



Brasília DF, 01 Agosto 2019

Going fast with GO



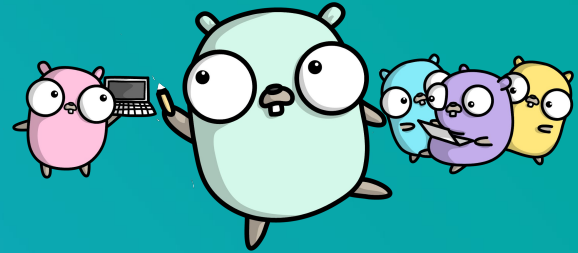
Rafael Passos

CS Researcher - UnB/IFB

GO Evangelist

SECTION ONE

Introduction



- **Suitable for anyone, from beginners to experienced programmers.**

Why?

- Easy to learn, and become productive
- Concepts are easy to understand
- Type and Memory Safe, Enforces Code Quality
- Modern Toolchain, fun to write!

For what Go should be used ?



- Micro Services, Web, Networking, Infrastructure, CLIs, Data Science, Cloud Computing, Cross Platform Apps

Why ? Built for the Cloud

Easy to maintain

Stable Syntax since 1.0 (2012)

Resource efficient and high performance

Supported by big companies (e.g. Google)

- Compiled to machine code: 32 e 64-bit x86, ARM, RISC-V, WebAsm...
- Cross Platform: Linux, Windows, Mac, Browser, IOS, Android, Arduino and other Microcontrollers (tinygo) ...
- Cross Compilation (for any OS/Arch)

- **Statically Typed:** zero variable type related runtime errors
- **Garbage Collected: safe** for any programmer
- Born with fixed **Code Styling** and **Formatting Tools** (GO fmt)
- Powerful **Tests** and **Benchmarks** are built into the language

SECTION TWO

Language Design

Authors:

Rob Pike (Bell Labs Plan 9 OS, Unix)

Ken Thompson (C Language, Unix, UTF-8)

Robert Griesemer (Java HotSpot, V8 JS Engine)

Thoughtful
Simple
Efficient
Reliable
Productive
Friendly

Deliberate and considerate.

Clear and precise.

Do more with less.

It just works.

Realize your vision, faster.

Accessible and welcoming.


Source : Official GO Brand Book v1.0 <https://blog.golang.org/go-brand>

Simplicity




All features
should be easy
to understand

Orthogonality



All features
should interact
in predictable
and consistent
ways.

Legibility

- 
- Simple understanding without over verbosity
 - Built with concepts familiar to any C or Java Developer

SECTION TREE

Dependencies Management

packages & modules

Always know where your dependencies come from

```
package main

import (
    "bytes"
    "context"
    "fmt"
    "log"
    "math/rand"
    "producer/db" // pacote interno

    "github.com/google/uuid"
    kafka "github.com/segmentio/kafka-go"
)
```

Packages are
imported using
the repository
URL

Arquivo de dependencias go.mod

```
module kafka_producer

go 1.12

require (
    github.com/google/uuid v1.1.1
    github.com/segmentio/kafka-go v0.3.0
)
```

```
module rabbitmq_producer

go 1.12

require (
    github.com/streadway/amqp v0.0.0-20190404075320...
```

Sem-Ver based
on repositories
releases

Want a specific
commit?
Commit Hash !

SECTION FOUR

TOOLING

A **Modern and Powerful language** should
have **Modern and Powerful tools**

GO code looks familiar wherever you GO

```
package main
import ("log")
func main(){helloMessage:="Hello Go"
log.Println(helloMessage)}
```

```
$ go fmt test.go
```

```
package main

import (
    "log"
)

func main() {
    helloMessage := "Hello Go"
    log.Println(helloMessage)
}
```

Auto code
formatting
directly from
the toolchain

Want a library ?

```
go get github.com/google/uuid
```

```
go get github.com/google/uuid@latest
```

```
go get github.com/google/uuid@v1.1.0
```

**Want to run with
one command ?**

go run hello.go

Want to build your app?

go build app.go

Want to Cross Compile ?

