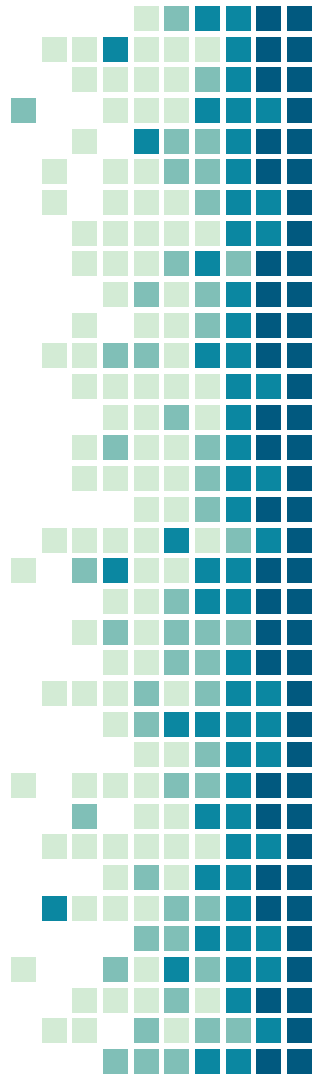


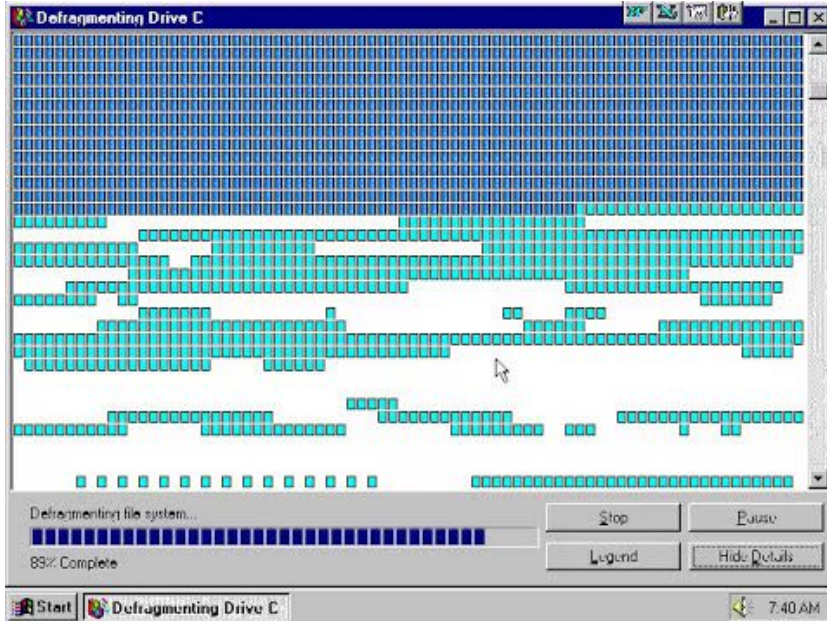
Welcome to the 10th Go devroom!



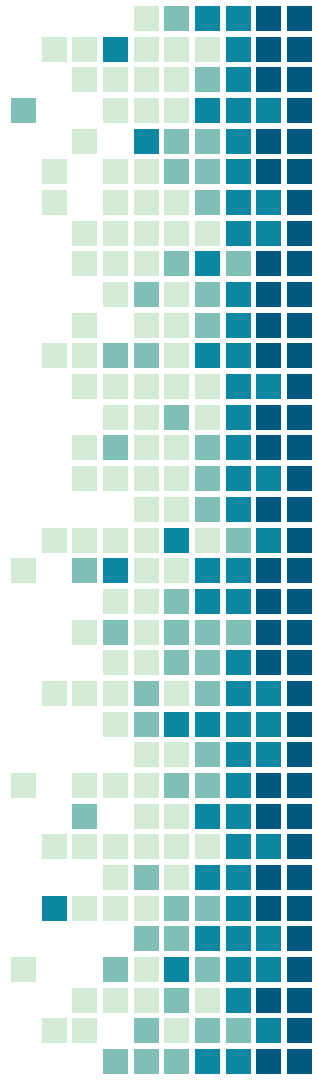
Do not leave empty spaces in between!



Welcome to the 10th Go devroom!

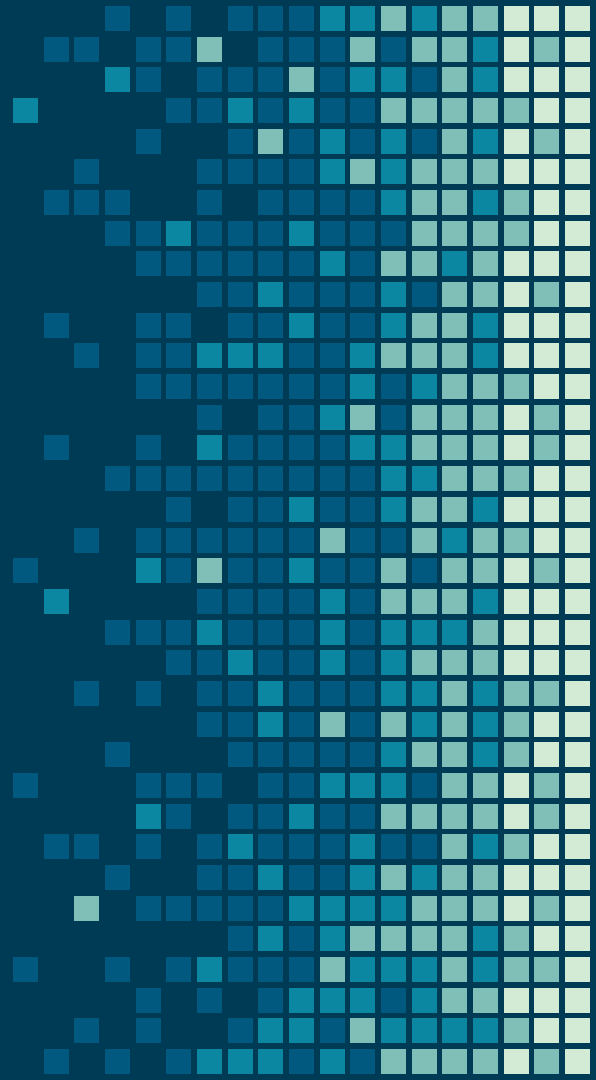


Please defragment to this side →



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



Agenda

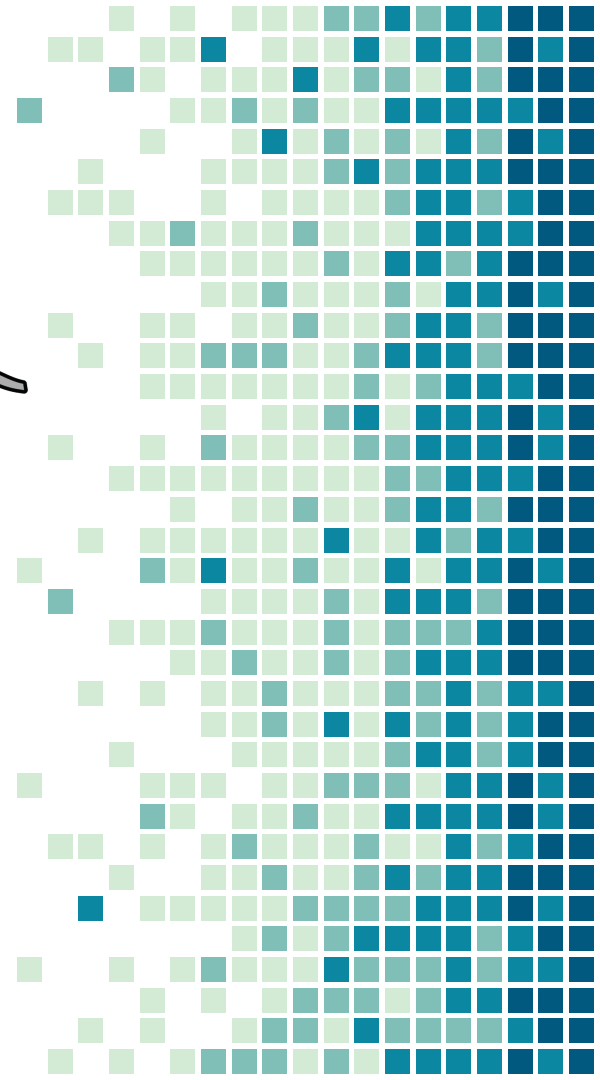
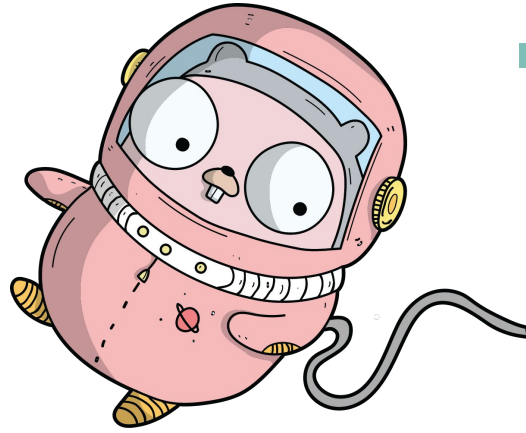
- Changes to:
 - the language
 - the standard library
 - the tooling
- Updates on design drafts
- Community update



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

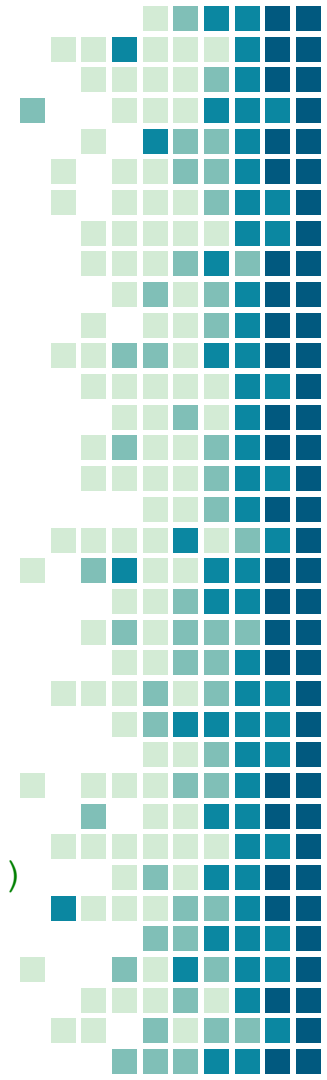
New built in functions

- `min()` and `max()` functions

```
m := min(2, 10) // m = 2
```

```
m := max(2, 10) // m = 10
```

```
f := max(0, float32(x)) // type of f is float32 (generics!)
```



New built in functions

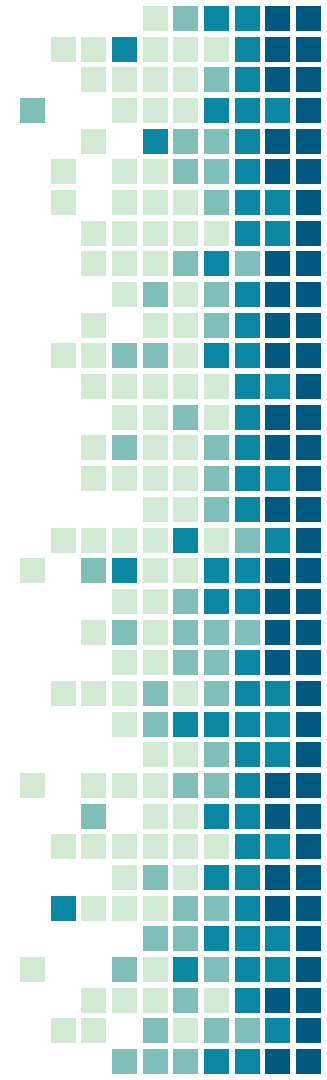
- `clear()` function
 - for slices and maps
 - sets all elements to the zero value

```
s := []int{1, 2, 3, 4, 5}
```

```
clear(s)
```

```
len(s) // 5
```

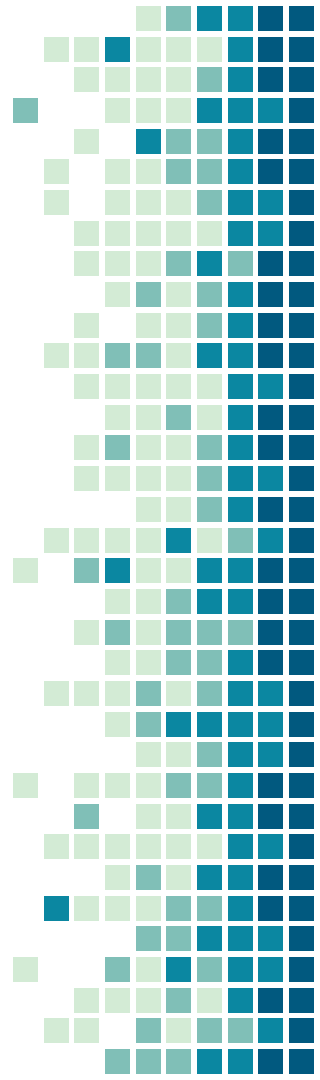
```
cap(s) // 5
```



New built in functions

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

```
s := []int{1, 2, 3, 4, 5}
clear(s)
for _, v := range s {
    println(v) // 0
}
```



Better imports

- 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

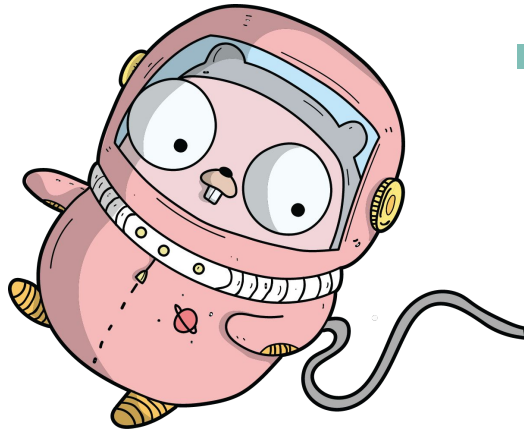


! breaking

Nil defer panic

- funcions



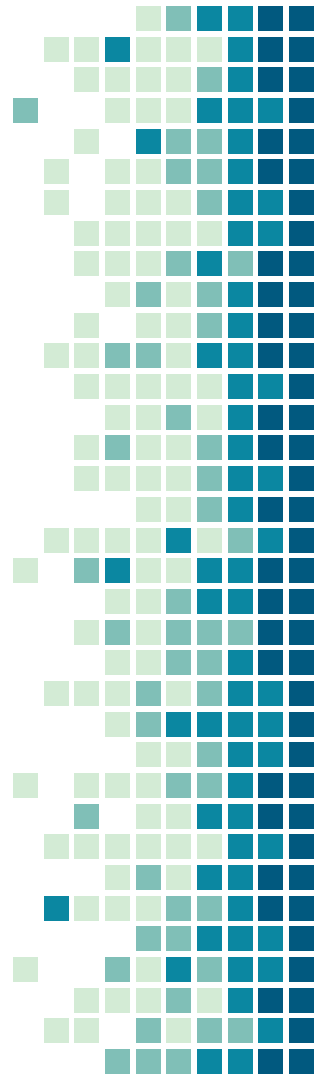
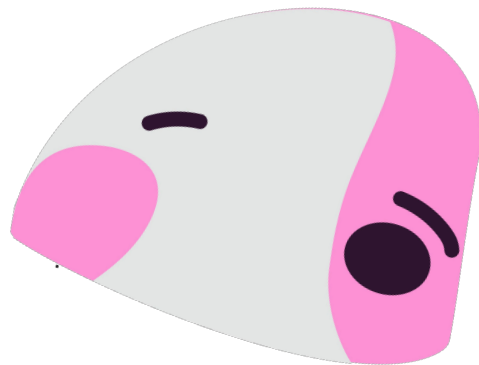


