# [Python编码规范](http://blog.csdn.net/yefengnidie/article/details/6721781)

2011-08-26 13:51 3219人阅读 [评论](http://blog.csdn.net/yefengnidie/article/details/6721781#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/yefengnidie/article/details/6721781#report)

[python](http://www.csdn.net/tag/python)[import](http://www.csdn.net/tag/import)[callback](http://www.csdn.net/tag/callback)[function](http://www.csdn.net/tag/function)[object](http://www.csdn.net/tag/object)[class](http://www.csdn.net/tag/class)

目录[(?)[+]](http://blog.csdn.net/yefengnidie/article/details/6721781)

|  |
| --- |
| 标准头部 #!/usr/bin/python 编码 所有的 Python 脚本文件都应在文件头标上 #-\*- coding:utf8 -\*- 。设置编辑器，默认保存为 utf8 格式。 main函数 应用程序要定义main函数而不要直接写脚本  def main():  dosomething()  if \_\_name\_\_ == "\_\_main\_\_":  main()  如果是全局变量，则需要显式加上"global"。 注释 Python 的注释分为两种的概念，一种是由 # 开头的"真正的"注释，另一种是 docstrings。前者表明为何选择当前实现以及这种实现的原理和难点，后者表明如何使用这个包、模块、类、函数（方法），甚至包括使用示例和单元测试。 坚持适当注释原则。对不存在技术难点的代码坚持不注释，对存在技术难点的代码必须注释。但与注释不同，推荐对每一个包、模块、类、函数（方法）写 docstrings，除非代码一目了然，非常简单。 注释块通常应用于跟随着一些（或者全部）代码并和这些代码有着相同的缩进层次。注释块中每行以'#'和一个空格开始（除非他是注释内的缩进文本）。注释块内的段落以仅含单个'#'的行分割。注释块上下方要有一空行包围。 缩进 Python 代码缩进为4个空格，在文件结尾声明  # Modeline {{{  # vim:set ts=4:  # vim600:fdm=marker fdl=0 fdc=0  # }}} 空格 空格在Python 代码中是有意义的，因为Python的语法依赖于缩进，在行首的空格称为前导空格。在这一节不讨论前导空格相关的内容，只讨论非前导空格。非前导空格在 Python 代码中没有意义，但适当地加入非前导空格可以增进代码的可读性。 1）在二元算术、逻辑运算符前后加空格：如 a = b + c； 2）在一元前缀运算符后不加空格，如 if !flg: pass； 3）":"用在行尾时前后皆不加空格，如分支、循环、函数和类定义语言；用在非行尾时后端加空格，如 dict 对象的定义 d = {'key': 'value'}。 4）括号（含圆括号、方括号和花括号）前后不加空格，如 do\_something(arg1, arg2)，而不是 do\_something( arg1, arg2 )； 5）逗号后面加一个空格，前面不加空格。 空行 1）在类、函数的定义间加空行； 2）在import不同种类的模块间加工行； 3）在函数中的逻辑段落间加空行，即把相关的代码紧凑写在一起，作为一个逻辑段落，段落间以空行分隔。 断行 行的最大长度不得超过 80 个字符的标准。折叠长行的方法有以下几种方法： 1）为长变量名换一个短名，如：  this.is.a.very.long.variable\_name = this.is.another.long.variable\_name  应改为：  variable\_name1 = this.is.a.very.long.variable\_name  variable\_name2 = this.is.another.variable\_name  variable\_name1 = variable\_name2  2）在括号（包括圆括号、方括号和花括号）内换行，如：  class Edit(CBase):  def \_\_init\_\_(self, parent, width,  font = FONT, color = BLACK, pos = POS, style = 0):  或：  very\_very\_very\_long\_variable\_name = Edit(parent, \  width, \  font, \  color, \  pos)  2）在长行加入续行符强行断行，断行的位置应在操作符前，且换行后多一个缩进，以使维护人员看代码的时候看到代码行首即可判定这里存在换行，如：  if color == WHITE or color == BLACK \  or color == BLUE:  do\_something(color); 命名常量 常量名所有字母大写，由下划线连接各个单词，如：  WHITE = 0XFFFFFF  THIS\_IS\_A\_CONSTANT = 1 变量 变量名全部小写，由下划线连接各个单词，如：  color = WHITE  this\_is\_a\_variable = 1  不论是类成员变量还是全局变量，均不使用m或g前缀。私有类成员使用单一下划线前缀标识。 变量名不应带有类型信息，因为 Python 是动态类型语言。如 iValue、names\_list、dict\_obj 等都是不好的命名。 函数 函数名的命名规则与变量名相同。 类 类名单词首字母大写，不使用下划线连接单词，也不加入 C、T 等前缀。如：  class ThisIsAClass(object):  pass 模块 模块名全部小写，对于包内使用的模块，可以加一个下划线前缀，如：  module.py  \_internal\_module.py 包 包的命名规范与模块相同。 缩写 命名应当尽量使用全拼写的单词，缩写的情况有如下两种： 1）常用的缩写，如 XML、ID等，在命名时也应只大写首字母，如 class XmlParser(object):pass 2）命名中含有长单词，对某个单词进行缩写。这时应使用约定成俗的缩写方式，如去除元音、包含辅音的首字符等方式，例如： function 缩写为 fn text 缩写为 txt object 缩写为 obj count 缩写为 cnt number 缩写为 num，等。 特定命名方式 主要是指 \_\_xxx\_\_ 形式的系统保留字命名法。项目中也可以使用这种命名，它的意义在于这种形式的变量是只读的，这种形式的类成员函数尽量不要重载。如  class Base(object):  def \_\_init\_\_(self, id, parent = None):  self.\_\_id\_\_ = id  self.\_\_parent\_\_ = parent  def \_\_message\_\_(self, msgid):  # ...略  其中 \_id、parent\_ 和 \_message\_ 都采用了系统保留字命名法。 语句import import 语句有以下几个原则需要遵守： 1）import 的次序，先import Python内置模块，再import第三方模块，最后import自己开发的项目中的其它模块；这几种模块中用空行分隔开来。 2）一条import语句import一个模块。 3）当从模块中 import 多个对象且超过一行时，使用如下断行法（此语法 py2.5 以上版本才支持）： from module import (obj1, obj2, obj3, obj4, obj5, obj6) 4）不要使用 from module import \*，除非是 import 常量定义模块或其它你确保不会出现命名空间冲突的模块。 赋值 对于赋值语言，主要是不要做无谓的对齐，如：  a = 1  variable = 2  fn = callback\_function  应该写成：  a = 1  variable = 2  fn = callback\_function 分支和循环 不要写成一行，如：  if !flg: pass  for i in xrange(10): print i  应该写成：  if !flg:  pass  for i in xrange(10):  print i 其他建议 避免在内存中处理大对象，由于python申请的内存不一定会返还给操作系统（有时会返还给python内存池 |