

# **Philosophy of Go**

10 years and growing...

**Egon Elbre**

Golang Estonia Meetup @ Starship

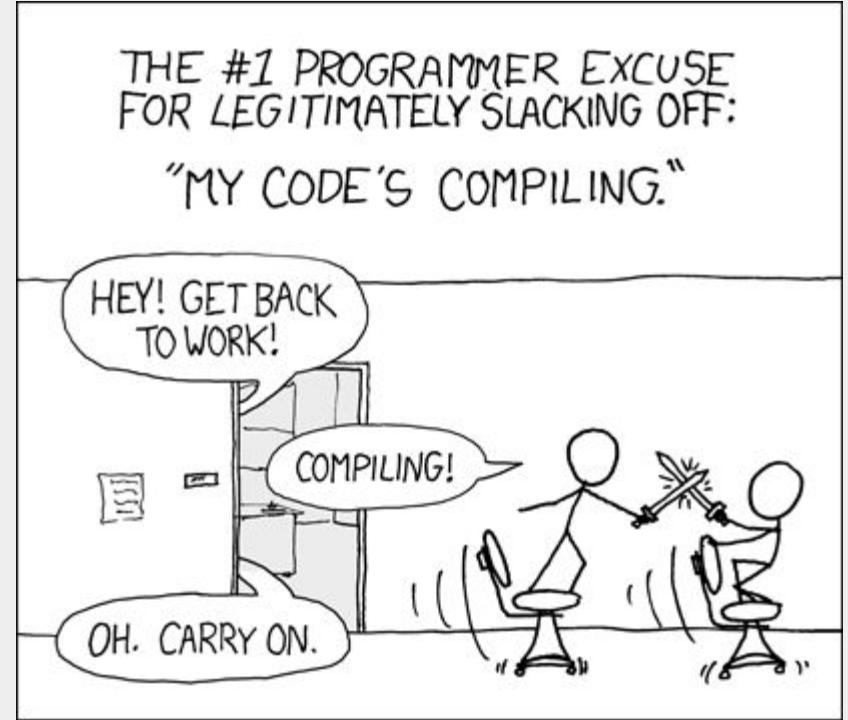
2019-11-22

# Where It All Started



# during a 45 minute build

September 2007



<https://xkcd.com/303/>

In the span of an hour at that talk [C++0x] we heard about something like 35 new features that were being planned. In fact there were many more, but only 35 were described in the talk. ...

Did the C++ committee really believe that was wrong with C++ was that it didn't have enough features?

Surely, [..], it would be a greater achievement to simplify the language rather than to add to it.

Of course, that's ridiculous, but keep the idea in mind.

- Rob Pike, [Less is Exponentially More](#)



Rob Pike



Ken Thompson



Robert Griesemer

# Problems

slow builds

uncontrolled dependencies

each programmer using a different subset of the language

poor program understanding (code hard to read, poorly documented ...)

duplication of

effo

rt

cost of updates

version skew

difficulty of writing automatic tools

cross-language builds

<https://talks.golang.org/2012/splash.article>

# Goals

It must work at scale, for large programs with large numbers of dependencies, with large teams of programmers working on them.

It must be familiar, roughly C-like. Programmers [...] most familiar with procedural languages, particularly from the C family. The need to get programmers productive quickly in a new language means that the language cannot be too radical.

It must be modern. C, C++, and to some extent Java are quite old,[...]. There are features of the modern world that are better met by newer approaches, such as built-in concurrency.

There are two ways of constructing a software design.

One way is to make it so simple that there are obviously no deficiencies.

And the other way is to make it so complicated that there are no obvious deficiencies.

- Tony Hoare, The emperor's old clothes - 1981



When the three of us got started, it was pure research. ...

We started off with the idea that all three of us had to be talked into every feature in the language,

so there was no extraneous garbage put into the language for any reason.

- Ken Thompson, 2017





Ian Lance Taylor



Russ Cox



Ian Lance Taylor

Rob Pike



Ken Thompson



Russ Cox



Robert Griesemer



Ian Lance Taylor

# Waiting for Good Design

**No** is temporary,  
**Yes** is forever.

9/24/2019

Google.com Mail - [go-dev] Generics



Robert Griesemer <gri@google.com>

---

**[go-dev] Generics**

12 messages

---

**Ian Lance Taylor** <iant@google.com>

To: go-dev@google.com

Sun, Aug 16, 2009 at 1:48 PM

I know we probably don't want to implement generics in Go at this point. But it may be worth thinking about what they would look like if we do implement them in the future.

It seems to me that the natural way to implement them is to make objects

# **Public Announcement**

November 10, 2009

# Hey! Ho! Let's Go!

Tuesday, November 10, 2009

Here at Google, we believe programming should be fast, productive, and most importantly, fun. That's why we're excited to open source an experimental new language called [Go](#). Go combines the development speed of working in a dynamic language like Python with the performance and safety of a compiled language like C or C++. Typical builds feel instantaneous; even large binaries compile in just a few seconds. And the compiled code runs close to the speed of C. Go lets you move fast.

