Go is 9 years old

Go is 9 years old

1.0 - 3/2012	• First version
1.1 - 5/2013	• Performance improve (30-40%). Race detector.
1.2 - 12/2013	• Better schedule. Test coverage.
1.3 - 6/2014	• Speed up GC.
1.4 - 12/2014	• Translate compiler from C to Go. Continuous stacks. New platform: Android.
1.5 - 8/2015	• Concurrency GC.
...	
1.8 - 8/2017	• Default GOPATH. Reduce GC pause. HTTP2.
...	
1.11 - 8/2018	• Go modules. New platform: WebAssembly.
...	• Go 2

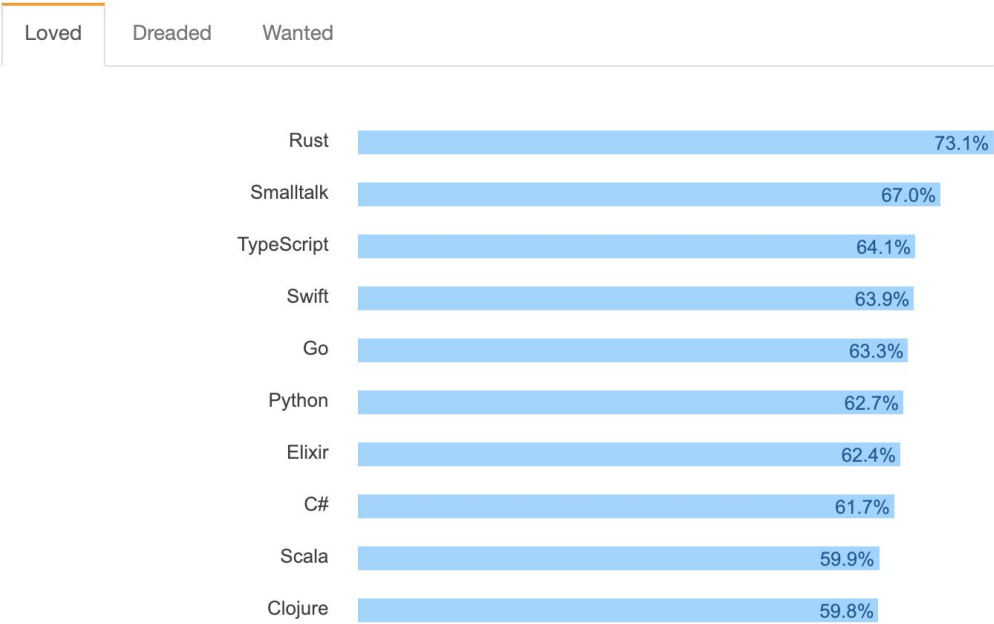# Hundreds of Golang meetups over the world

The fifteen most popular languages on GitHub

by opened pull request

GitHub is home to open source projects written in 337 unique programming languages—but especially JavaScript.

Language	# of Pull Requests
JAVASCRIPT	2.3M
PYTHON	1M
JAVA	986K
RUBY	870K
PHP	559K
C++	413K
CSS	335K
C#	326K
GO	285K
C	239K
TYPESCRIPT	207K
SHELL	206K
SWIFT	107K
SCALA	99K
OBJECTIVE-C	66K

OF PULL REQUESTS

Most Loved, Dreaded, and Wanted Languages

Loved | Dreaded | Wanted

Language	Loved
Rust	73.1%
Smalltalk	67.0%
TypeScript	64.1%
Swift	63.9%
Go	63.3%
Python	62.7%
Elixir	62.4%
C#	61.7%
Scala	59.9%
Clojure	59.8%

Companies using Go

Go is used as library for

Which of the following do you access from Go: (multiple choice)

3,784 (61%)		Open Source Relational DB (MySQL/PostgreSQL/CockroachDB)
2,400 (39%)		Memory Cache (Redis/memcache)
2,005 (32%)		Cloud Storage (S3/Google Cloud Storage/Azure Storage/Minio)
1,891 (30%)		Open Source NoSQL DB (MongoDB/Cassandra)
1,606 (26%)		Authentication and federation (SSO/LDAP/OAuth)
1,546 (25%)		Distributed Key-Value store (etcd/consul)
657 (11%)		Proprietary Relational DB (Oracle/DB2/MSSQL/Sybase)
459 (7%)		Distributed Lock Service (zookeeper)
1,367 (22%)		No response

