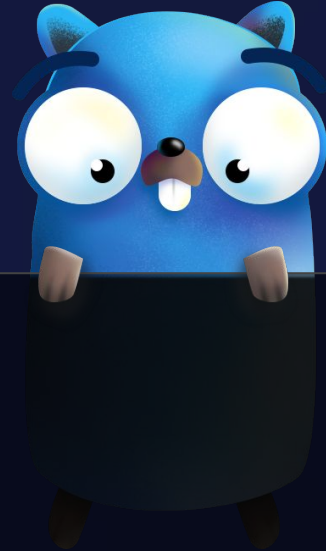


Code blocks example

```
Package main
import(
    "fmt"
)

Func main(){
    fmt.Println("This is Cool")
}
```





GCP and Go

Google Cloud

Synopsis

GCP and Go (golang) Talk covers some basics of the Google Go language. It also includes how to setup your development environment, build your first Go app, and covers Google Go strengths. We will also dive deeper into the aspects of Google Gos strengths such as portability, concurrency, and Go Cloud libs. The talk is one day in length. Audiences that benefit the most are automators, system administrators, and developers with 0-4 years experience.

Schedule

Hello, World (9 to 10)	Workstation setup (Go, IDE), your first app, language overview and exercises
Cloud Libs (10 - Noon)	Understand to the Go Cloud lib eco system, how they are built, and how to submit an issue.
Go Strengths (12:30 -2)	Concurrency and Portability
Open Kimono (2 - 3:30)	Refactor some code I wrote
Wrap Up (30 minutes)	References, books, blogs, and getting help



Google Cloud



Workstation setup

Google Cloud

Setup Go

- ▶ Install Go for your platform from <https://golang.org/dl/>
- ▶ Set up you folder structure <https://golang.org/doc/code.html>
- ▶ (Your Home Folder) / (gopath folder)
 - /src
 - /pkg
 - /bin
- ▶ This training will do all of its work under the src folder.



IDE Install Your Favorite IDE

- **Atom** - <https://atom.io>
- **Visual Studio Code** - <https://code.visualstudio.com/> (Recommended for Beginners)
- **Goland** - <https://www.jetbrains.com/go/specials/go/go.html?dclid=CJzE5LDG4NwCFvuwQodWy4Pow>
- **LiteIde** - <https://github.com/visualfc/liteide>
- **VIM** - <https://github.com/fatih/vim-go>
- **Emacs** - <https://github.com/dominikh/go-mode.el>

Explore Go commands

The commands are:

bug start a bug report

build compile packages and dependencies

clean remove object files and cached files

doc show documentation for package or symbol

env print Go environment information

fix update packages to use new APIs

fmt gofmt (reformat) package sources

generate generate Go files by processing source

get download and install packages and dependencies

install compile and install packages and dependencies

list list packages or modules

mod module maintenance

run compile and run Go program

test test packages

tool run specified go tool

version print Go version

vet report likely mistakes in packages

Getting the samples code and slides

Commands

- `cd $GOPATH/src/`
- `git clone https://github.com/goog-lukemc/gcp-train`
- Slide and docs are in `$GOPATH/src/assets`
- Source is in `$GOPATH/src/gcp-train/<item>`



Background

Google Cloud

O2

History

Who

Robert Griesemer
Rob Pike
Ken Thompson

Why

Combine the ease of a dynamic type language with the safety of the static type system.
https://golang.org/doc/faq#Is_Go_an_object-oriented_language
https://golang.org/doc/effective_go.html
<https://golang.org/doc/code.html>

Where are we now

1.13.1 Oct 2019
<https://golang.org/dl/>

More info

<https://talks.golang.org/2012/splash.article>
<https://tip.golang.org/doc/go1.11>
<https://golang.org/doc/devel/release.html>
<https://talks.golang.org/2015/gophercon-goevolution.slide#8>
<https://golang.org/doc/faq>

What's cool about Go (top 3)

► Concurrency: (more after lunch)

- Concurrency is not parallelism (https://www.youtube.com/watch?v=cN_DpYBzKso)
- Concurrency is about having the best design to maximize parallelism if it is available.

► Interfaces:

- A brief overview

► Portability:

- Go is not runtime interpreted (There is nothing to install on the target to execute a Go program.)
- A simple build switch can build the executable for any supported platform.

Coders vs. developers

Coders

Writing code to solve the problem in front of you. Slinging code - having fun!

Developers

Writing code to solve a problem for generic reuse. Writing small - having fun!

Idiomatic

What is this anyway? - https://golang.org/doc/effective_go.html



Hello, World

Google Cloud



~/gcp-train/hello

Review: main.go (In Editor)

Build: main.go (In Editor)

Build for any platform from any platform:

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

Running the file and flow testing: `go run main.go`

~/gcp-train/hello_flag

Review: main.go (In Editor)

Build: main.go (In Editor)

Build for any platform from any platform:

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

Running the file and flow testing: `go run main.go`

~/gcp-train/hello_struct

Review: main.go (In Editor)

Build: main.go (In Editor)

Build for any platform from any platform:

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

Running the file and flow testing:

- `go build`
- `go run main.go`



The basics

Google Cloud

04

errors ~/gcp-train/errors

Review: main.go (In Editor)

Build for any platform from any platform:

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

Running the file and flow testing: `go run main.go`

basic ~/gcp-train/basics

Review: main.go (In Editor)

Build for any platform from any platform:

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

Running the file and flow testing: `go run main.go`

basic ~/gcp-train/asciicoolness

Review: main.go (In Editor)

Build for any platform from any platform:

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

Running the file and flow testing: `go run main.go`

basic ~/gcp-train/basichttpserver

Review: main.go (In Editor)

Build for any platform from any platform:

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

Running the file and flow testing: `go run main.go`

Go Cloud Libs:

<https://github.com/googleapis/google-cloud-go>

Generated vs Custom

gRPC vs HTTP Wrappers

How to submit bugs and contribute



Lunch time

Google Cloud

05



Interfaces

Google Cloud



Interfaces

Web links

- <https://gobyexample.com/interfaces>
- <https://medium.com/golangspec/interfaces-in-go-part-i-4ae53a97479c>

Let's do some code review



Concurrency

Google Cloud



concurrency ~/gcp-train/basichttpserver

Review: main.go (In Editor)

Build for any platform from any platform:

- **Pi:** `env GOOS=linux GOARCH=arm go build -v main.go -o program-arm`
- **MAC:** `env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64`
- **Windows:** `env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe`

Running the file and flow testing: `go run main.go`

concurrency ~/gcp-train/sametime
~/gcp-train/mutex
~/gcp-train/sametime

Review: main.go (In Editor)

Build for any platform from any platform:

- **Pi:** env GOOS=linux GOARCH=arm go build -v main.go -o program-arm
- **MAC:** env GOOS=darwin GOARCH=amd64 go build -v main.go -o program-mac-amd64
- **Windows:** env GOOS=windows GOARCH=amd64 go build -v main.go -o program-windows-amd64.exe

Running the file and flow testing: go run main.go



Testing

Google Cloud



Let's make
something
TDD style

Survey: <https://goo.gl/forms/tj22lUtiqEODfHP2>

We done