

《代码英雄》第三季（1）： Python 的故事

代码英雄讲述了开发人员、程序员、黑客、极客和开源反叛者如何彻底改变技术前景的真实史诗。

什么是《代码英雄》

Command Line Heroes

代码英雄是世界领先的企业开源软件解决方案供应商红帽（Red Hat）精心制作的原创音频播客，讲述开发人员、程序员、黑客、极客和开源反叛者如何彻底改变技术前景的真实史诗。该音频博客邀请到了谷歌、NASA 等重量级企业的众多技术大牛共同讲述开源、操作系统、容器、DevOps、混合云等发展过程中的动人故事。

本文是《[代码英雄](#)》系列播客[第三季（1）：Python 的故事](#)的[音频](#)脚本。

benevolent dictator for life

导语：一位“仁慈的终身独裁者（BDFL）”的辞职，永久改变了 Python 语言的命运，Guido van Rossum 那个名为《移交权利》的备忘录，让大家的注意力集中到了语言发展这件事情上。

在这一期里，Emily Morehouse 将 Python 技术层面的拓展性和它的包容性社区联系在了一起。Michael Kennedy 解释了 Python 是如何做到在简单易学的同时，又强大到足以担当的起像 Youtube 和 Instagram 这样的大项目。而 Diane Mueller 则强调了 Python 社区是如何率先在科技领域传播了许多包容性的实践——包括社区主导的决策的兴起。

有时候，一位仁慈的终身独裁者可以让一个语言开始发展，但 Python 证明了，是社区让语言兴旺起来的。

00:00:06 - Saron Yitbarek:

在 2018 年 7 月 12 日的早晨，世界各地的 Python 社区成员起床之后，他们拿起一杯咖啡，打开了自己的电脑。随后一个接着一个地，看见了一条来自他们仁慈的独裁者的消息。

00:00:24:

Guido van Rossum，世界上最伟大的编程语言之一 Python 的创造者，也许没有之一。他面向 Python 社区的全体成员写下这条消息，我们不难想象出 Python 的粉丝们阅读这条消息时的画面。

00:00:38 - 多个配音演员:

我没想到在我已经如此努力的情况下，仍有这么多人对我的决策不满。我想把自己从决策过程中完全抽离出来，给自己一个永久的假期，让自己一辈子都不做仁慈的独裁者。你们都要靠自己了，我不会指定继任者。那么你们接下来要如何做呢，创立民主制度吗？或者保持无政府状态？

00:01:00 - Saron Yitbarek:

在这条简短却惊人的消息发布之后，Guido van Rossum，这个被 Python 社区追随了几十年的人.....好吧，他基本上就这样退出了。这条消息被命名为《移交权利》，它将永久的改变 Python 的格局。

00:01:19:

不过除此之外，它又引出一个问题，那就是不光是 Python，所有的这些编程语言在未来要怎样衍变与壮大呢？是否应该由一个仁慈的独裁者把控，赋予它们形状和一致性？又或者，在开源世界里，编程语言的发展与变化是否应该更像口语，取决于所有语言使用者的行为？Python 社区，这个成长速度远超其他语言的社区，需要找到这个问题的答案。

00:01:56:

我是 Saron Yitbarek，这里是《代码英雄》第三季，一个来自红帽的原创播客。在上一季的《代码英雄》里，我们探索了广袤天地，从[游戏主题](#)出发，到[失败的艺术](#)，再到[无服务器开发](#)，最后我们甚至追踪

了一台在火星表面，来自 NASA 的[火星车](#)。但是真正抓住每个人想象力的一集，似乎是 [Grace Hopper 的故事](#)。她参与开发的编译器创造出了世上第一个高级编程语言，COBOL。我们随后意识到，Grace Hopper 的故事不过是万千个塑造了软件开发与运维的世界的故事之一。新的编程语言使得我们可以连通人类与机器，它们为我们打开了一扇通往无限可能的大门。

00:02:51:

因此，第三季将全部围绕这些语言展开。我们会谈论 JavaScript、BASIC、Go 以及 Perl，当然，还有 Python。我们的旅程将从 Python 开始，顺着 Python 的故事，我们会学到一个与所有编程语言未来相关的至关重要的事实。

00:03:16:

在 Python 的仁慈的独裁者放弃王座之后，整个 Python 社区.....呃，有些迷茫。你要如何在独裁者退位之后组织工作呢？有人提议采用

Presbyterian church

长老会结构，不过没能奏效。要弄清楚 Python 是如何重新找到领导方式的，并了解这对其它语言的未来有什么意义，我们必须回到 Python 故事的源头。

00:03:46 - Guido van Rossum:

好吧，让我用 C 来完成全部代码，这事情变得有些枯燥。

00:03:51 - Saron Yitbarek:

本尊驾到，Guido van Rossum，Python 的仁慈的独裁者。Van Rossum 曾在为阿姆斯特丹著名的 Centrum Wiskunde & Informatica 工作数年，在那里他协助开发了 ABC 编程语言。现在，他将描述他使用 C 工作时，发现对一个全新编程语言产生需求的时刻。

00:04:13 - Guido van Rossum:

仍然感觉还是有着很多 bug，并且开发速度缓慢。我在想，呃，如果这里支持 ABC 语言，我可以在十五分钟内写出整个登录程序，然后我可以接着开发账户管理之类的功能，但是用 C 进行开发，却要花掉

我一整个星期。我开始设想，要提出一种可以在 **Amoeba** 环境中使用 **ABC** 语言特性的方法。

00:04:47 - Saron Yitbarek:

在研究编程语言的历史时我们发现，没有什么东西是全新的。它们全都是为了拼凑出解决方案而从旧语言借鉴来的。语言会不断的变化、发展、开枝散叶。当 **van Rossum** 被种种可能性弄得迷茫时，他设想出一个可以弥合 **C** 与 **Shell** 编程两者间隙的编程语言。一些场景下使用 **C** 语言往往是牛刀杀鸡，与此同时，用 **Shell** 脚本又常常让人感到繁琐，二者之间的最佳结合点，正是 **Python** 的栖身之处。当 **van Rossum** 在 1991 年首次发布 **Python** 时，它给大家都带来了启发，尤其是对系统管理员而言。这是一种前无来者，功能全面的脚本语言。

00:05:35 - Emily Morehouse:

第一次使用 **Python** 时，我立即就爱上了它。

00:05:39 - Saron Yitbarek:

这是 **Emily Morehouse**，目前 **Python** 担任核心开发者的五位女性之一。

00:05:44 - Emily Morehouse:

我认为当你从接触到的像 **C++** 这样的第一门语言跨越到 **Python** 时，发现二者之间如此显著的差异，会真的会意识到语言与其设计的优美之处。你不必去应付内存管理的毛糙的底层实现细节，它可以既快又好的构建一些项目，更不必说它还有着如此广泛的应用领域。

00:06:18 - Saron Yitbarek:

Python 吸引力的关键之处，就在于它的可扩展性。举个例子，像 **ABC** 这样的语言，在设计上是固化单一的，无法形成一个能够真正帮助改善它的社区。反观 **Python**，**van Rossum** 在设计之初就希望 **Python** 是开放的、易于扩展的。

00:06:37 - Emily Morehouse:

进行软件设计时，你往往需要用到一些现有的软件或系统，并且要让他们能够协同工作，其实软件设计的核心价值之一就在于确保它的可扩展性。

00:06:58 - Saron Yitbarek:

听起来不怎么费脑子，但并非每一个语言都像 Python 这样，有着与生俱来的强可扩展性。事实上，一门不具备可扩展性的语言，大概率会随着自身的发展而崩溃。

00:07:16 - Emily Morehouse:

Python 有一个非常有趣的设计，它允许在其内核上进行扩展。实际上，你可以在运行时环境上对系统的不同片段打补丁，假如你想要改变模块导入的方式，或者你想改变你的字符串类型或整数类型，Python 允许你用相当容易的方式去实现这些事。

00:07:44:

Python 可扩展性的核心是所谓的 C 扩展，或者说是 C 模块。因此，Python 实际上被设计出来的意图，是为你提供对其它语言的切入点。本质上来讲，如果你可以写出 C 扩展或 C 模块，可以用它去桥接其它数百种语言，那么你在某种角度上算是破解了 Python 的设计，再造了一个它。

00:08:16 - Saron Yitbarek:

这完全取决于用户根据自身需求调整语言的能力。所以说 Python，按照 Guido van Rossum 的设想，绝不会局限于独裁者的个人视野。他的《移交权利》备忘录由来已久。Van Rossum 明白社区影响力所能产生的力量，这是一种能够使所有人都聚集于同一顶“大帐篷”之下的力量。没错，他最终被冠以独裁者称号，但却是一名仁慈的独裁者。