Go is used for implementing

I write the following in Go: (multiple choice)

4,071 (65%)	API/RPC services (returning non-HTML)
3,921 (63%)	A runnable/interactive program (CLI)
3,027 (49%)	Web services (returning HTML)
2,766 (44%)	Agents and daemons (e.g, monitoring)
2,394 (38%)	Libraries or Frameworks
2,038 (33%)	Automation/scripts (e.g, deployment, configuration management)
2,030 (33%)	Data processing (pipeline, aggregation)
167 (3%)	I don't write in Go
176 (3%)	Other
70 (1%)	No response
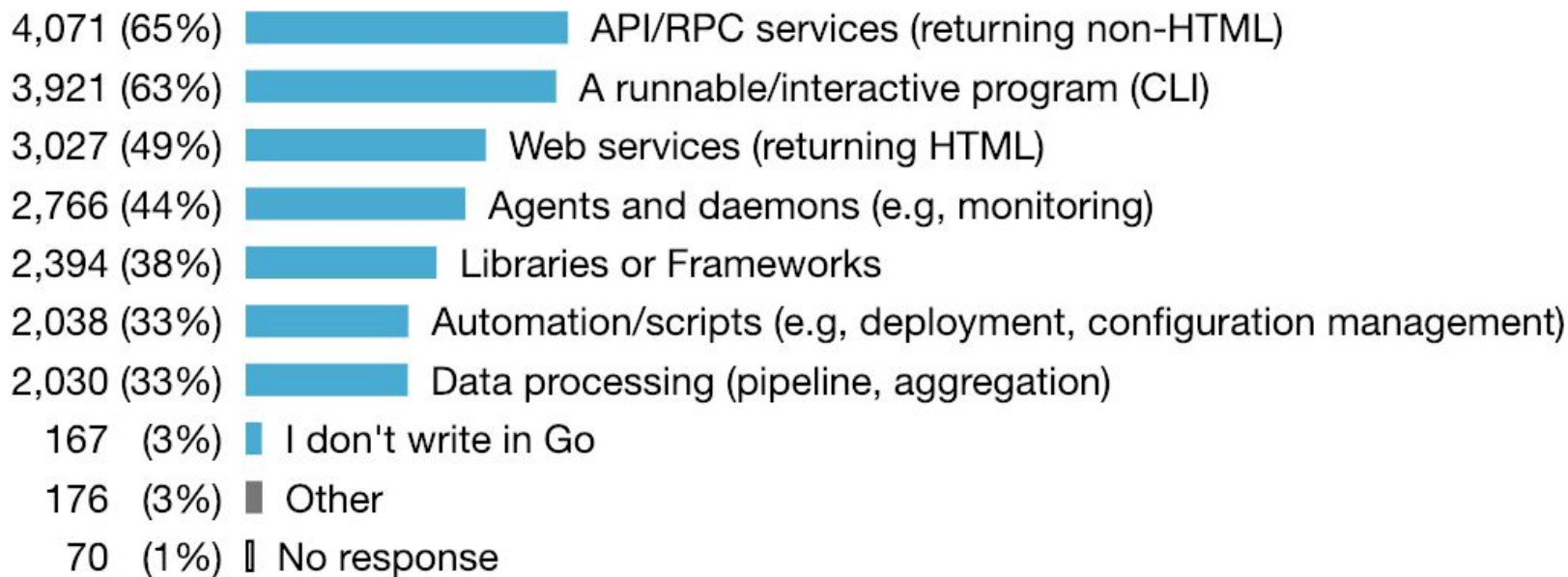
Which functionality implemented in Go?

