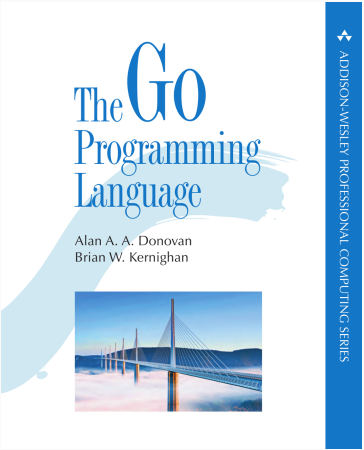
Programming in Go

Matt Holiday
Christmas 2020

The Book



ISBN 978-0-13-419044-0

"Anything with Brian Kernighan's name on it is worth reading."

— Matt Holiday

Why Go?

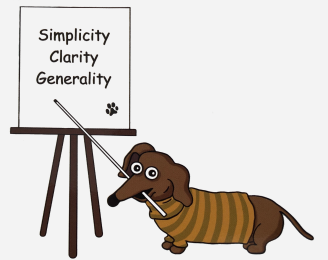
A better C from the folks who created Unix & C

"Go is about **language design in the service of software engineering**." — Rob Pike

from *Software Engineering at Google*

It's programming if "clever" is a compliment,
but it's software engineering if "clever" is an accusation. 😱

Simplicity



The Practice of Programming

"Programs must be written for people to read, and only incidentally for machines to execute." — Harold Abelson

"If our basic tool, the language in which we design and code our programs, is also complicated, **the language itself becomes part of the problem rather than part of its solution**." — Tony Hoare

Design goals

"Go is an attempt to combine the **ease of programming** of an interpreted, dynamically typed language with the **efficiency and safety** of a statically typed, compiled language." — *Go FAQ*

Overall goals:

- **simplicity**, **safety**, and **readability**
- ease of expressing algorithms
- orthogonality
- one right way to do things

A language that fits in your head

"One of the reasons I enjoy working with Go is that I can mostly hold the spec in my head — and when I do misremember parts it's a few seconds' work to correct myself. It's quite possibly the only non-trivial language I've worked with where this is the case."

— Eleanor McHugh

"**Clear is better than clever**" — Go Proverb

Go is boring

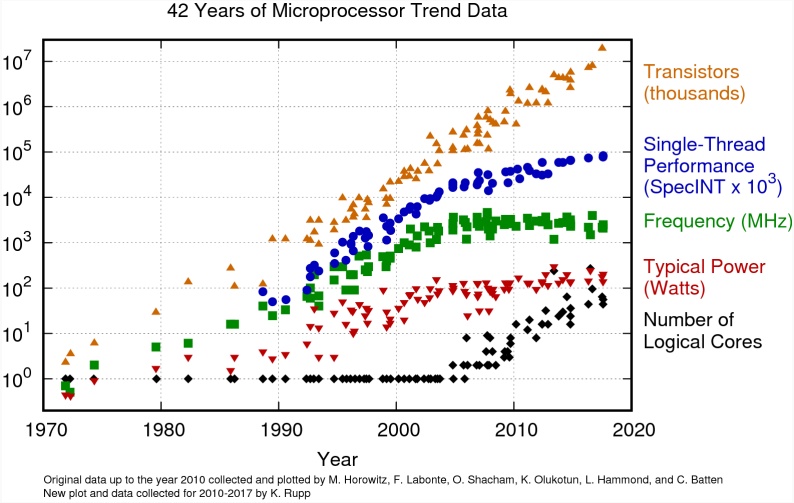
< SOFTWARE ENGINEERING

Go is Boring...And That's Fantastic!

A deep dive into why the world depends on simple, reliable, well-understood technologies

https://www.capitalone.com/tech/software-engineering/go-is-boring

Why did they do it?



Why did they do it?

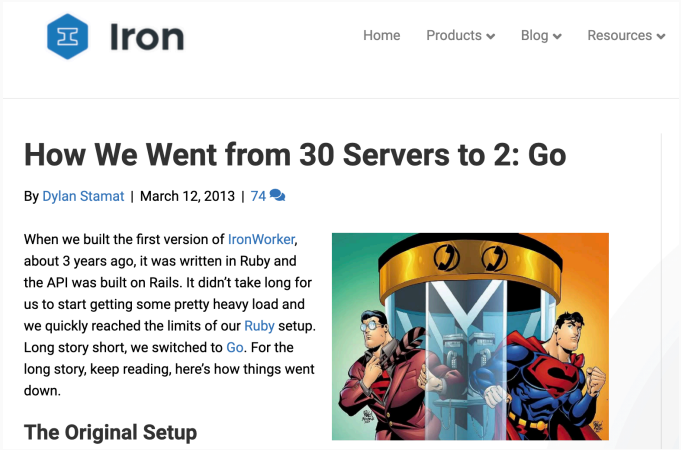
A new language for a new computing landscape:

- **multicore processors**
- networked systems
- massive clusters
- the web programming model (think REST)
- huge programs
- large numbers of developers
- long build times

##	Language	Year
1	JavaScript	1995
2	Python	1991
3	Java	1995
4	C#	2000
5	C++	1983
6	C	1972

Concurrency was an afterthought in older languages ☹

It’s just a computer you rent

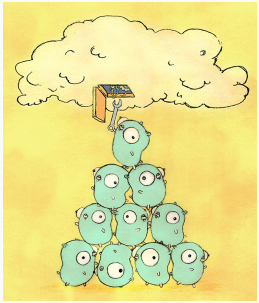


https://blog.iron.io/how-we-went-from-30-servers-to-2-go

Built for the cloud

It’s taking over the infrastructure/container/cloud world:

- Docker
- Kubernetes
- Helm
- Drone
- Rancher
- Prometheus
- Grafana
- CoreOS (etcd, flannel)
- CockroachDB
- DropBox
- CloudFlare



©Renée French

From bitly's engineering blog

"Think about it like this . . . if you can write something in Go

just as [quickly] as you could in Python and:

- gain the speed and robustness of a compiled, statically-typed language without *all* of the rope to hang yourself
- clearly express concurrent solutions to parallelizable problems
- sacrifice little to nothing in terms of functionality
- unambiguously produce consistently styled code (thank you `gofmt`)

What would you choose?"

https://word.bitly.com/post/29550171827/go-go-gadget

Once more, with feeling

“A language that doesn’t have everything is actually easier to program in than some that do.” — Dennis Ritchie