



GO 语言学习笔记

理论 + 实践

摘要

Go 语言爱好者的理论和实践

Gainovel

Go 语言学习笔记本

Go 语言学习笔记本，初步分为三个版块：初级、中级和高级，理论和实践相结合。

一、初级

1. Go 基础语法
2. Go 的日常使用

1.1 Go 语言简介

Go 编程语言是一个开源项目，旨在提高程序员的工作效率。

Go 语言表现力强、简洁、干净、高效。它的并发机制使得编写充分利用多核和网络机器的程序变得容易，而其新颖的类型系统使程序构造变得灵活和模块化。Go 可以快速编译为机器码，并且具有垃圾收集的便利性和运行时反射的强大功能。它是一种快速的、静态类型的编译语言，感觉像是一种动态类型的解释语言。

1.1.1 为什么使用 Go 语言？

1. Go 语言很简洁，只有 25 个关键字，使用上可以媲美 python
2. Go 程序可以直接编译成可执行文件，这一点和 C 语言很像
3. Go 比较好追源码，因为 Go 程序的写法相对比较固定，相比于 Java 和 Python 源码的难以看懂，看 Go 的源码会容易很多
4. Go 在语言层面支持协程，新手也可以简单的使用 go 关键字去开启协程并发处理任务

以上我最喜欢的点就是 Go 写代码很快而且可以直接编译成二进制，这让我可以快速实现一个想法并投入使用，因为二进制文件添加到环境变量中就可以直接使用。比如你可以开发一个类似于 Linux 上的 tree 命令，并把它用于 Windows 系统上。

1.1.2 安装 Go

1.1.3 Go 语言的 Hello World

在 main 包下定义一个 main 函数，输出 Hello World!

```
package main

import "fmt"

// go run github.com/gainovel/testcase/quickstart
func main() {
    fmt.Println("Hello World!")
}
```

1. 运行方式一：包的形式运行 `go run github.com/gainovel/testcase/quickstart`
2. 运行方式二：文件的方式运行 `go run main.go`

二、中级

1. Go 的进阶使用

三、高级

1. Go 实现原理

如果不考虑数据校验，那么整个数据迁移过程是这样的。



