
- Go 1.21 requires macOS 10.15 Catalina or later
- Go 1.22 is the last to support macOS 10.15 Catalina
- Go 1.22 adds experimental PowerPC64 support to OpenBSD
Changes to WebAssembly

- go:wasmimport directive can now be used to import functions from the WebAssembly host.
- The Go scheduler now interacts much more efficiently with the JavaScript event loop.
- Experimental support for the WebAssembly System Interface (WASI)
Changes to GOARM

- Values now support setting of soft-, hardware floating point
- eg. GOARM=7, softfloat
  GOARM=6, hardfloat
- Default: softfloat for v5 and hardfloat for v6&7.
Changes to Windows

- `File.Chdir` method now changes the current directory
- `TempDir` now uses `GetTempPath2W` when available
  - security hardening measure.
- `os` package supports files with UTF-16 names which can’t be represented UTF-8
- `Lstat` now resolves symbolic links
Changes to Windows

- Many more fixes in the os package...
  - It almost feels like somebody finally used Go on Windows...
THE FUTURE OF GO

Go 2.0 and Design Drafts
Rangefunc experiment

- preliminary implementation in 1.22
- Gives the generic type `Seq`

```go
type Seq[V any] func(yield func(V) bool)
type Seq2[K, V any] func(yield func(K, V) bool)
```
RangeFunc experiment

type Seq[V any] func(yield func(V) bool)

func MyOwnRange[E any](s []E) func(func(int, E) bool) {
    return func(yield func(int, E) bool) {
        for i := len(s); i < 0; i++ {
            // insert logic here
        }
    }
}

for v := range MyOwnRange(data) {
    // ...
}
Rangefunc experiment

type Seq[V any] func(yield func(V) bool)

func MyOwnRange[E any](s []E) func(func(int, E) bool) {
    return func(yield func(int, E) bool) {
        for i := len(s); i < 0; i++ {
            // insert logic here
        }
    }
}

for v := range MyOwnRange(data) {
    // ...
}
Rangefunc experiment

type Seq[V any] func(yield func(V) bool)

func MyOwnRange[E any](s []E) func(func(int, E) bool) {
    return func(yield func(int, E) bool) {
        for i := len(s); i < 0; i++ {
            // insert logic here
        }
    }
}

for v := range MyOwnRange(data) {
    // ...
}
COMMUNITY
Kris Nóva
Go Developer Network on Meetup

102,000
(-25,000)
Members

107
(-83)
Meetups

41
(-10)
Countries
WOMEN WHO GO & GOBRIDGE MEETUPS

41 chapters*
10 years ago... this happened

<table>
<thead>
<tr>
<th>Event</th>
<th>Speakers</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sunday</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Go Devroom Welcome</td>
<td>Andrew Gerrand</td>
<td>09:45</td>
<td>10:00</td>
</tr>
<tr>
<td>Iris Decentralized Messaging</td>
<td>Péter Szilágyi</td>
<td>10:05</td>
<td>10:35</td>
</tr>
<tr>
<td>Peer-to-peer based messaging for back-end service decentralization</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Camlistore</td>
<td>Brad Fitzpatrick</td>
<td>10:40</td>
<td>11:40</td>
</tr>
<tr>
<td>your personal storage system for life</td>
<td>Matthew Cottingham</td>
<td>11:45</td>
<td>12:15</td>
</tr>
<tr>
<td>Interfaces: a new leaf for an old book</td>
<td>Sugu Sougoumarane</td>
<td>13:00</td>
<td>14:00</td>
</tr>
<tr>
<td>Scaling with go: Youtube's Vitess</td>
<td>Elliott Stoneham</td>
<td>14:05</td>
<td>14:35</td>
</tr>
<tr>
<td>Write your own Go compiler</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>More adventures with go.tools/ssa</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Looking toward Go 1.3</td>
<td>Andrew Gerrand</td>
<td>14:40</td>
<td>15:10</td>
</tr>
<tr>
<td>What's coming in the next major release of Go</td>
<td>Matthew Cottingham</td>
<td>15:15</td>
<td>16:15</td>
</tr>
<tr>
<td>Porting Go to New Platforms</td>
<td>Aram Hävärmanu</td>
<td>16:20</td>
<td>17:20</td>
</tr>
<tr>
<td>Lessons learned from the Solaris port</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Go Lightning Talks</td>
<td>Andrew Gerrand</td>
<td>16:20</td>
<td>17:20</td>
</tr>
</tbody>
</table>
2014, a cold day in Brussels
Back in 2014 (10 years!)
Traditions never die!
2017 ... find Maartje!
2019 ...
2020: “moments” before a pandemic
2020: “moments” before a pandemic
2021: first online edition

