

Getting Started with Go

A Quick Introduction

4 May 2016

Jacob Walker

Gopher

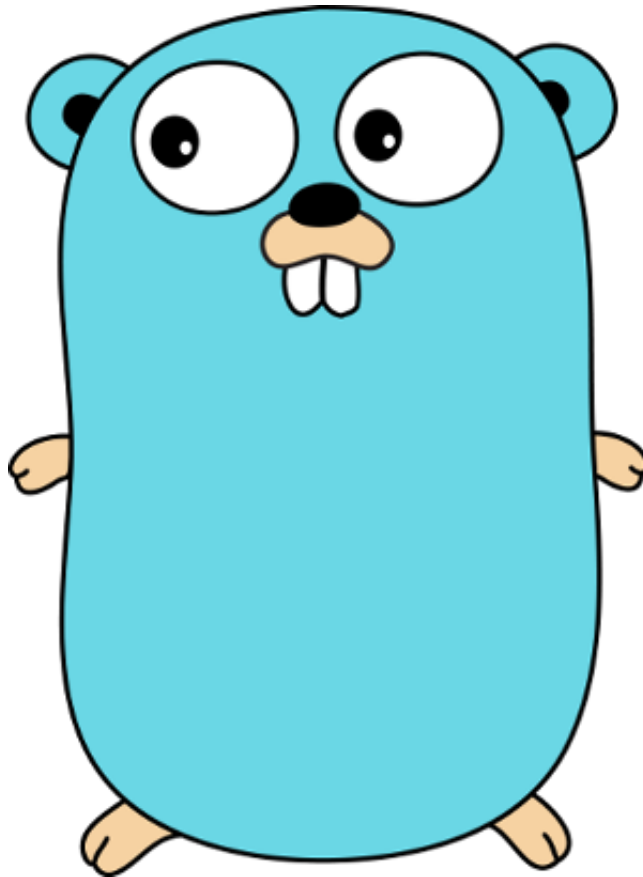
Background

Me

The majority of my experience is in PHP. I ventured into Ruby, Python, and Node.js. Dabbled with C++ and Android.

Found Go and felt at home.

Enter The Gopher



Gopher by [Renée French](http://www.reneefrench.com) (<http://www.reneefrench.com>)

Go Origins

Originally developed at Google by Ken Thompson, Rob Pike, and Robert Griesemer.

Relatively young: started in 2007 and announced in 2009.

Originally developed in response to pain points in C and C++, Go has become very popular among developers coming from dynamic general purpose languages.

What's It For?

Mostly

- Server software (web app back ends, APIs)
- Command line tools (Docker, ngrok, hugo, lots of DevOps stuff)

Kind of

- GUI desktop apps / games
- Mobile apps

Benefits

Compiled benefits without the costs

- Fast compilation (really fast).
- Easy to compile

```
go build
```

- Easy to cross compile

```
GOOS=linux GOARCH=amd64 go build
```

- Single statically linked binary. Build, copy, run, profit!

Simplicity



Dave Cheney

@davecheney



Following

Most programming languages start out aiming to be simple, but end up just settling for being powerful.

RETWEETS

33

LIKES

29



6:28 PM - 1 Dec 2014

Simplicity

- Only 25 keywords.
- Opinionated. Often only one way to do something.
- Only one looping construct for that can handle different styles of looping.
- Few needs for semicolons. Few need for parens.
- Mandatory braces. Have to be on same line.
- Double quotes for strings. Single quotes for single runes.
- Consistent, Predictable, Orthogonal

Tooling

- Built in testing, benchmarking, and documentation tools.
- Plethora of analysis / refactoring tools available.
- Universal code formatting style (gofmt).

gofmt

Core tool that enforces consistent coding style.

No more bike-shedding over style.

Tabs vs Spaces? Non-issue.

Expansive standard library

<http://golang.org/pkg/> (<http://golang.org/pkg/>)

- `net/http`: Build web servers and clients
- `text/template`, `html/template`: Templating tools
- `sync`: Tools supporting concurrency
- `encoding/json`: JSON Marshalling and Unmarshalling

More Notable Features

- Statically Typed
- Automatic garbage collection
- OOP Composition over Inheritance (No direct inheritance)
- Implicit interfaces
- First class unicode Support
- No exceptions
- No generics

Install Go

Setup Instructions (<https://gist.github.com/jcbwllkr/2cbe72427f9cbc41bfbb7972d62429d8>)

Let's Build an App!

1. Greet us
2. Greet us loudly
3. Greet a specified name
4. Use a random greeting based on time with switch
5. Use a slice of greetings and `math.rand`

Where do we go from here?