1 REVOLUTIONARY CHANGE TO THE LANGUAGE

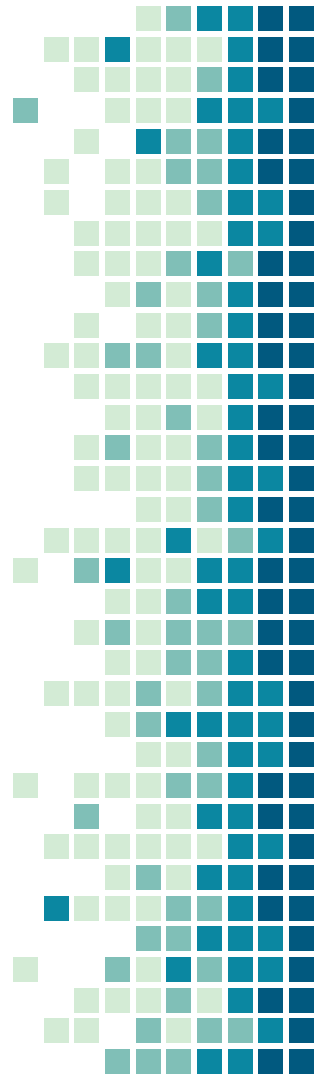
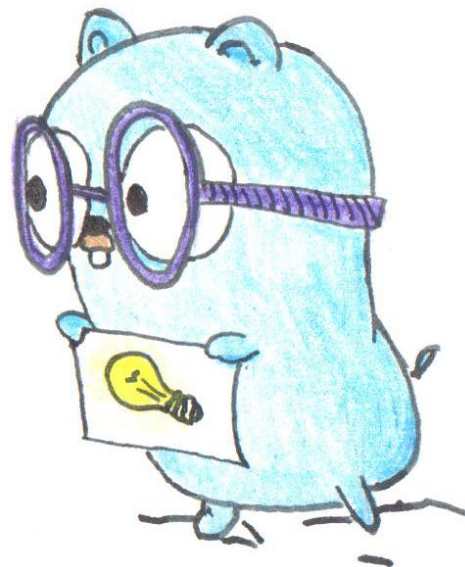
```
for i := 0; i < 10; i++ {  
    fmt.Println(i)  
}
```



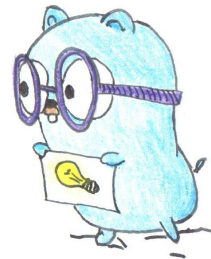
```
for i := 0; i < 10; i++ {  
    fmt.Println(i)  
}
```



```
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*

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

0

1

2

3

4

5

6

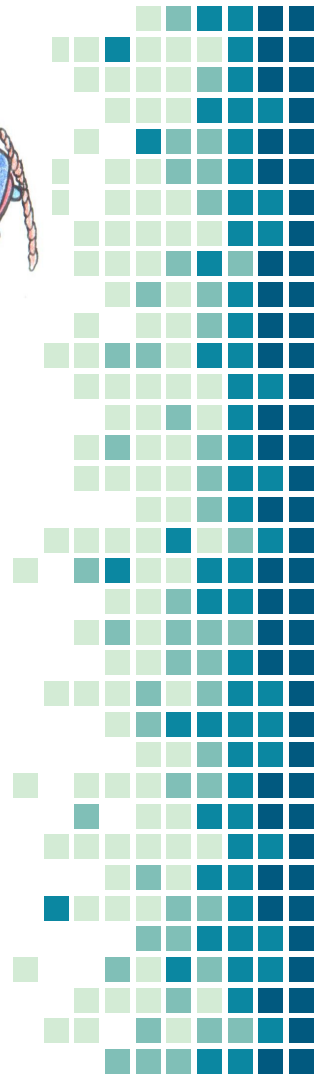
7

8

9

for loops

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



for loops



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

```
1.21 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

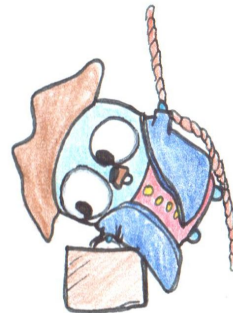
```
1.21 var zero *int

    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

1.22

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

for loops

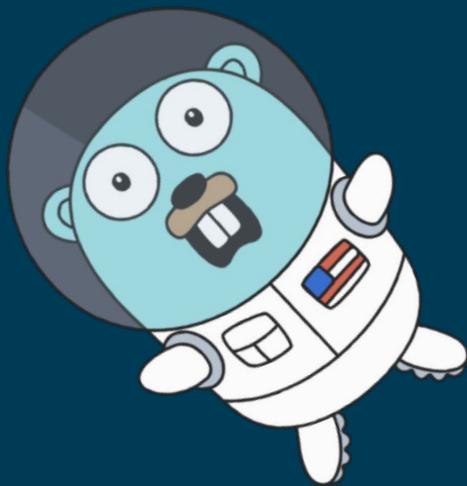


- Variables get created for each loop iteration
- Prevents value reuse bugs

1.22

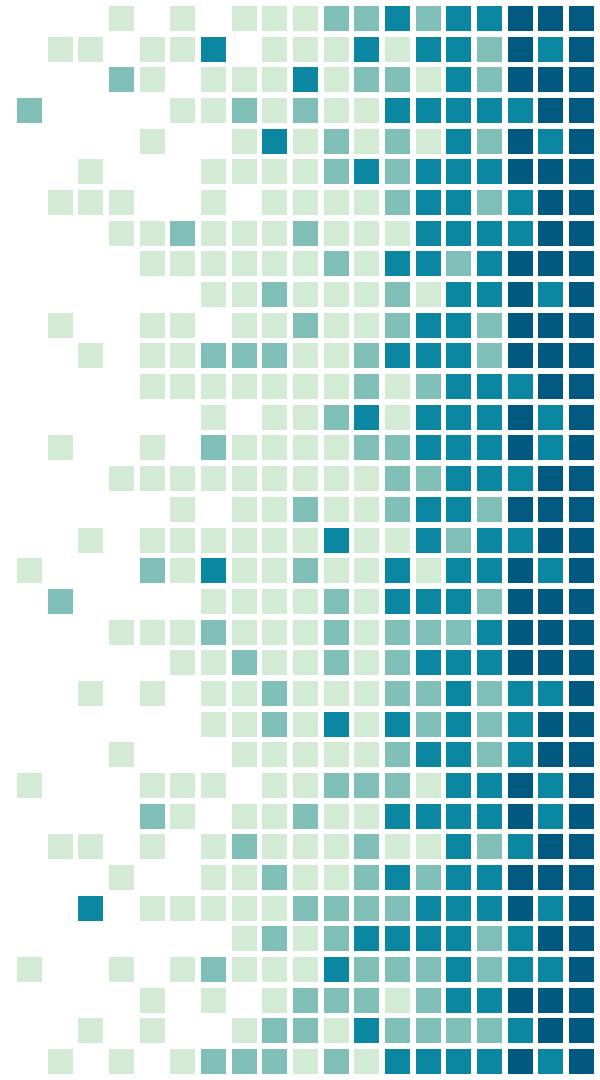
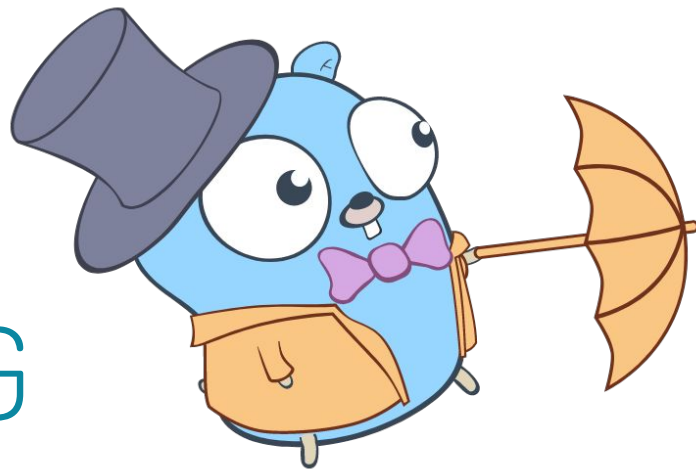
```
var zero *int
for i := range 100 {
    if i == 0 {
        zero = &i
    }
}
fmt.Println(*zero) // 0
```

! breaking

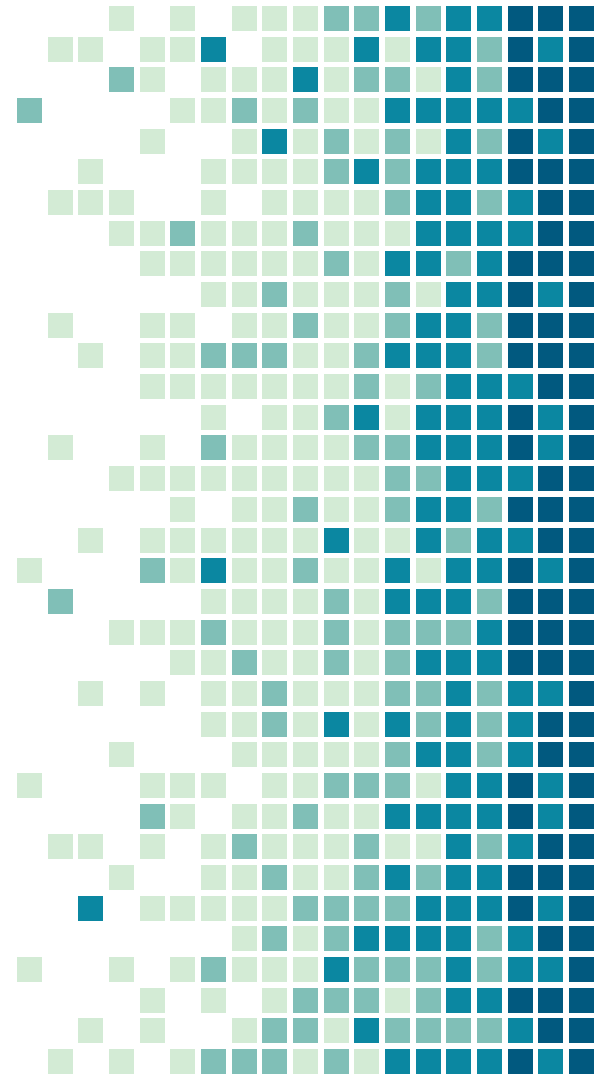
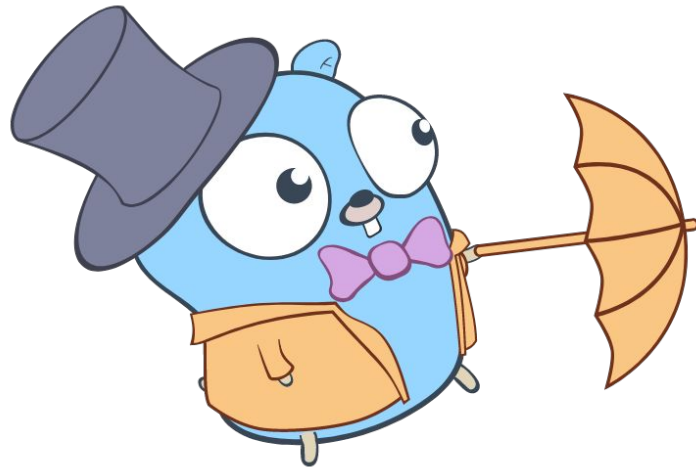


This page is intentionally left *void*

TOOLING



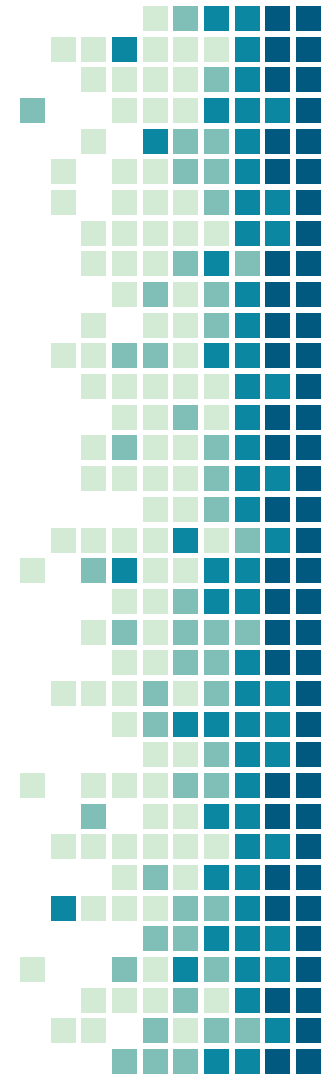
go vet



go vet: new warnings

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

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

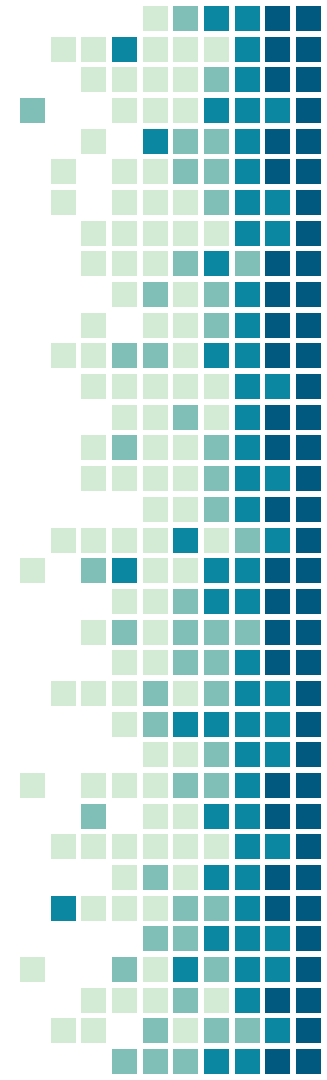


go vet: new warnings

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

```
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

```
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
}()
```



