**1. 代码风格规范​​**

**​​a. 强制要求​​**

1. ​**​缩进​**​：必须使用4个空格，禁止使用制表符（Tab）。
2. ​**​行长度​**​：代码行不超过79字符，文档字符串/注释不超过72字符。
3. ​**​命名​**​：
   * 类名用驼峰命名法（如MyClass），函数/变量用下划线（如my\_function）。
   * 常量全大写（如MAX\_LENGTH）。
4. ​**​导入顺序​**​：标准库→第三方库→本地模块，每组用空行分隔。
5. ​**​异常处理​**​：禁止裸露except:，必须指定异常类型（如except ValueError）。

**​​b. 推荐做法​​**

1. ​**​操作符空格​**​：二元操作符两侧加空格（如a = b + 1）。
2. ​**​空行​**​：函数/类间空两行，方法间空一行。
3. ​**​None比较​**​：用is或is not而非==。
4. ​**​字符串连接​**​：优先用''.join()而非+=。
5. ​**​类型检查​**​：用isinstance()而非type()。

**​​c. 允许做法​​**

1. ​**​引号选择​**​：单引号或双引号均可，但项目内需统一。
2. ​**​多行表达式​**​：可用括号或反斜杠换行。
3. ​**​行尾分号​**​：允许但不鼓励（如x = 1; y = 2）。
4. ​**​相对导入​**​：复杂包结构中允许显式相对导入（如from . import module）。
5. ​**​行内注释​**​：允许简短注释，但避免冗余。

**​​2. 文档与注释规范​​**

**​​a. 强制要求​​**

1. ​**​模块文档​**​：公共模块必须有文档字符串（"""Module description..."""）。
2. ​**​函数文档​**​：公共函数需用Google风格文档字符串，包含参数、返回值、异常说明。
3. ​**​类文档​**​：公共类需描述功能及主要方法。
4. ​**​版本标注​**​：重大API变更需注明版本号（如.. versionadded:: 1.2）。
5. ​**​类型注解​**​：公共API必须标注参数和返回类型（如def func(a: int) -> str）。

**​​b. 推荐做法​​**

1. ​**​示例代码​**​：文档字符串中提供使用示例。
2. ​**​术语表​**​：为技术术语添加对照表。
3. ​**​流程图解​**​：复杂逻辑用Mermaid等工具绘制流程图。
4. ​**​错误提示​**​：为可能失败的操作提供详细错误信息。
5. ​**​文档测试​**​：用doctest验证示例代码。

**​​c. 允许做法​​**

1. ​**​私有方法文档​**​：不强制但鼓励添加。
2. ​**​多语言注释​**​：允许非英语注释，但项目内需统一。
3. ​**​可视化结果​**​：文档中可包含图表输出示例。

**​​3. 开发与测试规范​​**

**​​a. 强制要求​​**

1. ​**​单元测试​**​：公共API必须覆盖单元测试。
2. ​**​测试覆盖率​**​：核心模块覆盖率≥80%。
3. ​**​代码审查​**​：所有代码需至少一名其他开发者审查。
4. ​**​CI/CD集成​**​：必须配置自动化测试和代码检查流程。
5. ​**​静态分析​**​：必须通过pylint/mypy检查。

**​​b. 推荐做法​​**

1. ​**​TDD​**​：推荐测试驱动开发模式。
2. ​**​性能测试​**​：对性能敏感模块进行基准测试。
3. ​**​依赖管理​**​：用requirements.txt或poetry明确依赖版本。
4. ​**​代码复用​**​：鼓励提取公共工具函数。

**​​c. 允许做法​​**

1. ​**​原型代码​**​：允许临时忽略部分规范，但需标记为# TODO: Refactor。
2. ​**​调试代码​**​：允许临时print调试，但提交前需移除。
3. ​**​实验性功能​**​：允许用@experimental装饰器标注未稳定功能。