An Exploration of Go

Admin

- Syllabus
  - Just to cover all the bases
- Who is here?
  - Reverse roll call

This Class

- What is this class all about?
  - A fairly deep dive into the go programming language
    - Fairly deep, won't be looking into the internals of the compiler
  - Go
    - a new(ish) programming language.

Why Go?

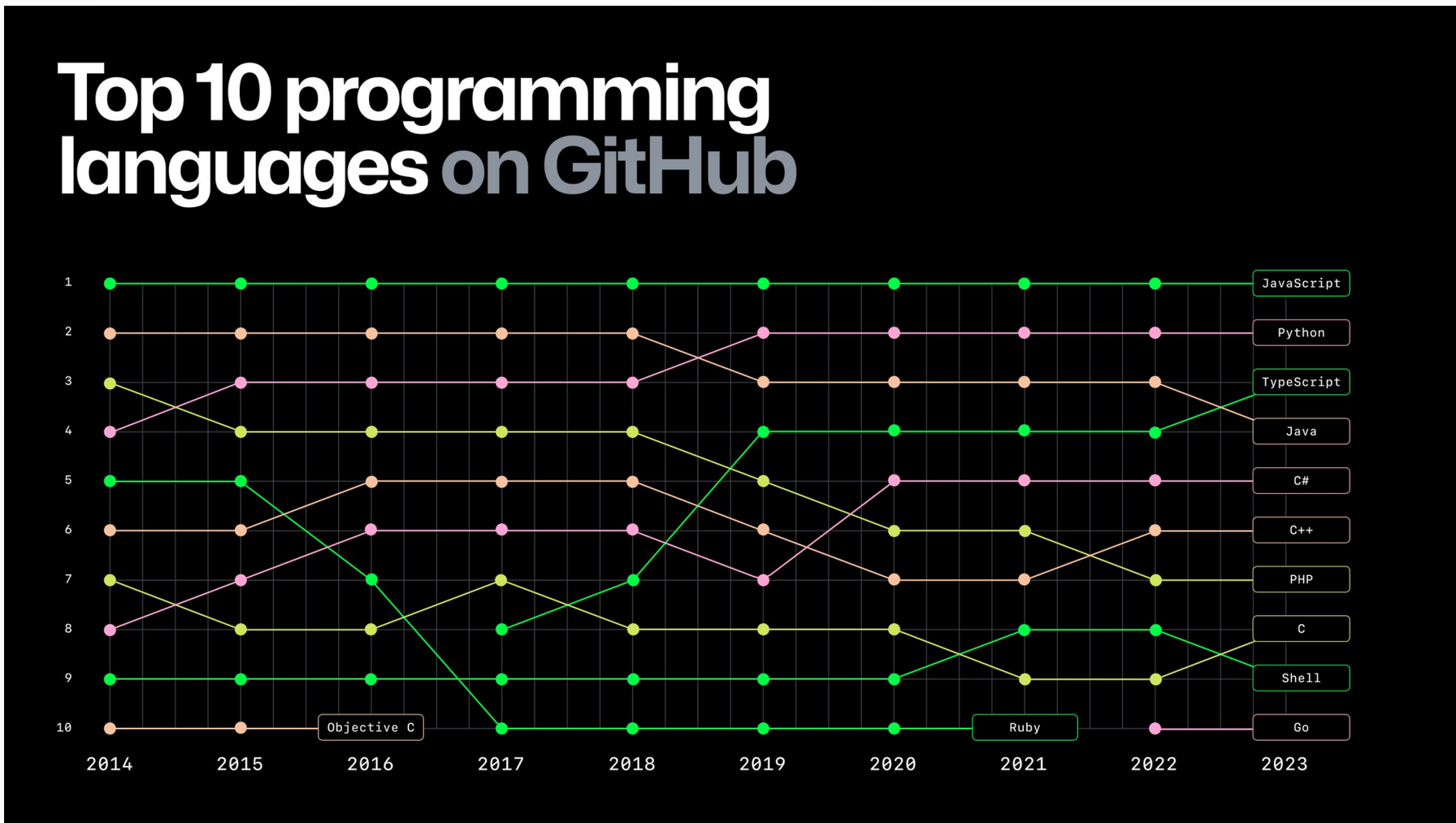
- 14 years ago I started to hear a lot of buzz about dynamically typed languages
  - So I ran a seminar on Python and Ruby – lots of buzz at time
- 10 years later dynamically typed languages were no longer ‘in’
  - So 6ish years ago I ran a rust vs go seminar.
    - Similar buzz about these languages
  - Students’ consensus:
    - Go is mostly fun and fairly easy to use
    - Rust is powerful and hard to use

Why Go?

- IEEE Programming language usage (probably best)
  - https://spectrum.ieee.org/the-top-programming-languages-2023
- And in recent open source projects it is really widely used
  - https://madnight.github.io/githut/#/pull_requests/2023/4
  - Number 3 in open source projects on github for all 4 quarters of 2023

Why Go?

- In All of Github (including non-open source) go is only a top 10 language.
-



Why this Class?