GOOS=windows GOARCH=amd64 go build app.go

GOOS=linux GOARCH=arm go build app.go

gomobile build -target=android app.go

Want a tool in your path ?

go install github.com/uber/kraken

Binary goes to:

- \$GOPATH/bin/kraken

In my case: /home/auyer/go/bin/kraken

Or

- /usr/local/bin/kraken If GOBIN=/usr/local/bin/

```
package main

import (
    "fmt"           // fmt unused, will be removed
                   // log needed, will be added
                   /* github.com/streadway/amqp needed,
                      will be added if found in your system,
                      error if not recognized
                   */
)

func main() {
    url := "amqp://amqp.local"
    conn, err := amqp.Dial(url)
    if err != nil {
        log.Fatalf("cannot dial: %v: %q", err, url)
    }
    conn.Close()
}
```

Automatic
insertion and
removal of
imported
libraries

CODE QUALITY CHECK WITH Go Report Card



```
go get github.com/gojp/goreportcard
```

```
goreportcard-cli -v
```

```
Grade: A+ (99.9%)
Files: 332
Issues: 2
gofmt: 100%
go_vet: 100%
gocyclo: 99%
gocyclo download/download.go:22
    warning: cyclomatic complexity 17 of function
download() is high (> 15) (gocyclo)

golint: 100%
ineffassign: 100%
license: 100%
misspell: 100%
```

Run several
static check
tools to keep
your code in
good shape!

Web Version :
<https://goreportcard.com/>

Go test



Test files should be in the same package of the functions, and are called `package_test.go`

```
package pow

import (
    "math"
    "testing"
)

func TestMathPow(t *testing.T) {
    res := math.Pow(2, 2)
    if res != 4 {
        t.Fail()
    }
}

func BenchmarkHandler(b *testing.B) {
    for n := 0; n < b.N; n++ {
        math.Pow(2, float64(n))
    }
}
```

BenchmarkPow 20000000 113 ns/op 0 B/op 0 allocs/op

go test .

go test -race .

go test -bench .

SECTION FIVE

Features

Concurrency with Goroutines

```
go doSomething()
```

- Lightweight,
 - Scalable,
 - Simpler
- Than
Threads!

Communication between goroutines



'Read-exactly-once' communication channels

```
package main

import ("fmt")

func helloName(name string, resultsChannel chan string) {
    resultsChannel <- "Hello " + name + " !"
}

func main() {

    channel := make(chan string)
    go helloName("Gopher", channel)
    result := <-channel
    fmt.Println(result)
}
```

<https://play.golang.com/p/BmLWIGDNwfC>

*“Do not
communicate by
sharing memory;
instead, share
memory by
communicating.”
– Rob Pike*

How many lines make a HTTP server?



```
package main

import (
    "fmt"
    "log"
    "net/http"
)

func função_olá(w http.ResponseWriter, req *http.Request) {
    fmt.Fprintf(w, "Olá !")
}

func main() {
    http.HandleFunc("/", função_olá)
    err := http.ListenAndServe("localhost:8080", nil)
    if err != nil {
        log.Fatal(err)
    }
}
```

Concurrent and
Non-Blocking !

Production
Ready with
stdlib only

<https://play.golang.org/p/G5v-OgjUbix>

ABSTRACTION THROUGH INTERFACES



You can define methods on any type/struct:

```
type geometry interface {  
    area() float64  
    perim() float64  
}  
  
type rect struct {  
    width, height float64  
}  
  
func (r rect) area() float64 {  
    return r.width * r.height  
}  
  
func (r rect) perim() float64 {  
    return 2*r.width + 2*r.height  
}  
  
func measure(g geometry) string {  
    return fmt.Sprintf("Area: %f, Perim: %f", g.area(), g.perim())  
}
```

*Interfaces are
collections of
method
signatures.*

Source:
<https://gobyexample.com/interfaces>

SECTION SIX

Who uses it ?

Notable Companies Using GO



Adobe, Alibaba, AT&T, Atlassian, Booking.com, Canonical, Cabify, CircleCI, Cloudflare, Comcast, Dell, Digital Ocean, Dropbox, Facebook, Google, IBM, Intel, Microsoft, Mozilla, PayPal, Pivotal, Twitter, Uber, SpaceX...

What about Brazil ?

