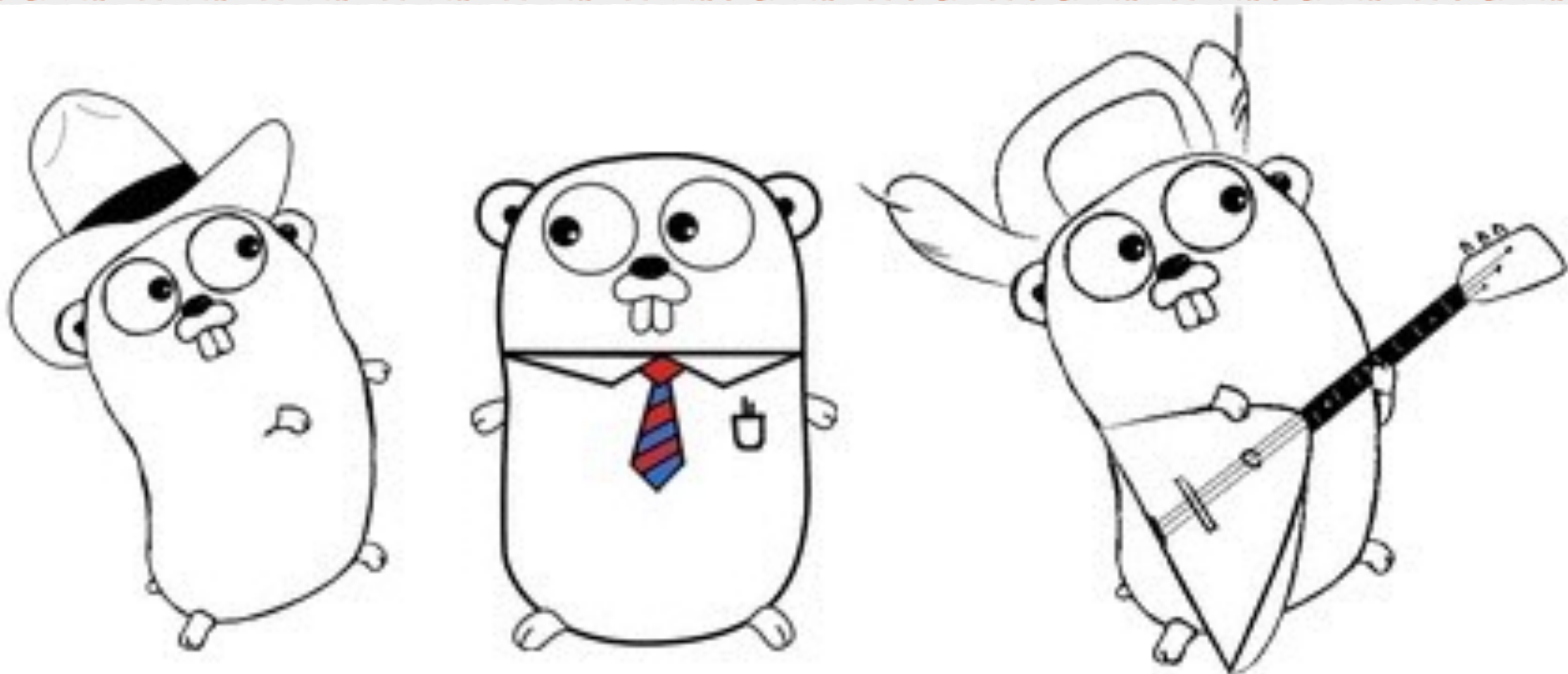


Go语言北京用户组简介 & Go语言速览

Go语言北京用户组 郝林

Gopher Meeting (Beijing, 2016-02-27)