But time for a change!
2021: first online edition

But time for a change!
2022: online, everywhere

Gophers around the world
- Go Devroom @ FOSDEM 2022

What are the generics?

type Book struct{}
func main() { var BookStore CRUD[Book] }

Why Go 1.18?
- Fuzzing

Overview
- Who am I?
- Why am I talking about this?
- Authorama-what?
- How do I engage work?
- Where did I spend them up?
2024

3D full immersion VR image available: LOOK AROUND YOU

* Apple Vision Pro Support for Go Devroom coming 2034**

** given availability of the Apple Vision Pro at said date
Conferences

- Go Devroom @ FOSDEM 2024 *(you are here)*
- GopherCon Europe Athenes  February 6-8
- Conf42 Go Online  April 25
- GothamGo New York  April 26
- GopherCon Brazil  May 9-19
- GopherCon Europe Berlin  June 17-20
- Gophercon Chicago  July 7-10
# 2024: Happy Birthday!

<table>
<thead>
<tr>
<th>Event</th>
<th>Speakers</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Saturday</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The state of Go</td>
<td>Maartje Eyskens</td>
<td>10:30</td>
<td>11:00</td>
</tr>
<tr>
<td>The secret life of a goroutine</td>
<td>Jesús Espino</td>
<td>11:00</td>
<td>11:30</td>
</tr>
<tr>
<td>You're already running my code in production: My simple journey to becoming a Go contributor.</td>
<td>Jonathan Hall</td>
<td>11:30</td>
<td>12:00</td>
</tr>
<tr>
<td>Single binary, full-stack provisioning</td>
<td>James (purpleidea)</td>
<td>12:00</td>
<td>12:30</td>
</tr>
<tr>
<td>Efficient Integration Testing in Go: A Case Study on Dapr</td>
<td>Josh van Leeuwen</td>
<td>12:30</td>
<td>13:00</td>
</tr>
<tr>
<td>Effortless Bug Hunting with Differential Fuzzing</td>
<td>Maciej Mionskowski</td>
<td>13:00</td>
<td>13:30</td>
</tr>
<tr>
<td>Maintaining Go as a day job - an update</td>
<td>Filippo Valsorda</td>
<td>13:30</td>
<td>14:00</td>
</tr>
<tr>
<td>How we almost secured our projects by writing more tests</td>
<td>Alessio Greggi</td>
<td>14:00</td>
<td>14:30</td>
</tr>
<tr>
<td>Dependency injection: a different way to structure a project</td>
<td>Dylan Reimerink</td>
<td>14:30</td>
<td>15:00</td>
</tr>
<tr>
<td>Putting an end to Makefiles in go projects with GoReleaser</td>
<td>Denis GERMAIN</td>
<td>15:00</td>
<td>15:30</td>
</tr>
<tr>
<td>REST in Peace: using generics to remove REST boilerplate</td>
<td>Tanguy Herrmann</td>
<td>15:30</td>
<td>16:00</td>
</tr>
<tr>
<td>Low code graphical apps with Go top to bottom!</td>
<td>Andrew Williams</td>
<td>16:00</td>
<td>16:30</td>
</tr>
<tr>
<td>Creating a multiplayer game in Go, from zero</td>
<td>Francesc Gil</td>
<td>16:30</td>
<td>17:00</td>
</tr>
<tr>
<td>Clearing Out the Cobwebs: Finding and Removing Dead Code. In Production.</td>
<td>Martin Kjellstrand</td>
<td>17:00</td>
<td>17:30</td>
</tr>
<tr>
<td>Smartwatch firmware... in Go? On TinyGo, small displays, and building a delightful developer experience</td>
<td>Ayke van Laethem</td>
<td>17:30</td>
<td>18:00</td>
</tr>
<tr>
<td>Go Without Wires Strikes Back</td>
<td>Ron Evans</td>
<td>18:00</td>
<td>18:30</td>
</tr>
<tr>
<td>Go Lightning Talks</td>
<td>Maartje Eyskens</td>
<td>18:30</td>
<td>19:00</td>
</tr>
</tbody>
</table>
Lightning talks

- Today 18:30-19:00
- 5 minute talks
- CfP open till 17:00 (5pm)
- https://gophers.love/light
Social Media

- #golang #FOSDEM
- @godevroom@fosstodon.org on the fediverse
THANKS!
Enjoy all talks!

Gophers by Renee French, Ashley McNamara, Egon Elbre, Sam Croswell and Miguel Molina, Marcus Olsson, Takuya Ueda
Glenda by Renee French