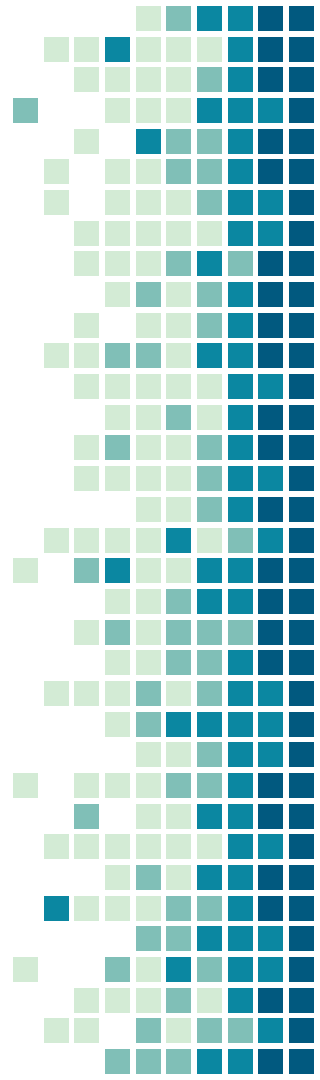
go vet: ~~new~~ removed warnings

- no longer reports references to loop variables outside the loop

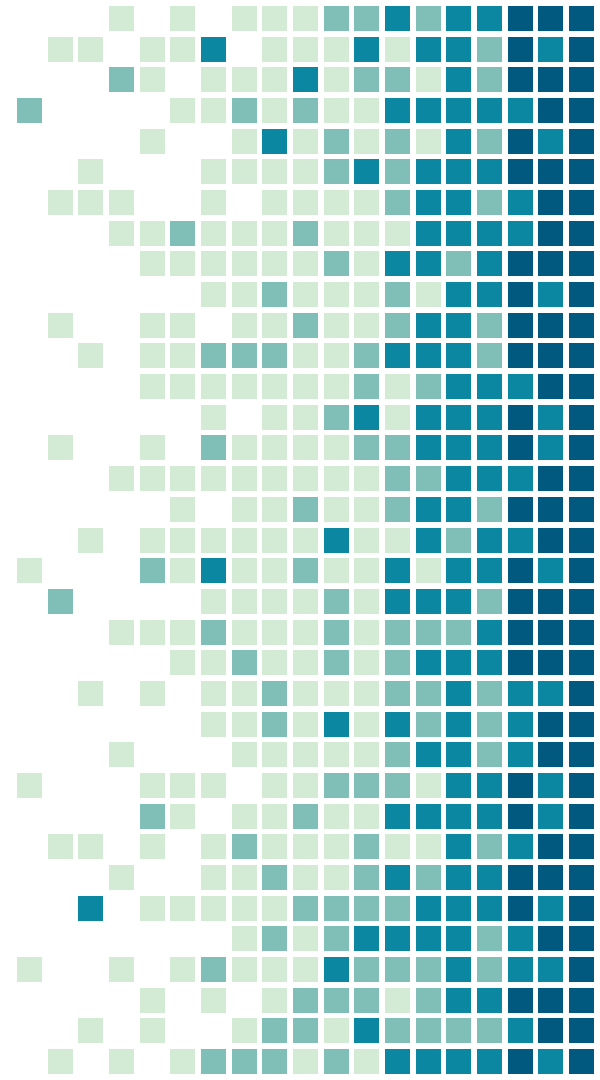
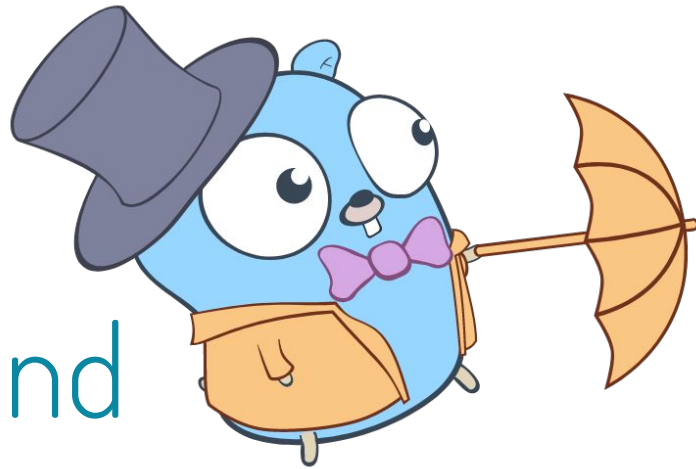
```
var zero *int

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

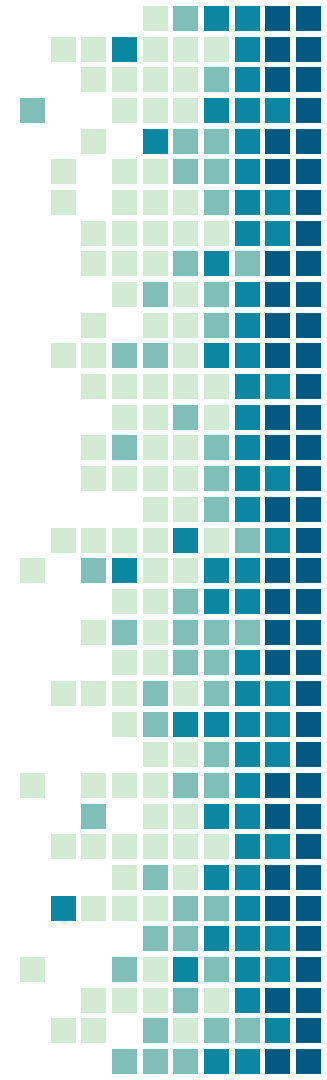
- `go test -cover` prints coverage summaries for packages without tests

1.21

```
? mymod/mypack [no test files]
```

1.22

```
mymod/mypack coverage: 0.0% of  
statements
```



Go Modules Vendoring

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



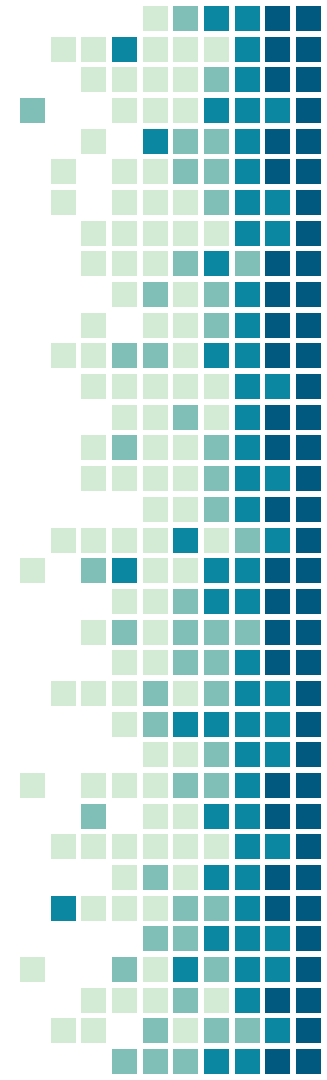
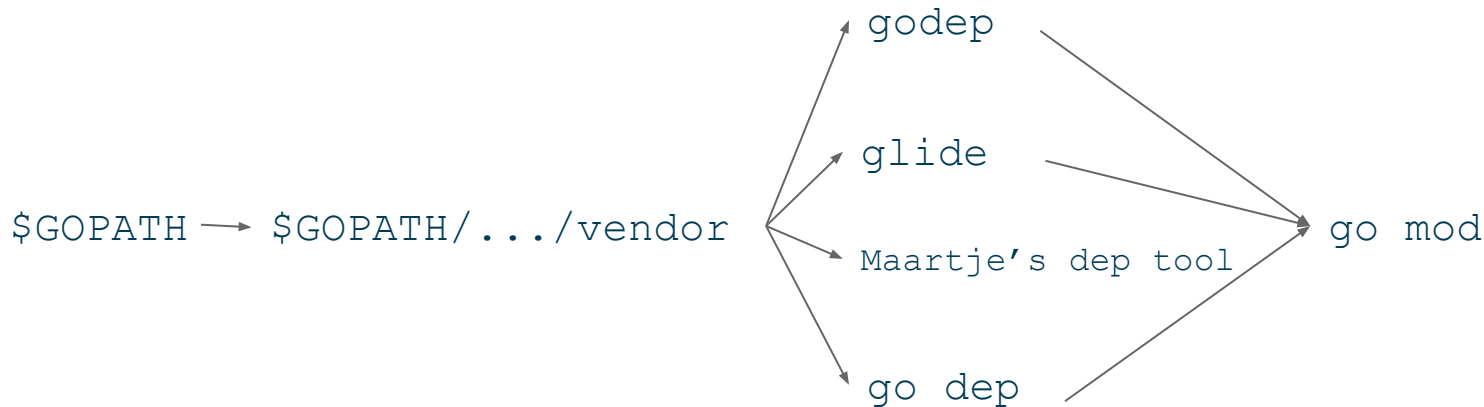
go mod

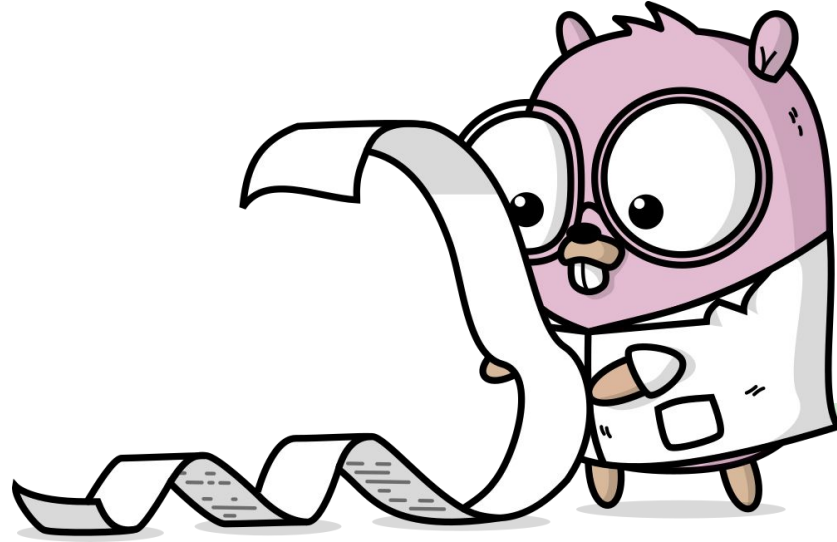
- `go mod init` no longer attempts to import module requirements from configuration files for other vendoring tools (such as `Gopkg.lock`).



Other vendoring tools??

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

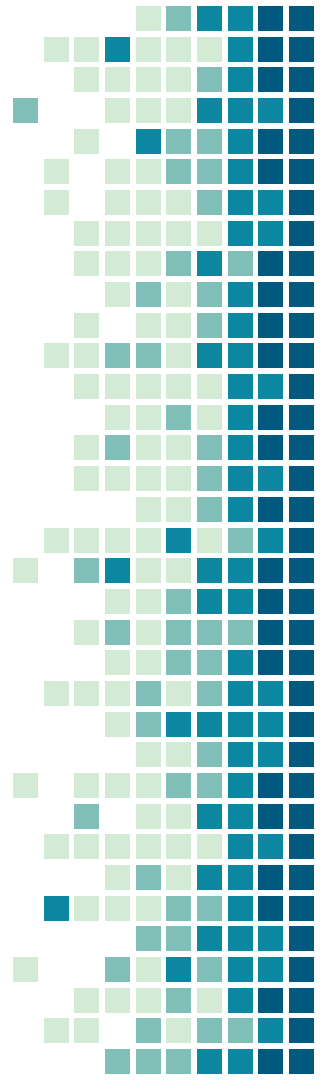




STANDARD LIBRARY

bytes

- `Buffer` type has two new methods
 - `Available() int`
 - `AvailableBuffer() []byte`
 - useful with the `Write` method to append directly



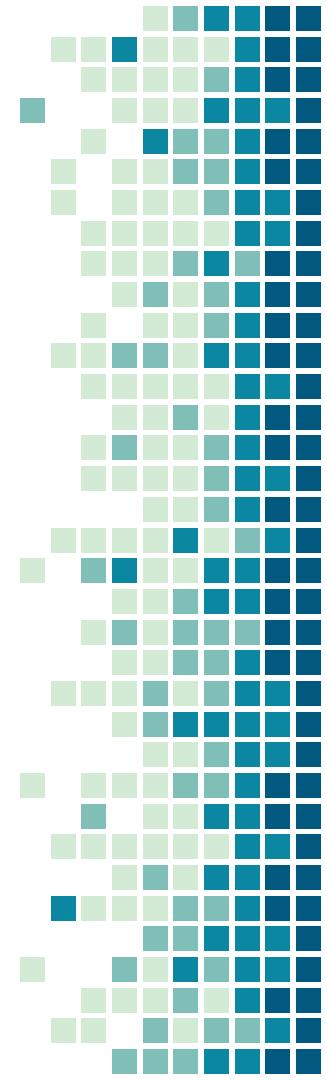
context

- `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

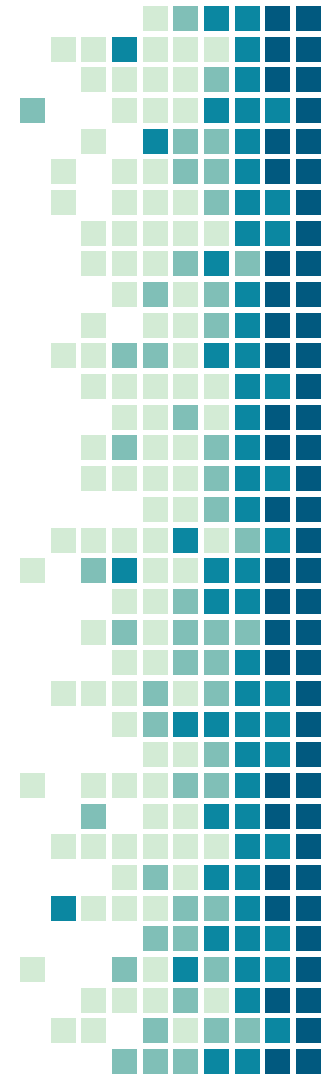
- `PublicKey.Equal` & `PrivateKey.Equal` now execute in constant time



crypto/tls

- You can now control the content of session tickets.
 - `VersionName()` returns TLS version
 - Better error messages using RFC 5246 and RFC 8446
- 1.22
- TLS1.2 is default minimum

! breaking



errors

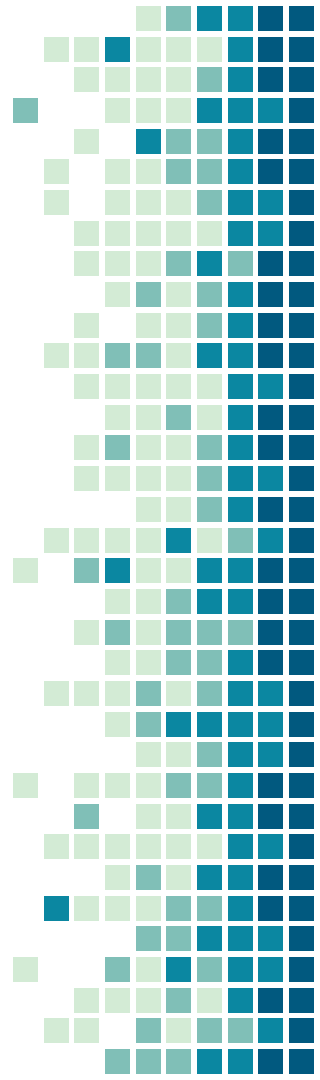
- `ErrUnsupported` a new standard error
 - HTTP 405 unsupported method
 - `os.Link()` on non hard-link filesystem



flags

 breaking

- 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



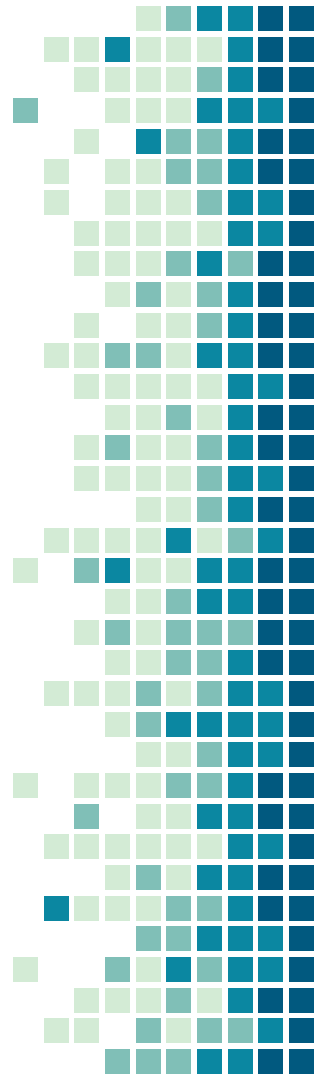
sync

- `OnceFunc()`
- `OnceValue()` & `OnceValues()`
- capture a common use of `Once()` to lazily initialize a value on first use



archive/tar&zip

- `Writer.AddFS` adds all of the files from an `fs.FS` to the archive



Enhanced HTTP routing

- net/http.ServeMux accept methods and wildcards

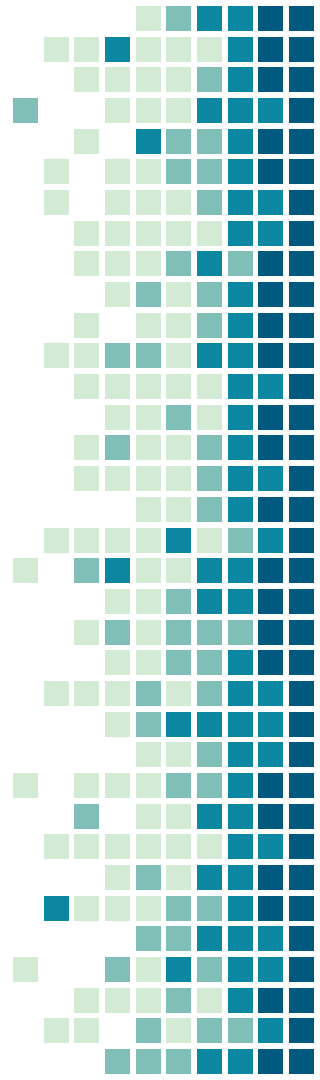
```
mux := http.NewServeMux()
mux.HandleFunc("/", func(w, r) {
    fmt.Fprintf(w, "Welcome to the Go Devroom")
})
```



Enhanced HTTP routing

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

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

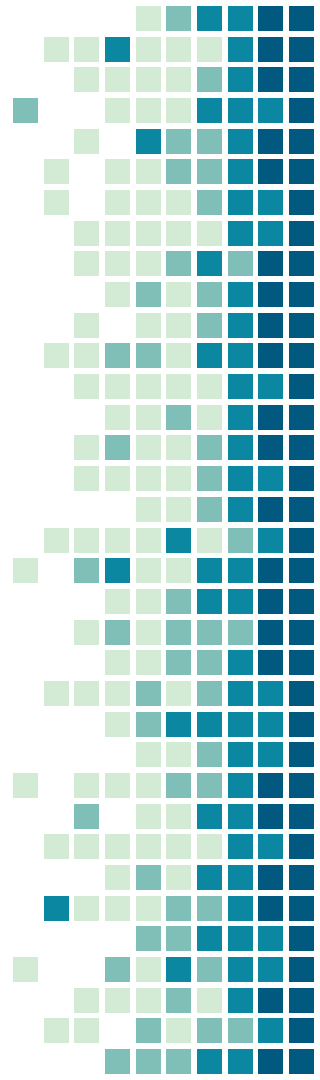


Enhanced HTTP routing

- net/http.ServeMux accept methods and wildcards