- Why am I doing this class?
  - Because this is a fantastic excuse to learn new promising programming language with enthusiastic fellow learners
  - Maybe Go will be huge someday
    - Has seen promising growth in the recent years
    -
  - Or maybe go will be the next ruby/visual basic (both flash in the pan then second tier languages)
  - But it will be fun to learn
    - And it can do really neat things

Very Punny

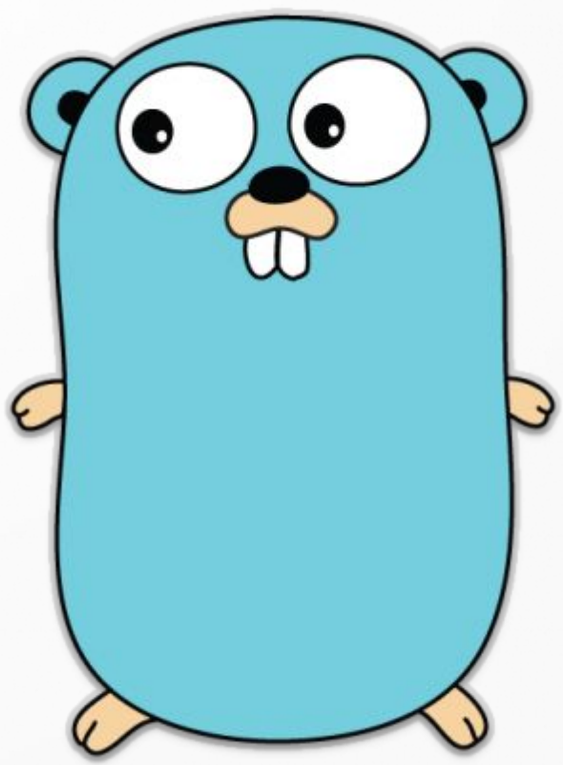
- You call a python programmer? .....
- What do you call a Go programmer?

Very Punny

- You call a python programmer? a pythonista
- What do you call a Go programmer?

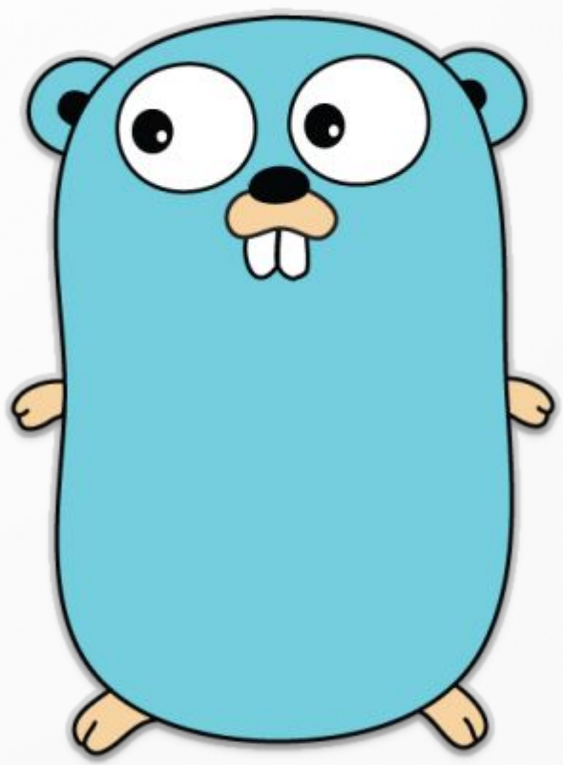
Very Punny

- What do you call a Go programmer?
  - A gopher
- The pun is better when you see the go tools



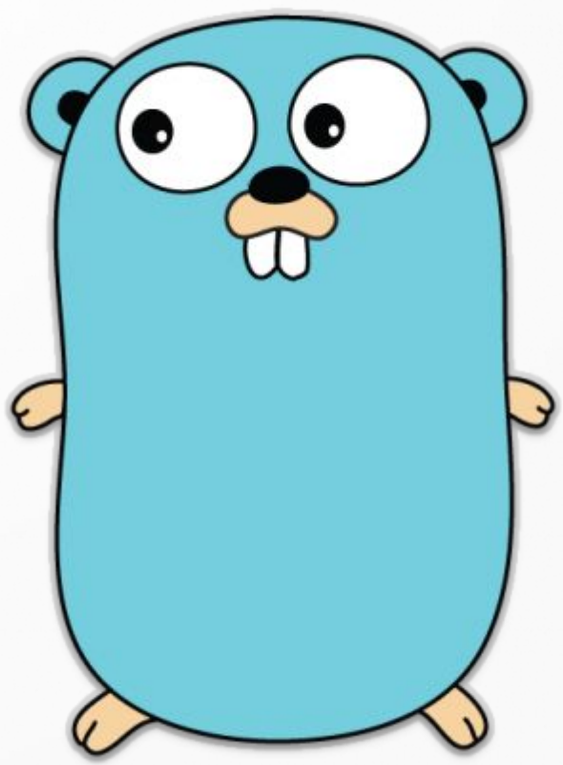
Very Punny

- What do you call a Go programmer?
  - A gopher
- The pun is better when you see the go tools
- How do you install a package at the python command line by default?