00:08:44 - Emily Morehouse:

我认为 Python 社区变得如此多元化，原因之一就是 Guido。Python 现在能够拥有女性核心开发者，正是因为 Guido 希望有这样的改变，并一手推动其实现。

00:09:01 - Saron Yitbarek:

Python 软件基金会的主席 Naomi Ceder 曾在一次演讲中说：“Python，因语言而来，为社区而留。”这可能正是 Guido van Rossum 留下的最棒的礼物。不仅仅是 Python 语言，他更是为整个 Python 社区都创造了空间。他不仅在语法意义上为 Python 提供了可扩展性，某种角度来讲，他也在社会意义上提供了扩展性。永远都为新来者留有位置。

00:09:29 - Emily Morehouse:

Python 有着大量不同的应用领域，而社区又如此多元化。。所以它真的、真的扩展了社区的边界。

00:09:51:

Emily Morehouse 是 Python 核心开发者与 Cuttlesoft 的工程总监。

00:09:59:

Python 降世之后，它就开始了前所未有的增长。我看到一个 Stack Overflow 上的图表，统计了每门语言被提及的次数，对应着 Python 的那条线仿佛坐上了火箭。2018 年，在 Google 搜索 Python 的人数比搜索金-卡戴珊的还多。另一个令人振奋的消息是，Python 开始和 Java、C 以及 C++ 之类的语言争夺最常用语言的头衔。

00:10:26:

那么，这些人们对 Python 的偏爱究竟来源何处呢？为了找到答案，我找到 Michael Kennedy，他活跃在 Python 时代的中心。他主持着两个专注于 Python 的播客：Talk Python to Me 和 Python Bytes。我们会在展示内容里放上对应的链接以便大家查看。Michael 将要和我一起讨论 Python 是如何真正迈出大步的。

00:10:52 - Michael Kennedy:

如果你看过分析报告或总结报告一类的东西，会发现 2012 年是一个重要拐点，而发生在 2012 年左右最具意义的事情莫过于，数据科学社区换掉了 R 之类的东西，开始真正专注于 Python。这件事发生以

后，Python 社区就有了更充足的劲头，也有了更多的机器学习库。像许多流行的机器学习库，都会首先考虑使用 Python 来实现，此后才会考虑其他语言。

00:11:22 - Saron Yitbarek:

嗯，我也这么认为。当我想到 Python 时，我知道它可以被用作 Web 开发，我也认识许多仍然在使用 Python 来写 Web 应用的人。但我感到 Python 如今真正的核心更多是在数据科学领域。你认为是什么导致了这件事情发生？为什么数据科学社区抛弃了那些，或者我不应该叫抛弃，而是远离了 R 之类的工具。

00:11:44 - Michael Kennedy:

对，正是如此。

00:11:45 - Saron Yitbarek:

从何而来呢？

00:11:46 - Michael Kennedy:

我认为这种转变中是有两件事在起作用。其中之一就是 Python 作为一种“真正的”——引号引起来的——编程语言，它既可以写一些简单的东西，例如构建图形或数据分析工具等等，但是，它也可以用来构建 Instagram 和 YouTube 以及其他类似的复杂应用……

00:12:08 - Saron Yitbarek:

是的，显而易见。

00:12:09 - Michael Kennedy:

相对于 R …… 之类的东西而言。对，这很显而易见。很多应用都是由 Python 来编写的。所以他们之前在使用一些其它的编程语言时，例如 R，这门专精数据科学工作的，科学统计类的编程语言，如果他们想要写一个 Web 应用来展示他们的结果，那么，他们要用来实现呢？Node 或者 Python 还是……这导致他们没办法一直使用那些语言。

00:12:31 - Saron Yitbarek:

对，说的很好。

00:12:31 - Michael Kennedy:

所以说 Python 拥有相当棒的能力 基本上，它是一种真正的编程语言，这是其一。第二点是，Python 十分独特，被我称为.....全频谱语言。我所谓的全频谱的意思是指，假如我是一名生物学家、天文学家或别的什么，当我想要处理一些数据，想要加载一个 CSV 文件并运行一些指令得到图像时，我无需理解什么是类、静态方法、主函数、编译或者链接。你不需要像一些编程语言那样，为了入门而去学会这些。我只用写几行代码，键入指令，程序就可以运行。