```
mux := http.NewServeMux()

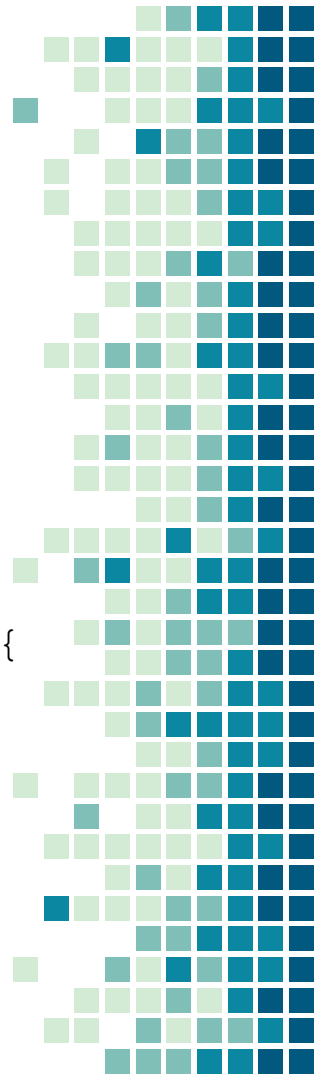
mux.HandleFunc("GET /talk/{id}", func(w, r) {
    if r.PathValue("id") == "0" {
        fmt.Fprintf(w, "State of Go")
    }
})
```



Enhanced HTTP routing

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

```
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

```
mux.HandleFunc("GET /home/", ...) // matches /home/*
```

```
mux.HandleFunc("GET /home/{$}", ...) // matches /home/
```

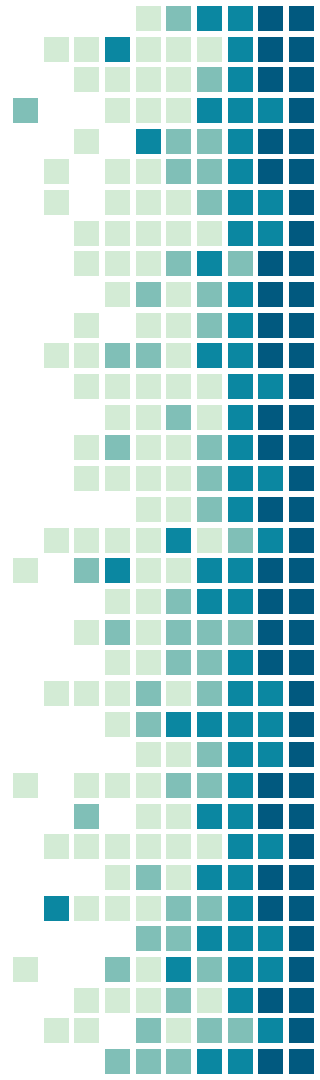


Enhanced HTTP routing

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

! breaking

- patterns with "{" and "}" behave differently

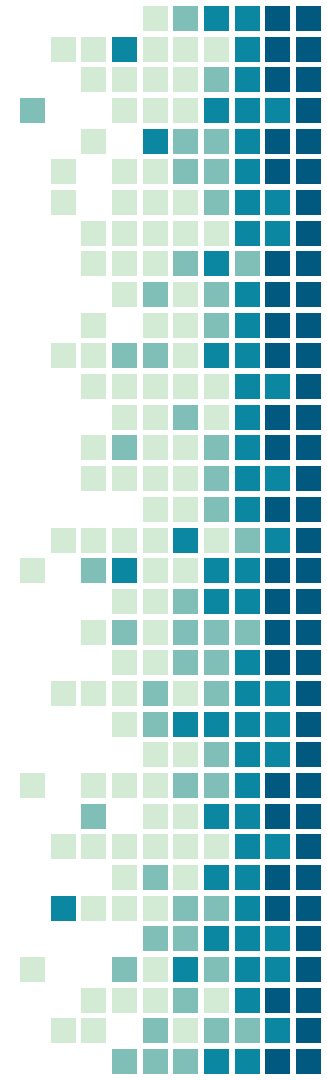


net/http

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

! breaking

- server & client reject an invalid empty `Content-Length` header





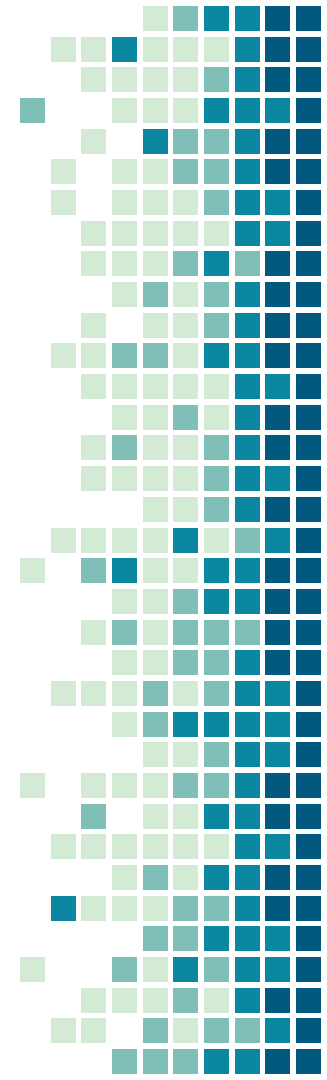
NEW PACKAGES



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

State of Go @ FOSDEM 2023

<https://go.golang.org/proposal/+master/design/56345-structured-logging.md>



Structured logging

State of Go @ FOSDEM 2023

```
import "log/slog"

func main() {
    slog.SetDefault(slog.New(slog.NewTextHandler(os.Stderr)))
    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))
}
```



Structured logging

State of Go @ FOSDEM 2023

```
import "log/slog"

func main() {
    slog.SetDefault(slog.New(slog.NewTextHandler(os.Stderr)))
    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

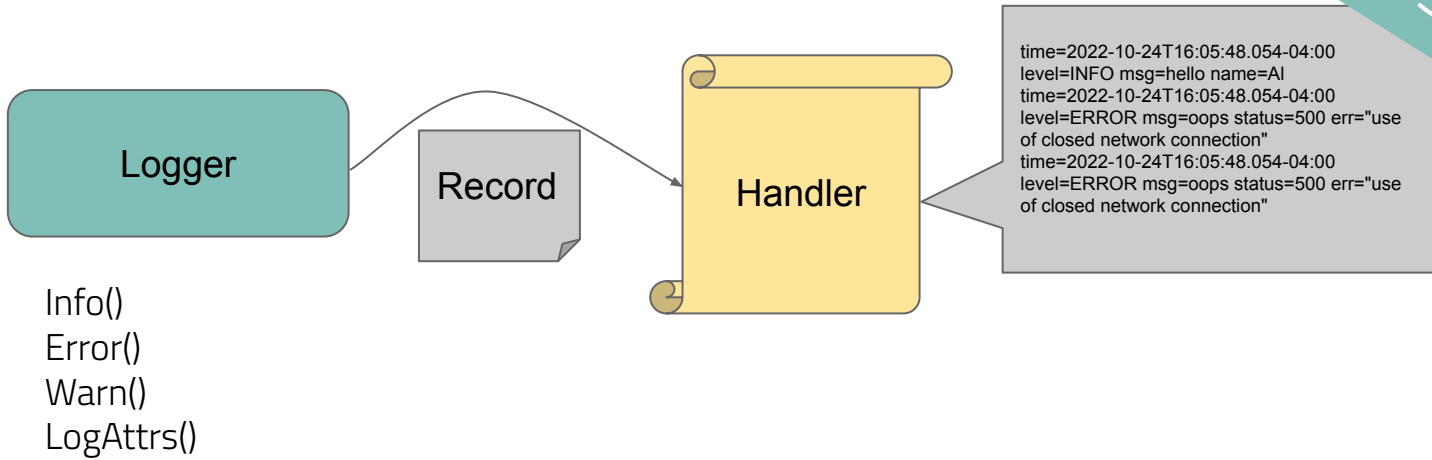
time=2022-10-24T16:05:48.054-04:00 level=ERROR msg=oops status=500 err="use of closed network connection"

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



Structured logging

State of Go @ FOSDEM 2023



```
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

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

<https://go.goglesource.com/proposal/+master/design/56345-structured-logging.md>

State of Go @ FOSDEM 2023

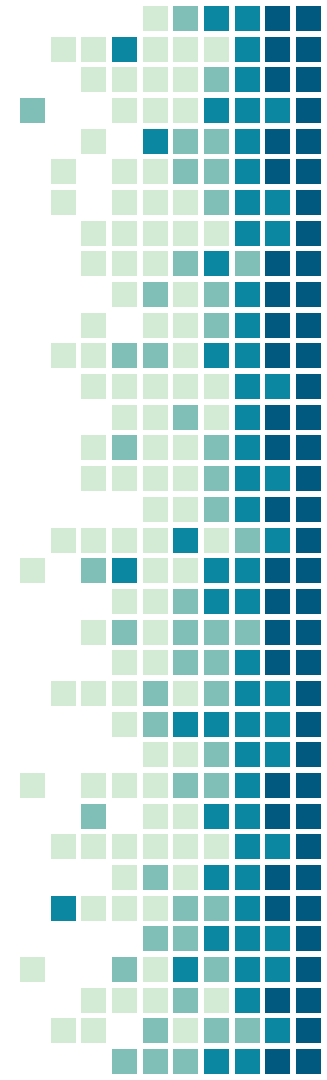




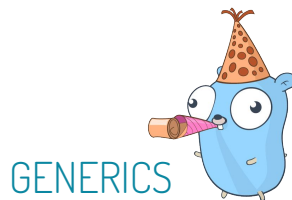
Structured logging

- <https://pkg.go.dev/testing/slogtest>

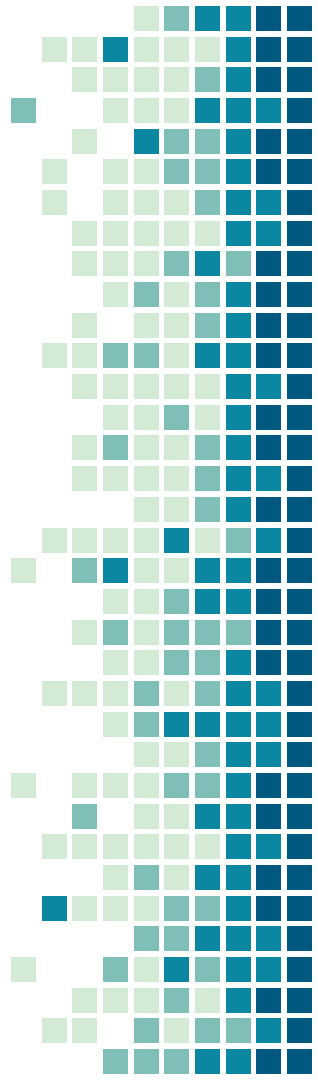
<https://go.golang.org/proposal/+master/design/56345-structured-logging.md>



