# 1.  概述

[(Edit)](http://www.elias.cn/Python/PythonStyleGuide?action=edit&s=2&auto=y&from=Python.PythonStyleGuide" \o "edit section:  Python 语言方面的准则)

## 1.1  Python 语言方面的准则

1. [pychecker](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#pychecker): 建议使用
2. [导入模块和包](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Module_and_package_imports): 可以，**但不要 import \***
3. [完整路径导入](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Packages): 可以
4. [异常处理](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Exceptions): 可以
5. [全局变量](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Global_variables): 谨慎使用
6. [内嵌/本地/内部类和函数](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Nested_Local_Inner_Classes_and_Functions): 可以
7. [List Comprehensions](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#List_Comprehensions): 可以用，如果**简明易懂**的话
8. [默认迭代器和运算符](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Default_Iterators_and_Operators): 可以
9. [生成器](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Generators): 可以
10. [使用 apply、 filter、 map、 reduce](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Using_apply_filter_map_reduce): 对**one-liner**来说可以
11. [Lambda 函数](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Lambda_functions): 对**one-liner**来说可以
12. [默认参数值](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Default_Argument_Values): 可以
13. [Properties](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Properties): 可以
14. [True/False 求值](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#True_False_evaluations): 可以
15. [布尔内置类型](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Boolean_built_in_type): 可以
16. [String 方法](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#String_Methods): 可以
17. [静态域](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Lexical_Scoping): 可以
18. [函数和方法修饰符](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Function_and_Method_Decorators): **适度**使用
19. [线程](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Threading): **不要用**（[Google App Engine](http://code.google.com/appengine/kb/general.html#libraries)不支持）
20. [高级特性](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Power_features): **不要用**

## 1.2  Python 编码风格方面的准则

所有文件都保持一致风格时程序要容易维护得多，以下是 Melange 项目的标准 Python 编码风格规范：

1. [分号](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Semicolons): **避免使用**
2. [每行长度](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Line_length): 最多**80**列
3. [圆括号](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Parentheses): **吝啬地用**
4. [缩进](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Indentation): **4** 空格（**不要用 tab**），**和**[**PEP8**](http://www.python.org/dev/peps/pep-0008/)**有所区别**
5. [空行](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Blank_lines): 对函数和类用 **2** 个空行分隔，对类的方法用 **1** 个空行
6. [空格](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Whitespace): 在行内吝啬地使用
7. [Python 解释器](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Python_Interpreter): 用[Google App Engine](http://code.google.com/appengine/docs/python/purepython.html)支持的那个： #!/usr/bin/python2.5
8. [注释](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Comments): \_\_doc\_\_ strings、块注释、行内注释
   * [\_\_doc\_\_ Strings](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Doc_strings)
   * [模块](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Modules)
   * [函数和方法](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Functions_and_Methods)
   * [类](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Classes)
   * [块注释及行内注释](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Block_and_Inline_Comments)
9. [类](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Classes): 继承 object
10. [字符串](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Strings): 避免重复使用 + 和 +=
11. [TODO style](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#TODO_style): TODO(username): 使用一种一致的风格
12. [import 分组、排序和整理](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Imports_grouping_and_order): 一行一个，按包名字分组顺序放置，按字母顺序排序
13. [语句](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Statements): 一行一个，避免使用[分号](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide" \l "Semicolons)
14. [访问控制](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Access_control): 轻量化风格可以用 foo，否则用 GetFoo() 和 SetFoo()
15. [命名](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Naming): 用 foo\_bar.py 而不是 foo-bar.py，和[**PEP8**](http://www.python.org/dev/peps/pep-0008/)**有些区别**
16. [程序入口](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Main): if \_\_name\_\_ == '\_\_main\_\_':
17. [总结](http://www.elias.cn/Python/PythonStyleGuide?from=Develop.PythonStyleGuide#Conclusion): 看看你周围是什么