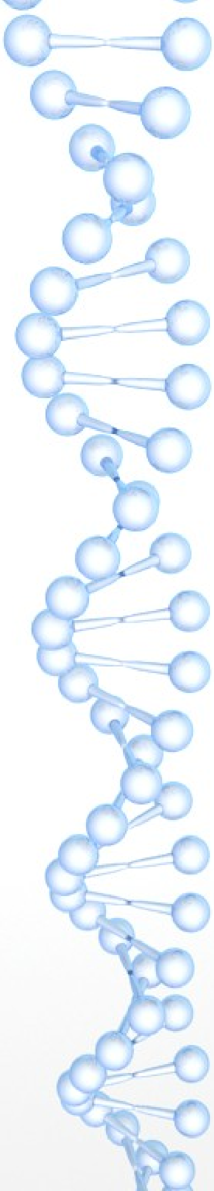
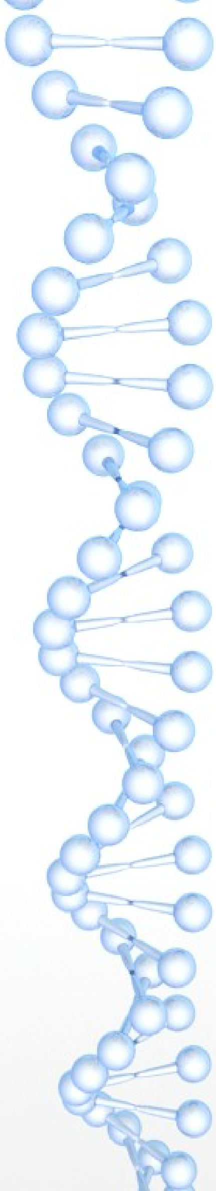
golang



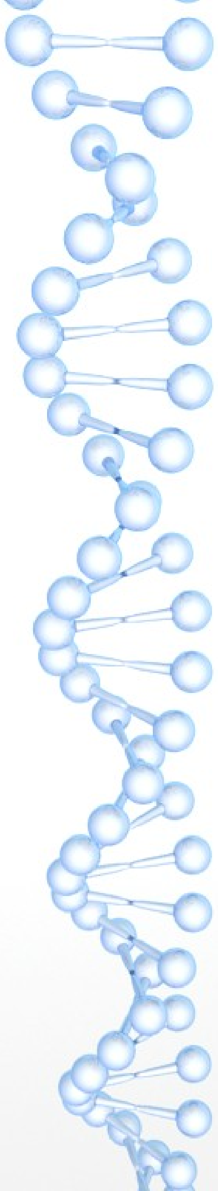
Ravi Jaya



Intro....

- Go language is a programming language initially developed at Google in the year 2007.
  - Robert Griesemer, Rob Pike, and Ken Thompson

Robert Griesemer, Rob Pike and Ken Thompson

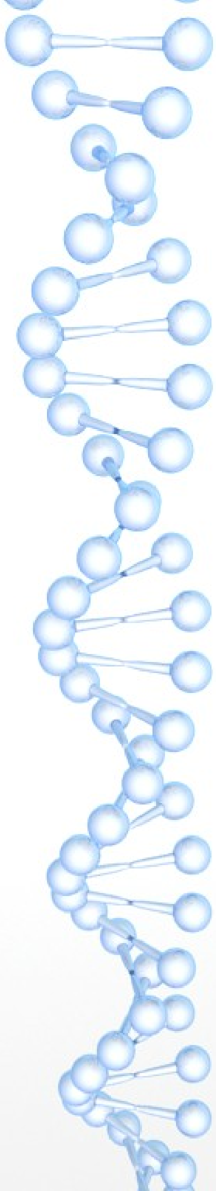


features....

- statically-typed language having syntax similar to that of C.
- garbage collection
- type safety
- dynamic-typing capability,
- built-in types such as
  - variable length arrays
  - key-value maps.
- Concurrency

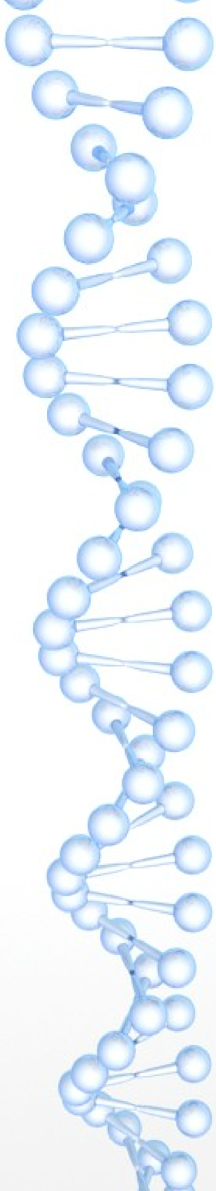
Go is as programming language

- Compiler doesn’t issue warnings, just errors
- Unused local variables are an error
- fmt is Go’s single way to format output



