

# One Does Not Simply Iterate over Mordor

---

GopherC@N 2025  
*Israel*

# miki @tebeka



CEO,  
CTO,  
UFO



GopherCON  
Israel

Platinum  
Programmer

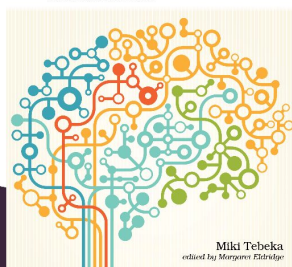
Effective Go Recipes  
Fast Solutions to Common Tasks



Miki Tebeka  
edited by Margaret Eshelby

Platinum  
Programmer

Go Brain Teasers  
Exercise Your Mind



Miki Tebeka  
edited by Margaret Eshelby

Platinum  
Programmer

Python Brain Teasers  
Exercise Your Mind



Miki Tebeka  
edited by Margaret Eshelby

Platinum  
Programmer

Pandas Brain Teasers  
Exercise Your Mind



Miki Tebeka  
edited by Margaret Eshelby

GO  
Meetup



vit

# Range Over Function

Go 1.23, August 2024

# Generators

Python 2.3, July 2003



## type Seq

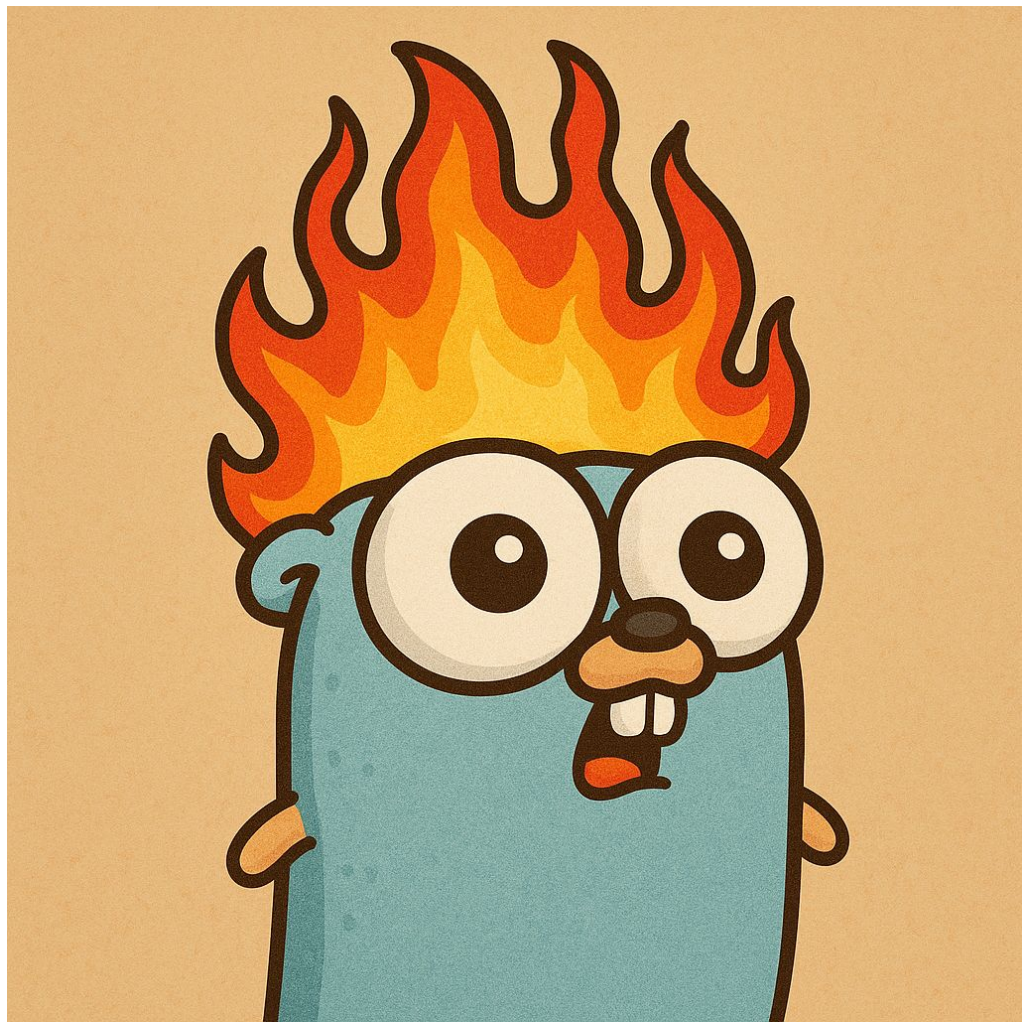
```
type Seq[V any] func(yield func(V) bool)
```

