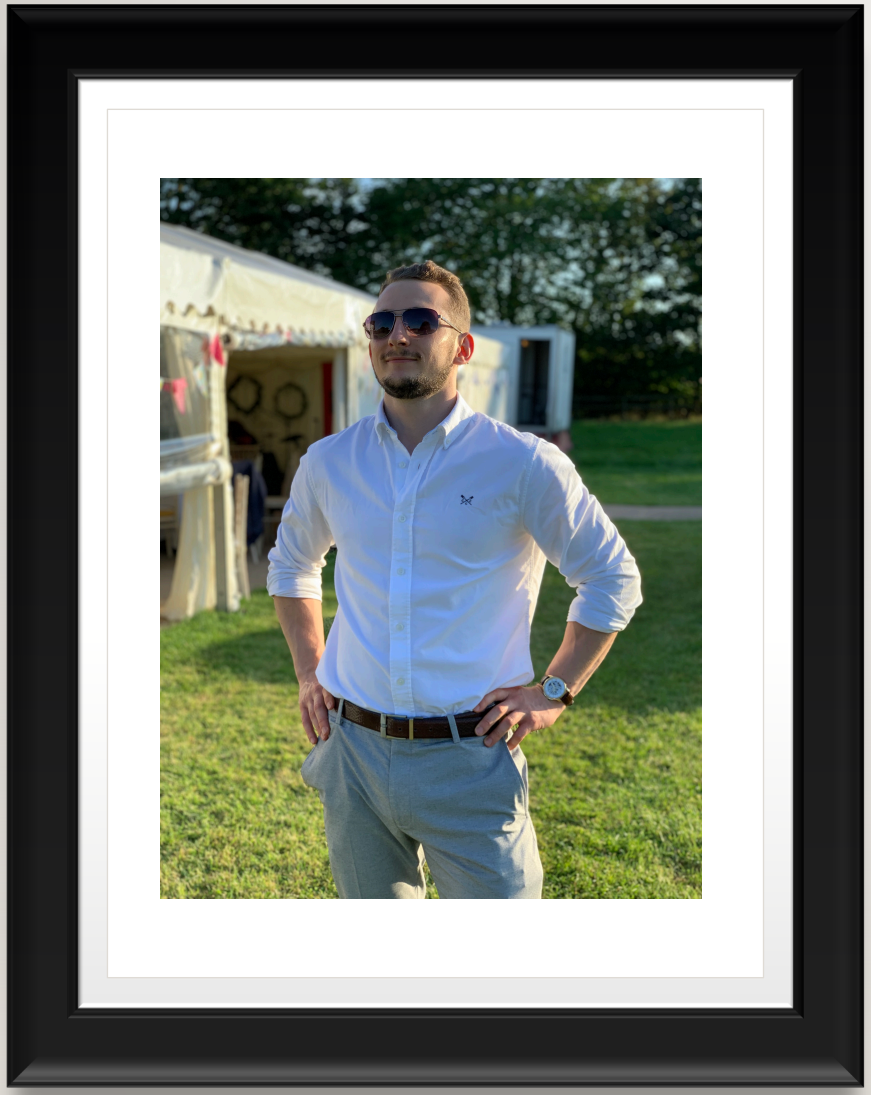


INTRODUCTION TO GO

LIAM HAMPTON

Part of the UKI developer advocate team focusing on Go.

Started as a Node.js developer in the Eclipse Codewind open source initiative.

@LiamConroyH

WHATS THE HISTORY?

- Name derived from the animal
- Open source
- Created by Google Engineers and launched in 2009
- Statically typed language for the multi core processor

@LiamConroyH

FEATURES

- Concurrency logic
- Minimalistic approach = simple
- Type-safe – code only has access to authorized memory locations!
- Garbage collected – frees up memory / less chance of a memory leak
- Fast compilation – runs directly on underlying hardware / doesn't use a VM

