编程

- 只要它激励你，竞争性编程是可以的。
- 编程就像写作。编程是一种创造性活动。
- 做自己的项目。写技术博客。为一个你将维护多年的项目编程，就像维护一个长达 10 年的博客一样。
- 通常，你不需要追求当前热门的技术，因为许多趋势几年后就会消退。
- 追求你的好奇心，为自己编程。
- 尝试为自己创建程序。它们不是工作任务。
- 如果你经常在编程时感到不开心，那么你做得方式不对。
- iOS、Android、后端、前端、AI 都很好。你至少可以尝试用它们做一个小项目，或者学习几个月。
- 调试是关于怀疑。不要相信你代码的每一行；你可以想出更好的方法来做到这一点。
- 在编程中，即使是一个字符或一行日志也很重要。它们告诉你一些信息。
- 使用编程制作产品供他人使用。拥有用户是有趣的。
- 你不需要太苛刻。几百个真正喜欢你产品的用户比几万个只是稍微喜欢你产品的用户更好。
- 记住你为什么进入编程领域，永远不要忘记它。
- 将编程中的知识应用到生活的每个方面。它们是一样的。批量处理或逐个处理。如何将工作分解成单元。每个应用背后的底层技术。网络请求背后的细微细节。
- 抽象和逻辑思维。细节导向的思维。思考每个解决方案。
- 真理就是真理。通常，计算机不会错。电路不会错。编译器不会错。当有 bug 时不要感到难过。
- 追求简洁和简单的解决方案。简单是最终的复杂。你需要努力思考，留下本质，去除多余的部分。
- 对于编程语言，能完成工作的语言就可以了。我个人推荐 Java 和 Python。
- 关注阴王，https://www.yinwang.org。他是编程中少有的天才之一，尽管他说天才不存在。
- 编程的知识和原则可以很容易地应用于语言学习、硬件修理、生活技巧和科学研究。
- 对于大多数编程任务，你不需要复杂的数学知识，高中数学就够了。
- 反思你几年前的旧代码，或者维护一个长期的代码项目。它会教会你很多东西。
- 如果你失去了对编程的热情，就做其他事情一段时间。
- 测试的时机很重要。自然地去做。你经常不需要为你的项目编写测试。尝试不写测试，写单元测试，写集成测试，写 API 测试。明智地比较它们。
- 尝试 AI 代码编辑器。经常使用 ChatGPT 或其他聊天机器人。现在 AI 工具易于使用，你可以专注于更有创意或更重要的部分。