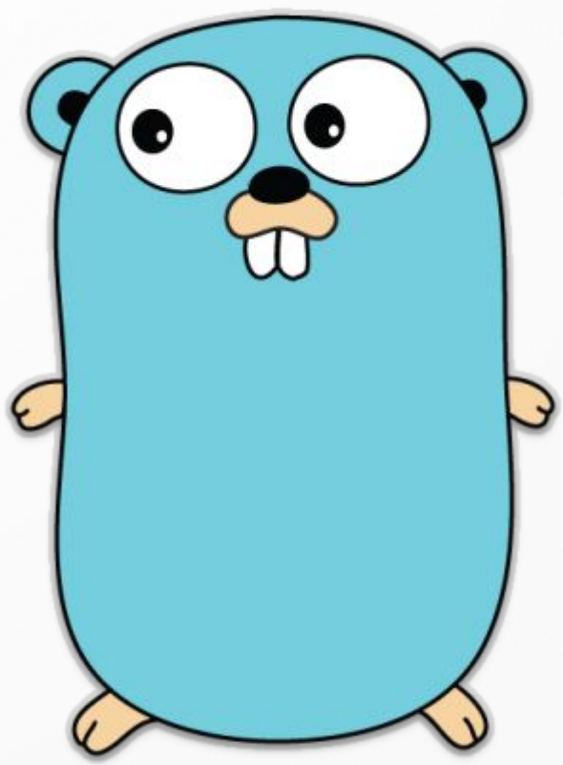
Very Punny

- What do you call a Go programmer?
  - A gopher
- The pun is better when you see the go tools
- How do you install a package at the python command line by default?
  - pip install <package>



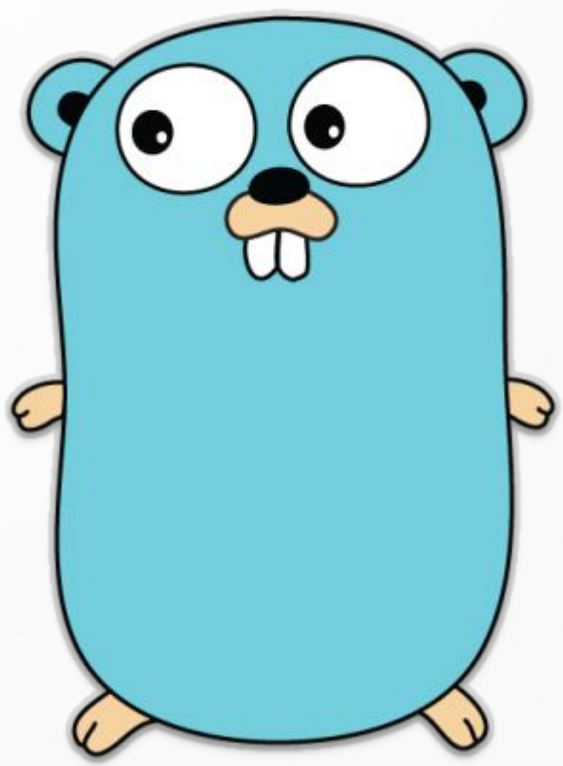
Very Punny

- What do you call a Go programmer?
  - A gopher
- The pun is better when you see the go tools
- How do you install a package at the go command line?



Very Punny

- What do you call a Go programmer?
  - A gopher
- The pun is better when you see the go tools
- How do you install a package at the go command line?
- go get <repo>
- gofer this, gofer that
  - Though modules are replacing go get.



Hello world

- Following the tradition set down by Brian Kernighan for BCPL, lets look at the “hello world” program for these languages
  - See the minimal program that produces output

Go (Golang) Hello World

- The canonical hello world from https://tour.golang.org/welcome/1
- package main
- import "fmt"
-
- func main() {
- fmt.Println("Hello, 世界 ")
- }
- Note the deliberate highlighting of multilanguage support

boring

- Go sets out to be ‘boring’
  - Code in industry/real world is often read far more often than written.
  - Go wants to present easy to read code to your colleagues.
- Go 1.x compatibility promise
  - Go adds features, but doesn’t break any existing code.
- Deliberately slow to add features:
  - https://drewdevault.com/2019/03/25/Rust-is-not-a-good-C-replacement.html

Lets try go in golang

- Command line possible,
  - Post go 1.11 module support
- I suggest IDE usage.
  - We’ll start with go module usage (discuss)
  - Might go with go dep later
- Lets create a go project in golang

Trial run

- Now create file and do hello world
- Run – what happens?

Assignment

- End of intro
  - Now go install the compiler
    - at least go 1.21 (go 1.22 comes out in feb and has some changes to for loops – I don’t know if we will need that one or not)
  - Install IDE tooling – get goland from jetbrains
    - As students you can get the professional version for free.
  - Install Git if you haven’t already.
  - Send me your github username/id (even if you did it for a previous class)
  - See resources page.
  - Read preface and chapter 1 in Learning Go