00:13:14:

但是，你也可以用它去搭建像 Instagram 之类的软件。它可以成长为绝对专业的系统，你能使用，但是你不会在一开始就被逼迫着，去理解大型应用的一切深层抽象，你可以根据自身需要来使用它。这有意义吗？

00:13:30 - Saron Yitbarek:

对，对。说的非常贴切。刚才我们谈论到 2012 年的拐点时，我搜索了一些关于 Python 的资料，Python 的确是世界上被 Google 次数最多的编程语言。

00:13:42 - Michael Kennedy:

哇。

00:13:43 - Saron Yitbarek:

你感觉它现在确实被大家青睐，并且仍在成长之中吗？

00:13:47 - Michael Kennedy:

我认为它真的被青睐并处于成长之中。在我们谈论到的过去几年里，有相当多的企业集团都选择使用 Python。而他们曾经的选择是

.NET、Java，甚至是 C 。这就是回答，并且现在 Python 开始走出属于自己的路，我认为它在某些场景下被格外倚重，关于这个，我指的是那些数据科学家们。就像，显然我们用到的 Jupyter Lab 那类炫酷笔记本应用，这些的东西，全都是 Python 造就的。

00:14:17:

数据科学没有祖传代码的负担。如果我想开发一个新的项目，用来探索广告活动或科学结果，就不会对一堆陈旧的东西，保持着庞大的依赖关系。模型和数据都会过时，因此对于数据科学世界而言，切换技术栈或保持最新状态要容易得多。

00:14:37 - Saron Yitbarek:

说的不错。

00:14:39 - Michael Kennedy:

嗯，谢谢。

00:14:42 - Saron Yitbarek:

嗯，听上去 Python 不会很快停止成长，而是保持增长的趋势，甚至这种增长的势头仍在向前发展。你认为对这种增长与前进的势头影响最大的事情是什么呢？

00:14:52 - Michael Kennedy:

我感觉这就像滚雪球。Python 有如此多我们可以使用的库和软件包。现在更是多到难以置信。一两年后，这个数字大概只有 10 万。而现在我们已经有了 17 万个包或项目。

00:15:10 - Saron Yitbarek:

哇。

00:15:10 - Michael Kennedy:

你可以仅仅写几行代码，然后说，噢，我要做机器学习。有人在会议上向我们展示了一个例子，她们训练一个机器学习系统，给它提供大量人脸的图像，并告诉系统他们拥有什么类型的眼睛。是圆眼睛？或者椭圆形眼睛之类的东西。似乎这会帮助你决定妆容之类的。

00:15:30 - Saron Yitbarek:

噢，哇。

00:15:31 - Michael Kennedy:

这位女士的演讲十分精彩，然后她说，这是训练模型的代码，让大家提问。而代码从头到尾只有十五行。

00:15:40 - Saron Yitbarek:

哇。

00:15:40 - Michael Kennedy:

接着你看，就像她说的那样，你提供一张照片，模型就会知道你的眼睛像什么。

00:15:44 - Saron Yitbarek:

噢，天哪。

00:15:45 - Michael Kennedy:

这种类型的项目势头十足。这些简洁，却又极具能量的东西，你可以直接通过一些包来导入，这真是厉害极了。

00:15:53 - Saron Yitbarek:

哇，这好酷。

00:15:54 - Michael Kennedy:

是不是很疯狂。

00:15:56 - Saron Yitbarek:

好吧。让我们暂停一下对话。我们会在稍后听到更多 Michael 的观点，但我现在想要回头来强调一些事情。使 Python 这些令人惊叹的特质成为可能的源头，Python 社区。Python 成功的一个决定性因素就是这个庞大的、反应迅速的社区。

00:16:21:

此时此刻，正如我们在 van Rossum 离开时看到的那样，社区的规模已经庞大到让人难以接受。想象一下你不得不背着整个语言包袱的样子。某种角度来说，吸引了如此庞大的社区，让保有一个终生独裁者的主意变得站不住脚。Van Rossum 未曾设想他的语言会收到如此巨大的反响，但是，几乎完全自发的，社区成员将 Python 的邮箱列表、新闻组、网站，以及最终通过 PEP Python Enhancement Proposals 讨论语言变化的流程汇集在一起。PEP 是 Python 增强建议 的缩写。