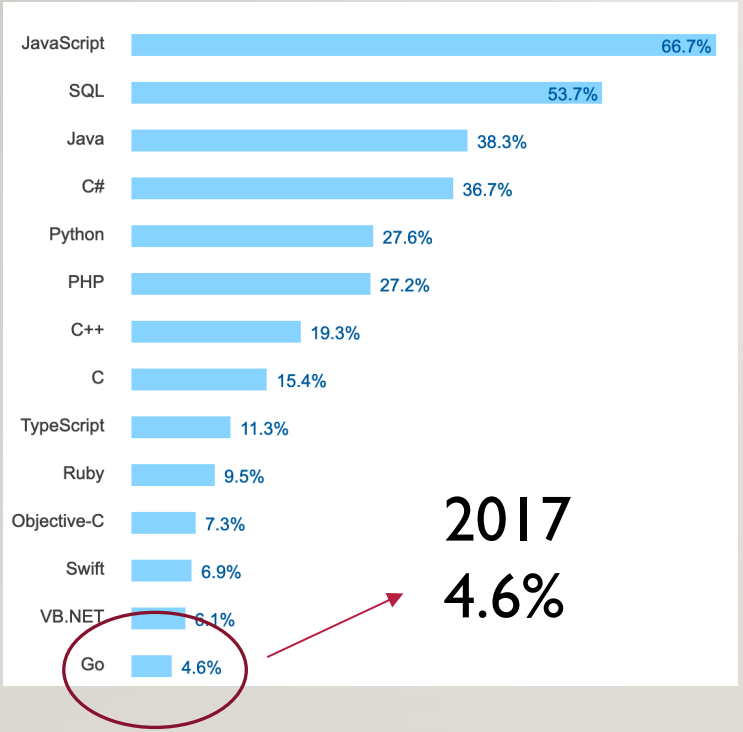
@LiamConroyH

FEATURES

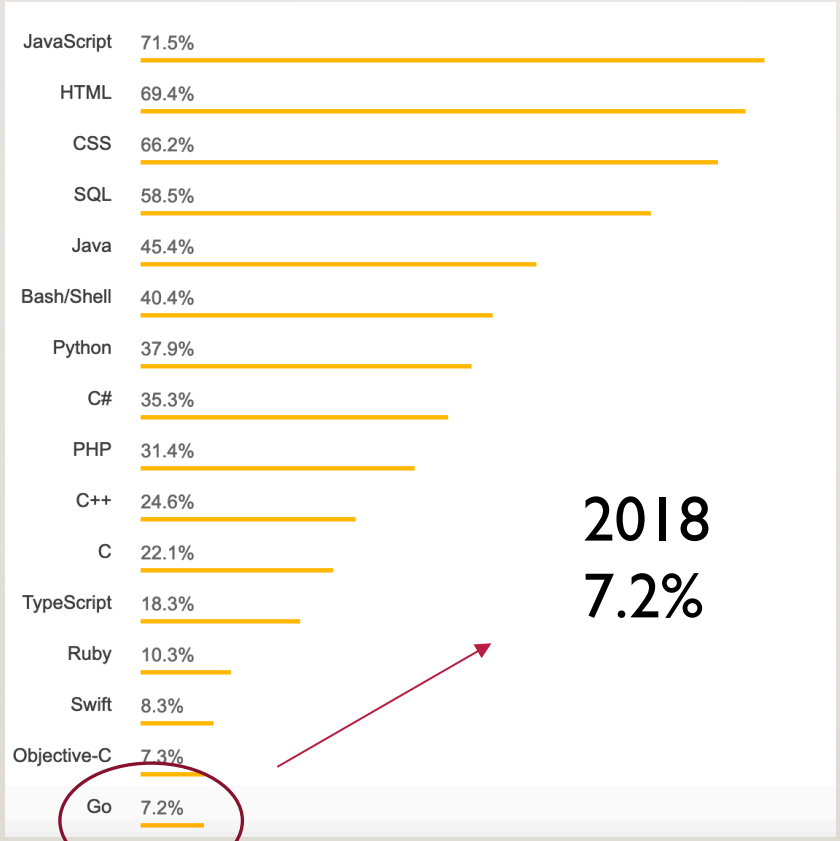
- Cross platform
- No classes or inheritance it uses Structs instead
- Multiple return values (e.g return data, nil)
- Built in tools to test and benchmark
- C + go = Cgo – Go packages can call C code!