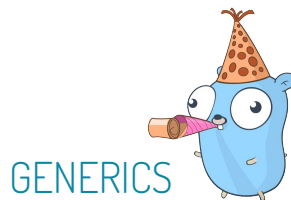
maps package



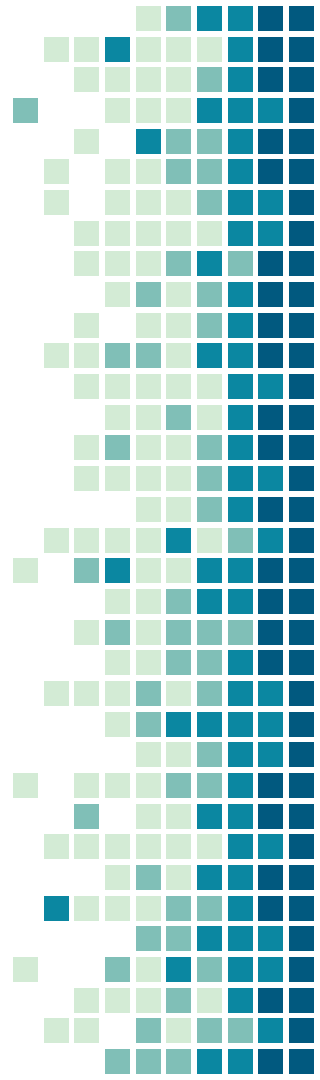
- `Clone()`
- `Copy(dst, src)`
- `DeleteFunc(m, func(K, V) bool)`
- `Equal(m1, m2) bool`
- `EqualFunc(m1, m2, func() bool) bool`
- <https://pkg.go.dev/maps>



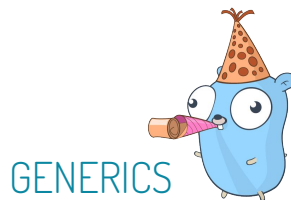
slices package



- `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



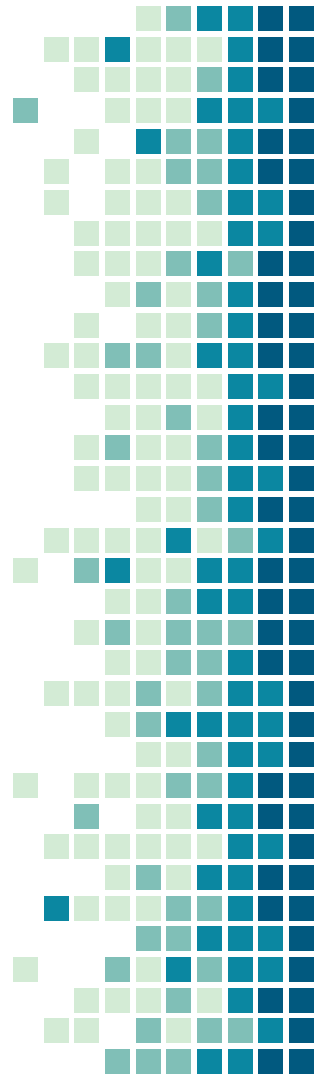
- `Compare(x, y) int`
 - -1 less, 0 equal, +1 more
- `Less(x, y) bool`

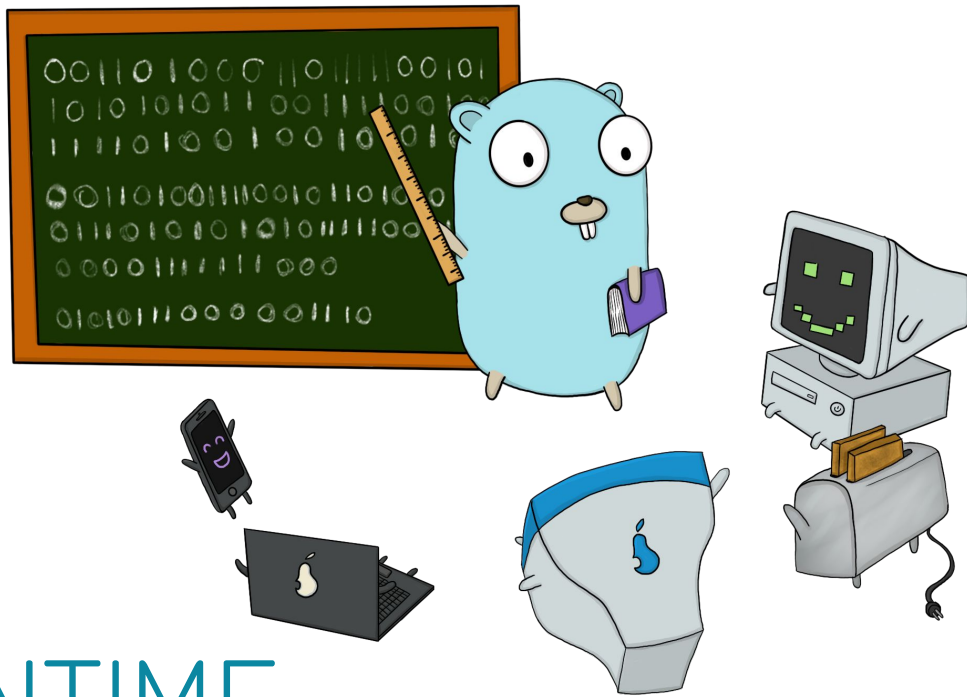
1.22

- `Or(x, y)` returns the first non zero value.
- <https://pkg.go.dev/cmp>

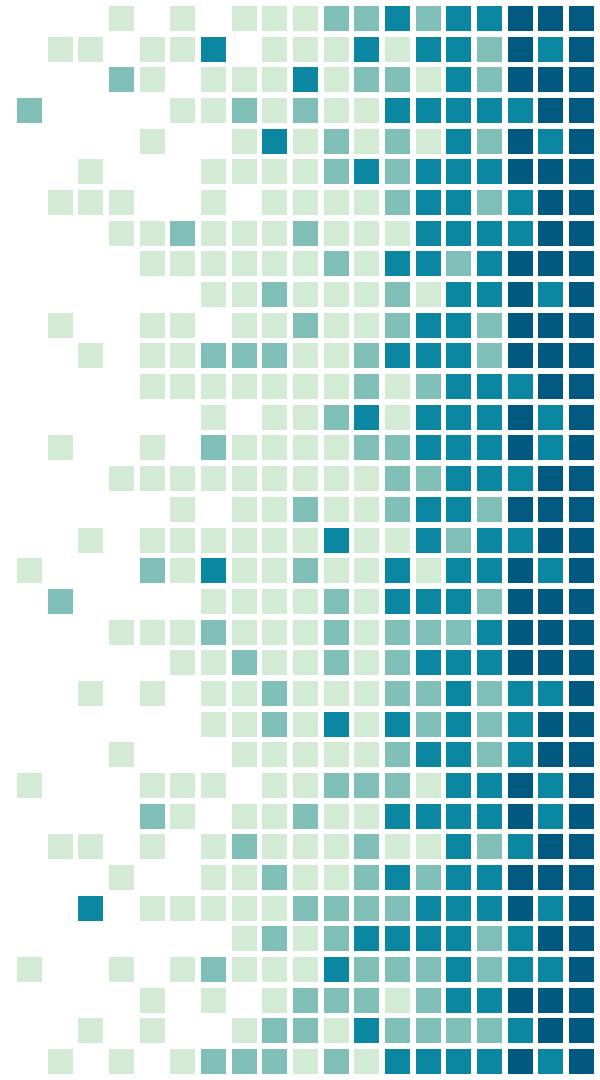
math/rand/v2

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





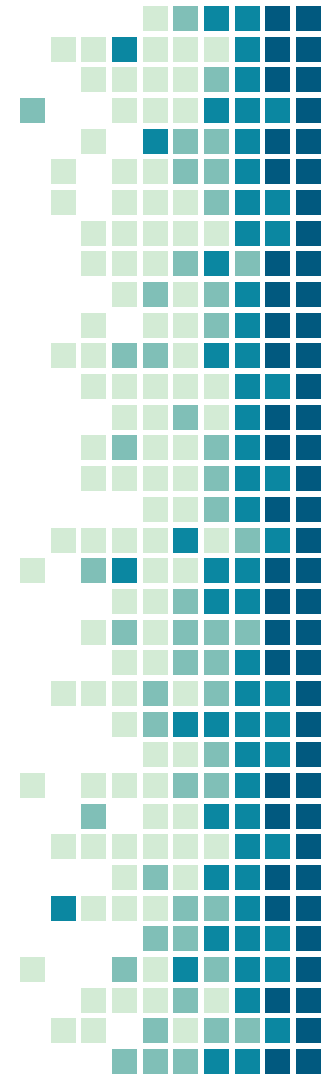
RUNTIME



runtime/trace

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

*on amd64 and arm64



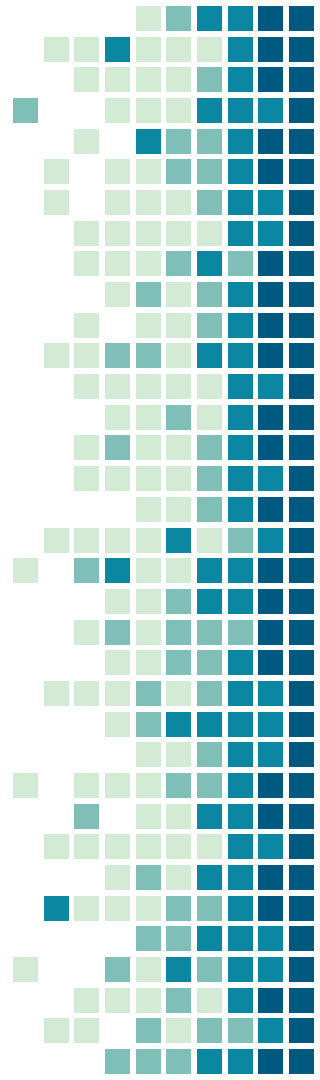
PGO

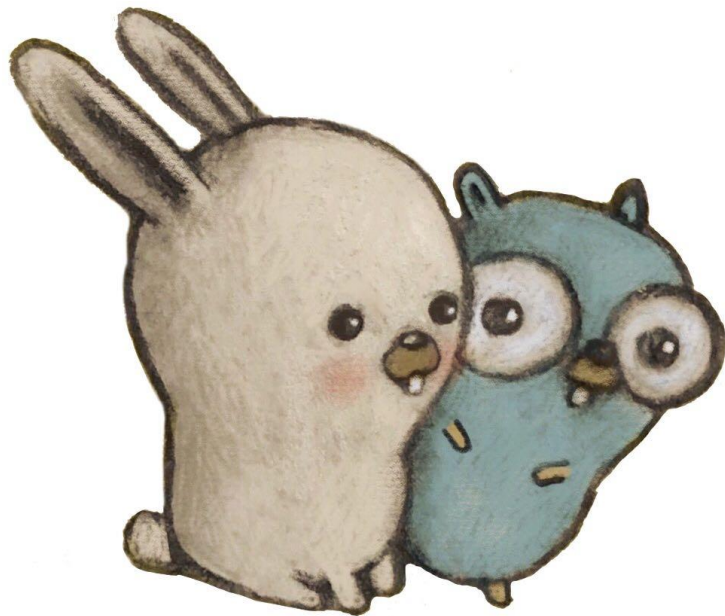
- The `-pgo` build flag now defaults to `-pgo=auto`
 - Used if `default.pgo` is found
- Build speed by up to 6%, largely thanks to building the compiler itself with PGO.



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





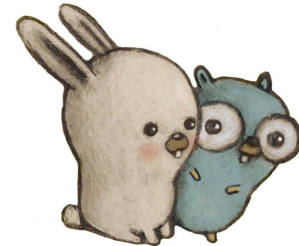
PORTS



PORTS

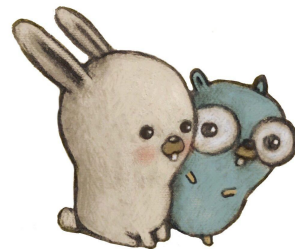
- Go1.21 requires Windows 10 (server 2016) or later
- Go 1.21 requires macOS 10.15 Catalina or later
- Go 1.22 is the last to support macOS 10.15 Catalina

- Go 1.22 adds experimental PowerPC64 support to OpenBSD



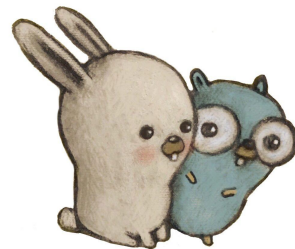
Changes to WebAssembly

- `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)



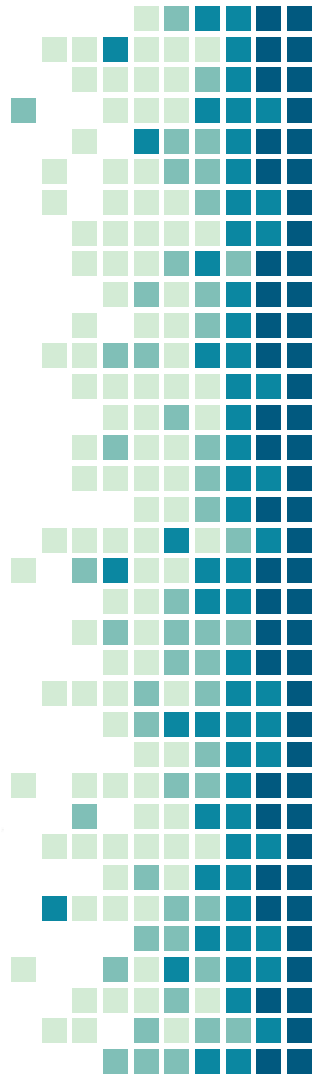
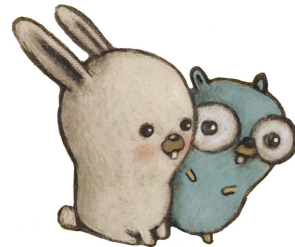
Changes to GOARM

- 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

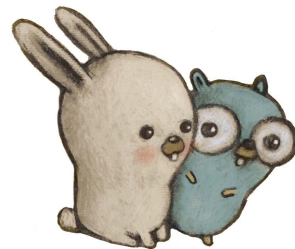
- `File.Chdir` method now changes the current directory
- `TempDir` now uses `GetTempPath2W` when available
 - security hardening measure.
- `os` package supports files with UTF-16 names which can't be represented UTF-8
- `Lstat` now resolves symbolic links