00:17:05:

所以，尽管有着独裁者的名号，van Rossum 仍致力于搭建一个用户可以真正传达意见，参与构建的语言。我敢打赌，尽管他在离开的那一刻感到沮丧，但 van Rossum 明白，一个活跃的社区给他的语言带去的好处，要远胜他离去而折损的。

00:17:25 - Diane Mueller:

我叫 Diane Mueller。

00:17:27 - Saron Yitbarek:

Diane 是红帽云平台的社区开发总监。在过去的 30 年里，她目睹了开源社区的强劲发展，尤其是 Python 社区，给她留下了极为深刻的印象。

00:17:42 - Diane Mueller:

Python 社区做的太棒了.....他们带来了大量行为准则的概念，关于会议、多样性奖金，所有这类东西。通过引入不同的声音以及不同的观

点，我们得到了一个更好、更具创新性的项目，它必定能够留存得更久，并有望为更多的人带去更好的体验。

00:18:03:

即便他们犯错了，也会开诚布公的进行处理。在看到这种精神弥漫进硅谷和初创公司的兄弟文化中之后，通过与社区的合作，Python 就像带我回到了我创业的地方，围绕着它的社区也像是回到了往日。它真的相当鼓舞人心，非常惊艳。

00:18:33 - Saron Yitbarek:

能够鼓舞如此多的人，缘由是 Python 在最初就重新定义了成为社区一员的含义。我讲过 Guido van Rossum 即使是在引退之际，仍倡导了在社区里对女性的拥护。此外他也在更宽泛的意义上帮助了社区的拓展。

00:18:50 - Diane Mueller:

个人为集体带来的远远不止是代码贡献。绝大多数社区管理者或是项目主管，都把精力集中在如何促进大家为他们的项目做出贡献。而在 Python 社区里，大家会鼓励你参与文档工作，或是帮助会议的举办，以及对多样性的促进。这里有许多不同类型的事让你参与，都可以使你成为 Python 社区的一份子。

00:19:19:

所以这个想法，即社区贡献不仅仅局限于代码，像参与活动、相互学习和教育，以及大量的文档工作，这些对大多数人而言都是融入 Python 社区的方法。

00:19:35 - Saron Yitbarek:

当然，我们还有许多路要走。

00:19:37 - Diane Mueller:

精英阶层仍然注重于技术。没人会怀疑这个。但我想你也看见社区管理和社区管理者的理念了.....我们都是社区重要的一份子，而不是被

雇佣来干事的。

00:19:55:

对 Diane 而言，van Rossum 正式放弃他独裁者角色的决定，是全局变化的一部分。这代表着编程语言的建设开始摆脱陈旧与单一。

00:20:07:

我想我们可能已经摆脱了这种模式，尽管在一段时间里，我经常听见人说：“是的，我终生都是这个项目的仁慈独裁者。”但我并不认同。

00:20:21:

Diane Mueller 是红帽社区发展总监。

00:20:28:

在 Guido van Rossum 发出那份令人瞠目结舌的《移交权利》之后，Python 社区自身便成为了权力中心。随着项目的成熟发展，出现新的治理模式是正常的，并且正如我们所见，这些人已经做好准备，要来管理自己的语言了。但我仍然好奇，整件事最终是如何收场的？van Rossum 退场之后究竟发生了什么。让我们回到与 Michael Kennedy 的对话中来寻找一些答案吧。

00:20:59:

.....他离开了 Python，社区在没有他的情况下过得怎么样呢？

00:21:05 - Michael Kennedy:

嗯，社区的状态其实还不错，但我们陷在一个.....制高点，有点像是卡住了。运行时和语言基本上陷停滞不前。有一些有趣的提案，有些比较麻烦，有些又挺简单。比如，嘿，要不然我们每一年发布一次 Python，而不是每 18 个月，这样跟年度会议绑在一起，语言变化会变得更具有可预测性。诸如此类的事情。

00:21:33 - Saron Yitbarek:

噢，嗯。

00:21:33 - Michael Kennedy:

这些都是无法做出定夺的。因为在他引退之后，还没有方法去做决策。他基本上只留下一句，我去度假了，全靠你们了，你们必须得弄清楚怎么才能继续运行下去。我不会告诉你们怎样去做决策或怎么去运营。麻烦现在是你们的了。

