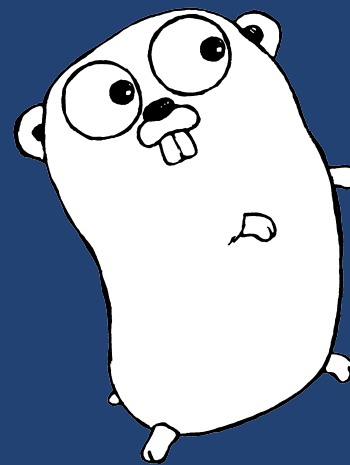

MAKING WEBAPPS WITH GO

AKA IMPORT "NET/HTTP"



WHOAMI?

- George
- Don't do webapps often, but YOLO.
- I like Go
 - And Python
 - And Ruby
 - And Clojure
 - Not PHP though
 - PHP is *literally worse than Oracle*



OHOHOHOHOH.ALSO.

- I'm sarcastic.
- And feel free to interrupt with questions or tell me to slow down.
- I don't do well with 15 minute presentations.
 - And there's a lot to cover
 - I'll have to Dart through it all.
 - Ha. Ha. Ha.



COOL. LET'S GO

(Laugh now)

WHAT IS GO?

- Systems language.
- Statically typed.
- Not stupid.
 - In the 'C' way
 - In the 'PHP' way

NOT STUPID IN THE 'C' WAY

- All arrays/vectors/lists/whatever-you-want-to-call-them auto-resize
- Built-in dict (map)
- Tons of built-in libs for all sorts of things (like Python, Ruby, etc)
- Strings actually exist (SHAM-WOW)
- Garbage Collected
- Strongly typed
 - `int a = *(int*)"hello, world!"` panics, not silently succeeds



NOT STUPID IN THE 'PHP' WAY

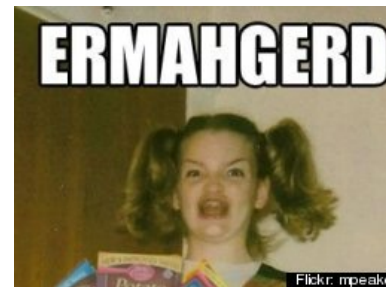
- Statically typed
 - You can't do:

```
myNum := 3  
myNum = "hello!"
```
- If you try to do `"str1" == 12345`, it fails **at compile time**
 - **AHHH COMPILES NOOOOOOOOOOOOOO aKLFJDKLFJSDKLFJLKFJDKLSJ**
 - Doesn't throw at runtime
 - Doesn't not-throw at runtime either
 - (and return False.Thanks PHP.That's TOTALLY WHAT I WANTED.)



ONE LAST BIT...

- Yes, compiles. Not the way you're used to.
 - You'll see how ezipz it is in T-minus 30 seconds
 - Yes, ezipz is a word.



OUR FIRST PROGRAM – A WEB SERVER!

- Ezpz.
 - Just like the Systems one, right?
 - No.

WHERE DO COMPILE?

- ``go run foo.go`` automagically compiles `foo.go`, and runs the resulting binary
- Yes, it feels scripting language-y
- Yes, you can let go of ``make``, ``javac``, ``ghc``, ...

MOAR!

- As much as “Hello, World!” is basically the next Facebook, let’s do more.
- First, let’s download a third-party lib to help us out.



WHAT IS THIS WIZARDRY?!

- Like Python/Ruby, Go has a built-in package manager
 - ``go get github.com/dyreshark/yolo``
 - Uses Git to fetch repo
 - Puts it in special place (see: `$GOROOT/src/github.com/dyreshark/yolo`)
 - Now all I need to do is:

```
import "github.com/dyreshark/yolo"

func hasSwag() bool {
    return yolo.Swagarific;
}
```

COOL. DIFFERENT PAGES.AND TEMPLATING.

- Using our newfound magic, let's do stuff

COOL. USERS + LOGIN PAGE.



EVERY LOGIN PAGE NEEDS HTTPS

- Go has such character
 - So swag.

HOW ABOUT WE MAKE A CHATROOM?



AND THEN THE SKIDDIES COME OUT TO PLAY...