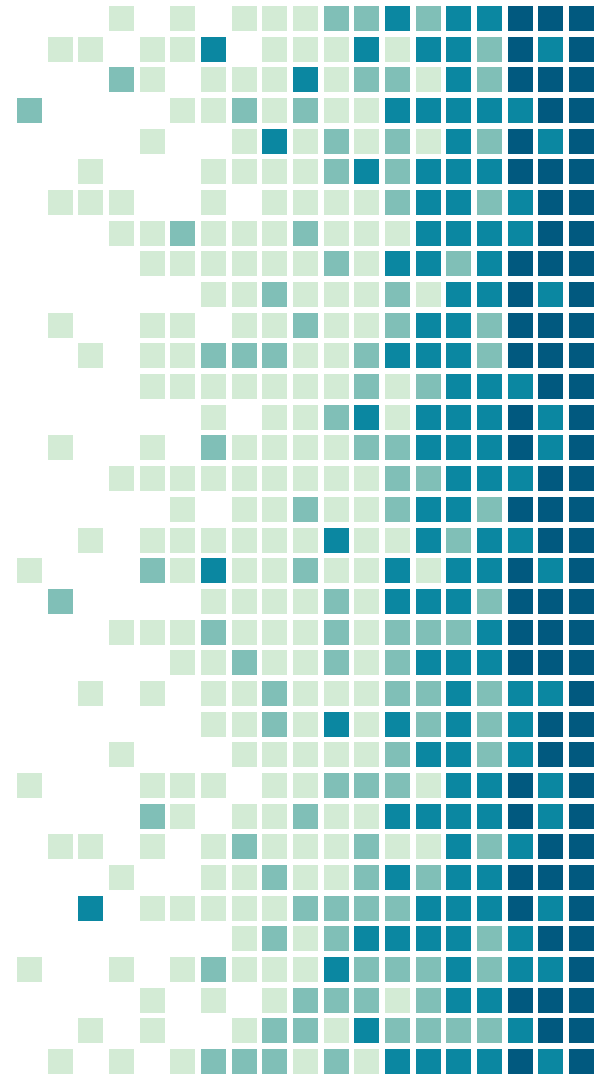
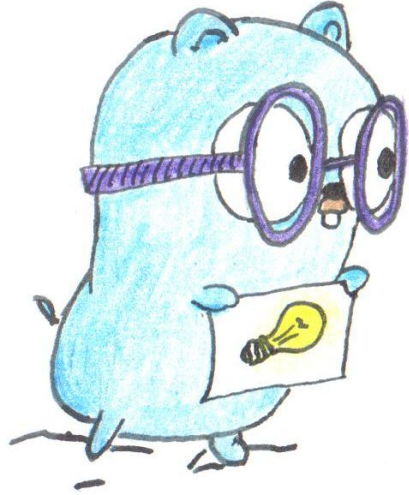


Changes to Windows



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





THE FUTURE OF GO

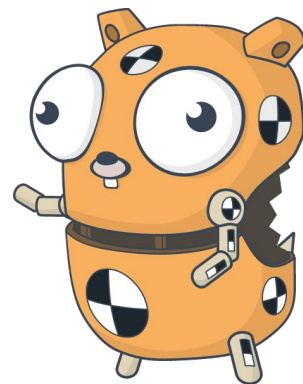
Go 2.0 and Design Drafts

Rangefunc experiment

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

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

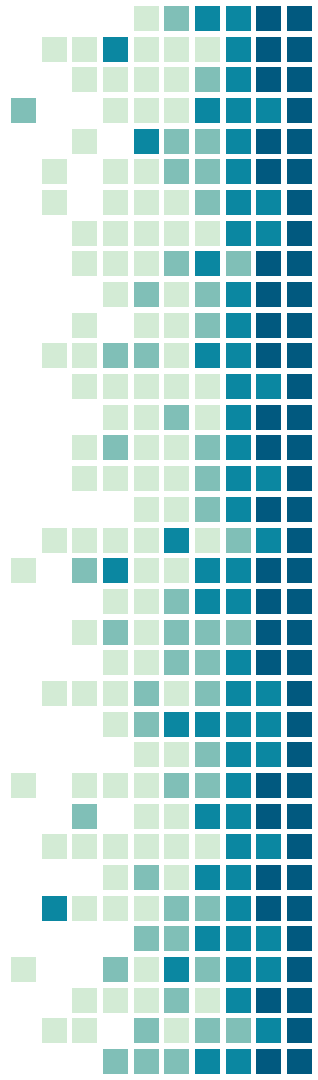
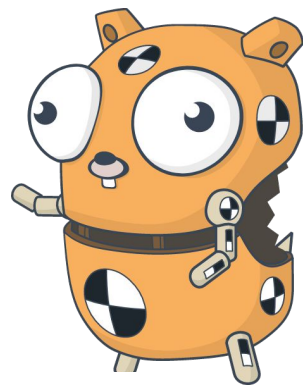
```
type Seq2[K, V any] func(yield func(K, V) bool)
```



Rangefunc experiment

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

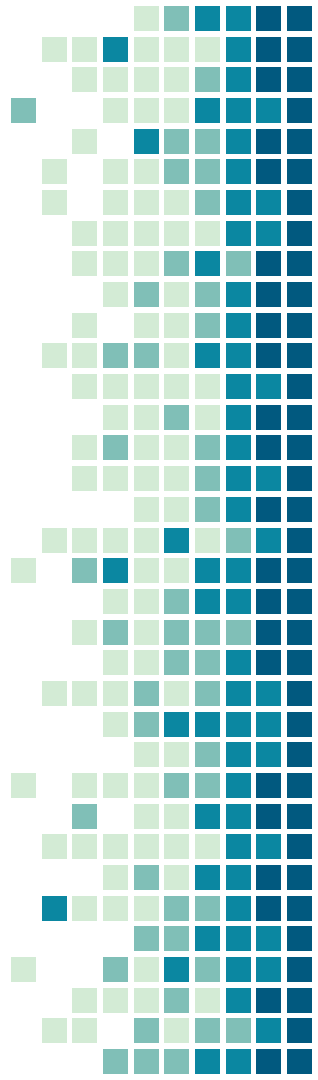
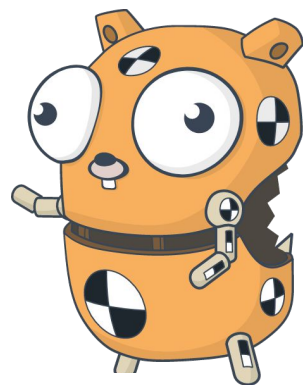
```
func MyOwnRange[E any](s []E) func(func(int, E) bool) {  
    return func(yield func(int, E) bool) {  
        for i := len(s); i < 0; i++ {  
            // insert logic here  
        }  
    }  
}  
  
for v := range MyOwnRange(data) {  
    // ...  
}
```



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
```

```
        }
```

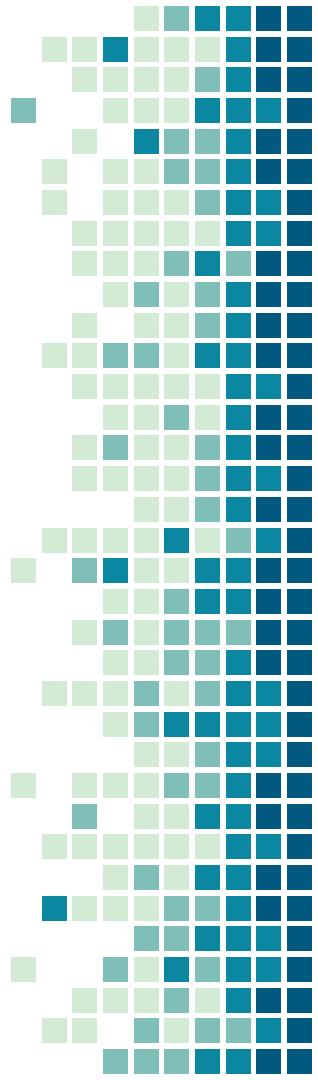
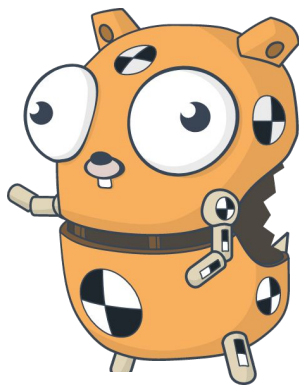
```
    }
```

