

指 的增强型提案，包含极为丰富的知识，是程序员必须要理解和学习的。本项目旨在从 中汲取素材，进行归类、对比、拓展，以期帮助开发人员从底层构建起含技巧、领域等维度的知识体系。

## （ 代码风格指南 ）简介

是 官方推荐的代码风格规范，目标是让代码在团队协作中更 一致、更 可读、更 易维护。它不改变语义，但能显著降低 看懂代码 的成本。

### 核心原则

可读性优先：优先写 容易被人读懂 的代码，而不是 看起来很聪明 的代码。

一致性优先：同一项目内保持一致（哪怕与某条规则略有偏离），通常比 严格但混乱 更重要。

### 常见规则速览

缩进：使用 个空格；不要使用 （或确保编辑器将 转为空格）。

行长度：传统建议每行不超过 字符（注释 文档字符串常见上限）；现代工程中也常见 等团队约定。

空行：顶层函数 类之间通常空两行；类内方法之间空一行。

导入：

标准库、第三方库、本地应用导入分组并用空行分隔；

尽量在文件顶部导入；

避免 。

命名（常见约定）：

：函数、变量、模块；

（ ）：类；

：常量；

受保护成员以 标识。

空白符：避免多余空格；运算符两侧通常留空格；括号内侧不留空格。

注释与文档：注释说明 为什么这么做；公共 建议写 （可参考 的 约定）。

### 落地建议（自动化优先）

风格规范最容易在 靠自觉 时失效，建议在项目里用工具自动化约束：

格式化：

静态检查 风格检查： （可替代 覆盖大量传统工具）

导入排序： （或使用 的 规则）

若本仓库后续加入可执行代码与 ，可在此基础上补充具体的配置文件与执行命令。