## type Seq2

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

[pkg.go.dev/iter](https://pkg.go.dev/iter)





## func Tick

```
func Tick(d Duration) <-chan Time
```

Before Go 1.23, this documentation warned that the underlying Ticker would never be recovered by the garbage collector, ...

[pkg.go.dev/time](https://pkg.go.dev/time)



# CODE

`tick/tick.go`

Load logs from  
compressed  
JSON file.

```
$ zcat logs.json.gz | wc
```

```
5350200 16045852 770589496
```

```
$ zcat logs.json.gz | head -1
```

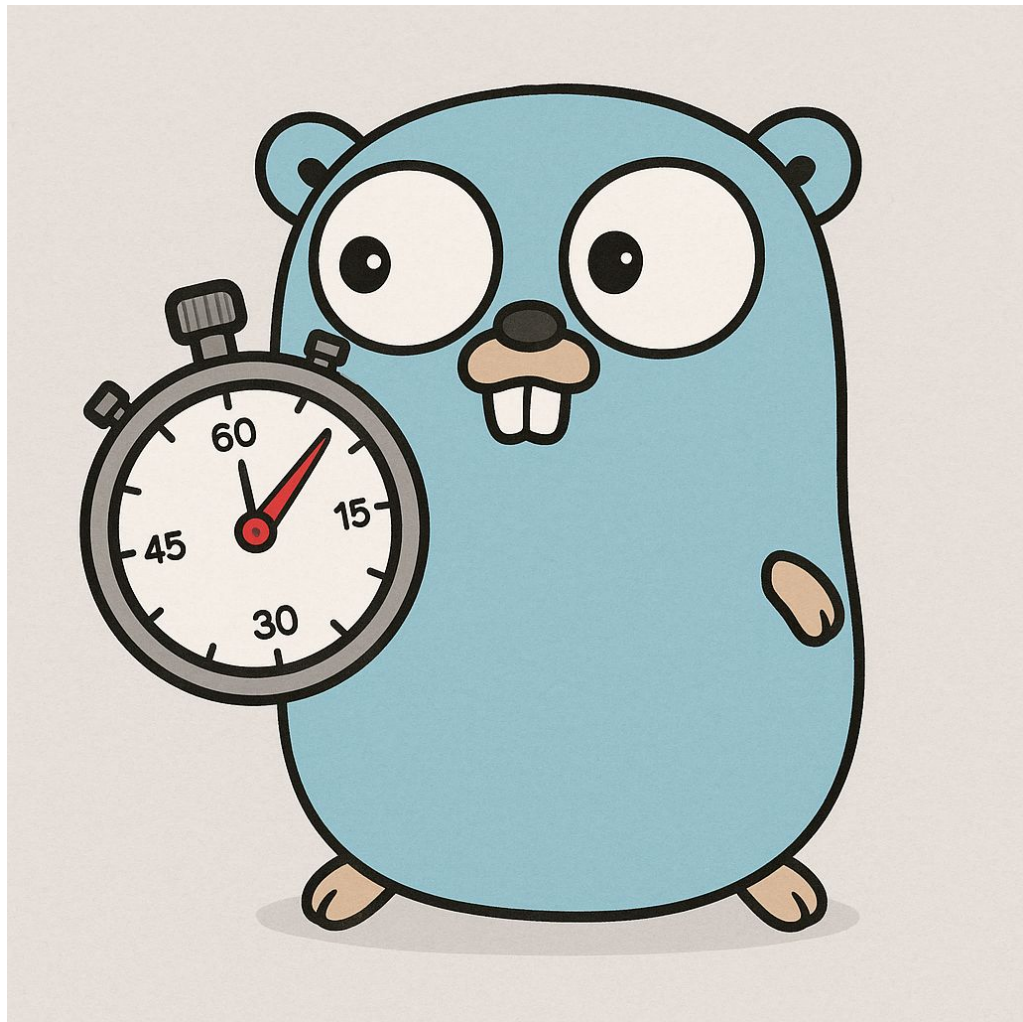
```
{"host": "199.72.81.55", "time":  
"2024-07-01T00:00:01Z", "request": "GET /history/apollo/  
HTTP/1.1", "status": 200, "bytes": 6245}
```

