Code blocks example

```
Package main
import(
“fmt”
)

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

Intro to Golang

October 2018

Google Cloud

Schedule

Hello, World (9 to noon)	Workstation setup (Go, IDE), your first app, language overview and exercises
Concurrency (1:30 - 2:30)	Develop a basic understanding of two of the most valuable features in Golang
Golang Challenge (2:30 - 3:30)	Write some code for fame
Mini Hackathon (3:30 - 5:30)	References, books, blogs, and getting help
Wrap Up (30 minutes)	References, books, blogs, and getting help

Google Cloud

Workstation setup

01

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.

Google Cloud

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=CJzE5LDG4NwCFeVuwQodWy4Pow
- **LiteIde** - https://github.com/visualfc/liteide
- **VIM** - https://github.com/fatih/vim-go
- **Emacs** - https://github.com/dominikh/go-mode.el

Google Cloud

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/go-train
- Slide and docs are in $GOPATH/src/assets
- Source is in $GOPATH/src/go-train/<item>

Google Cloud

Background

02

Google Cloud

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

1.11 Oct 2018
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

Google Cloud

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:

- We will not cover these in the intro - level 200

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.

Google Cloud

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

Google Cloud

Hello, World

03

Google Cloud

~/go-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

Google Cloud

~/go-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

Google Cloud

~/go-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

Google Cloud

The basics

04

Google Cloud

errors ~/go-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 ~/go-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

Google Cloud

basic ~/go-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 ~/go-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

Lunch time

05

Google Cloud

Interfaces

06

Google Cloud

Interfaces

Web links

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

Google Cloud

Concurrency

07

Google Cloud

concurrency ~/go-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 ~/go-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/tj22llUtigE0DfHP2

Google Cloud

Go + Google Cloud

07

Google Cloud

Mini Hackathon

https://cloud.google.com/appengine/docs/standard/go111/

Appengine Event Writer

Google Cloud

We done

Google Cloud