Get Going with Golang

Beaver Barcamp 17

8 April 2017

Cody Malick

Getting Started

- Why Go?
- Quick technical details
- Download and Install Go
- Hello World
- Concurrency

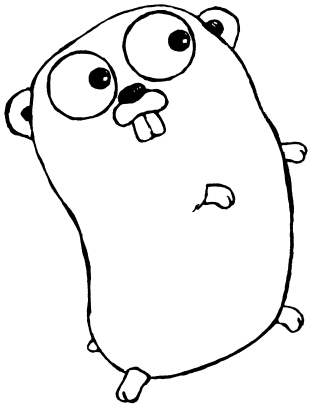
Why Go?

Go, released to the public in 2009, was designed to be simple, fast, and powerful. Go is completely open source and has support on all major platforms (Linux/Mac/Windows/FreeBSD).

Check it out:

github.com/golang (https://github.com/golang)

Go had a few specific design goals:
- Ease of programming
- Efficient compilation
- Performant execution
- Simple concurrency



Quick technical details

Go is:

- A compiled language
- Statically typed
- Garbage Collected

Go is not:

- An Object Oriented Programming Language, there is no such thing as a class

Download and Install Go

Install Docs:

golang.org/doc/install (https://golang.org/doc/install)

Downloads Page:

golang.org/dl/ (https://golang.org/dl/)

Debian (Go 1.6)

```
$sudo apt-get install golang
```

Fedora (Go 1.8)

```
$ sudo dnf install golang
```

CentOS (Go 1.6)

```
$ yum install golang
```

Hello World!

```
package main

import (
    "fmt"
)

func main() {
    fmt.Printf("Hello World!\n")
}
```