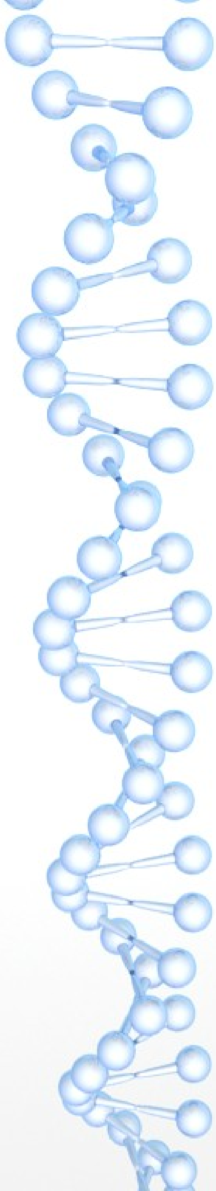
Like C

- Uses its own compiler to create itself
- Strongly typed (must specify data type in commands)
- Used by developers of C and Unix



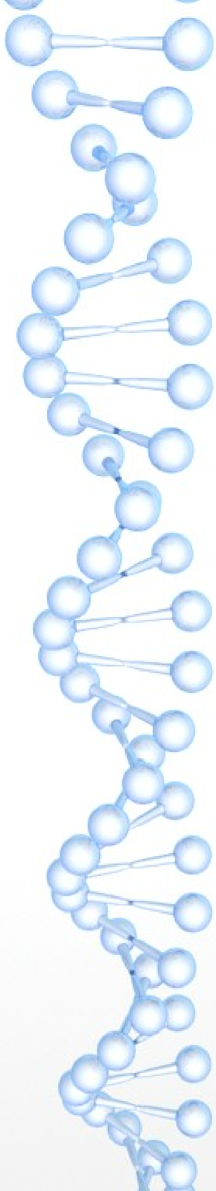
Unlike C

- Go has no pointer arithmetic



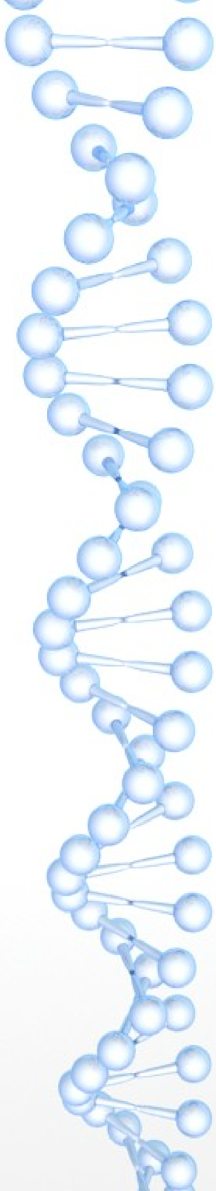
Like both Java and C

- needs to be built by a compiler (not interpreted like Python)
- double slashes for comments



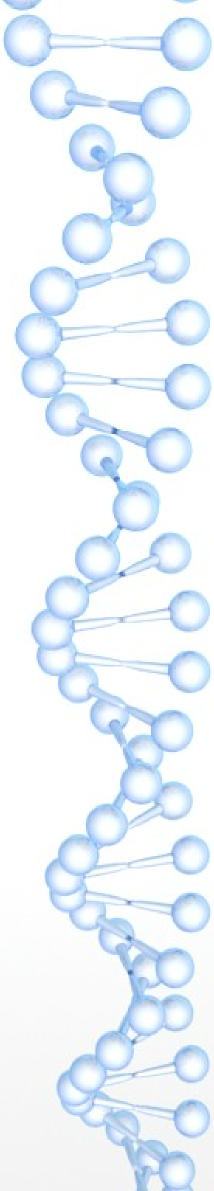
Like Java

- Is memory safe by default
- Networking is baked into the standard library and runtime
- Collects garbage automatically

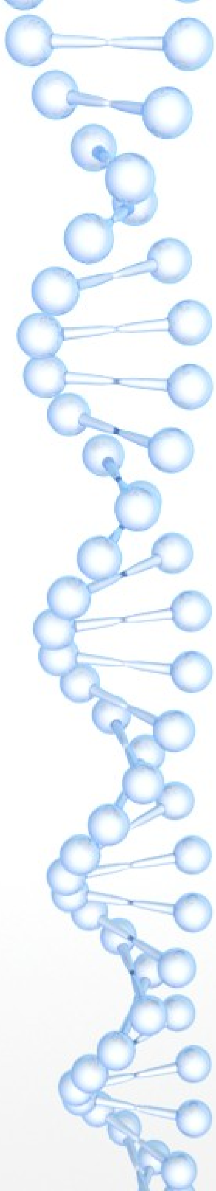


Unlike both

- fast compile to native code
- low GC pause (100 microseconds)
- Memory is zeroed if not explicitly initialized
- there are no parentheses surrounding the three components of the for statement

