

Python 规范 2023141461156 李宗泽

a. 强制

1. 缩进使用 4 个空格:严格禁止使用制表符 (Tab) 做缩进,每一级代码块必须使用 4 个空格。
2. 所有源文件必须使用 UTF-8 编码,且在文件顶部不需要添加 `# -*- coding: utf-8 -*-` 注释
3. 任何代码行最长不得超过 80 个字符 (不含换行符)。
4. 函数和类之间保留两个空行。
5. 类内部方法之间保留一个空行
6. 所有导入必须放在文件顶部,紧跟模块注释 (module docstring) 之后
7. 所有公共函数、类和模块都必须包含文档字符串 (docstring), 描述功能、参数和返回值。
8. 命名空间导入要用别名时严禁简写为单字符
9. 命名约定: 模块与包用小写字母,必要时可以使用下划线;类名使用 CapWords (驼峰式首字母大写);常量使用全大写 CAPS_WITH_UNDERSCORES
10. 禁止在代码中留下 TODO、FIXME 等未解决标记,提交前必须清理或明确添加对应 issue 链接
11. 函数参数列表和调用参数列表超过一行时,要换行并对齐
12. 模块顶层禁止带有可执行代码,测试或演示代码,应放在 `if __name__ == '__main__':` 块中
13. 强制在文件末尾保留一个空行: 每个 py 文件末尾必须以换行符结束,且不能包含额外的空行。
14. 强制禁止 print 调试,需要使用日志库 logging 模块进行日志记录。
15. 禁止在函数体内定义可变对象作为默认参数
16. 禁止多行 lambda 表达式,如需多行逻辑,必须使用普通的 def 函数
17. 布尔表达式中禁止 “`== True`” 或 “`== False`”
18. 所有常量值声明应在文件开头
19. 如需多行逻辑,必须使用普通的 def 函数。
20. 类的公有属性禁止直接访问私有属性
21. 类的继承列表不超过一行时,与类名同一行,继承过长时才换行,且新行与左括号对齐
22. 所有的异常必须继承自 Exception 基类
23. 避免裸露的 except: 捕获,必须捕获指定异常类型,不能使用 `except:` 或 `except Exception:`,除非确有必要,并且要在注释中说明原因。
24. 模块导入时禁止循环依赖
25. 注释必须使用 英文 (项目若需双语,可在英文后添加中文说明) 首字母大写,末尾不加句号。

b. 推荐

- 26. 注释优先使用 “# ” 而非 “''' '''” 做行内说明
- 27. 函数长度建议不超过 50 行
- 28. 模块级常量建议按类型分组，并用空行隔开不同类别
- 29. 在循环中若频繁使用常量或函数返回值，可考虑将其提到循环外部
- 30. 不要在同一行中混合布尔运算符与赋值
- 31. 单元测试文件与生产代码文件分开管理，建议使用 tests/ 目录，测试文件命名以 test_*.py 或 *_test.py 结尾。
- 32. 对于公共库，请在 setup.py 中明确列出依赖包及版本范围
- 33. 字典遍历时，若只需要键或值，优先使用 dict.keys() 或 dict.values()，避免 for key in my_dict: 后再 my_dict[key]
- 34. 禁止在循环中直接修改列表，可改为使用 for item in list.copy(): 或 列表推导创建新列表
- 35. 推荐对长字符串或 SQL 语句使用三引号多行字符串，并对齐左侧缩进
- 36. 推荐使用 pathlib.Path 管理文件路径，而非直接使用字符串拼接
- 37. 推荐使用 enumerate() 而非 range(len()) 遍历序列
- 38. 推荐对异常捕获后，使用 raise 保留原始 Traceback
- 39. 在配置文件或常量文件中，推荐使用 dotenv 或 配置解析库管理环境变量
- 40. 模块内部导入按使用场景排序：标准库-->第三方库-->本地模块，即使在同一组中，也按代码逻辑用途进行适当分类，并保留空行分隔。

c. 允许

- 41. 单行 if 语句在逻辑简单时可使用三元表达式
- 42. 模块内部临时调试可使用 print()，但提交前务必移除或替换为日志
- 43. 在脚本级别允许使用 argparse 等库解析命令行参数,但禁止直接使用 sys.argv 做复杂解析。
- 44. 私有方法可使用单下划线 _ 前缀，但如果确实需要暴露给测试，可在测试代码中直接导入
- 45. 允许在满足可读性的情况下使用行尾注释，但每行不超过 80 字符
- 46. 允许在同一文件中嵌套定义小型辅助函数（nested function），仅限函数逻辑清晰且生命周期仅限于外层函数时采用。
- 47. 对于一些一次性脚本，可将 main() 逻辑直接写在顶层，但需包含 if __name__ == '__main__': 保护
- 48. 允许根据项目需要选择不同的测试覆盖率阈值，但建议不低于 80%
- 49. 允许在私有包内部对部分函数使用 CamelCase 命名（若与第三方库保持一致）
- 50. 允许在项目中使用第三方代码格式化工具（如 Black、YAPF），但产出结果需符合上述强制/推荐规范
- 51. 允许在临时分支进行大型重构或风格转换，但必须在合并前进行全量代码审查，确保符合本规范
- 52. 允许在类内部定义 slots 以节省内存