Which of the following functionality have you implemented (multiple choice)

3,262 (52%)	Writing logs/metrics
3,123 (50%)	Reading/updating configuration
2,771 (45%)	User login and authentication
2,748 (44%)	Process to process communication
2,504 (40%)	Service authentication/authorization
2,056 (33%)	Health checking
1,138 (18%)	Keys & secret maintenance
831 (13%)	Distributed caching
532 (9%)	Distributed tracing
1,269 (20%)	No response
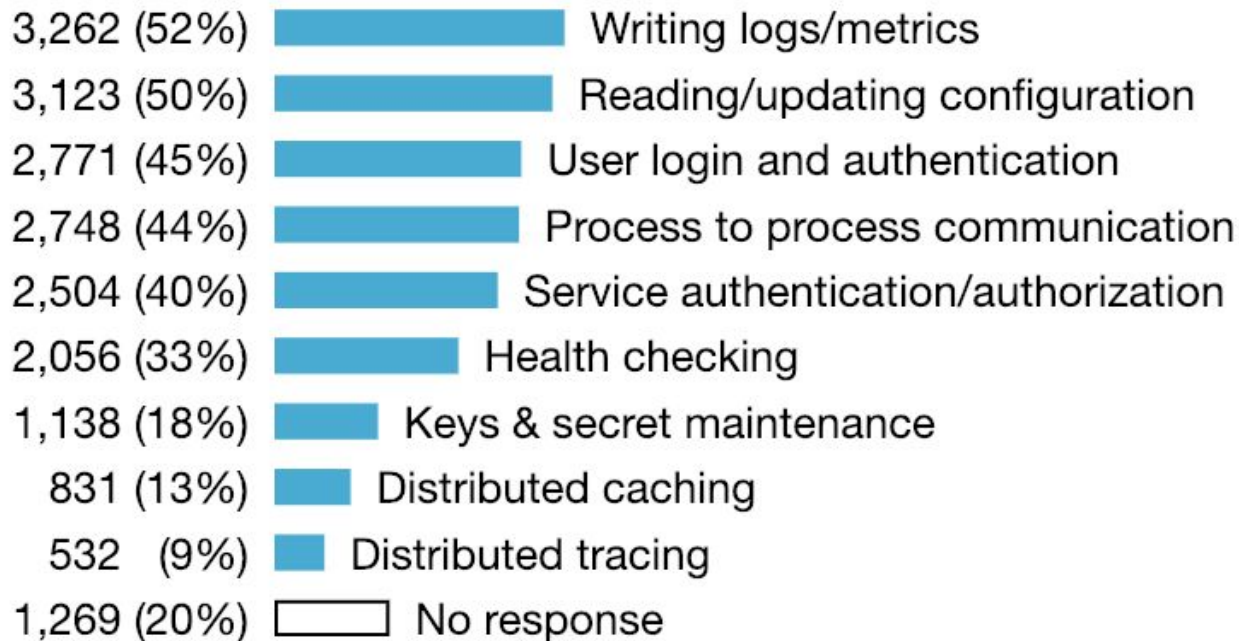
Go has nothing special!

- Go is not as **elegant** as **Python**
- Go does not excellent at **metaprogramming** like **JavaScript** or **Ruby**
- Go does not have **macro**
- Go does not have VM to execute code cross-platform

```go
func handleRequest(w http.ResponseWriter, h *http.Request) {
    query, err := parse(w)
    if err != nil {
        return err
    }

    items, err := db.getItems(query)
    if err != nil {
        return err
    }

    data, err := json.Marshal(items)
    if err != nil {
        return err
    }

    // ...
}
```

```go
func convertProducts(items []*db.Product) []*api.Product {
    result := make([], len(items))
    for i, item := range items {
        result[i] = convertProduct(item)
    }
    return result
}

func convertCustomers(items []*db.Customer) []*api.Customer {
    result := make([], len(items))
    for i, item := range items {
        result[i] = convertCustomer(item)
    }
    return result
}

// ...
```

```
// get the max price
let max = items.reduce((a,item) => a < item.price ? item.price : a, 0)
```

```go
func getProductMaxPrice(items Products) int {
    max := 0
    for _, item := range items {
        if item.price > max {
            max = item.price
        }
    }
    return max
}


// get the max price
maxPrice := getProductMaxPrice(products)
```

The most value of Go is

Get the works done!

