|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **级别** | **类别** | **规范项** | **示例 / 说明** |
| 1 | 强制 | 命名规范 | 变量、函数使用 snake\_case，类名使用 PascalCase | my\_var, class UserModel: |
| 2 | 强制 | 缩进 | 使用 4 个空格缩进，禁止使用 Tab | 遵循 PEP8 |
| 3 | 强制 | 注释 | 所有类、函数必须有 docstring 或说明注释 | """说明...""" |
| 4 | 强制 | 文件命名 | 文件使用全小写 + 下划线命名 | user\_model.py |
| 5 | 强制 | 每行长度 | 每行不得超过 100 字符 | 超过需换行 |
| 6 | 强制 | 异常处理 | 禁止使用裸 except:，必须指定具体异常类型 | except ValueError: |
| 7 | 强制 | 日志 | 禁止使用 print()，应使用 logging 模块 | logging.info() |
| 8 | 强制 | 安全 | 禁止使用 eval()、exec() 等高风险函数 | 会造成注入风险 |
| 9 | 强制 | 类型注解 | 公共函数接口必须添加类型注解 | def add(x: int) -> int: |
| 10 | 强制 | 引入顺序 | import 顺序：标准库 > 第三方库 > 本地模块，每组之间空一行 | 遵循 PEP8 |
| 11 | 强制 | 空值判断 | 判断是否为空使用 if not var 或 is None，禁止 == None | if x is None: ✅ |
| 12 | 强制 | 单元测试 | 每个模块必须编写测试文件 | test\_xxx.py |
| 13 | 强制 | 代码格式化 | 提交前必须运行 black 或 autopep8 | 统一风格 |
| 14 | 强制 | 依赖管理 | 所有依赖必须写入 requirements.txt，禁止直接 pip install | 支持 CI/CD |
| 15 | 强制 | 主函数保护 | 脚本执行入口必须使用 if \_\_name\_\_ == "\_\_main\_\_" | 防止 import 执行 |
| 16 | 强制 | logging 栈信息 | 捕获异常必须带 traceback，使用 logging.exception() | 可快速定位问题 |
| 17 | 推荐 | 命名规范 | 常量应使用全大写 + 下划线命名 | MAX\_LIMIT = 100 |
| 18 | 推荐 | 函数体长度 | 函数不超过 50 行，尽量保持单一职责 | 保持简洁 |
| 19 | 推荐 | 类体长度 | 类不超过 300 行，超过应考虑分拆 | 避免过度耦合 |
| 20 | 推荐 | 文件分层 | 项目结构分为 models、services、api 等逻辑层 | 便于协作 |
| 21 | 推荐 | 空行风格 | 函数之间空 1 行，类之间空 2 行 | 提高可读性 |
| 22 | 推荐 | 返回值统一 | 函数返回值类型统一，例如永远返回 list，而非 list 或 None | 减少判断分支 |
| 23 | 推荐 | 模块导入 | 尽量使用绝对导入，避免相对导入 | from project.module import X |
| 24 | 推荐 | 数据拷贝 | 传递 list/dict 时视情况使用 copy() 或 deepcopy() | 防止副作用 |
| 25 | 推荐 | 并发控制 | 多线程共享数据使用 threading.Lock() 或 RLock() 控制并发 | 避免数据竞争 |
| 26 | 推荐 | 异步编程 | IO 密集型代码优先使用 asyncio，推荐使用 async/await | 高并发性能更好 |
| 27 | 推荐 | CLI 工具 | 构建命令行工具建议使用 argparse 或 click | 结构清晰 |
| 28 | 推荐 | 配置分离 | 配置写入 config.yaml 或 .env，不写死在代码中 | 便于部署和管理 |
| 29 | 推荐 | 数据结构 | 使用 dataclass 或 pydantic 定义结构体 | 更安全 |
| 30 | 推荐 | 类型注解 | 项目中推荐全面启用 mypy 或 Pyright 静态检查 | 保证类型正确 |
| 31 | 推荐 | code review | 所有代码必须至少由 1 人 review 后合并 | 提高质量 |
| 32 | 推荐 | 注释语言 | 中文注释允许但建议风格统一，例如统一格式为 # 说明： | 避免混乱风格 |
| 33 | 允许 | 工具脚本 | 非主流程可临时写单个脚本，但应移入 tools/ 目录管理 | 可维护性 |
| 34 | 允许 | 交互开发工具 | 学习/调试可使用 Jupyter Notebook、IPython | 不用于生产 |
| 35 | 允许 | TODO / FIXME | 开发中可使用 # TODO: / # FIXME:，上线前应清理 | 追踪遗留问题 |
| 36 | 允许 | 类型忽略 | 少数场景可使用 # type: ignore 跳过类型检查 | 如 monkey patch 场景 |
| 37 | 允许 | assert 断言 | 只在测试中使用 assert，禁止用于业务逻辑 | 不可替代异常 |
| 38 | 允许 | lambda 表达式 | 可用于简单函数定义，避免多层嵌套 | key=lambda x: x.name |
| 39 | 允许 | 私有成员命名 | 私有方法或属性使用 \_ 前缀约定，不作强制封装 | \_internal\_method() |
| 40 | 允许 | logging 灵活级别 | 可使用 DEBUG/INFO/WARNING/ERROR 灵活控制日志级别 | 适配开发与生产 |