# CODE

`log/log.go`

# CODE

`slice/slice.go`





# CODE

bench/bench.go

# Map

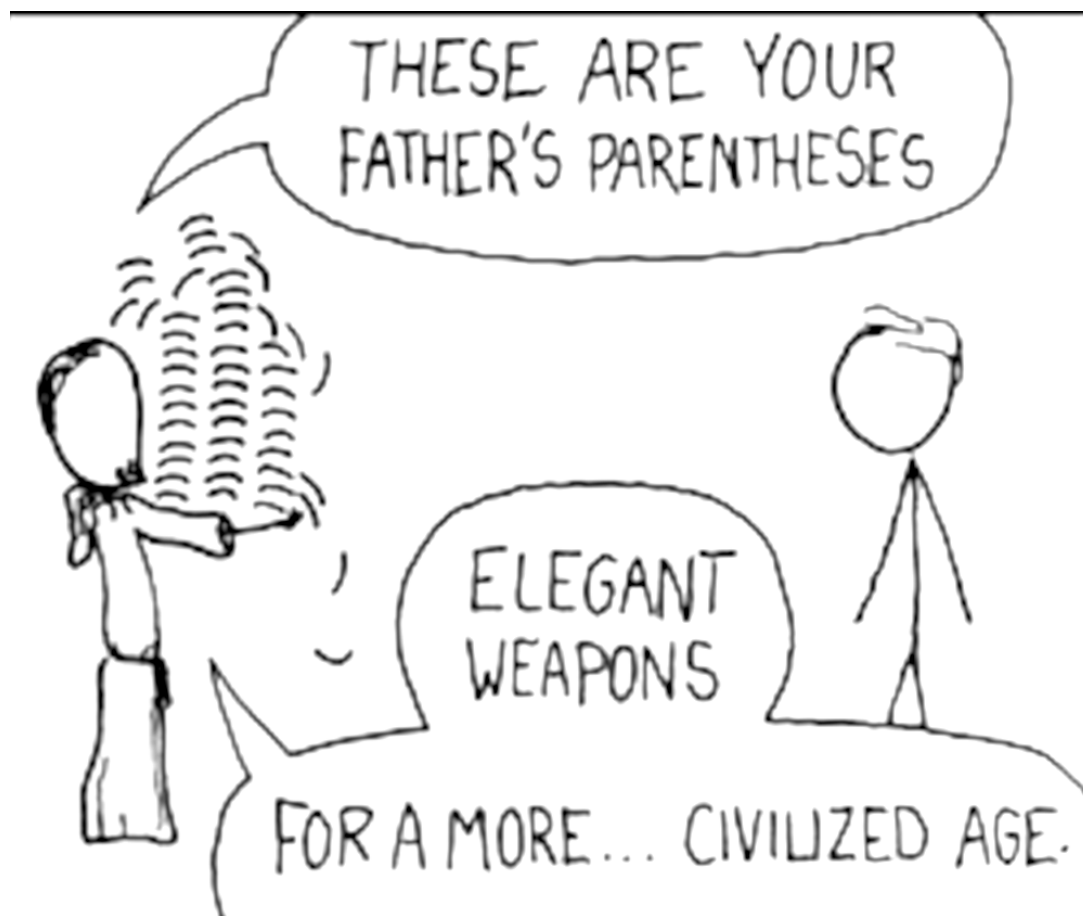
lower: ["Suite", "Up"] -> ["suite", "up"]

# Filter

$\geq 0$ : [2, -1, 3, 4] -> [2, 3, 4]