NOOOOOO IT ESCAPES IT ALL FOR US

- We don't have to do *anything extra* to be protected against attacks that take sites down every day
 - HERESY I SAY, HERESY!

SUMMARY OF THE HURRICANE THAT WAS THE LAST 10 SLIDES

- Made a webserver
- Downloaded 3rd party repo
- Used it to make landing page/login page
- Added HTTPS ezpz
- Added chatroom
- Added custom JSON tags so we could serialize/deserialize
- ...And templates auto-save us from lots of pretty crazy things for free.



ISS024E012814

BUT GEOOOOOOOOOOOOOOOOOOORGE

- I CAN DO THIS IN PYYYYYYYYYYYYYTHON WHY WOULD I USE GOOOOOOOOOOOO
 - Feel free to replace PY..YTHON with RU..UBY/PHP..P/CLO..OJURE/etc.

SAFETY AND PERFORMANCE

- First, safety.
 - “Bro I optimized my program. It doesn’t work anymore, but it’s SO FREAKING FAST”

OMG GUISE I JUST LAUNCHED A RAILS WEBSITE LOOK AT IT





SO HOW DOES GO PROTECT US?

- Simple. Statically typed.
- Compiler verifies you're not doing any crazy trippy stuff at compile time
 - Which is generally better than it happening at runtime.
- Will this catch **everything**?
 - No.
- Does it eliminate an **entire class of bugs**?
 - Mostly
 - You can still shoot yourself in the foot if you try.

SO HOW ELSE DOES GO PROTECT US?

- Returns errors instead of throwing.

Decoding int:

```
n, err := strconv.Atoi(s)
```

```
if err != nil { /* Someone gave you a bad int! */ }
```

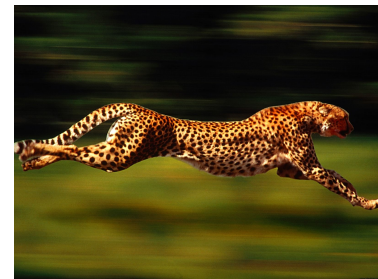


Y DIS GUD?

- You have to explicitly ignore it, unlike
 `n = int(s)`
 - If `s` isn't a valid int, this throws.
 - Oh also, if `s` isn't a string at all, this will throw.
 - You know. Because it can be anything.

PERFORMANCE

- Suffice to say: interpreted languages are bad at going fast.
- Go is compiled! And good at Going fast!
 - Also has amazing concurrency features that make it go *crazy fast* when dealing with the web/your hard drive
 - So much so, Google powers **all** of dl.google.com with it (or so they say)
 - But that's beside the point.



SHOULD I USE GO?

- Depends entirely on what you want
 - Do you want...
 - Everything already done (admin console, autogenerated pages to deal with db, enforced file structure, etc.)?
 - An incredibly mature framework that's been around since 2003?
 - Tons of tutorials everywhere that point you in different directions?
 - Do you not care about...
 - Speed (so long as it isn't this-takes-30-hours-to-make-a-page slow)
 - Types verified at compile-time (because you always make perfect unit tests)
 - Being locked down by a bigger framework

NO.THIS IS A NO-GO.

bad pun # 95

SHOULD I USE GO?

- Do you want...
 - Simple framework you can build on top of?
 - Blazing fast execution, compile-time-checked type-safety?
 - Solid documentation and a few good guides to push you in the right direction?
- Do you not care about...
 - Having the most mature libraries (so your code will magically get faster each release)
 - Having to “compile” and deal with a static type system (that’s not stupid like C’s or Java’s)

PROBABLY.

- If you're primarily gunning for "minimalist API", consider Flask too.
 - I've used it a few times in the past, it's really legit
 - Señor Barnes is giving a talk on it soon
 - You should go [to his talk]. He's funnier than I am.
 - And really smart too.



SUMMARY

- Go isn't for everyone
 - Gee don't I sound insightful by saying "X isn't for everyone"?
 - I'm basically Socrates.
- It's great if you want less bugs + faster code.
 - And don't mind that it's new/you're not given half of your website already
- But we **did** "make" a chat website with logging in/user registration/etc. in not-many LOC
 - Which is admittedly really cool for a "systems language"

THANKS, FRIENDS!

