

Python 技术管理规范文档

分类：a、强制；b、推荐；c、允许

一、代码风格

- 1. a. 缩进必须用 4 个空格，禁止 Tab 和混合缩进。
- 2. a. 单行长度不超过 120 字符，超长需换行并使用括号对齐。
- 3. a. 导入分组顺序：标准库→第三方库→本地模块，组间空一行。
- 4. a. 禁止使用通配符导入（``from module import``）。
- 5. b. 函数/类间用 2 空行分隔，类内方法用 1 空行分隔。
- 6. a. 二元运算符两侧加空格（``x = y + z``），但切片操作不加（``a[1:3]``）。

二、命名规范

- 7. a. 变量/函数名用蛇形命名（``snake_case``），类名用驼峰（``UpperCamelCase``）。
- 8. a. 常量名全大写（``MAX_LENGTH``），模块名全小写（``data_utils.py``）。
- 9. a. 避免单字符变量名（除循环计数 ``i`/`j``）。
- 10. b. 私有成员前缀下划线（``_internal_var``），双下划线仅用于避免冲突（``__mangled``）。
- 11. a. 布尔变量名以 ``is_`/`has_`` 开头（``is_valid``）。

三、函数与类

- 12. a. 函数参数最多 5 个，超限需用 ``kwargs`` 或重构。
- 13. a. 类方法首个参数必须为 ``self``，类方法为 ``cls``。
- 14. b. 函数长度不超过 50 行，类长度不超过 300 行。
- 15. a. 避免可变默认参数（用 ``None`` 替代 ``def func(a=[])``）。
- 16. b. 使用 ``@dataclass`` 简化数据容器类（Python 3.7+）。

四、异常处理

- 17. a. 禁止裸 ``except:``，必须指定异常类型（``except ValueError``）。
- 18. a. 资源操作必须用 ``with`` 语句（文件、锁等）。
- 19. b. 自定义异常继承 ``Exception`` 而非 ``BaseException``。
- 20. a. 异常消息需包含具体错误信息（``raise ValueError("Invalid ID: {id}")``）。

五、类型与注解

- 21. a. 公共接口必须添加类型注解。
- 22. b. 复杂类型用 ``typing`` 模块（``List[Dict[str, int]]``）。

- 23. a. 避免`Any`类型，除非必要场景（如动态装饰器）。
- 24. c. 允许在测试代码中省略类型注解。

六、并发与异步

- 25. a. 多线程共享变量必须用锁（`threading.Lock`）。
- 26. b. 优先使用`asyncio`替代`threading`处理 I/O 密集型任务。
- 27. a. 异步函数必须用`async`/`await`声明，禁止混用阻塞调用。
- 28. a. 线程池/进程池必须显式管理生命周期（`with ThreadPoolExecutor() as pool`）。

七、性能优化

- 29. a. 循环内避免重复计算（如`len(list)`提前缓存）。
- 30. b. 字符串拼接用`f-string`或`"".join()`，避免`+`循环。
- 31. a. 列表推导式优先于`map()`/`filter()`（除非涉及已有函数）。
- 32. b. 大数据处理用生成器（`yield`）替代返回列表。

八、测试规范

- 33. a. 核心逻辑必须用`pytest`覆盖单元测试。
- 34. a. 测试用例名需描述行为（`test_get_user_with_invalid_id_fails`）。
- 35. b. 使用`monkeypatch`管理环境变量，而非直接修改`os.environ`。
- 36. a. 异步测试必须用`pytest-asyncio`的`@pytest.mark.asyncio`。

九、文档与注释

- 37. a. 公共模块/类/函数必须写 docstring（Google 风格）。
- 38. b. 复杂算法添加行内注释，解释“为什么”而非“做什么”。
- 39. a. 废弃代码用`# DEPRECATED:`标注并说明替代方案。
- 40. c. 允许用`# type: ignore`跳过静态检查，但需附原因。

十、安全与防御

- 41. a. 用户输入必须过滤/转义（防 SQL 注入/XSS）。
- 42. a. 敏感数据禁止硬编码，必须从环境变量读取（`os.getenv("API_KEY")`）。
- 43. a. 反序列化数据必须验证结构（如 Pydantic 模型）。
- 44. b. HTTPS 请求需验证证书（`requests.get(..., verify=True)`）。

十一、工具与质量

- 45. a. 代码必须通过`flake8` + `mypy`静态检查（CI 集成）。
- 46. b. 使用`black`自动格式化代码，确保风格一致。
- 47. a. 依赖必须声明在`requirements.txt`或`pyproject.toml`中。
- 48. b. 优先用`pathlib`替代`os.path`处理路径。