

一小时入门 Golang

如何学习一门语言

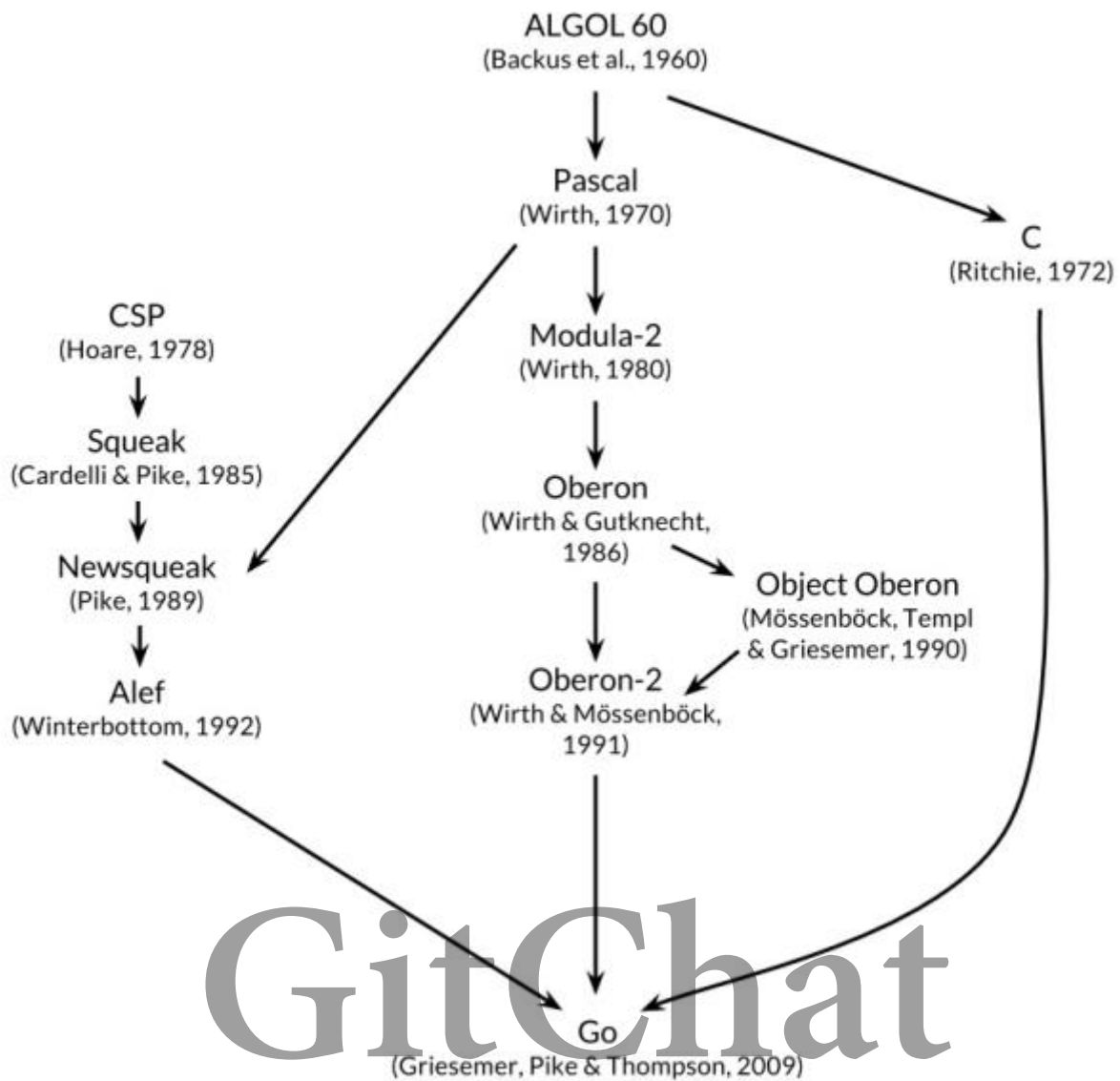
学习一门新的开发语言最重要的就是做到三点：

1. 基础知识
2. 学习抄代码
3. 学习写代码

Go 语言

Go 是 Google 开发的一种静态强类型、编译型、并发型，并具有垃圾回收功能的编程语言。为了方便搜索和识别，有时会将其称为Golang。摘自维基百科。

Go 语言的演进



服务开发语言有哪些？

- C/C++
- Java
- Python
- Golang
- php
- Rust
- NodeJS
- Erlang
- Ruby
- ...