# Reduce

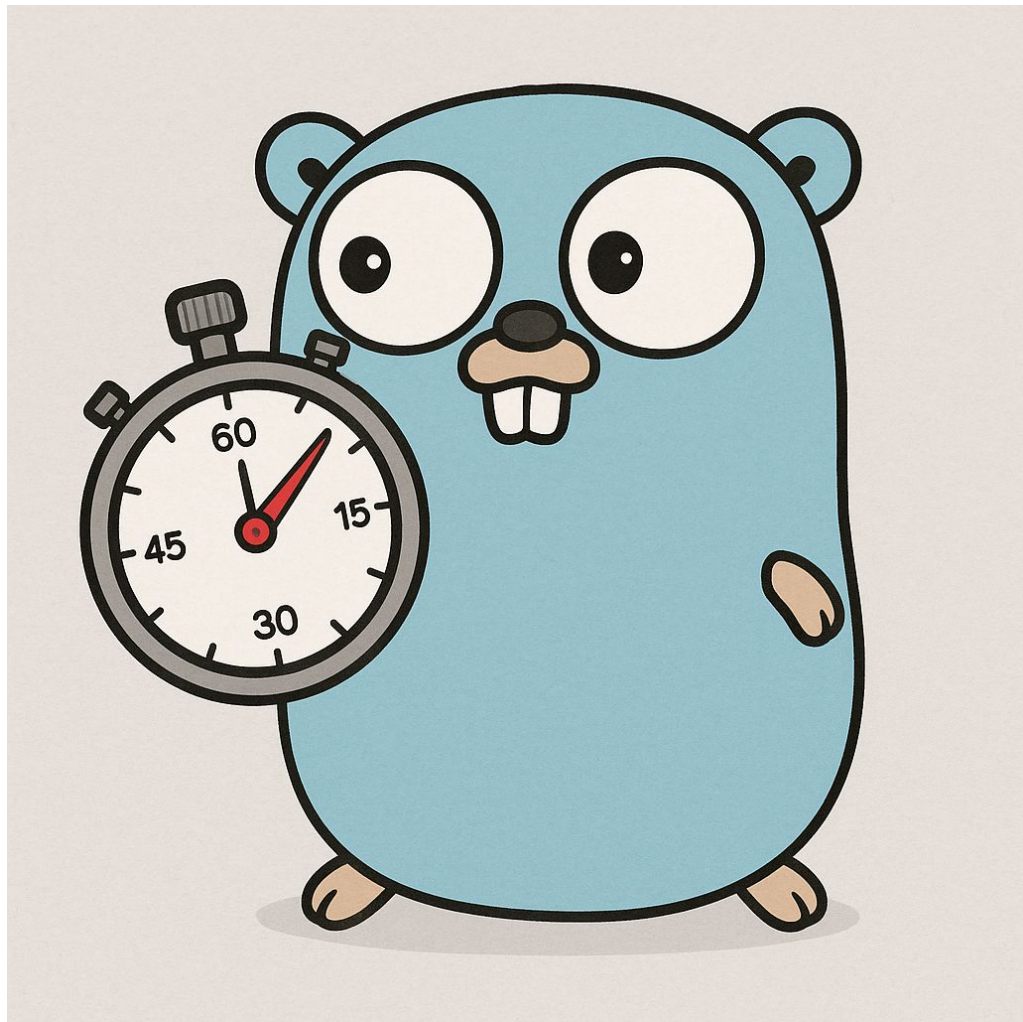
sum: [1, 2, 3, 4]  $\Rightarrow$  10

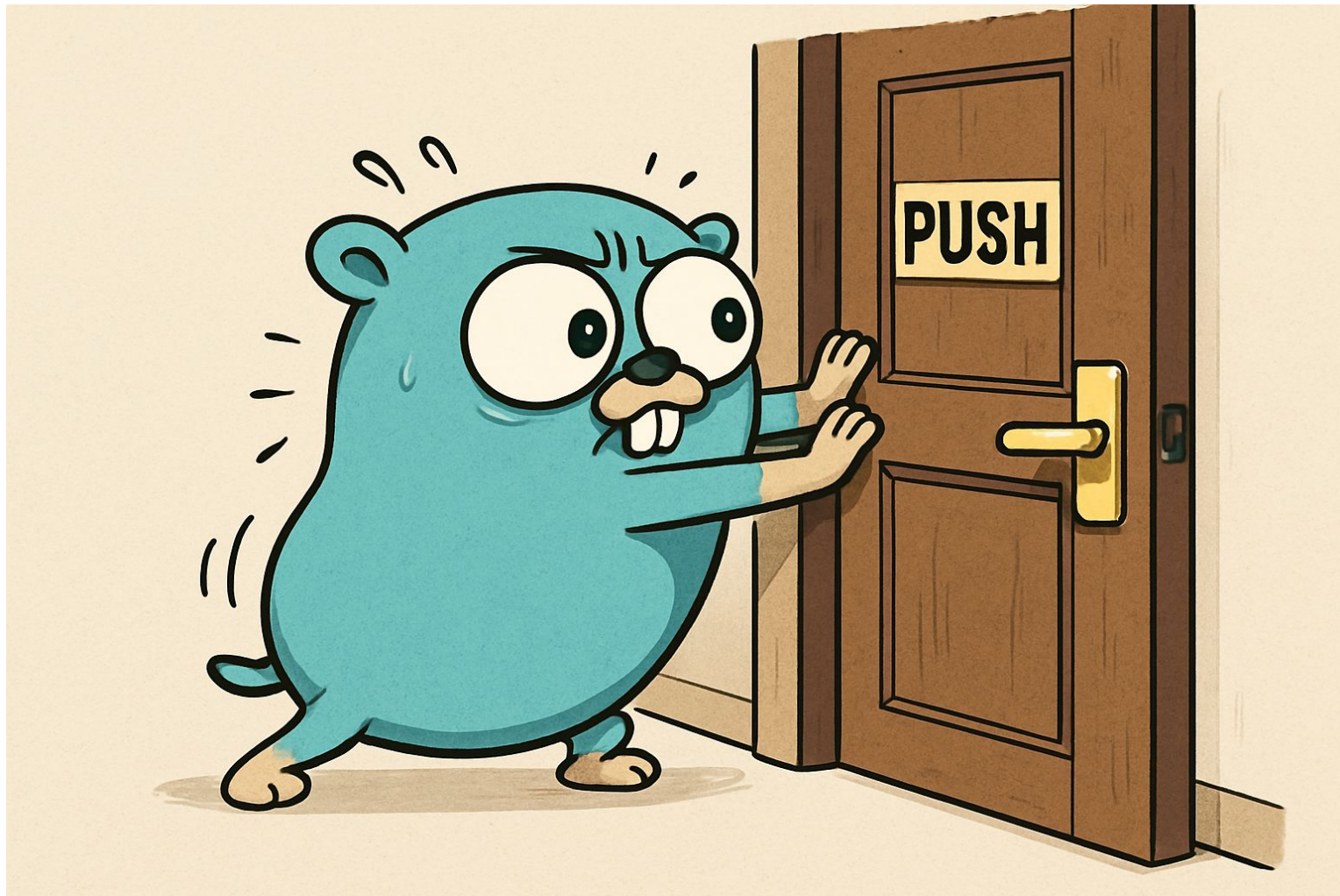


<https://xkcd.com/297/>

# CODE

`lazy/lazy.go`







# CODE

push / push . go

# CODE

`pull/pull.go`



## func Collect

```
func Collect[E any](seq iter.Seq[E]) []E
```

Collect collects values from seq into a new slice and returns it.

## func Values

```
func Values[Slice ~[]E, E any](s Slice) iter.Seq[E]
```

Values returns an iterator that yields the slice elements in order.