```
}
```

```
for v := range MyOwnRange(data) {
```

```
    // ...
```

```
}
```



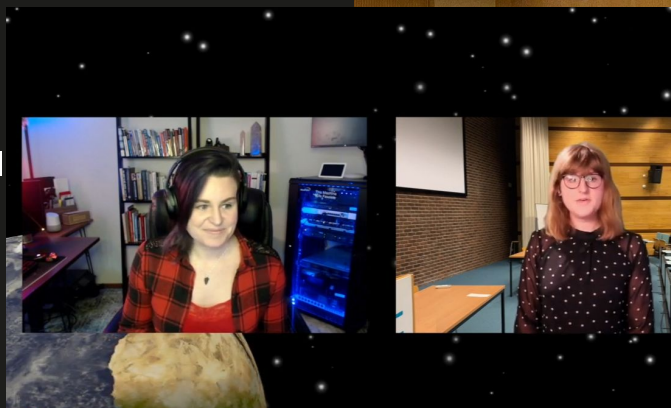


COMMUNITY





Carolyn Van Slyk



Kris Nóva



Go Developer Network on Meetup

102 000

(-25000)

Members

107

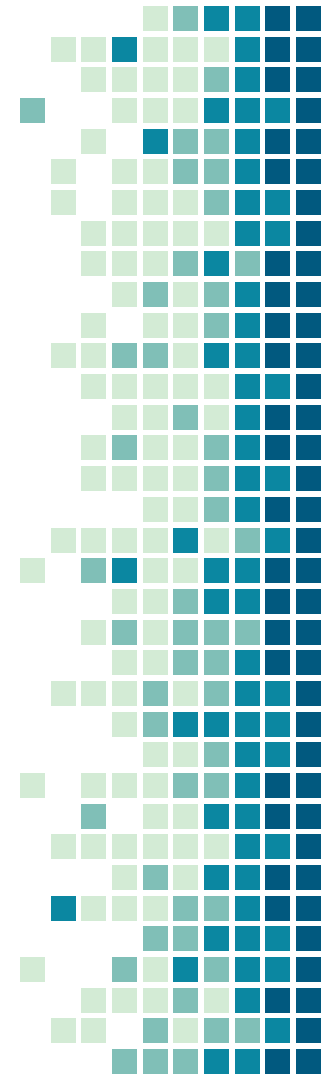
(-83)

Meetups

41

(-10)

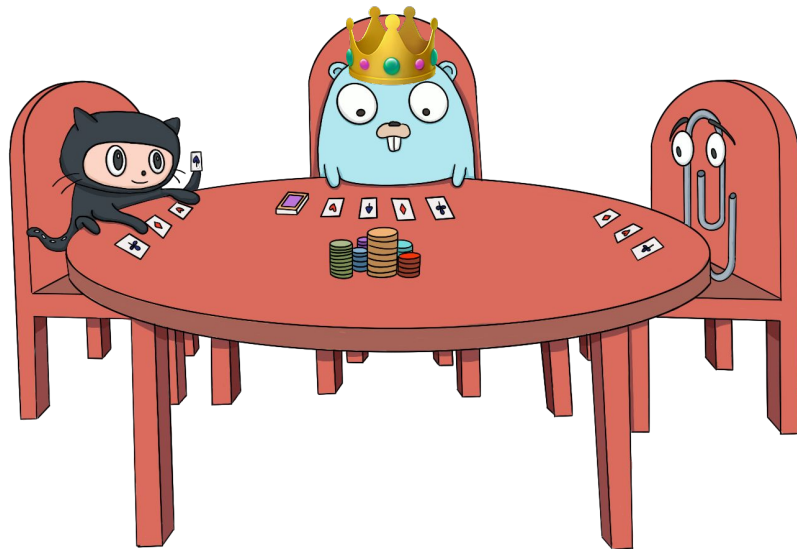
Countries



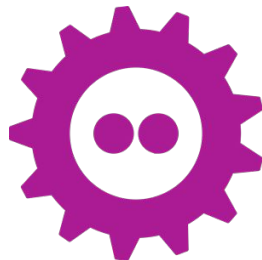
WOMEN WHO GO & GOBRIDGE MEETUPS



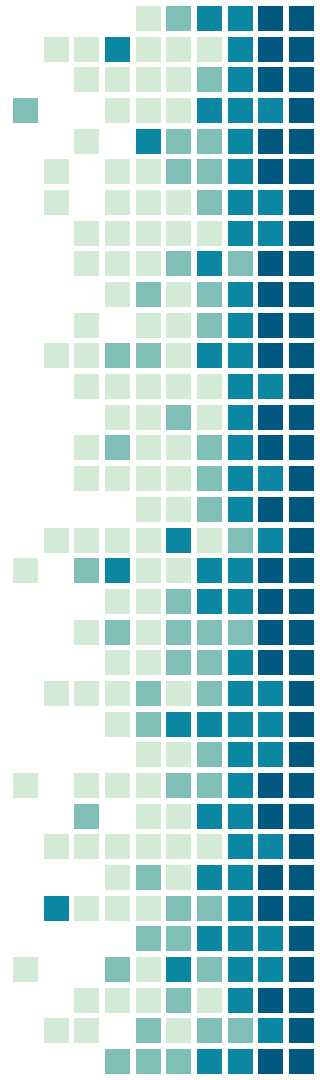
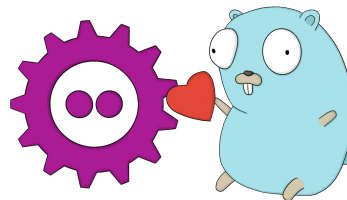
41 chapters*



FOSDEM

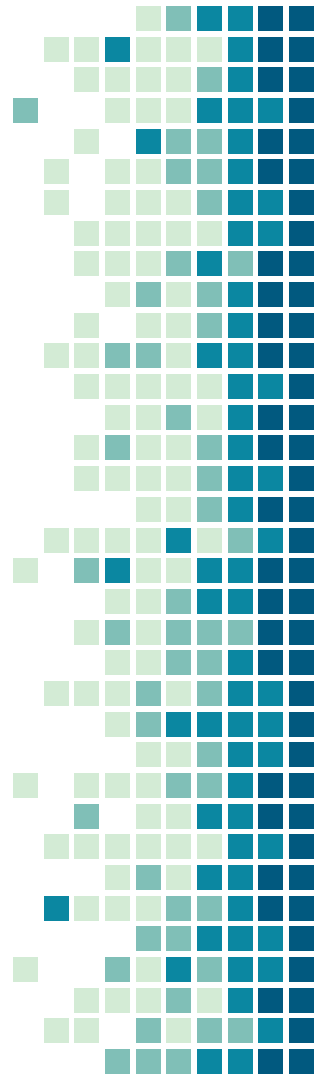
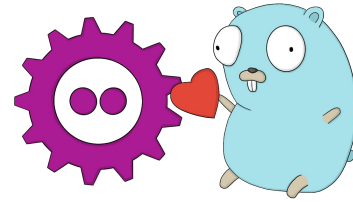


10 years ago... this happened

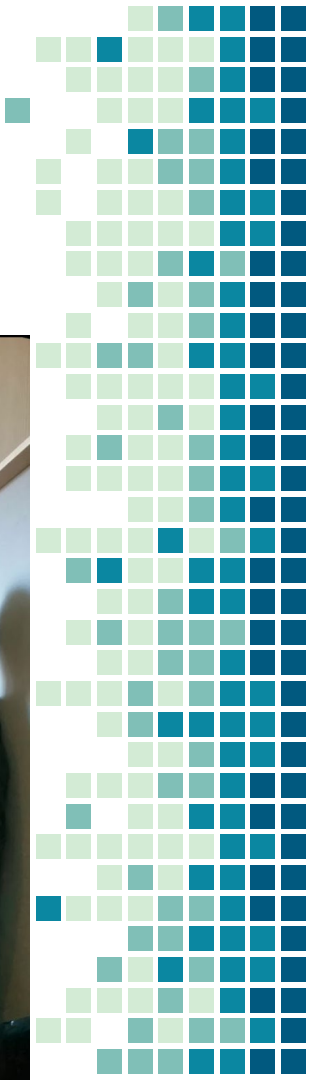
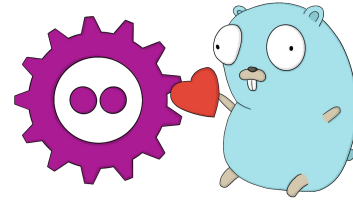


Event	Speakers	Start	End
Sunday			
Go Devroom Welcome	Andrew Gerrand	09:45	10:00
Iris Decentralized Messaging <i>Peer-to-peer based messaging for back-end service decentralization</i>	Péter Szilágyi	10:05	10:35
Camlistore <i>your personal storage system for life</i>	Brad Fitzpatrick	10:40	11:40
Interfaces: a new leaf for an old book	Matthew Cottingham	11:45	12:15
Scaling with go: Youtube's Vitess	Sugu Sougoumarane	13:00	14:00
Write your own Go compiler <i>More adventures with go.tools/ssa</i>	Elliott Stoneham	14:05	14:35
Looking toward Go 1.3 <i>What's coming in the next major release of Go</i>	Andrew Gerrand	14:40	15:10
Porting Go to New Platforms <i>Lessons learned from the Solaris port</i>	Aram Hävärmanu	15:15	16:15
Go Lightning Talks	Andrew Gerrand	16:20	17:20

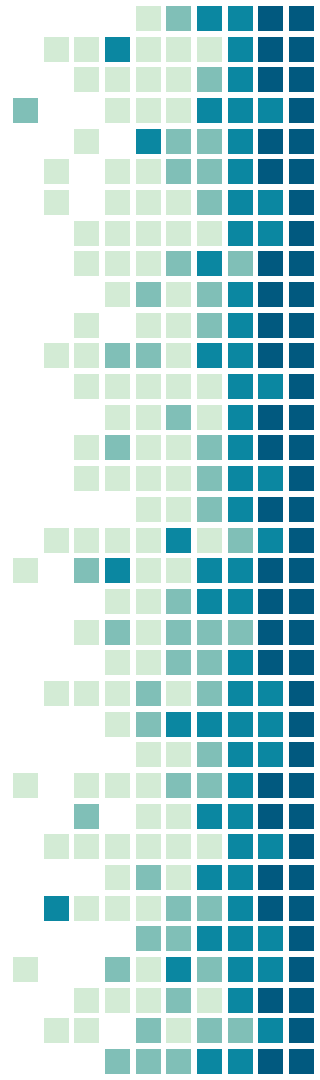
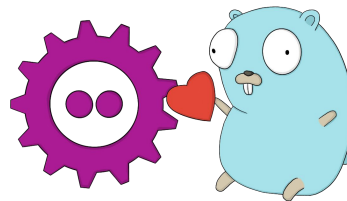
2014, a cold day in Brussels



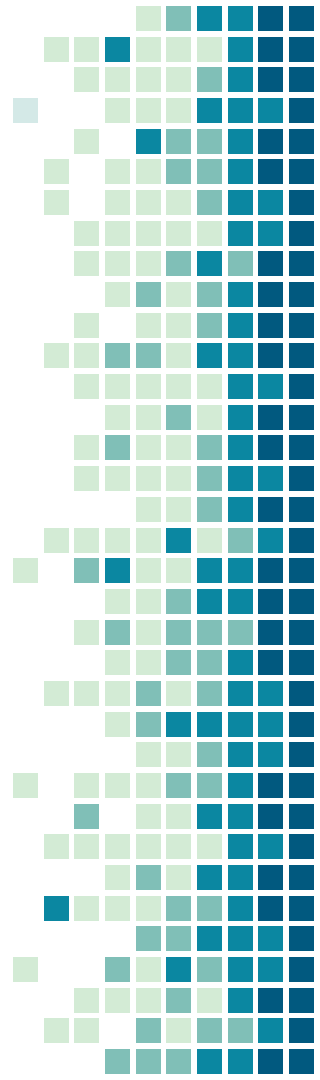
Back in 2014 (10 years!)



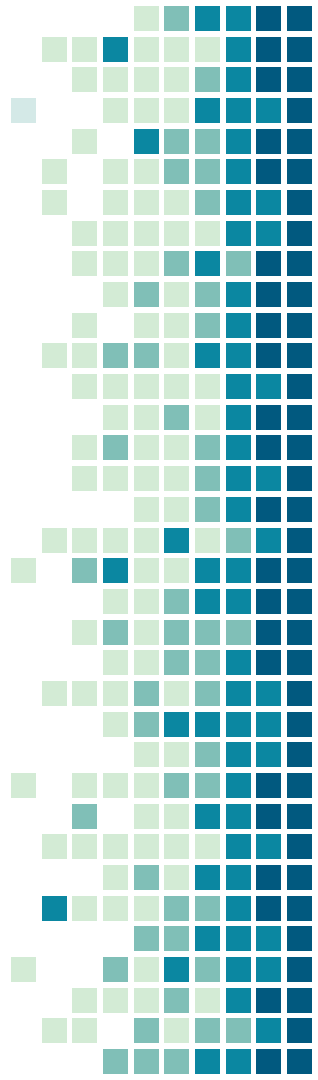
Traditions never die!



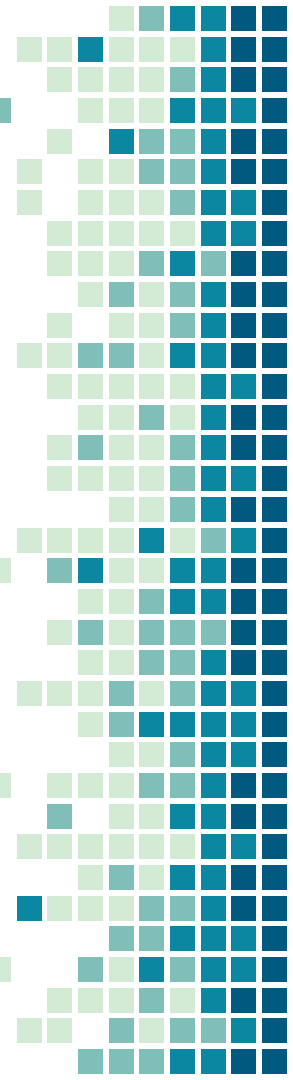
2015 ...



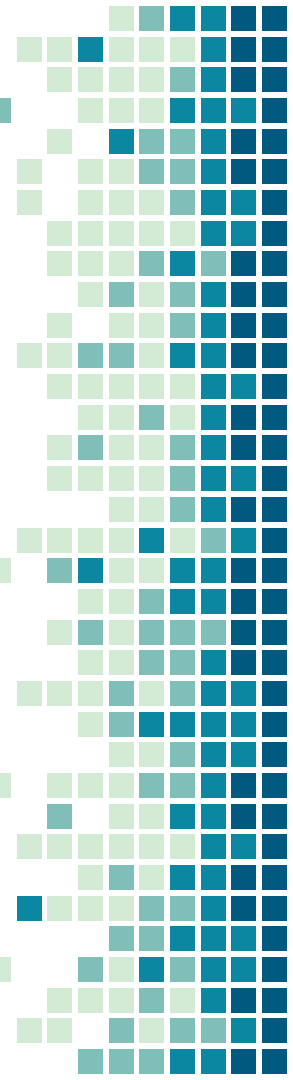
2016 ...



2017 ... find Maartje!



2017 ...



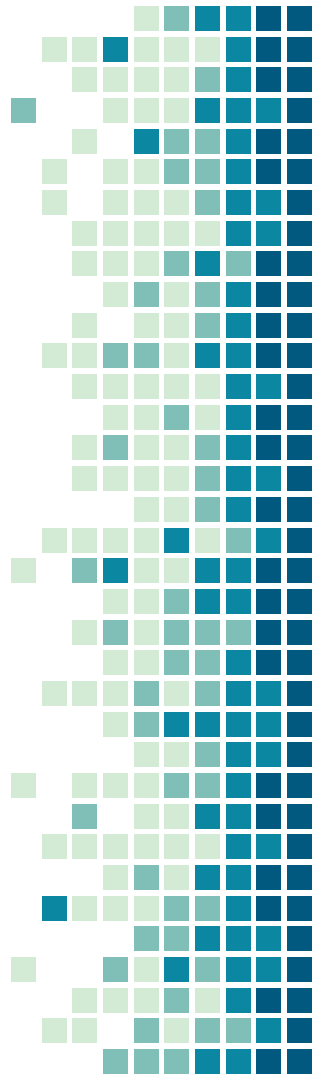
2018 ...



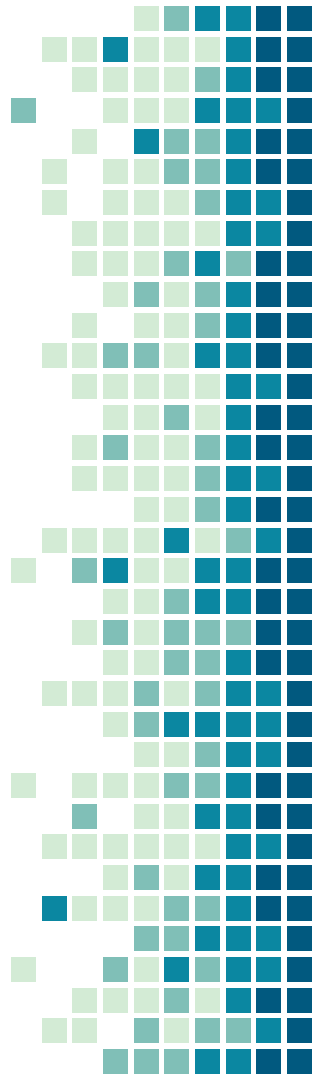
2018 ...



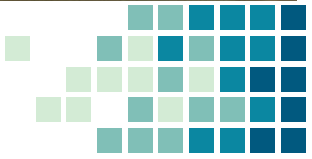
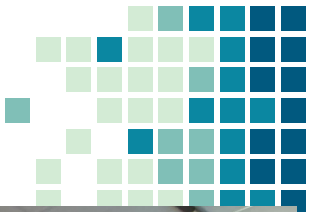
2019 ...



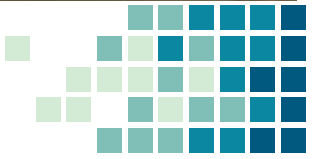
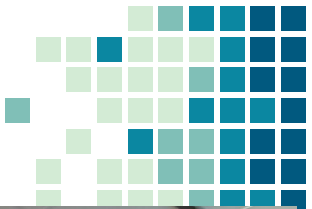
2019 ...



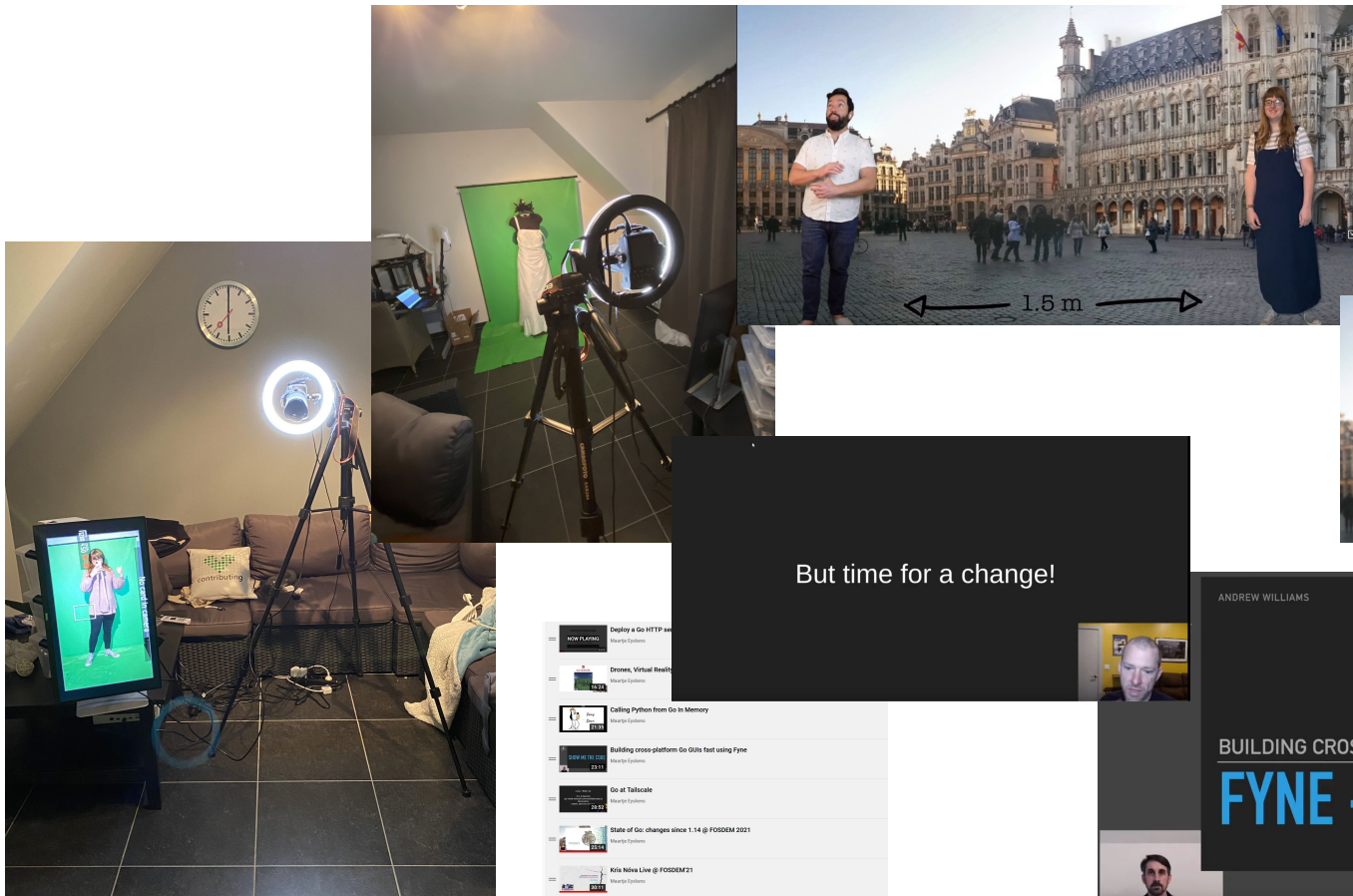
2020: “moments” before a pandemic



2020: “moments” before a pandemic



2021: first online edition



Go 1.16

io/fs

- Implementers:
 - os.DirFS
 - embed.FS
 - archive/zip.Reader
- Consumers:
 - {html,text}/template
 - net/http.Dir

But time for a change!

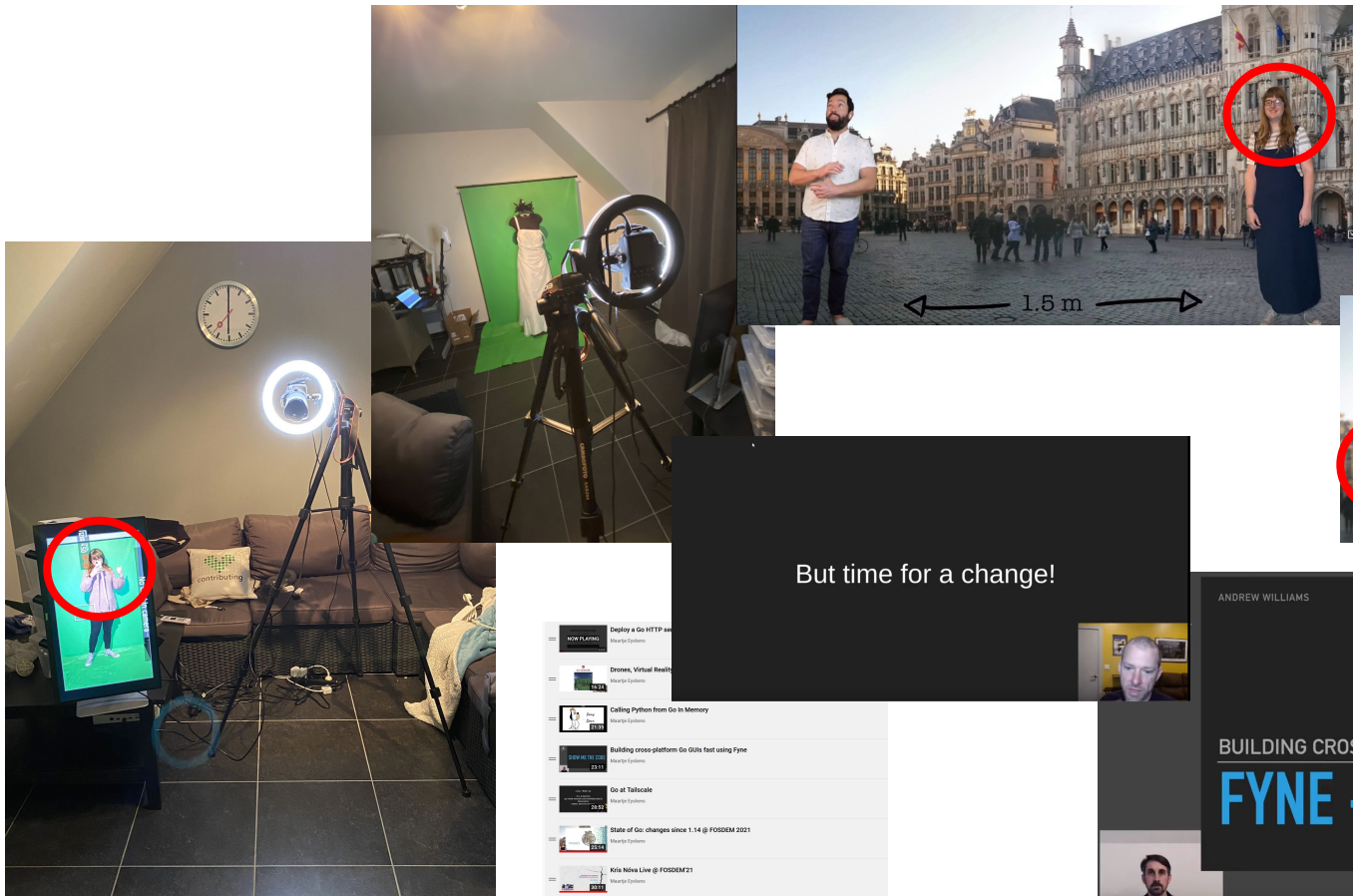
- Now Playing: Dijkstra & Go HTTP as...
- Drones, Virtual Reality...
- Calling Python from Go in Memory...
- Building cross-platform Go GUIs fast using Fyne
- Go at Tallcote
- State of Go: changes since 1.14 @ FOSDEM 2021
- Kris Niva Law @ FOSDEM21

ANDREW WILLIAMS FOSDEM 6 JANUARY 2021

BUILDING CROSS-PLATFORM GUIs FAST USING FYNE

FYNE - WHAT'S NEW?

2021: first online edition



Go 1.16

io/fs

- Implementers:
 - os.DirFS
 - embed.FS
 - archive/zip.Reader
- Consumers:
 - {html,text}/template
 - net/http.Dir

But time for a change!

- Now Playing: Dijkstra & Go HTTP as...
- Drones, Virtual Reality...
- Calling Python from Go in Memory...
- Building cross-platform Go GUIs fast using Fyne
- Go at Tallcote
- State of Go: changes since 1.14 @ FOSDEM 2021
- Kris Niva Lav @ FOSDEM21

ANDREW WILLIAMS FOSDEM 6 JANUARY 2021

BUILDING CROSS-PLATFORM GUIs FAST USING FYNE

FYNE - WHAT'S NEW?

2022: online, everywhere



Gophers around the world

Go Devroom @ FOSDEM 2022