Run

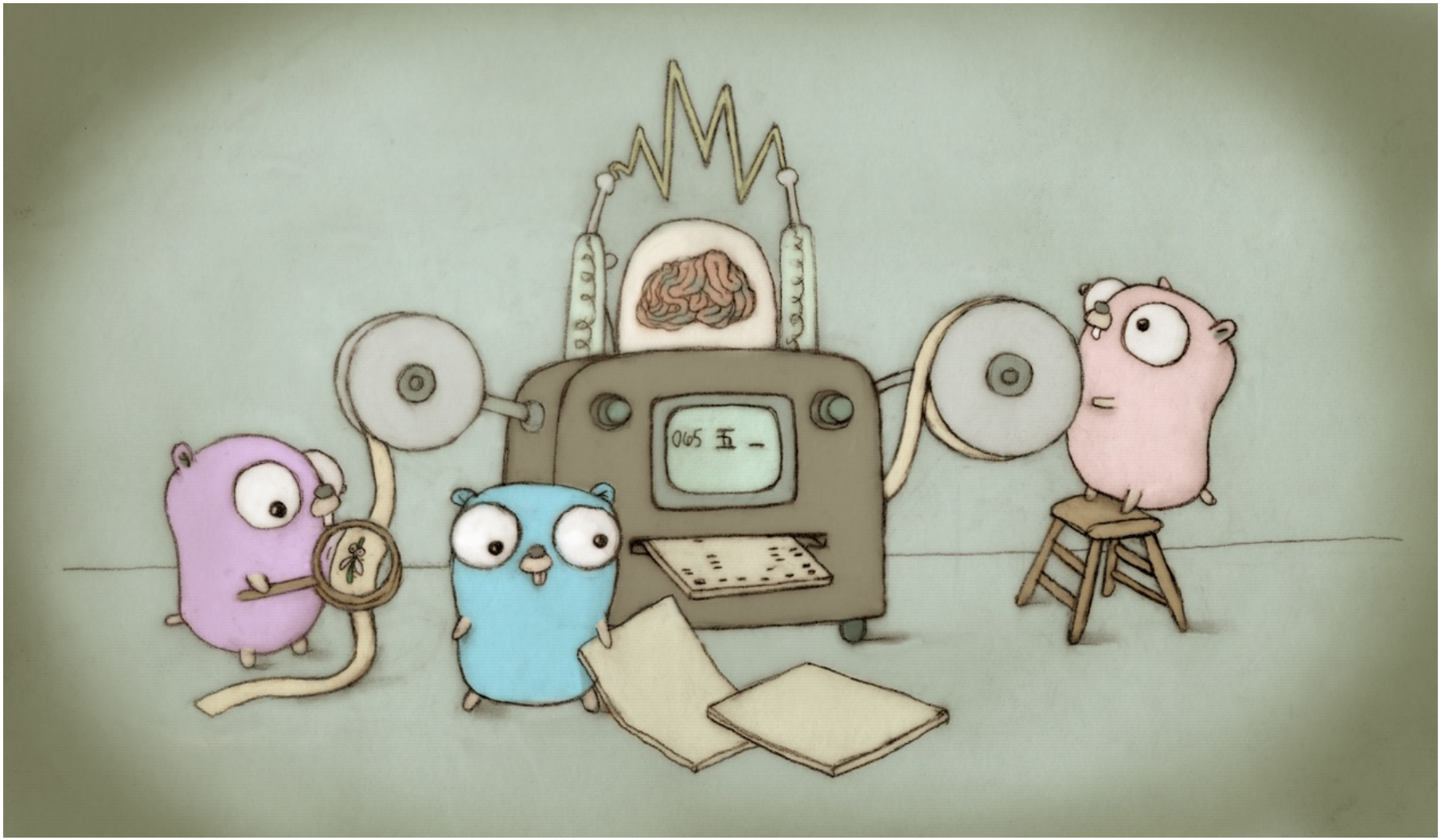
To build:

```
$ go build
```

To run:

```
$ ./main
Hello World!
```

We did it!



Concurrency

Do not communicate by sharing memory; instead, share memory by communicating.

golang.org/doc/effective_go.html#sharing (https://golang.org/doc/effective_go.html#sharing)

Finish this slide

Goroutines

Lightweight Threads, shared address space

Mutexes, Semaphores, and sharing memory

The general method of communicating is through shared memory. This is done by regulating access to critical sections of code using mutexes, semaphores, or other locking mechanisms.

TODO: C pthread example

The Go Solution: Channels

While Go has mutexes and semaphores, idiomatic Go uses channels for thread communication
where possible.

Share memory by communicating:

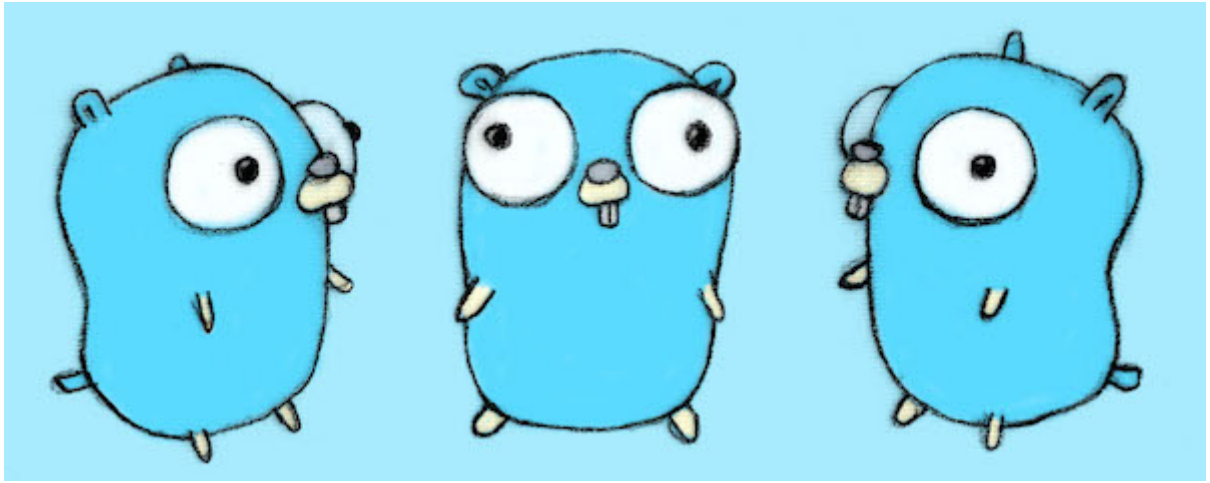
```
// Simple integer channel
jackie := make(chan int)

go func() {
    for i := 0; i < 10; i++ {
        jackie <- i
        time.Sleep(time.Second)
    }
}()

go func() {
    for {
        value := <- jackie
        fmt.Println(value)
    }
}()
```

Slide Download

cmalloc.com/public/static/slides.pdf (http://cmalloc.com/public/static/slides.pdf)



Thank you

8 April 2017

Tags: Golang, Hello World, Introduction (#ZgotmplZ)

Cody Malick

cody.malick@gmail.com (mailto:cody.malick@gmail.com)