[pkg.go.dev/slices](https://pkg.go.dev/slices)

## func Values

```
func Values[Map ~map[K]V, K comparable, V any](m Map) iter.Seq[V]
```

Values returns an iterator over values in m. The iteration order is not specific

## func All

```
func All[Map ~map[K]V, K comparable, V any](m Map) iter.Seq2[K, V]
```

All returns an iterator over key-value pairs from m. The iteration order is not

[pkg.go.dev/maps](https://pkg.go.dev/maps)

## **func SplitSeq**

```
func SplitSeq(s, sep string) iter.Seq[string]
```

## **func FieldsSeq**

```
func FieldsSeq(s string) iter.Seq[string]
```

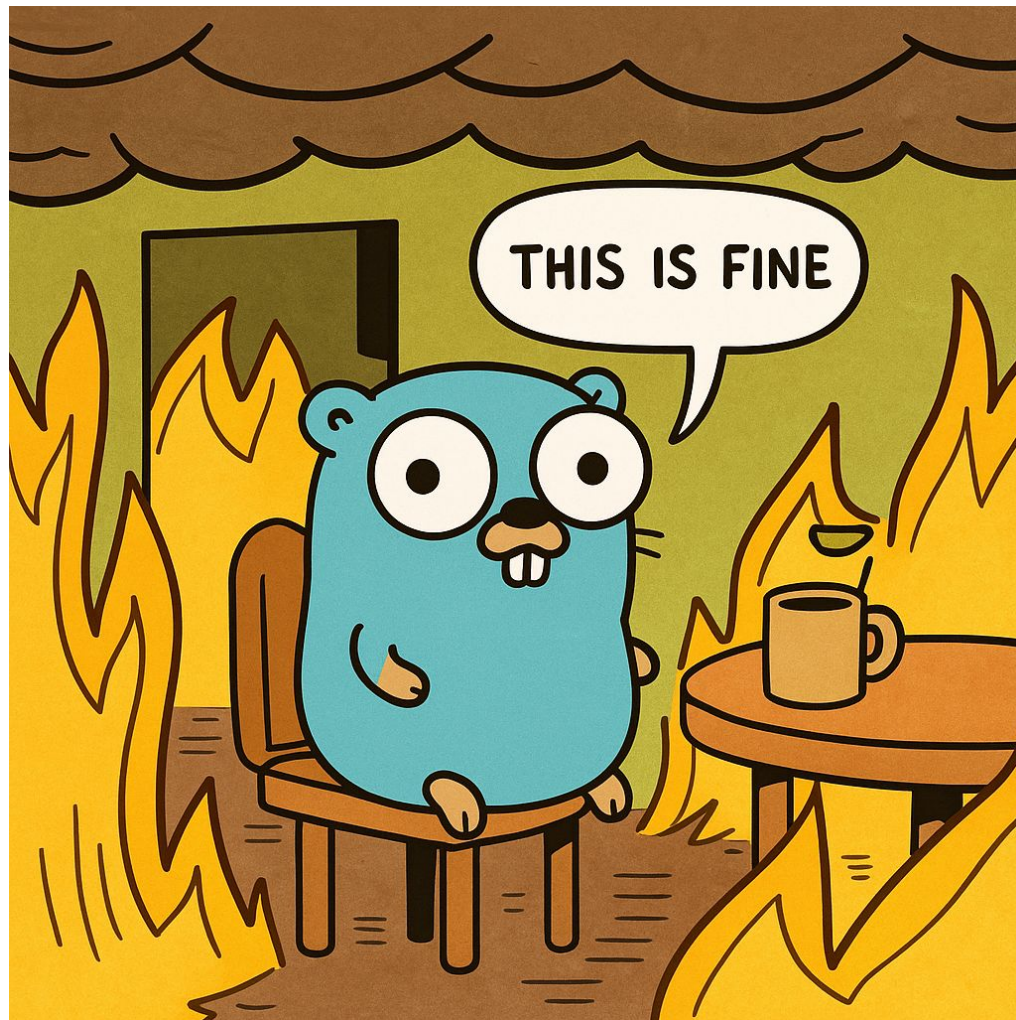
[pkg.go.dev/strings](https://pkg.go.dev/strings)





# TANSTAAFL

- Performance (\*)
- Harder to debug
- Error handling
- Resource management



# CODE

`order/order.go`

# CODE

`leak/leak.go`



# Questions?



# Range-Over Functions in Go ArdanLab Blog





[github.com/tebeka/talks/tree/main/gcil-2025](https://github.com/tebeka/talks/tree/main/gcil-2025)