

# Why Go?

NANOG 69 Washington, DC

7 February 2017

James Boswell

Principal Engineer, Charter Communications

Thank you!

# what this is not....

Not here to bash Python .... I ❤️ Python

Not a tutorial

Not a comprehensive review, just my experiences

Not a plug for Google .... (psst...Google ping me ;-)



Follow along here:

[talks.godoc.org/github.com/jamesboswell/nanog69/whygo.slide](https://talks.godoc.org/github.com/jamesboswell/nanog69/whygo.slide)

(<http://talks.godoc.org/github.com/jamesboswell/nanog69/whygo.slide>)

# So why did I start in Go?

## Problems with Python

I wanted to write Python 3, but....

- Community is split Python 2 vs Python 3
- 2.7 still supported for 3 more years (officially\*) and then ???
- many modules not ported from 2 to 3
- module compatibility....ok so virtual env everything

Also challenges with scaling code concurrently

*"Hard"* to do things concurrently

Multiprocessing package works... until it doesn't

A former colleague told me to learn Go...we need Go people here...

**CHALLENGE ACCEPTED**

# Hello world in Go

```
package main

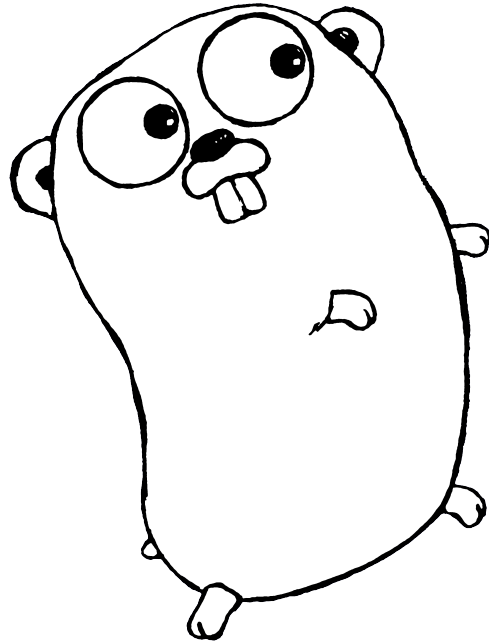
import "fmt"

func main() {
    fmt.Println("Hello world!")
}
```

Run



Why Go?



# Unicode support with no fuss

```
package main

import "fmt"

var hello = []string{"Hello", "สวัสดี ครับ", "你好", "😊"}

func main() {
    for _, h := range hello {
        fmt.Printf("%s, world!\n", h)
    }
}
```

Run

Go is fully UTF-8 compliant!

```
Hello, world!
สวัสดี ครับ, world!
你好, world!
😊, world!
```

# Type Safety

## Sample Python code

```
#!/usr/bin/env python2
number = 16
print(number, type(number))
number = '42'
print(number, type(number))
```

Run

```
$ python2 badinteger.py
16
42
```

Can you avoid this in Python? Yes you can but you **MUST** declare your types

An integer is always an integer

```
func main() {
    number := 16
    fmt.Println(number)
    number = "42"
    fmt.Println(number)
}
```

Run





# Dynamic Typing fun...

*Dynamic* compilation can result in errors that exist in code, but are not detected

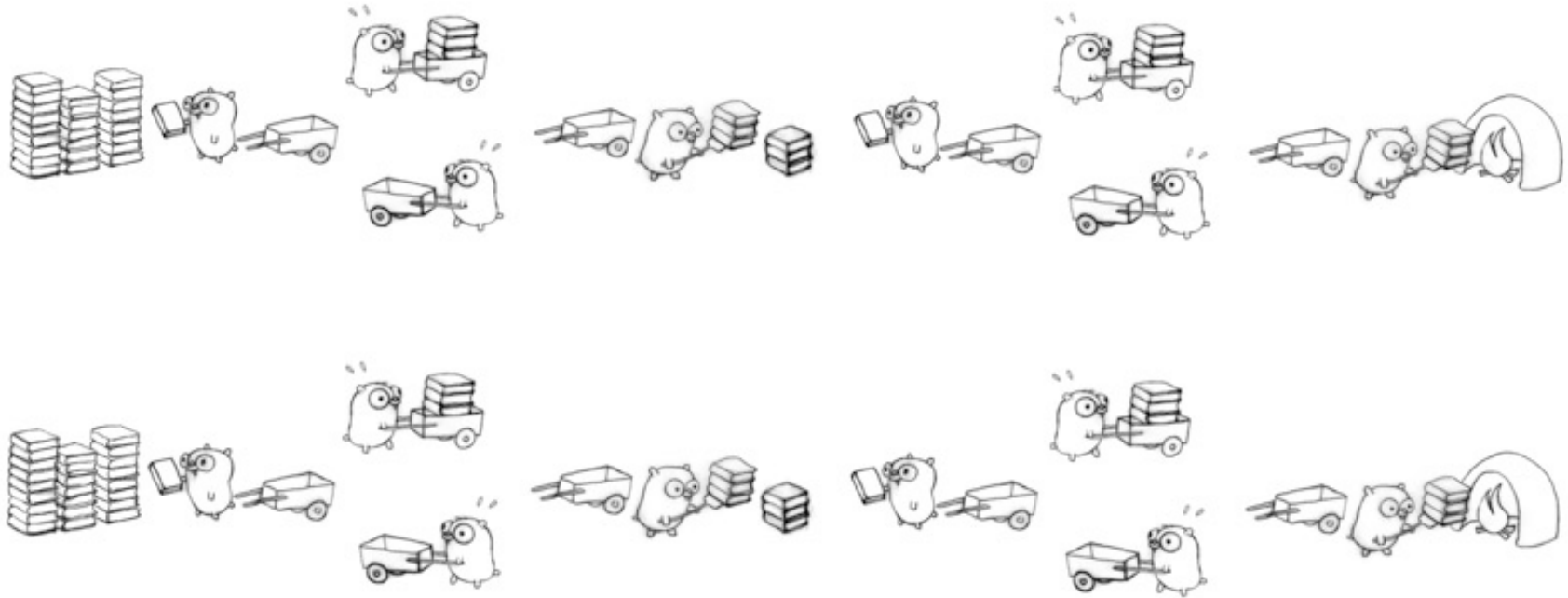
```
# highly contrived silly example
def junk():
    afdfaf
    asdfd
    fdafd / dsfsaf()
```