欢迎各位**Gopher!**

——GopherBeijing

关于Go语言北京用户组

□ GopherBeijing的主旨

- 推广Go语言以及相关技术
- 帮助Go语言用户更好的使用相关技术和服务
- 促进Go语言用户及相关公司之间的交流和接洽
- 凝聚社区的力量，并服务于社区！

关于Go语言北京用户组

□ 2015.01.10

第一次聚会

代表了用户组的成立



关于Go语言北京用户组

□ 2015.06.06

国内第一场以Go/
~~~~~

Docker技术为主题的

Hackathon





# 关于Go语言北京用户组

□ 2015.09.12

科技寺之旅



GopherBeijing的技术聚会常态化



# 关于Go语言北京用户组

---

□ 2015.11.08

## GDG 北京 DevFest

1. Go语言简介&2015年发展趋势-郝林
2. Golang工具链选择-水羽哲

# 关于Go语言北京用户组

□ 2016.01



Gopher下午茶北京首秀



# 关于Go语言北京用户组

---

□ 2016.02.27

Go语言开发者聚会

(GopherBeijing & 北京GDG)



# 关于Go语言北京用户组

---

## □ GopherBeijing将要做的事

- Gopher 技术聚会
- Gopher 技术培训
- Gopher Code Lab (或线上Hackathon)
- Gopher 跨界茶话会
- Golang 播客 (golang.fm, 马全一发起)



# GopherCon China 2016

---

- 4月，北京
- Go语言大咖云集，包括 Go team 成员
- Topic囊括Go语言在各个领域的应用
- 收费的，不过物超所值



# 关于我

---

- 郝林，年龄奔过34，码龄10++，Go语言爱好者
- 8年Java程序设计经验（转角遇到Go）
- 《Go并发编程实战》《Go命令教程》《Go语言第一课》，？
- 除了上述，也捣鼓些技术培训和社区运营的事儿
- 目前在微赛时代（微影时代旗下）任平台架构经理



# 这不是最后一页

Go语言北京用户组的微信公众号  
(golang-beijing)





“Go是一种开源的编程语言，它旨在使人们能够方便地构建简单、可靠、高效的软件。”

—摘自Go语言官网



# Go语言第一眼

---

- Google出品，码界先驱开创
- 跨平台、开源、稳步更新、向后兼容
- 静态强类型、编译型、**拥有原生的并发支持**
- 语法简约，支持多种编程范式（FP+OOP）
- 提供众多强大的标准命令，覆盖软件全生命周期

# Go语言第二眼

---

- 25个关键字
- 19个二元操作符，2个一元操作符
- 18个基本数据类型，8个高级数据类型
- 406个标准库代码包 (Go 1.6)
- System + Application(Web,...) + Mobile + Embedded



# Go语言哲学

---

- 崇尚简约，单刀直入
- 极少的配置，更多的约定俗成
- 严谨的编码规范，但为程序设计留出大空间
- 通讯 -> 共享状态，共享状态 -> 通讯（线程间）
- 软件开发是一个工程级别的事情

# Go语言学习途径

---

- ❑ 官网一定要看和学：[golang.org](http://golang.org) + [godoc.org](http://godoc.org)
- ❑ 参与国际社区：[golang-nuts](http://golang-nuts) + [Go Forum](http://GoForum.com) + [reddit /r/golang](http://reddit.com/r/golang) + [Golang News](http://GolangNews.com)
- ❑ 网上的免费教程超多，比如：《Go语言第一课》
- ❑ 优秀Go项目索引：[awesome-go](http://awesome-go.com) + [awesome-go-China](http://awesome-go-china.com)
- ❑ 一定要至少加入一个国内的Go语言技术组织！



# 第一个Go程序[3]

---

```
func main() {  
    flag.Parse()  
    http.HandleFunc("/",  
        func(w http.ResponseWriter, r *http.Request) {  
            name := r.FormValue("name")  
            fmt.Fprintf(w, "Hello, %s!\n", name)  
        })  
    log.Fatal(http.ListenAndServe(serverAddr, nil))  
}
```

# 第一个Go程序[2]

---

// 代表服务程序监听地址的标记名称。

```
const FLAG_NAME_SERVER_ADDR = "addr"
```

// 服务程序监听的地址。

```
var serverAddr string
```

```
func init() {
```

```
    flag.StringVar(  
        &serverAddr, FLAG_NAME_SERVER_ADDR,
```

```
        ":8080", "The address of server.")
```

```
}
```



# 第一个Go程序[1]

---

```
package main
```

```
import (  
    "flag"  
    "fmt"  
    "log"  
    "net/http"  
)
```

# 第一个Go程序[4]

---

```
$ go run hello.go -addr=:9000
```



# 第一个Go程序[5]

---

```
$ curl http://localhost:9000/?name\=Robert  
Hello, Robert!
```

# 第一个Go程序[6]

---

## □ 重点复述:

- 命令源码文件: 属于main包并包含无参无果的main函数的文件。另外还有库源码文件和测试源码文件。
- 程序实体的原生访问权限有两级: 代码包私有和共有。名称首字母的大小写决定了其权限。另外还有模块私有。
- 源码文件可由代码包组织起来。import <代码包的导入路径> 即可将代码包及其中的程序实体导入到当前文件并使用。



# 第一个Go程序[7]

---

## □ 后续：

- 修改第一个Go程序，编译&运行/直接运行它，并观察不同
- 了解GOROOT、GOPATH、GOBIN等环境变量的含义
- 为第一个Go程序编写和运行测试，并学习使用各种Go命令
- 编写更多的代码，并用代码包组织起来
- 把代码包上传到Github，并让自己及他人使用你的代码包

《Go命令教程》

# Goroutine

---

- 用户级线程，有别于内核线程
- 轻量级，单机上百万个也是小case
- 快速，*go func { ... }()*
- 独特的协作机制——Channel



# Channel

---

- FIFO队列
- 类型化数据（元素）传递, *chan int64*
- 固定长度, *ch := make(chan int64, 1)*
- 并发安全, *ch <- 100* 和 *e := <-ch*
- 多Goroutine共用

# 重点推荐几个Go项目

---

- ☐ Web框架: Beego, Gin
- ☐ HTTP/2 web server: Caddy
- ☐ 分布式计算框架: Glow
- ☐ KV存储系统: etcd
- ☐ DB: TiDB
- ☐ MQ: NSQ



“这才是最后一页”

-出自台上的那个人