The most value of Go is

Get the works done!

- Write code
- Pass tests
- Watch your business grow
- Sleep well!

Really?

- Single & static binary deployment
- Optimized for readability and team collaboration
- No bikeshed (gofmt)
- Minimal surprise
- Full development environment: go doc, go build, go build, go test, oracle

↑ Honest question: what do you consider Go's biggest strength?
31
↓ (self.golang)

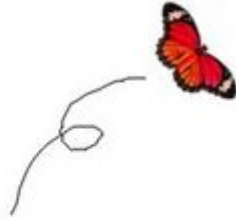↑ [−] **Queue29** 37 điểm 1 tháng trước đây
↓ Static binary deploys.

↑ [−] **thesnowmancometh** 35 điểm 1 tháng trước đây
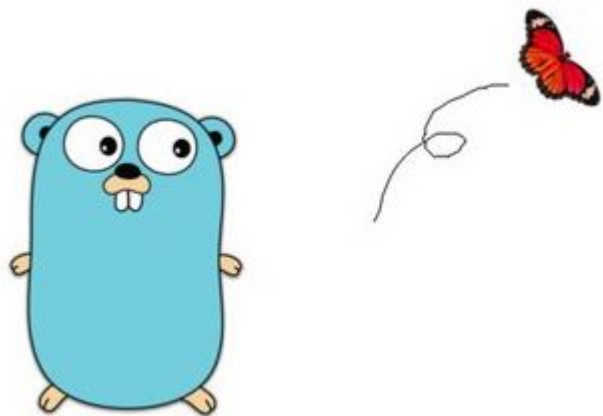↓ I never realized the benefits of having a small language.

↑ [−] **CorgiMann** 20 điểm 1 tháng
↓ Go is very minimalistic.

Go 2
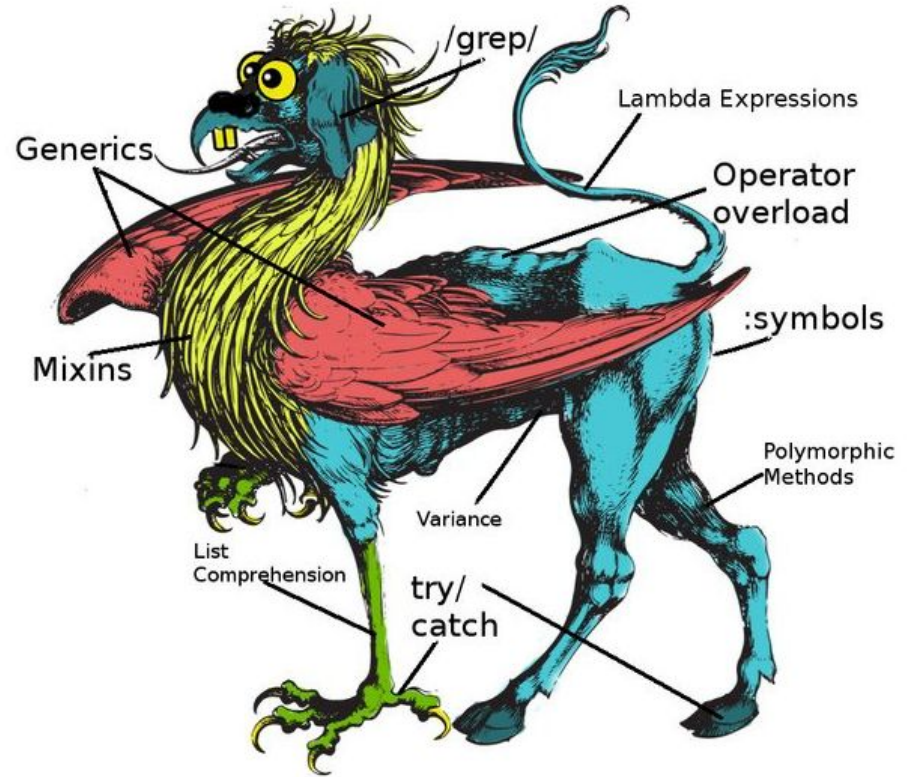
Go 2

Welcome to GopherCon 2018