

Python 技术管理规范文档

1. 代码风格

a. 强制

1. 必须遵循 PEP 8 风格指南，包括缩进(4 个空格)、行长度(不超过 79 字符)等
2. 必须使用 UTF-8 编码
3. 导入模块必须分组并按以下顺序：标准库、第三方库、本地应用/库，每组间空一行
4. 类名必须使用驼峰命名法(CamelCase)，函数和变量名必须使用小写加下划线

(snake_case)

5. 必须为所有公共模块、函数、类和方法编写文档字符串(docstring)

b. 推荐

6. 推荐使用类型注解(Type Hints)，特别是对于公共 API
 7. 推荐使用 f-string 进行字符串格式化(Python 3.6+)
 8. 推荐在运算符前后和逗号后添加一个空格
 9. 推荐使用_作为无意义变量名(如 for _ in range(10))
- #### c. 允许
10. 允许在行尾使用\来换行，但推荐使用括号包裹的隐式续行
 11. 允许使用 from module import *，但仅限于在__init__.py 中或测试代码中

2. 代码组织

a. 强制

12. 每个 Python 文件必须包含一个 if __name__ == '__main__': 块用于测试
13. 必须避免循环导入
14. 必须将测试代码放在单独的 tests 目录中

b. 推荐

15. 推荐使用绝对导入而非相对导入
16. 推荐将大型项目组织为包(package)而非单个模块
17. 推荐为每个主要功能创建单独的模块

c. 允许

18. 允许在小型项目中使用脚本式组织方式
19. 允许在合理情况下使用相对导入(如在包内部)

3. 异常处理

a. 强制

20. 必须捕获特定的异常，而不是使用裸露的 except:
21. 必须清理资源(如文件、网络连接)使用 with 语句或 try/finally
22. 必须避免使用异常进行流程控制

b. 推荐

23. 推荐自定义异常时继承自 Exception 而非 BaseException
24. 推荐在捕获异常后记录日志或进行适当处理
25. 推荐在适当情况下使用上下文管理器(context manager)

c. 允许

26. 允许在顶层捕获所有异常并记录日志(如长时间运行的服务)
27. 允许在测试代码中使用裸露的 except 语句

4. 性能与安全

a. 强制

28. 必须避免 SQL 注入，使用参数化查询而非字符串拼接

- 29. 必须避免使用 `pickle` 处理不受信任的数据
- 30. 必须对用户输入进行验证和清理
- b. 推荐
- 31. 推荐使用生成器(generator)处理大数据集
- 32. 推荐使用内置函数和库而非手动实现(如使用 `json.loads` 而非 `eval`)
- 33. 推荐使用 `lru_cache` 装饰器缓存昂贵函数的结果
- c. 允许
- 34. 允许在性能关键路径使用局部变量缓存属性访问
- 35. 允许在明确需要时使用全局变量
- 5. 测试与文档
- a. 强制
- 36. 必须为所有公共 API 编写单元测试
- 37. 必须保持测试代码与生产代码相同的质量标准
- 38. 必须更新文档字符串以反映代码变更
- b. 推荐
- 39. 推荐使用 `pytest` 而非 `unittest`
- 40. 推荐为复杂算法编写文档说明
- 41. 推荐使用 `doctest` 为示例代码提供测试
- c. 允许
- 42. 允许在原型阶段暂时跳过测试
- 43. 允许在内部工具中使用简化的文档
- 6. 工具与依赖
- a. 强制
- 44. 必须使用 `requirements.txt` 或 `Pipfile` 明确声明依赖
- 45. 必须固定生产环境依赖的版本
- 46. 必须使用虚拟环境隔离项目依赖
- b. 推荐
- 47. 推荐使用 `pre-commit hooks` 自动检查代码风格
- 48. 推荐使用 `black` 自动格式化代码
- 49. 推荐使用 `mypy` 进行静态类型检查
- c. 允许
- 50. 允许在小型项目中使用系统 Python 环境
- 51. 允许在开发环境中使用依赖的最新版本
- 7. 并发与异步
- a. 强制
- 52. 必须了解 GIL 限制, CPU 密集型任务考虑 `multiprocessing`
- 53. 必须避免在多线程中共享可变状态
- b. 推荐
- 54. 推荐使用 `asyncio` 而非多线程处理 I/O 密集型任务
- 55. 推荐使用线程池/进程池而非手动管理线程/进程
- c. 允许
- 56. 允许在简单脚本中使用基本的多线程
- 57. 允许在明确需要时使用低级线程 API
- 8. 面向对象

a. 强制

58. 必须使用@property 装饰器而非 getter/setter 方法

59. 必须优先使用组合而非继承

b. 推荐

60. 推荐使用数据类(dataclasses)简化类定义(Python 3.7+)

61. 推荐使用__slots__优化内存使用

c. 允许

62. 允许在适当情况下使用多重继承

63. 允许使用动态属性访问(__getattr__等)