为什么选择 Go ?

- 天生的服务器编程语言。
- 天生的并发模型。

除此之外，还有以下特点：

1. Go 具有 Python/Ruby 的开发效率，同时又有 C 语言的运行性能（不过还是有一定差距的）。
2. 静态语言。
3. 简单（只有 25 个关键词）。
4. 内存管理，简洁语法。
5. 标准类库，规范统一（gofmt, golint...）。
6. 文档全面，免费开源。
7. 快速编译，易于部署（二进制文件包可直接运行）。

Go 语言中会使用到的 25 个关键字或保留字：

1	2	3	4	5
break	default	func	interface	select
case	defer	go	map	struct
chan	else	goto	package	switch
const	fallthrough	if	range	type
continue	for	import	return	var

最简单的 Go 程序：

```
hello.go
package main

import "fmt"

func main() {
    fmt.Println("hello, world")
}
```

命令行运行 `go build` , 然后在同级目录下命令行 `./hello` 即可在控制台输出"hello,world".

Go 安装演示

Go 安装有多种办法：

我这里只给大家演示一种：解压缩包后配置环境变量。

```
export GOROOT=$HOME/go # go 语言的源路径
export GOPATH=$HOME/gopath # gopath 路径
export PATH=$PATH:$GOROOT/bin:$GOPATH/bin # 设置path
```

其他更多的 Go 安装可参考：

<https://github.com/astaxie/build-web-application-with-golang/blob/master/zh/01.1.md>

Go 命令：go build、go install、go test、go get、gofmt、godoc等等。

GitChat

Go is a tool for managing Go source code.

Usage:

```
go command [arguments]
```

The commands are:

build	compile packages and dependencies
clean	remove object files
doc	show documentation for package or symbol
env	print Go environment information
bug	start a bug report
fix	run go tool fix on packages
fmt	run gofmt on package sources
generate	generate Go files by processing source
get	download and install packages and dependencies
install	compile and install packages and dependencies
list	list packages
run	compile and run Go program
test	test packages
tool	run specified go tool
version	print Go version
vet	run go tool vet on packages

Use "go help [command]" for more information about a command.

Additional help topics:

c	calling between Go and C
buildmode	description of build modes
filetype	file types
gopath	GOPATH environment variable
environment	environment variables
importpath	import path syntax
packages	description of package lists
testflag	description of testing flags
testfunc	description of testing functions

Use "go help [topic]" for more information about that topic.

官方指南视频：[A tour of go](#)

Go 开发工具

- vim (经常使用)
- vscode (经常使用)
- IntelliJ idea
- sublime text 3
- Goland
- LiteIDE

- Atom

(大家可以打开他们的官网看看哪一个适合你自己)

个人推荐：VSCode，如果你是vim党的话，那应该不用我推荐了吧。

Go 基本结构

- 基本结构: function, if, switch, for, select
- 代码重用: package, 不允许循环引用
- 数据类型: struct+method, array, slice, map
- 抽象机制: interface
- 并发: goroutine, channel
- 内存管理: GC
- 错误处理: error, panic, recover
- 资源释放: defer
- reflection ...

基本的控制结构:

- if-else 结构
- switch 结构
- select 结构，用于 channel 的选择
- for (range) 结构

多返回值应用

Go 语言的函数经常使用两个返回值来表示执行是否成功：返回某个值以及 true 表示成功；返回零值（或 nil）和 false 表示失败。

当不使用 true 或 false 的时候，也可以使用一个 error 类型的变量来代替作为第二个返回值：成功执行的话，error 的值为 nil，否则就会包含相应的错误信息（Go 语言中的错误类型为 error: var err error）。

这样一来，就很明显需要用一個 if 语句来测试执行结果；由于其符号的原因，这样的形式又称之为 comma,ok 模式（pattern）。

Go 基本数据类型

基本类型

整数：

int8 (-128 -> 127)

int16 (-32768 -> 32767)

int32 (-2,147,483,648 -> 2,147,483,647)

int64 (-9,223,372,036,854,775,808 -> 9,223,372,036,854,775,807)