```
Size capacity: 8  
Stored data: [1,0,3,4,6,0,0,0,]  
Array Memory address: 0xc0001a548  
Size length: 4  
Size capacity: 4  
Stored data: [0,3,4,6,0,0,0,]
```

Go 1.18

what are the generics?

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

Why Go 1.18?

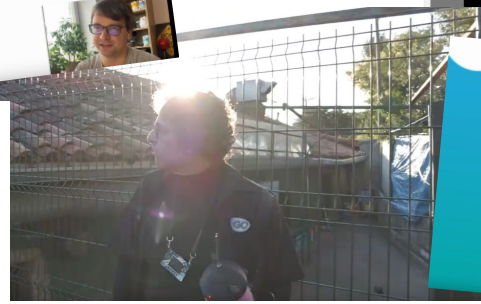
Fuzzing

Generics

Migrating from PHP 7.2.x to PHP 7.3.x

Control flow structures will now generate a warning. In PHP such continue they behave as continue 2 in other languages.

```
package main  
import ("crypto/sha256")  
  
func main() {  
    var msg [1024]byte  
    data := []byte(msg[:])  
    var hash [32]byte  
    sha256.Sum256(data, hash[:])  
    sha256.Sum256(msg, hash[:])  
}
```

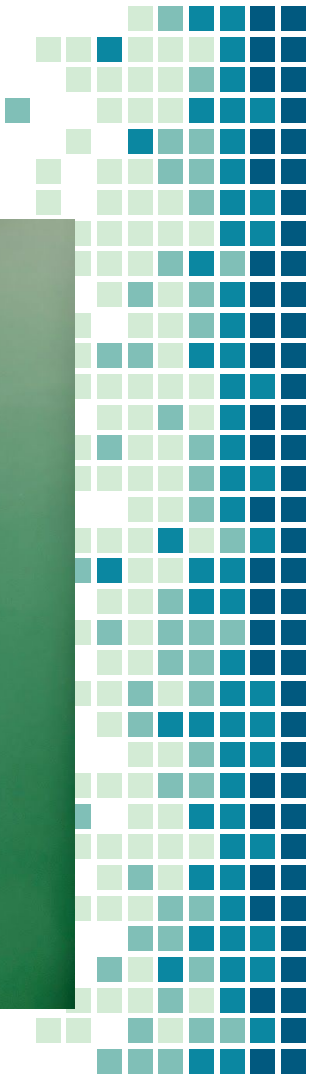


Overview

- Who am I?
- Why am I talking about this?
- Automa-what?
- How do regexps work?
- Where did I speed them up?



2023

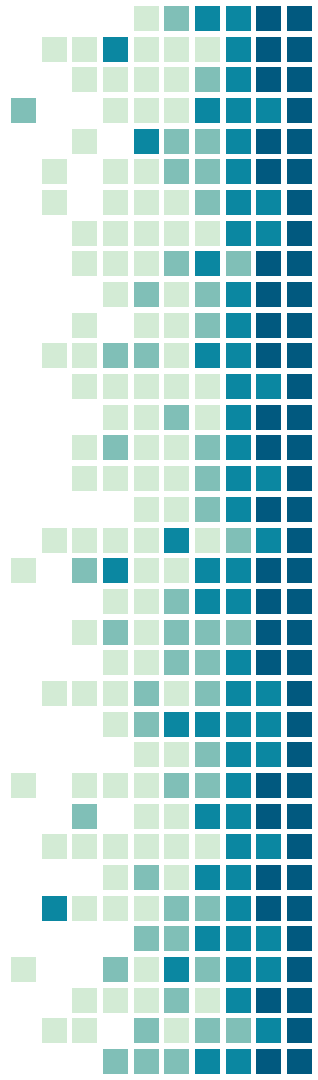


2024

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




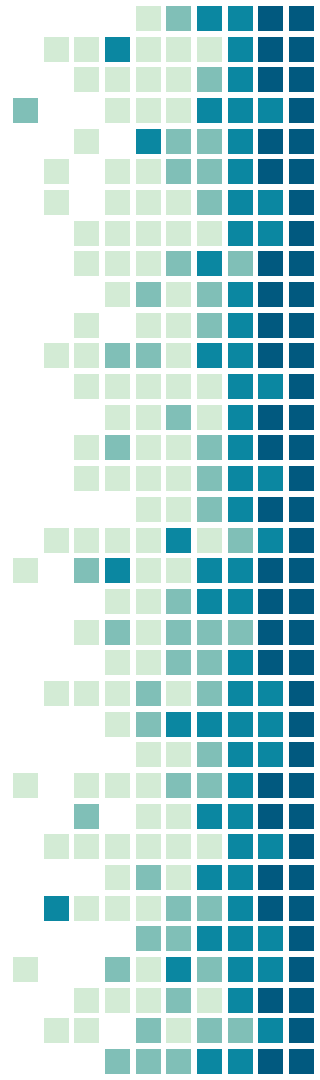
Conferences

- 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



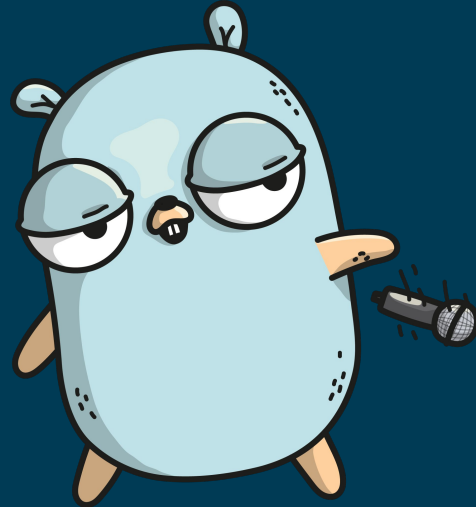
2024: Happy Birthday!

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



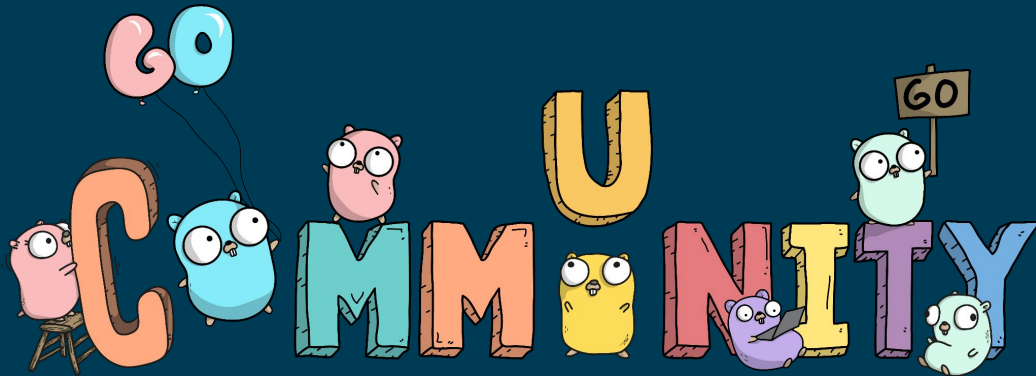
Lightning talks

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

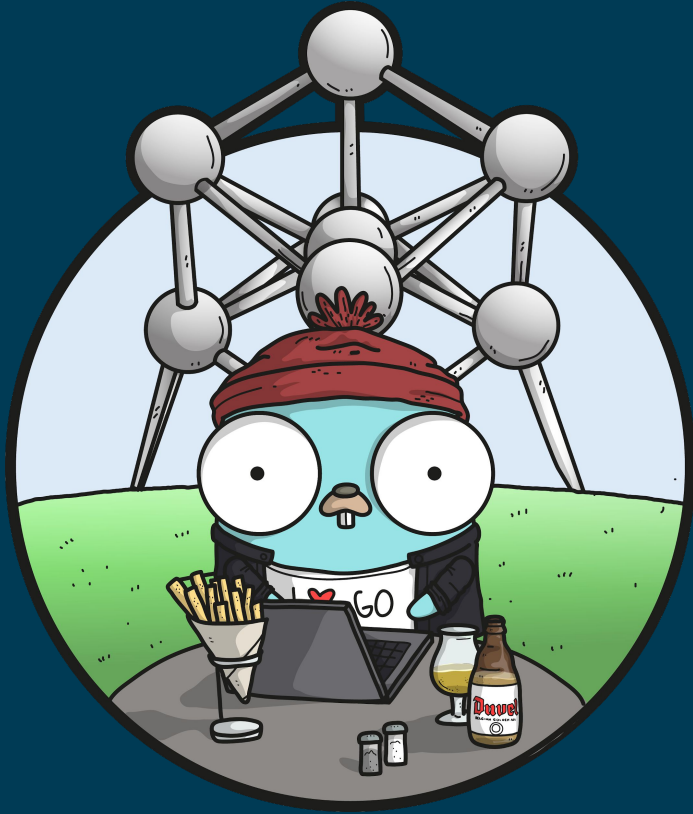


Social Media

- #golang #FOSDEM
- @godevroom@fosstodon.org on the fediverse



THANKS!
Enjoy all talks!



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

110 Glenda by Renee French