Go is a great language for systems programming with support for multi-processing, a fresh and lightweight take on object-oriented design, plus some cool features like true closures and reflection.

Want to write a server with thousands of communicating threads? Want to spend less time reading blogs while waiting for builds? Feel like whipping up a prototype of your latest idea? Go is the way

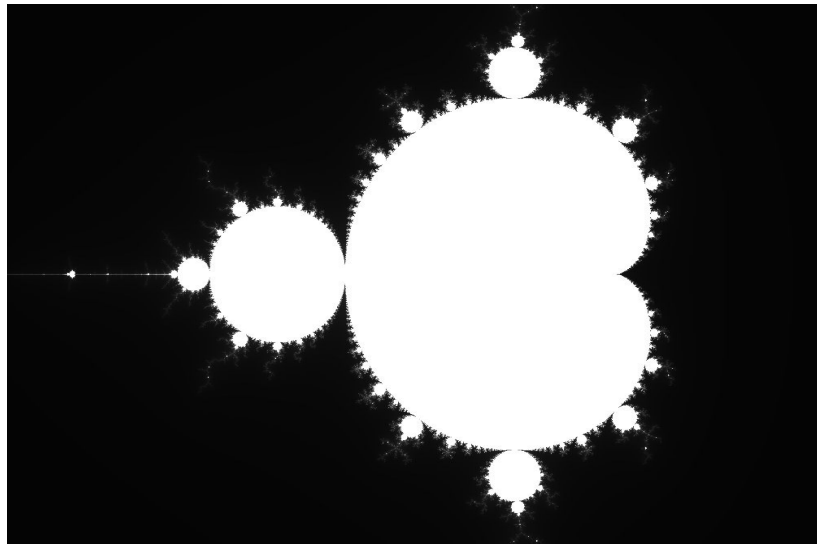


```
package main
```

```
import ( "image";  
        "image/png";  
        "bufio";  
        "fmt";  
        "os";  
        "math";  
        "time"; )
```

```
type PixelCalc struct { x int;  
                        y int;  
                        cx float64;  
                        cy float64; }
```

```
func PointIteration(in chan *PixelCalc,ready chan int,img *image.RGBA){  
    for {  
        pixelCalc := <- in;  
  
        xt := float64(0);  
        yt := float64(0);  
        x := float64(0);  
        y := float64(0);
```



# Readability

If a language has too many features, you waste time choosing which ones to use.

The code is harder to understand simply because it is using a more complex language.

Preferable to have just one way, or at least fewer, simpler ways.

Features add complexity. We want simplicity.

Features hurt readability. We want readability.

Readability is paramount.

- Rob Pike, [Simplicity is Complicated](#)

**gofmt**

**gofmt**'s style is no one's favorite,  
yet **gofmt** is everyone's favorite.

# Code is written for others to read.

Common language has more value than personal preferences.

**20% language for 80% of the effect.**

C11 is **683** pages. C++14 is **1354** pages.

Java 7 is **604** pages. Java 11 is **774** pages.

Go Language Specification is **90** pages.

No inheritance

=> Composition over Inheritance....

=> better separation of Interfaces.

No generics

=> Concrete over Abstract.

No circular dependencies

=> Code can be understood in isolation

# Go 1.0

March 28, 2012



# Compatibility Promise

Programs written to the Go 1 specification will continue to

compile and  
run correctly,  
unchanged,

over the lifetime of that specification.

What makes Go 1.0 so important is that it came with a promise that users' programs would continue to compile and run without change for the indefinite future. [ ... ]

Go 1.0 was far from perfect — there were many things that could have been done better, [ ... ] — but the promise of true stability more than compensated for any such weaknesses.

It's not just about promising compatibility, you have to deliver it.

- Rob Pike, 2019, [Computational Reproducibility](#)

# Early Adoption

2013-2014

Go 1.2

# dl.google.com

Rewritten from C++ to Go, because codebase was impenetrable.

*future maintainers slowly violated unwritten rules*

*C++ single-threaded event-based callback spaghetti*

*hard to know when/where code was running, or what "blocking" meant*

By Brad Fitzpatrick (of LiveJournal, memcached, OpenID, Perl ...)