You will not catch this until **junk()** is called

```
Traceback (most recent call last):
  File "junk.py", line 7, in <module>
    junk()
junk()
NameError: global name 'afdfaf' is not defined
```



# Concurrency



# A non concurrent example

```
var langs = []Lang{
    Lang{name: "Go", url: "http://golang.org"},
    Lang{name: "Python", url: "http://python.org"},
    Lang{name: "Perl", url: "https://www.perl.org/"}}
```

```
func count(name, url string) {
    start := time.Now()
    r, err := http.Get(url)
    if err != nil {
        fmt.Printf("%s: %s", name, err)
        return
    }
    n, _ := io.Copy(ioutil.Discard, r.Body)
    r.Body.Close()
    fmt.Printf("%s\t%d bytes \t[%s]\n", name, n, time.Since(start))
}
```

Run

# A concurrent example

## Communicate with channels

```
func count(name string, url string, results chan string) {
    start := time.Now()
    r, err := http.Get(url)
    if err != nil {
        results <- fmt.Sprintf("%s: %s", name, err)
        return
    }
    n, _ := io.Copy(ioutil.Discard, r.Body)
    r.Body.Close()
    results <- fmt.Sprintf("%s\t%d bytes \t[%s]\n", name, n, time.Since(start))
}
```

Run

(Everything here is standard lib! [batteries included\*])

# What can you do with Go?

This presentation is done with Go [present]

GoBGP

- integrate a BGP daemon with another application

Custom packet analyzer (gopacket)

- capture packets, and do *something* with them

SNMP

- gosnmp

NETCONF

- go-netconf, go-junos

gRPC

- **streaming telemetry!!**

# SNMP example

```
// setup our session
gosnmp.Default.Target = 10.1.2.3
gosnmp.Default.Community = "shhhsecret"
// connect
err := gosnmp.Default.Connect()
// handle a failure
if err != nil {
    fmt.Printf("Connect err: %v\n", err)
    os.Exit(1)
}
defer gosnmp.Default.Conn.Close()

// perform a BulkWalk and send to PrintValue func
err = gosnmp.Default.BulkWalk(oid, poller.PrintValue)
if err != nil {
    fmt.Printf("Walk Error: %v\n", err)
    os.Exit(1)
}
```

## Who uses Go?

- Docker.io
- Uber
- Cloudflare
- Soundcloud
- Google (of course)

## Apps

- Prometheus
- Grafana
- InfluxDB
- **Docker**

and many more...

# Community

[Gophers Slack](https://invite.slack.golangbridge.org) (https://invite.slack.golangbridge.org)

[reddit.com/r/golang](https://reddit.com/r/golang) (https://reddit.com/r/golang)

[golang-nuts on Google Groups](https://groups.google.com/forum/#!forum/golang-nuts) (https://groups.google.com/forum/#!forum/golang-nuts)



# Thank you

James Boswell

Principal Engineer, Charter Communications

[jim@jimboswell.net](mailto:jim@jimboswell.net) (mailto:jim@jimboswell.net)

[@jimb on gophers slack](http://twitter.com/jimb%20on%20gophers%20slack) (http://twitter.com/jimb%20on%20gophers%20slack)

<https://invite.slack.golangbridge.org/> (https://invite.slack.golangbridge.org/)

<http://talks.godoc.org/github.com/jamesboswell/nanog69/whygo.slide>

(<http://talks.godoc.org/github.com/jamesboswell/nanog69/whygo.slide>)

# Thank you!

