# Python 技术管理规范文档

**规范等级说明：**

* **a. 强制**：必须遵守，违反即视为代码错误或需整改；
* **b. 推荐**：建议遵守，可视情况酌情；
* **c. 允许**：可选择采用，供特定场景使用。

## 一、命名规范（8 条）

| **编号** | **规则** | **级别** |
| --- | --- | --- |
| 1 | 模块名使用全小写字母，必要时使用下划线分隔（如：data\_loader.py） | a |
| 2 | 类名使用驼峰命名法，首字母大写（如：DataProcessor） | a |
| 3 | 函数名、变量名使用小写加下划线命名（如：get\_data()） | a |
| 4 | 常量名使用全大写加下划线（如：MAX\_RETRIES = 3） | a |
| 5 | 私有变量/函数前加单下划线（如：\_internal\_func()） | b |
| 6 | 避免使用单字符命名（除非是数学公式或循环变量） | b |
| 7 | 避免命名冲突，如 list, str 等内建类型 | b |
| 8 | 临时变量可使用 \_ 作为占位 | c |

## 二、代码格式（10 条）

| **编号** | **规则** | **级别** |
| --- | --- | --- |
| 9 | 使用 4 个空格进行缩进 | a |
| 10 | 每行最多 100 个字符（尽量不超过 80） | a |
| 11 | 类、函数、模块之间使用 1~2 行空行隔开 | b |
| 12 | 文件末尾必须保留一个换行符 | a |
| 13 | 运算符两侧加空格（如：a = b + c） | a |
| 14 | 函数定义时，参数之间用逗号+空格分隔 | a |
| 15 | 括号内不要出现多余空格（如：func(x, y) 而非 func( x, y )） | a |
| 16 | 不使用制表符（Tab）进行缩进 | a |
| 17 | 导入模块顺序：标准库 -> 第三方库 -> 自定义模块 | a |
| 18 | 多个导入语句应分行书写，不使用 import x, y | b |

## 三、注释与文档（9 条）

| **编号** | **规则** | **级别** |
| --- | --- | --- |
| 19 | 所有模块、类、函数必须有 docstring（使用三引号） | a |
| 20 | 函数 docstring 应说明参数、返回值和异常 | b |
| 21 | 注释应为完整句子，首字母大写，句末加句号 | b |
| 22 | 代码逻辑复杂处添加行注释（以 # 开头） | b |
| 23 | 避免使用无意义的注释（如：# define variable） | a |
| 24 | 中文注释必须确保编码为 UTF-8 且无乱码 | a |
| 25 | TODO 应注明责任人和完成时间 | b |
| 26 | 不在生产代码中保留 DEBUG 注释或 print | a |
| 27 | 所有外部接口（API）必须添加注释说明用途 | a |

## 四、异常与日志（5 条）

| **编号** | **规则** | **级别** |
| --- | --- | --- |
| 28 | 尽量使用具体异常类型（如：ValueError） | a |
| 29 | 不使用裸 except:，应写为 except Exception as e: | a |
| 30 | 所有异常必须记录日志或抛出，不可静默处理 | a |
| 31 | 使用 logging 模块而不是 print 打日志 | a |
| 32 | 日志应区分等级（debug/info/warning/error） | b |

## 五、函数与类设计（5 条）

| **编号** | **规则** | **级别** |
| --- | --- | --- |
| 33 | 函数长度建议控制在 50 行以内 | b |
| 34 | 一个函数只做一件事，避免过多 if-else 嵌套 | a |
| 35 | 尽量使用组合而非继承（prefer composition over inheritance） | b |
| 36 | 类的职责应单一，不应承载过多逻辑 | a |
| 37 | 不在函数体外定义可变全局变量 | a |

## 六、测试与可维护性（3 条）

| **编号** | **规则** | **级别** |
| --- | --- | --- |
| 38 | 所有核心功能必须写单元测试，测试覆盖率不低于 80% | a |
| 39 | 测试用例应覆盖边界条件与异常情况 | b |
| 40 | 单元测试文件名以 test\_\*.py 命名，使用 pytest/unittest 框架 | b |