

## 23 结束语 你的 Go 语言成长之路

---

我们从 Go 语言的基础知识，到底层原理，再到实战，相信你已经学会了如何使用 Go 语言，并可以上手做项目了。这一路走来，非常感谢你对学习的坚持，以及对我的支持。

在本专栏的最后，我会和你聊下 Go 语言的前景，以及对于你学习 Go 语言编程和在今后职业发展方面，我的一些建议。

### Go 语言的发展前景

随着这几年 Docker、K8s 的普及，云原生的概念也越来越火，而 Go 语言恰恰就是为云而生的编程语言，所以在云原生的时代，它就具备了天生的优势：易于学习、天然的并发、高效的网络支持、跨平台的二进制文件编译等。

CNCF（云原生计算基金会）对云原生的定义是：

- 应用容器化；
- 面向微服务架构；
- 应用支持容器的编排调度。

我们可以看到，对于这三点有代表性的 Docker、K8s 以及 istio 都是采用 Go 语言编写的，所以 Go 语言在云原生中发挥了极大的优势。

在涉及网络通信、对象存储、协议等领域的工作中，Go 语言所展现出的优势要比 Python、C/C++ 更大，所以诸如字节跳动、腾讯等很多大厂都在拥抱 Go 语言的开发，甚至很多公司在业务这一层也采用 Go 语言来开发微服务，从而提高开发和运行效率。

总体来说，对 Go 语言的前景我还是比较看好的，所以本专栏是你 Go 语言学习的敲门砖，接下来我建议你可以对这一语言进行更加系统和全面的学习。

### Go 语言学习建议

关于 Go 语言的学习，我建议从**官方文档**和**官方作者著作的书**开始，这样你可以看到“原汁原味”的讲解。其实不只 Go 语言，任何一门语言都应该是这样，官方的内容是比较权威的。

基于官方文档入门后，你就可以参考一些第三方大牛写的相关书籍了。阅读不同人写的 Go 语言书籍，你可以融会贯通，更好地理解 Go 语言的知识点。比如在其他书上看不懂的内容，换一本你可能就看懂了。

阅读书籍还有一个好处是让你的学习具备**系统性**，而非零散的。现在大部分的我们都选择碎片化学习，其实通过碎片化的时间，系统地学习才是正确的方式。

不管是通过书籍、官网文档，还是视频、专栏的学习，我们都要结合示例进行练习，不能只用眼睛看，这样的学习效率很低，一定要将代码动手写出来，这样你对知识的理解程度和只看是完全不一样的，在这个过程中你可以**通过写加深记忆、通过调试加深理解、通过结果验证你的知识**。

有了这些基础后，就可以看一些实战类的书籍、文章和视频了，这样你不只是学会了 Go 语言，还能用 Go 语言做项目，了解如何编码、分库、微服务、自动化部署等。

不管是学习 Go 语言还是其他编程语言，都要阅读源代码，通过阅读源代码了解底层的实现原理，以及学习他人优秀的代码设计，进而提升自己在 Go 语言上的技术能力。

当然一个工程师“源于代码”，但不能“止于代码”。

## 不止于编程语言

无论你是想走技术专家路线，还是技术管理路线，要想更多地发挥自己的价值，必然是要带人的，因为一个人再怎么努力、技术如何厉害，也比不上**多人团队的协作**。

所以，当你工作 3 年具备骨干的能力后，就要开始尝试带人、做导师了，把自己学习编程的经验教给新人，让他们少走弯路，同时也能锻炼自己带人的能力，协调更多的人一起做事情。

这样当你有 5 年、7 年，甚至以上工作经验的时候，你的团队会越来越壮大，在团队中你所发挥的价值也越来越大；而在个人方面，你也可以做架构设计、技术难点攻关等更有价值的事情。

关于技术编程人员的成长，我有过一次分享。我把成长经历分为 9 个阶段，每一个阶段需要哪些技术，如何提升自己的段位，都有详细的介绍，你可以在[《技术编程人员成长的 9 个段位》](#)中查看。

## 总结

具备自我驱动力，以及学习能力的人，在职场中的竞争力都不会太差。

希望这个专栏可以很好地帮到你，让你学到 Go 语言的知识，让你在职场中更具备竞争力。