Required Reading

- [Take the tour](https://tour.golang.org) (<https://tour.golang.org>)
- [How to Write Go Code](https://golang.org/doc/code.html) (<https://golang.org/doc/code.html>)
- [Effective Go](https://golang.org/doc/effective_go.html) (https://golang.org/doc/effective_go.html)

More Reading!

- [Code Review Comments](https://github.com/golang/go/wiki/CodeReviewComments) (<https://github.com/golang/go/wiki/CodeReviewComments>)
- [The spec](https://golang.org/ref/spec) (<https://golang.org/ref/spec>)
- [The FAQ](https://golang.org/doc/faq) (<https://golang.org/doc/faq>)
- [The Go Blog](https://blog.golang.org/index) (<https://blog.golang.org/index>)
- Read the standard library source

Use the tools

- `gofmt` (or `goimports`) on save
- `golint` looks for style errors
- `go vet` looks for suspicious / dangerous code
- `godoc` browse Go Doc of all locally installed packages
- `gometalinter` (<https://github.com/alecthomas/gometalinter>)

Community

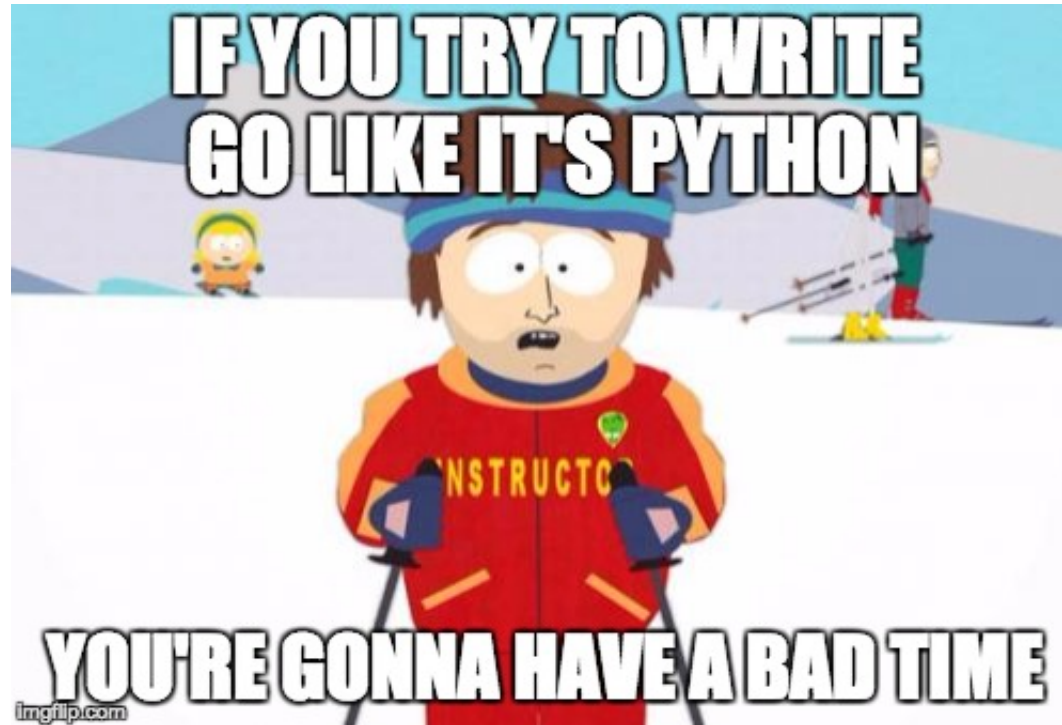
- #go channel on devICT / WWC Slack
- [Gopher Slack](http://invite.slack.golangbridge.org/) (http://invite.slack.golangbridge.org/)
- [Exercism.io](http://exercism.io/) (http://exercism.io/) Go exercises with community sourced feedback
- [Monthly Go Challenge](http://golang-challenge.com/) (http://golang-challenge.com/)
- Open Wichita Infant Info project

Other Resources

- [Awesome Go](https://github.com/avelino/awesome-go) (https://github.com/avelino/awesome-go) (curated list of great community packages)
- [Go By Example](https://gobyexample.com/) (https://gobyexample.com/) Go tutorials
- [Golang weekly](http://golangweekly.com/) (http://golangweekly.com/) free weekly newsletter
- RSS Feed of StackOverflow questions tagged go

Misc & Best Practices

Write Idiomatic Go. Don't try to write Python/Ruby/Java.



Thank you

Jacob Walker

Gopher

<http://jacob-walker.com> (<http://jacob-walker.com>)

[@jcbwlkr](http://twitter.com/jcbwlkr) (<http://twitter.com/jcbwlkr>)