@LiamConroyH

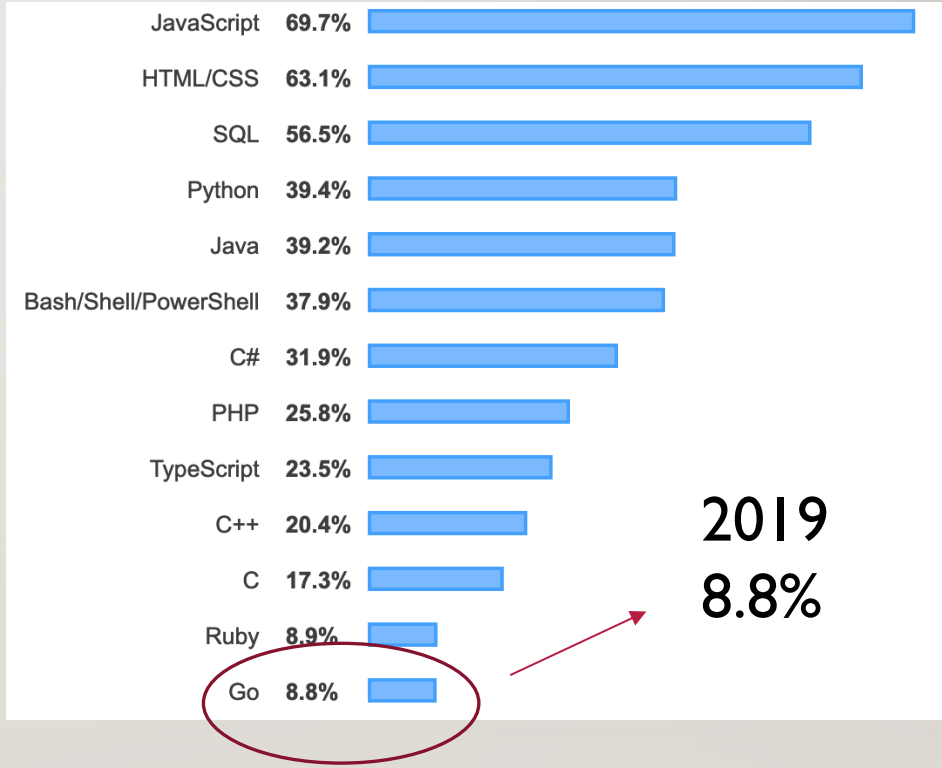
POPULARITY
(AS PER STACK OVERFLOW DEVELOPER SURVEY)



2017
4.6%



2018
7.2%



2019
8.8%

@LiamConroyH

WHERE IS IT BEING USED?

- Google – Lots of projects across the board (Kubernetes, Chrome, Proxy for YouTube etc)
- Uber – Geofence service
- Twitch – Used for the most loaded systems
- Dropbox – Used to scale its systems
- SoundCloud – Used for static analysis in realtime
- Docker - CLI

@LiamConroyH

WHY IS IT BEING USED?

- It is a great server-side language for infrastructure
- Easy to scale up with
- Easy package management
- Smart standard library – includes most of what you'll need to get started right out the box

@LiamConroyH

WHY MIGHT YOU NOT USE IT?

- It's a young language
- No standard GUI libraries
- Does it have a niche like JS and Python? – is it dangerous to fully invest in it without a specific need?

@LiamConroyH

PROJECT STRUCTURE

- Go code must be contained to a workspace – This workspace will include 3 top level directories. You will often see the follow 3 directories under the patent directory '/go'.
  - /src – contains the Go source code
  - /pkg – contains package objects
  - /bin – contains executables
- Environment Variables
  - GOROOT – Where your Go SDK is located on your machine
  - GOPATH – Defines the root of your workspace. Usually / by default it is '~/go'

@LiamConroyH

WHAT IS A STRUCT?

- Typed collection of fields
- Composite data type
- Similar to classes in OO languages
- Good for grouping data

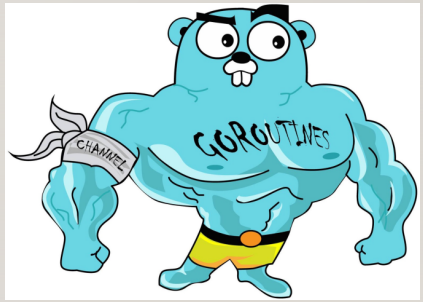
```
12    type Credentials struct {
13        ConsumerKey       string
14        ConsumerSecret    string
15        AccessToken       string
16        AccessTokenSecret string
17    }
```

```
23    func GetCredentials() Credentials {
24        creds := Credentials{
25            AccessToken:       os.Getenv("ACCESS_TOKEN"),
26            AccessTokenSecret: os.Getenv("ACCESS_TOKEN_SECRET"),
27            ConsumerKey:       os.Getenv("CONSUMER_KEY"),
28            ConsumerSecret:    os.Getenv("CONSUMER_SECRET"),
29        }
30
31        return creds
32    }
```

@LiamConroyH

CONCURRENCY

- Its super easy to create a new thread for execution
- By using the go construct before a function executes, it executes it within a new thread - commonly known as a go-routine
- Go-routine memory stacks grow / shrink on demand – can have 100's/1000's at a time, making them very cheap
- Its not parallelism but enables it – composition of independently executing functions