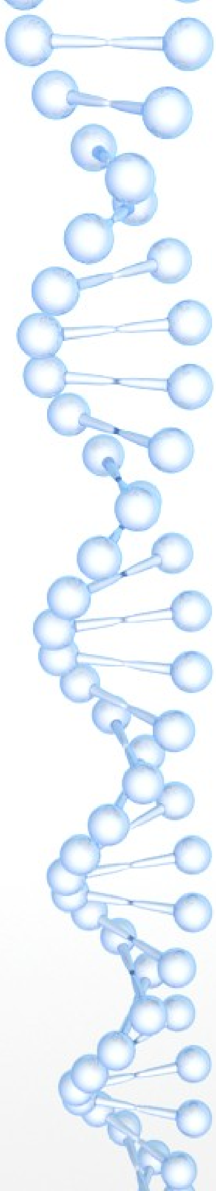


Let’s get started.



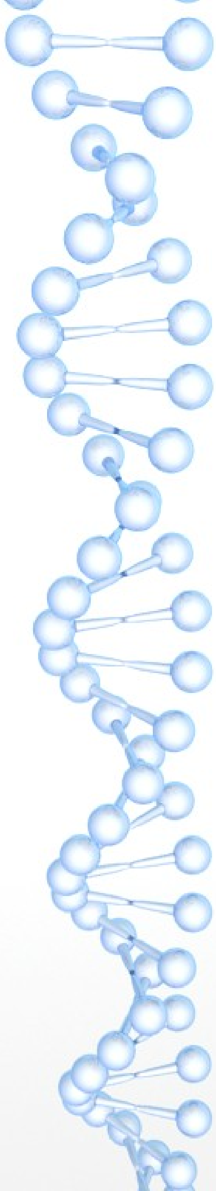
Code organization

- Overview
  - Go programmers typically keep all their Go code in a single workspace.
  - A workspace contains many version control repositories (managed by Git, for example).
  - Each repository contains one or more packages.
  - Each package consists of one or more Go source files in a single directory.
  - The path to a package's directory determines its import path.

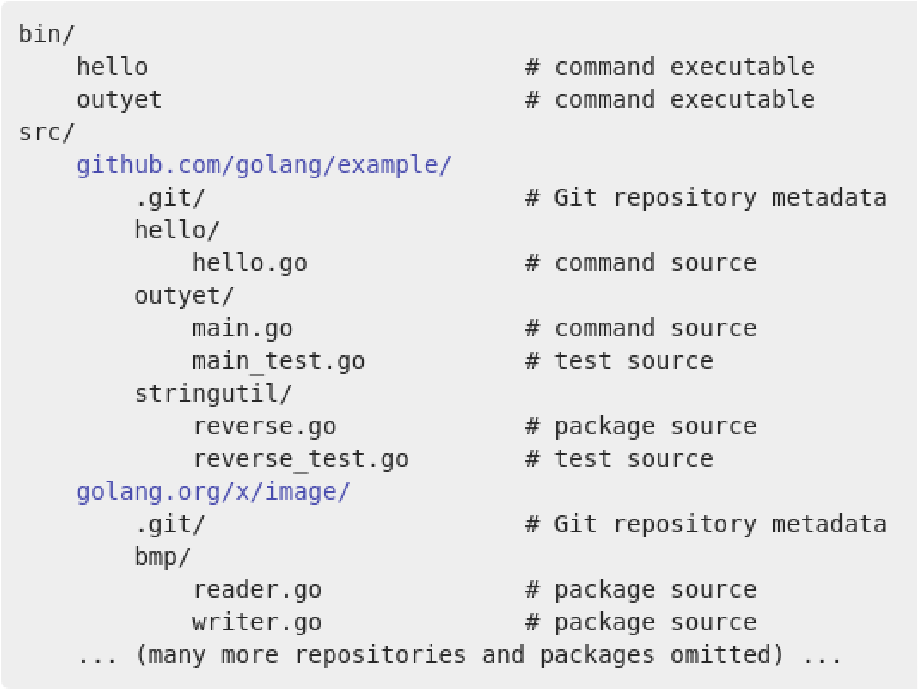


Workspaces

- A workspace is a directory hierarchy with two directories at its root:
  - src contains Go source files, and
  - bin contains executable commands.



workspaces



```
bin/
    hello                          # command executable
    outyet                         # command executable
src/
    github.com/golang/example/
        .git/                      # Git repository metadata
        hello/
            hello.go               # command source
        outyet/
            main.go                # command source
            main_test.go           # test source
        stringutil/
            reverse.go             # package source
            reverse_test.go        # test source
    golang.org/x/image/
        .git/                      # Git repository metadata
        bmp/
            reader.go              # package source
            writer.go              # package source
    ... (many more repositories and packages omitted) ...
```