

PEP

PEP是每一个Python程序员必须知道的。

PEP是Python Enhancement Proposals的缩写（Python增强提案）。一个PEP是一份为Python社区提供各种增强功能的技术规格，也是提交新特性，以便让社区指出问题，精确化技术文档的提案。

每一个Python版本的新特性或变化都是将PEP提案通过社区决策层讨论、投票决议，通过后才有我们看到的功能变化。

Guido van Rossum被戏称为"仁慈的独裁者"(benevolent dictator for life, BDFL)。他会亲自签署并确认每一份PEP。但在2018年7月，由于PEP 572（Assignment Expressions）这个提案，他已经卸任BDFL。没有继任者，没有管理原则，PEP路在何方？

注：PEP 572影响到了几乎所有的Python用户，但相当多的人不认可

PEP 0

<https://www.python.org/dev/peps/>

PEP规范列表，PEP文档索引

PEP 1 -- PEP Purpose and Guidelines

PEP协议指南

PEP 8

由Python之父编写的Python 编码风格指导。被Python社区广泛采纳，标准库也按照这个规范编写。

<https://www.python.org/dev/peps/pep-0008/>

<http://www.magedu.com/70951.html>

<https://yq.aliyun.com/articles/626638>

PEP 20

Python之禅

import this

PEP 257

文档字符串的规范

<https://www.python.org/dev/peps/pep-0257/>

<https://my.oschina.net/LuCastiel/blog/1552148>

PEP8016

Python社区治理方案，这是Python之父卸任BDFL之后，社区从几种方案中投票胜出的方案。

PEP 8016 治理方案采用指导委员会模式，其特点是引导治理的迭代，该方案由 Nathaniel J. Smith 和 Donald Stufft 提出。

PEP 8016 中提出了不信任投票，也就是弹劾机制，可将任期内的当权者赶下台；它严格限定了在委员会里，只允许少于 50% 的成员是企业（5 人委员会里最多有 2 个）；并且关注到核心开发者的选举/淘汰、如何更新治理提案等问题。

而且，PEP 8016 中也提出了新的 PEP 流程，目前的 PEP 流程是提案人确定 PEP 的选题方向，提案人负责收集与整合来自整个社区的反馈。然后，相关领域的专家们汇总全部讨论，并开启为期 14 天的审查，之后进行社区投票。如果一个 PEP 很有争议，任何专家成员都可发起动议来拒绝通过它，这需要超过 2/3 的票数。

PEP 8016 的 PEP 流程：指导委员会在必要时可直接地批准/否决 PEP，但最好是设置流程来避免这样做决策，例如，将决策权委派给团队或者 BDFL 代表。

2019年2月4日，为期2周的投票后，Python社区选出来了指导委员会的5名成员。

5 人分别是：

- Barry Warsaw：自1995年起成为核心开发者之一，荣获 2014 年的弗兰克·威利森纪念奖。目前供职于 LinkedIn（已被微软收购，也即供职于微软），业余爱好是音乐和太极。
- Brett Cannon：自2003年起成为核心开发者之一，荣获 2016 年的弗兰克·威利森纪念奖。曾担任 Python 软件基金会的执行副主席。目前供职于微软，负责 VSCode 的 Python 插件项目。
- Carol Willing：Python 核心开发者，Jupyter 核心开发者及 Jupyter 的指导委员会成员。自由职业，兴趣在于科研及教育项目。
- Guido van Rossum：Python 的创始人，被称为“Python 之父”，长期领导 Python 社区的发展，直到此次的退位风波。目前供职于 Dropbox。
- Nick Coghlan：自2005年起成为核心开发者之一。目前供职于 Tritium