<https://talks.golang.org/2013/oscon-dl.slide#1>

# **dl.google.com - rewritten in Go**

*less than half the code*

*more testable*

*same CPU usage for same bandwidth*

*... but can do much more bandwidth*

*... and more than one CPU*

*starts up instantly (not 24 hours)*

<https://talks.golang.org/2013/oscon-dl.slide#1>



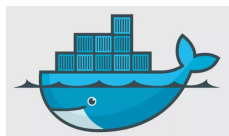
Packer



YouTube Vitess



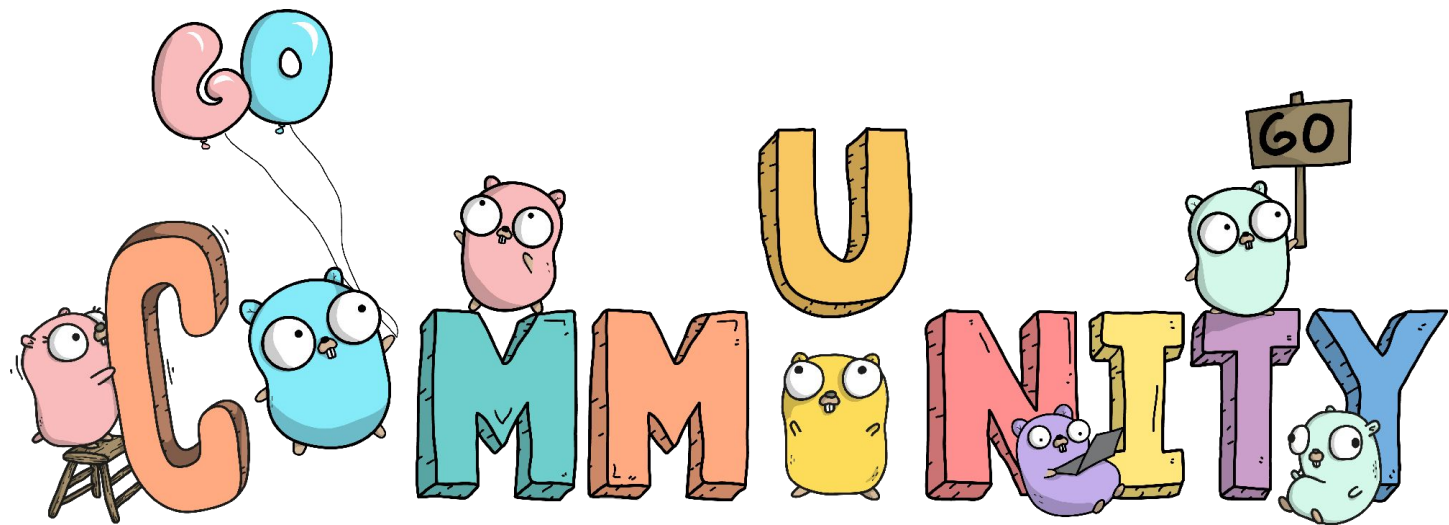
CoreOS



Docker



Kubernetes



Be friendly and welcoming

Be patient

Be thoughtful

Be respectful

Be charitable

Avoid destructive behavior





Users

**1.15-1.9M**

Contributors

**2100**

Conferences

**28**

# Go Developers

46 countries • 159 groups • 93563 members



## Companies using Go



Medium

NETFLIX



Uber



Booking.com

@mail.ru

dailymotion

Community is everywhere



Docker is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers



Hugo is one of the most popular  
open-source static site generators





A dead simple 2D game library in Go

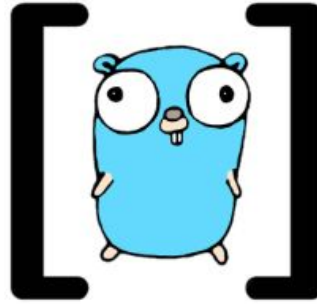


Caddy 2 is a powerful, enterprise-ready, open source web server with automatic HTTPS written in Go



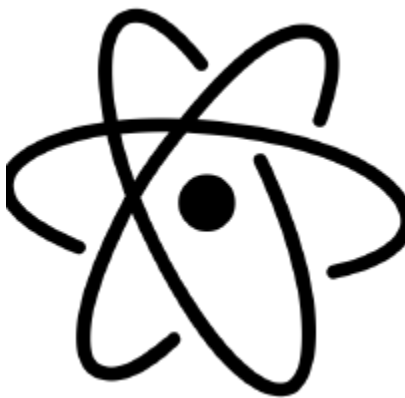
geth

Official Go implementation of the Ethereum protocol



## Gonum Numerical Packages

Consistent, composable, and comprehensible scientific code



go-hep

Go-based software for the  
High Energy Physics community.



## Deep Learning in Go

Write and evaluate mathematical equations involving multidimensional arrays easily.



Gobot is a framework for robots, drones, and  
the Internet of Things (IoT),  
written in the Go programming language



TinyGo brings the Go programming language to microcontrollers and modern web browsers.





Gio implements portable immediate mode GUI.

iOS/tvOS, Android, Linux (Wayland/X11),  
macOS, Windows, FreeBSD and  
browsers (Webassembly/WebGL).

# **Philosophy**

Waiting for Good Design.

Code is written for others to read.

Stability has enormous benefits.

Explicit is clearer than implicit.

Code that grows with grace.

Friendly and welcoming.



Go isn't perfect, nothing is.

Go was born out of frustration with complexity.

Go language is intentionally boring.

*... but, as a benefit ...*

*... but, as a benefit ...*

Go helps code to be

Maintainable

Readable

Obvious

Maybe when your language is compiling you can try something new ...

Maybe when your language is compiling you can try something new ...



**Chewxy**

@chewxy

Follow



I messed up my opencv install on my computer. So I recompiled it from scratch. By the time it finished compiling I had written the function I wanted in pure `#golang`. So uh, thanks for the waste of time/computational resources?

8:28 AM - 5 Oct 2019

8 Retweets 30 Likes



2



8



30