@LiamConroyH

CHANNELS

- The magic that connects concurrent go-routines
- Easy to send values into channels from one go-routine and receive those in another go-routine
- To pass a value to a channel use <- operator on the **right** side
- To read off of the channel use <- on the **left** side

@LiamConroyH

CHANNELS - EXAMPLE

```
1  package main
2
3  import "fmt"
4
5  func main() {
6
7          fmt.Println("Main() started")
8
9          c := make(chan string) // make the channel c
10
11         go greet(c) // start go-routine and exectute greet()
12
13         c <- "World" // pass a value onto the channel
14
15         fmt.Println("Main() stopped")
16
17 }
18
19 func greet(c chan string) {
20
21         fmt.Println("Hello " + <-c + "!") // read from the channel
22
23 }
```

```
main() started
Hello World!
main() stopped

Program exited.
```

@LiamConroyH

CROSS-PLATFORM YOU SAY?

- When you build/install an executable is created - code is compiled into binary
  - "go build –o test" creates an executable of your code in the current dir called test
  - "go install –o test" creates an executable of your code in the /bin dir (accessible from anywhere if its on your GOPATH)
- All imports are included
- Because of this, the host system does not need a runtime to execute the binary!
- To see the list run the command $ go tool dist list

@LiamConroyH

WHAT ARE YOU DOING IN THIS WORKSHOP?