99, MercadoLivre, Catho, C6 Banck, Dafiti, Elo 7, Globo.com, LEVEE, Loggi, Hotel Urbano, Viva Real, Magazine Luiza, Nic.br, Rede Bandeirantes, PagSeguro, PMMG, Quinto Andar, SiBBR.gov.br, SumUp, TOTVS, Vórtx, Walmart Brasil ...

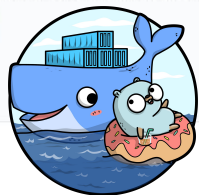
Fontes: <https://github.com/golang/go/wiki/GoUsers> e <https://www.meetup.com/golangbr>



Docker

Docker enables developers and IT operations to build, secure and manage applications without technology or infrastr...

San Francisco, CA <https://www.docker.com> info@docker.com Verified



Repositories 137

Packages

People 46

Projects 2

Find a repository...

Type: **All**

Language: **All**

docker-ce

Docker CE

[git](#) [docker](#) [golang](#) [moby](#)

[Go](#) [794](#) [3,073](#) [0](#) [1](#) Updated 1 hour ago

engine

Forked from moby/moby
docker engine release repository fork of moby

[Go](#) [Apache-2.0](#) [15,679](#) [299](#) [0](#) [27](#) Updated 2 hours ago

cli

The Docker CLI

[docker](#) [cli](#)

[Go](#) [Apache-2.0](#) [797](#) [1,559](#) [326](#) (3 issues need help) [122](#) Updated 5 hours ago

app

Make your Docker Compose applications reusable, and share them on Docker Hub

[docker](#) [kubernetes](#) [helm](#) [applications](#) [compose](#)

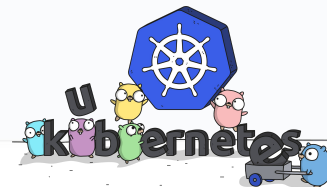
[Go](#) [Apache-2.0](#) [114](#) [1,207](#) [41](#) [9](#) Updated 7 hours ago



Kubernetes

Kubernetes

<https://kubernetes.io> Verified



Repositories 69

Packages

People 473

Projects 27

Pinned repositories

kubernetes

Production-Grade Container Scheduling and Management

[Go](#) [56k](#) [19.5k](#)

enhancements

Features tracking repo for Kubernetes releases

[Go](#) [729](#) [425](#)

community

Kubernetes community content

[Go](#) [4.4k](#) [2.2k](#)

website

Kubernetes website and documentation repo:

[HTML](#) [1.5k](#) [5.2k](#)

test-infra

Test infrastructure for the Kubernetes project.

[Go](#) [1.5k](#) [963](#)

examples

Kubernetes application example tutorial

[Shell](#) [2.3k](#) [1.6k](#)

Find a repository...

Type: **All**

Language: **All**

People



test-infra

Test infrastructure for the Kubernetes project.

[k8s-sig-testing](#)

[Go](#) [Apache-2.0](#) [963](#) [1,455](#) [287](#) (35 issues need help) [98](#) Updated 1 minute ago

kubernetes

Production-Grade Container Scheduling and Management

[containers](#) [kubernetes](#) [go](#) [cncf](#)

[Go](#) [Apache-2.0](#) [19,487](#) [55,972](#) [2,159](#) (60 issues need help) [1,066](#) Updated 7 minutes ago

node-problem-detector

Top languages

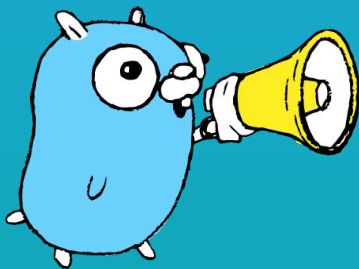
[Go](#) [Shell](#) [Python](#) [H](#)

Most used topics

[k8s-staging](#) [kubernetes](#)
[containers](#) [go](#) [k8s-si](#)

People





Synthesis

Efficient, reliable, scalable, simple.

Easy to learn. Familiar Concepts.

Code Quality at its core

Strict rules = Easy to maintain

High performance with low footprint

It's growing, and it's not going away.

TIME FOR A CHALLENGE



Brasília DF, 01 Agosto 2019

Thank you for your time



Rafael Passos

CS Researcher - UnB/IFB

GO Evangelist