无符号整数：

uint8 (0 -> 255)

uint16 (0 -> 65,535)

uint32 (0 -> 4,294,967,295)

uint64 (0 -> 18,446,744,073,709,551,615)

浮点型 (IEEE-754 标准)：

float32 (+- 1e-45 -> +- 3.4 * 1e38)

float64 (+- 5 * 1e-324 -> 107 * 1e308)

int 型是计算最快的一种类型。

整型的零值为 0，浮点型的零值为 0.0

结构化的 (复合的)，如：struct、array、slice、map、channel；

只描述类型的行为的，如：interface

Go 开发规范

这一块内容，我们不需要多说，直接上go官方标准。

- gofmt
- golint
- ...
- gometalinter
- goreportcard

Go 单元测试与测试覆盖率

- goconvey
- overall

- goveralls

举个例子

```
math.go
package goconvey

import (
    "errors"
)

// Add 加法
func Add(a, b int) int {
    return a + b
}

// Subtract 减法
func Subtract(a, b int) int {
    return a - b
}

// Multiply 乘法
func Multiply(a, b int) int {
    return a * b
}

// Division 除法
func Division(a, b int) (int, error) {
    if b == 0 {
        return 0, errors.New("被除数不能为 0")
    }
    return a / b, nil
}
```

单元测试的例子：

```
math_test.go
package goconvey

import (
    "testing"

    . "github.com/smartystreets/goconvey/convey"
)

func TestAdd(t *testing.T) {
    Convey("两数相加", t, func() {
        So(Add(0, 1), ShouldEqual, 1)
    })
}
```



```

}

func TestSubtract(t *testing.T) {
    Convey("两数相减", t, func() {
        So(Subtract(0, 1), ShouldEqual, -1)
    })
}

func TestMultiply(t *testing.T) {
    Convey("两数相乘", t, func() {
        So(Multiply(3, 7), ShouldEqual, 21)
    })
}

func TestDivision(t *testing.T) {
    Convey("两数相除", t, func() {

        Convey("除以非 0 数", func() {
            num, err := Division(1024, 8)
            So(err, ShouldBeNil)
            So(num, ShouldEqual, 128)
        })

        Convey("除以 0", func() {
            _, err := Division(10, 0)
            So(err, ShouldNotBeNil)
        })
    })
}

```

Go 持续集成

在GitHub上做持续集成非常简单，只需要添加 `.travis-ci.yml` ；

如果是Gitlab 8.0以上也非常简单，只需要添加 `.gitlab-ci.yml` 。

可以参考我的文章：

<https://github.com/yangwenmai/how-to-add-badge-in-github-readme>

Go 标准库/标准包

比较常用的有：

```

fmt
time
strings
strconv
io/ioutil

```

encoding/json

net/http

database/sql

Go 标准库中文文档：<http://cngolib.com>

Go 哪些开源项目值得学习？

- golang/go
- pingcap/tidb
- flike/kingshard
- astaxie/beego
- 其他 (influxdb,etcd,grafana,promethues)...
- 更多可以去 avelino/awesome-go 这里找自己感兴趣的。

Go 可能的坑

1. gc 问题(其实一般场景大家遇不到)
2. defer 问题 (FILO)
3. nil 与 interface
4. 没有泛型 (reflect , 代码生成)
5. := 导致的变量覆盖
6. 没有自动类型转换 (var i int; var j int32 = 1; i = j; (语法错误))
7. 其他...

1. [50 Shades of Go: Traps, Gotchas, and Common Mistakes for New Golang Devs](#)

2. [Golang开发新手常犯的50个错](#)

推荐资料

1. Go 示例 (<https://gobyexample.com>)
2. Go 向导 (<https://tour.golang.org/welcome/1>)

3. Go 向导 (<https://go-tour-zh.appspot.com/welcome/1>)
4. Go 入门指南 (<http://wiki.jikexueyuan.com/project/the-way-to-go/>)
5. Go 构建Web应用 (<https://github.com/astaxie/build-web-application-with-golang>)
6. Go 学习笔记：无痕 (<https://github.com/qyuhen/book>)

这里的Go向导很不错，大家可以一步一步的跟着练习。

最后希望这篇文章能帮助到初学者。

GitChat