

# golang教学课程表

## 课程标准

1. 每节课不超过30分钟，尽量控制在15分钟以内
2. 每节课节奏为
  - a. 理论/现象，是什么，解决什么（3分钟）
  - b. 编码演示怎么用（十分钟）
  - c. 其他用法（1分钟）
  - d. 总结（1分钟）
3. 每节课必须按上面节奏产出pdf课件

## 一、入门基础

聚焦Go语言核心语法与工具链，搭建基础认知框架

### 1. Go语言基础语法

- 课时1: Go环境搭建（Windows/macOS/Linux）与IDE配置（VS Code/GoLand）
- 课时2: 变量、数据类型与常量（基本类型、复合类型、类型转换）
- 课时3: 运算符（算术/逻辑/位运算/比较运算符）与表达式优先级
- 课时4: 流程控制（if-else条件判断、for循环、switch-case分支）
- 课时5: 函数（定义/调用/参数传递/返回值、匿名函数与闭包）
- 课时6: 指针（内存地址、指针操作、指针作为函数参数）
- 课时7: 错误处理（error类型、defer语句、panic/recover机制）

### 2. Go语言数据结构与算法包

- 课时1: 数组（固定长度、遍历、多维数组）与切片（动态数组、扩容机制、切片操作）
- 课时2: 映射（Map、键值对存储、遍历、并发安全问题）
- 课时3: 排序算法（sort包的使用）

### 3. Go语言面向对象编程

- 课时1: 结构体（定义/初始化/字段访问、嵌套结构体）
- 课时2: 方法（绑定结构体的函数、值接收者vs指针接收者）

- 课时3: 接口（定义/实现、多态性、空接口interface{}的应用）
- 课时4: Go的“面向对象”特性（无继承、通过组合实现代码复用）
- 课时5: 包（package）与导入（import机制、包作用域）

## 4. Go语言标准库与工具链

- 课时1: 常用标准库（fmt输入输出、os文件操作、io流处理）
- 课时2: 字符串处理（strings/split/join/trim等函数、正则表达式regexp）
- 课时3: 时间与日期（time包、时间格式化、定时任务）
- 课时4: Go工具链（go mod依赖管理、go build编译、go run运行、go test测试）
- 课时5: 调试，分析工具（pprof，日志库）

## 二、进阶应用

深化Go语言核心能力，覆盖并发、网络、Web开发等高频场景

### 1. Go语言并发编程

- 课时1: 并发模型（CSP理论、Goroutine轻量级线程）
- 课时2: Channel（通信机制、无缓冲/有缓冲Channel、关闭与遍历）
- 课时3: 同步原语（sync.Mutex互斥锁、sync.WaitGroup等待组、sync.RWMutex读写锁）
- 课时4: 并发模式（生产者-消费者、扇入/扇出、Pipeline）
- 课时5: 并发安全（避免竞态条件、atomic原子操作、并发测试）

### 2. Go语言网络编程

- 课时1: 网络基础（TCP/IP协议、三次握手、端口概念）
- 课时2: HTTP编程（net/http包、创建HTTP服务器、处理请求与响应）
- 课时3: WebSocket（gorilla/websocket库、实时通信场景）

### 3. Go语言Web开发基础

- 课时1: Web开发流程（需求分析、原型设计、前后端分离）
- 课时2: 路由系统（原生http.ServeMux、第三方库gin/iris基础）

- 课时3: 中间件（身份认证、日志记录、CORS跨域、请求限流）
- 课时4: RESTful API设计（资源定位、HTTP方法语义、状态码规范）

### 三、企业级项目实战

从0到1落地真实项目，掌握工程化开发全流程

#### 1. Go项目实战基础

- 课时1: 项目启动准备（需求文档）
- 课时2: 项目技术架构（技术文档）
- 课时3: 数据库选型（Mysql）
- 课时4: Git协作流程（分支策略、Pull Request、代码评审）
- 课时5: 服务配置管理（config.yaml文件、环境变量注入）

#### 2. 项目实战分析设计

- 课时1: 模块设计（功能边界定义）
- 课时2: 技术方案选型（比如选择gin还是其他、是否用ORM）
- 课时3: 数据库设计（表结构、索引优化、外键约束）
- 课时4: 接口设计（API文档编写、Swagger/OpenAPI生成）
- 课时5: 技术评审（可扩展性、性能瓶颈预判、容错设计）
- 课时6: 项目管理与分工排期（各角色分工和进入开发）

#### 3. 项目实战开发

- 课时1: 核心功能实现（比如用户登录、订单创建、数据查询）
- 课时2: 单元测试（go test编写测试用例、覆盖率统计）
- 课时3: 集成测试（模块间接口联调、Mock服务使用）
- 课时4: 调试与优化（pprof性能分析、慢SQL排查、内存泄漏修复）
- 课时5: 初次部署（Docker镜像打包、云服务器上线）

### 四、企业级项目深入

#### 1. 数据库组件

- 课时1: ORM框架 (GORM基础CRUD、关联关系映射、事务处理, Gorm)
- 课时2: 数据库优化 (索引策略、分库分表、读写分离)
- 课时3: NoSQL数据库 (MongoDB基础、文档存储场景)
- 课时4: 数据库连接池 (配置调优、连接泄漏预防)

## 2. 缓存组件

- 课时1: Redis基础 (数据类型、常用命令、持久化机制)
- 课时2: 缓存集成 (go-redis库、缓存读取/写入策略)
- 课时3: 缓存问题解决 (缓存穿透/击穿/雪崩、布隆过滤器)
- 课时4: 分布式锁 (Redis Redlock算法、解决并发冲突)

## 3. 消息队列

- 课时1: 消息队列原理 (异步解耦、流量削峰、最终一致性)
- 课时2: Kafka基础 (Topic/Partition/Consumer Group、生产者/消费者)
- 课时3: RabbitMQ基础 (Exchange/Queue/Binding、路由模式)
- 课时4: 消息可靠性 (ACK机制、消息重试、死信队列)
- 课时5: 消息积压处理 (扩容消费者、优化消费逻辑、监控告警)

# 五、微服务与持续集成

## 1. 微服务

- 课时1: 微服务架构概念 (服务拆分原则、单一职责)
- 课时2: 服务发现与配置中心 (Consul/Etcd使用、服务注册与发现)
- 课时3: 链路追踪 (Jaeger/Zipkin、分布式请求链路分析)

## 2. CI/CD

- 课时1: CI/CD流程设计 (持续集成→持续交付→持续部署)
- 课时2: CI工具 (Jenkins/GitLab CI/Drone基础配置)
- 课时3: 自动化构建 (Docker镜像打包、多阶段构建优化)
- 课时4: 自动化部署 (Kubernetes部署、滚动更新与回滚)

## 答疑服务

### 1. 日常学习技术咨询

- 课时1：语法/环境/工具问题（比如“指针怎么用？”“go mod怎么解决依赖冲突？”）
- 课时2：项目开发问题（比如“并发场景下数据不一致怎么办？”“缓存穿透怎么解决？”）
- 课时3：错误排查技巧（比如通过pprof找性能瓶颈、通过日志定位接口报错）

### 2. 面试与就业指导

- 课时1：常见面试题解析（基础语法/并发/网络/项目经验）
- 课时2：简历优化（突出Go项目经验、量化成果、技术栈匹配）
- 课时3：面试技巧（技术面应对、HR面沟通、薪资谈判）
- 课时4：职业规划（Go开发岗位方向、进阶路径、行业趋势）