

# PEP 8风格指南

PEP是Python Enhancement Proposal的缩写，通常翻译为“Python增强提案”。每个PEP都是一份为Python社区提供的指导Python往更好的方向发展的技术文档，其中的第8号增强提案（PEP 8）是针对Python语言编订的代码风格指南。尽管我们可以在保证语法没有问题的前提下随意书写Python代码，但是在实际开发中，采用一致的风格书写出可读性强的代码是每个专业的程序员应该做到的事情，也是每个公司的编程规范中会提出的要求，这些在多人协作开发一个项目（团队开发）的时候显得尤为重要。我们可以从Python官方网站的[PEP 8链接](#)中找到该文档，下面我们对该文档的关键部分做一个简单的总结。

## 空格的使用

1. 使用空格来表示缩进而不要用制表符（Tab）。这一点对习惯了其他编程语言的人来说简直觉得不可理喻，因为绝大多数的程序员都会用Tab来表示缩进，但是要知道Python并没有像C/C++或Java那样的用花括号来构造一个代码块的语法，在Python中分支和循环结构都使用缩进来表示哪些代码属于同一个级别，鉴于此Python代码对缩进以及缩进宽度的依赖比其他很多语言都强得多。在不同的编辑器中，Tab的宽度可能是2、4或8个字符，甚至是其他更离谱的值，用Tab来表示缩进对Python代码来说可能是一场灾难。
2. 和语法相关的每一层缩进都用4个空格来表示。
3. 每行的字符数不要超过79个字符，如果表达式因太长而占据了多行，除了首行之外的其余各行都应该在正常的缩进宽度上再加上4个空格。
4. 函数和类的定义，代码前后都要用两个空行进行分隔。
5. 在同一个类中，各个方法之间应该用一个空行进行分隔。
6. 二元运算符的左右两侧应该保留一个空格，而且只要一个空格就好。

## 标识符命名

PEP 8倡导用不同的命名风格来命名Python中不同的标识符，以便在阅读代码时能够通过标识符的名称来确定该标识符在Python中扮演了怎样的角色（在这一点上，Python自己的内置模块以及某些第三方模块都做得并不是很好）。

1. 变量、函数和属性应该使用小写字母来拼写，如果有多个单词就使用下划线进行连接。
2. 类中受保护的实例属性，应该以一个下划线开头。
3. 类中私有的实例属性，应该以两个下划线开头。
4. 类和异常的命名，应该每个单词首字母大写。
5. 模块级别的常量，应该采用全大写字母，如果有多个单词就用下划线进行连接。
6. 类的实例方法，应该把第一个参数命名为self以表示对象自身。
7. 类的类方法，应该把第一个参数命名为cls以表示该类自身。

## 表达式和语句

在Python之禅（可以使用`import this`查看）中有这么一句名言：“There should be one -- and preferably only one -- obvious way to do it.”，翻译成中文是“做一件事应该有而且最好只有一种确切的做法”，这句话传达的思想在PEP 8中也是无处不在的。

1. 采用内联形式的否定词，而不要把否定词放在整个表达式的前面。例如`if a is not b`就比`if not a is b`更容易让人理解。
2. 不要用检查长度的方式来判断字符串、列表等是否为None或者没有元素，应该用`if not x`这样的写法来检查它。
3. 就算if分支、for循环、except异常捕获等中只有一行代码，也不要将代码和if、for、except等写在一起，分开写才会让代码更清晰。
4. import语句总是放在文件开头的地方。
5. 引入模块的时候，from math import sqrt比import math更好。
6. 如果有多个import语句，应该将其分为三部分，从上到下分别是Python标准模块、第三方模块和自定义模块，每个部分内部应该按照模块名称的字母表顺序来排列。