Build a simple Golang application

Deploy app into Cloud Foundry

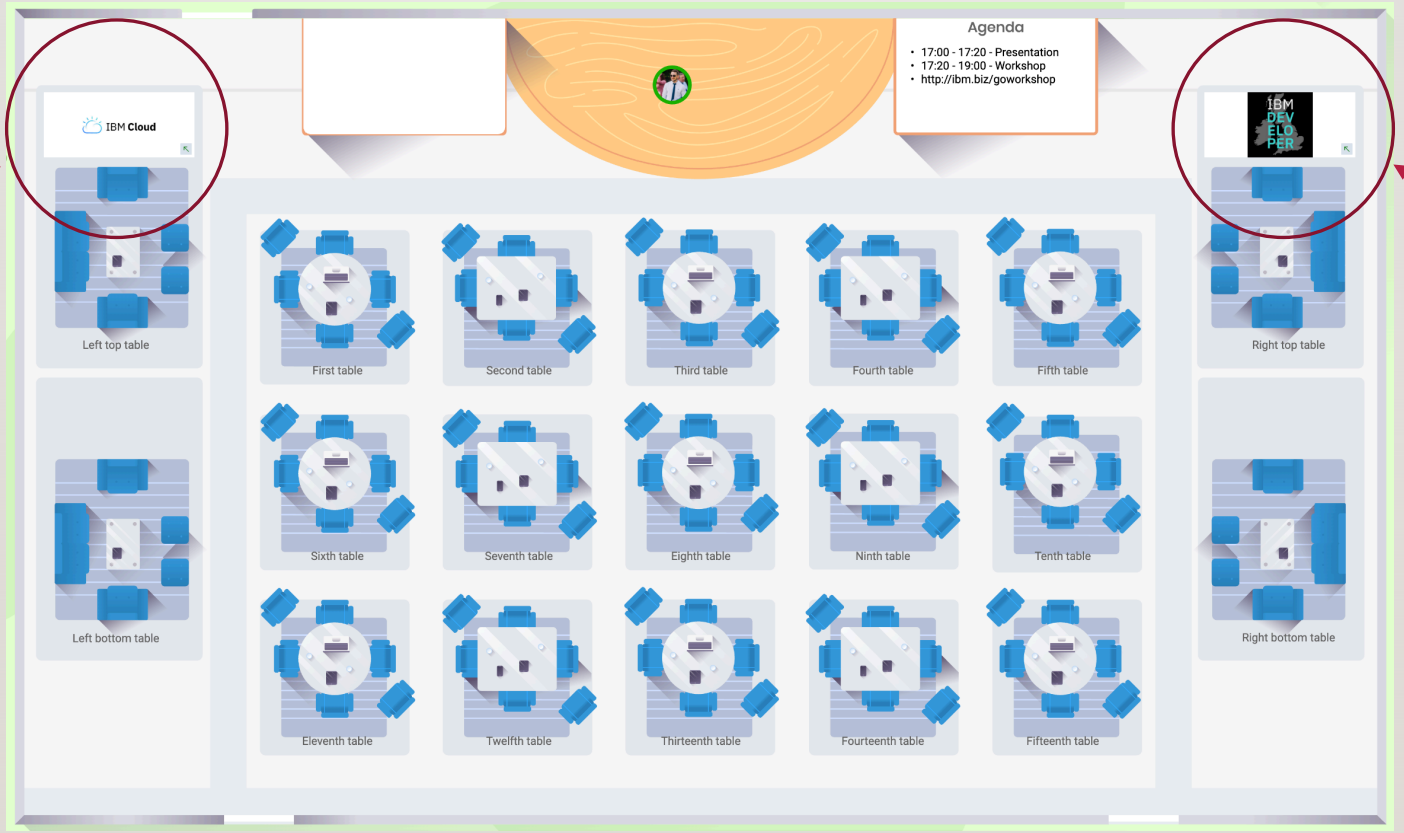
Call an API

Call Twitter API & redeploy app

@LiamConroyH

WORKSHOP MATERIALS

IBM Cloud sign up



GitHub repository

@LiamConroyH

IBM CODE MEETUP - CODE OF CONDUCT

A safe, respectful, comfortable and harassment-free environment for attendees

ibm.biz/CodeofConduct

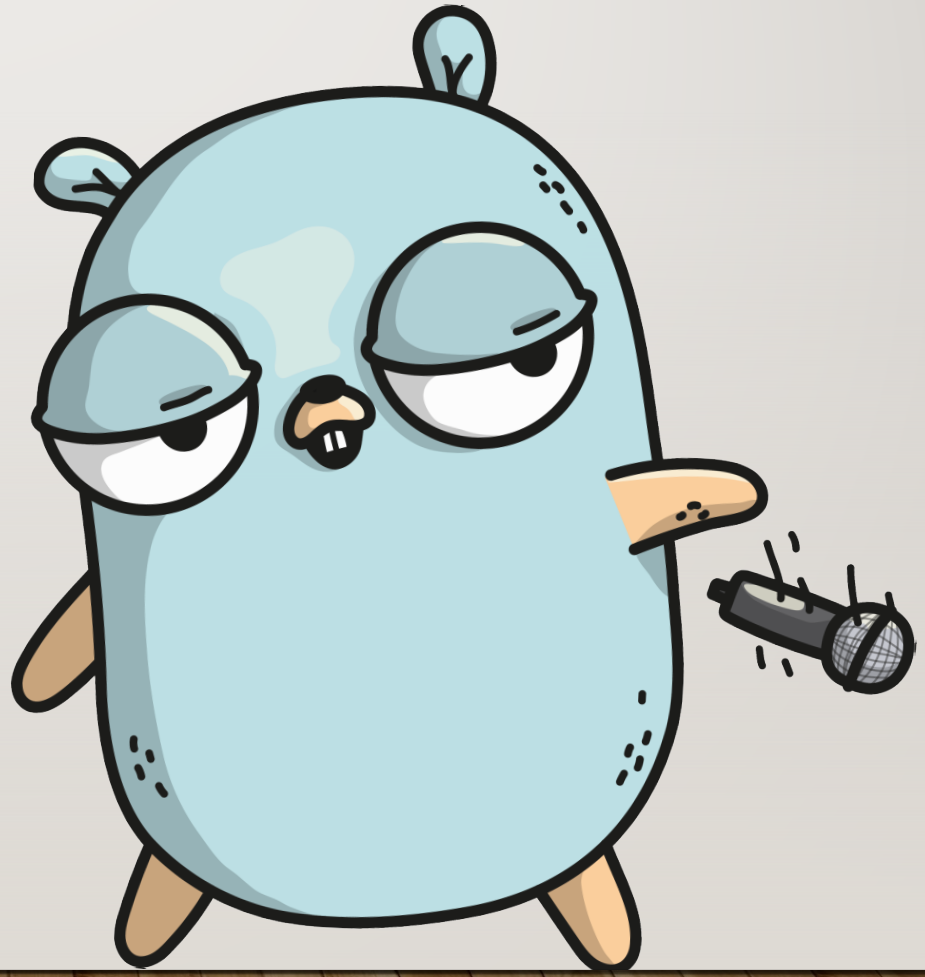
- **Our Code**
- ➢ Be nice
- ➢ Be Respectful
- ➢ Participate but don't disrupt
- ➢ Don't break the law

- **Report Your Concerns**
- Your instructor
- Any IBM'er
- DM via Meetup.com
- DM via @codemeetup
- Email stewaa3@uk.ibm.com
- Phone or text 07802765745

- **We will act on incidents**
- During events – listen & act
- 2 weeks – investigate
- 30 days – further action

- **What we can do**
- ➢ Ask you to leave
- ➢ Written warning
- ➢ 3 months breather
- ➢ Remove from the group

WORKSHOP
LETS CODE



@LiamConroyH