00:21:48 - Saron Yitbarek:

哇。这听起来非常有戏剧性，但仔细一想。还记得那些

Python Enhancement Proposals

Python 增强建议（PEP）吗，那些用于接受社区反馈的 PEP？它们可以拯救大伙。

00:22:02 - Michael Kennedy:

有一系列他们试图为 Python 社区确立的新治理模型。好吧，最大的新闻是他们最终决定选择其中之一，叫做指导委员会，委员会由五人组成——我相信他们都有着平等的投票权——他们最近选举出了五个人。。所以，这不是一个人的责任，而是所有的人的责任。

Steering Council

00:22:23:

我感到十分高兴的一件事是， Guido van Rossum 也是他们的一员。他引退后说，我不能作为.....所有这些人们想要改变和反馈的压力的唯一的发泄口。但是他还在，他没有完全抛弃这门语言。他仍然是一个核心开发者，并且加入了指导委员会。因此他仍然保留了发言权，但无需再独自承受一切，这真是非常棒。

00:22:47 - Saron Yitbarek:

我很好奇，这一切在现实世界里是怎么奏效的？我感觉如果我是指导委员会的一员，和语言的创造者坐在一起，我可能会倾向于赞同他所说的任何意见。

00:22:58 - Michael Kennedy:

对，确实如此。在平局的情况下，最终取决于 Guido。

00:23:03 - Saron Yitbarek:

嗯，的确。

00:23:03 - Michael Kennedy:

我不确定。但我认识指导委员会的一些人，他们一直以来都是社区的贡献者和开发者，可能超过了.....在代码水平上，比 Guido 还多十五年。他们也都是非常深入的参与者，并且相当有主见，所以.....

00:23:23 - Saron Yitbarek:

.....而且相当投入。

00:23:25 - Michael Kennedy:

对，投入巨大。所以我认为这是行之有效的。另外，我觉得 Guido 的态度是，我仍然想参与其中，但不愿把自己的意志强加于人，否则一切都和原先没什么两样.....我认为他可能倾向于选择一个更轻松的立场。

00:23:43 - Saron Yitbarek:

好的。我想知道，你是否认为在语言的初创阶段，为了项目能够启动运行，以便语言可以变得激进，从而得到那些突破性的好处，拥有一个 benevolent dictator for life 终生仁慈独裁者（BDFL）的管理模型是必要的？

00:24:00 - Michael Kennedy:

我认同。我有考虑过，大多数由委员会进行的设计，并不惊艳。在早期，有太多决策需要进行，语言如何工作？要不要用分号？怎么做这，怎么做那？所有这些内容都很难由委员会来决定，但是 Python 至今已经有 25 年的历史了。它有这么多人参与其中。我认为现在，它执行的是一个非常好的模型。

00:24:29 - Michael Kennedy:

他们还有过辩论，是否应该换上一个替补的 BDFL，这次我们又要选出谁来做我们的领袖呢？好在他们最后决定反对这个提议。

00:24:37 - Saron Yitbarek:

好。如果 BDFL 的位置是极其重要的，我想知道，社区需要依赖他多久呢？听上去 Guido 是自行做出决定：嘿，太过分了，这不是一个可以延续的状态，我不干了。但是，假如他没有做出这样的决策，我想知道，是否有一个最佳的时机让这个人退出，让大家走向一个更民主的状态呢？

00:25:01 - Michael Kennedy:

嗯，一定会有的，对吧？我认为这个时机应该存在。一个人难以与社区、技术的脉搏以及新的趋势一直保持紧密联系，说个数，大概 40 年。这是件极其困难的事，因此一定要有这个转换。我不能确切的说究竟在什么时候，但我认为必须得等到其他人相比 BDFL 做出了更多的贡献。随着核心贡献者和开发者越来越多，然后你就，嗯，我在度假，看看这些新的事情发生了，它还能活下来。类似这样的事情。

00:25:39 - Saron Yitbarek:

嗯。就好像是社区在准备好后会自己告诉你。

00:25:42 - Michael Kennedy:

对，正是如此。

00:25:48 - Saron Yitbarek:

由于 Python 社区仍在自己的生命历程中，因此这里就是我们暂时告一段落的地方。Michael Kennedy 的两个播客会持续追踪 Python 之后的历程。欢迎订阅 Talk Python to Me 和 Python Bytes。

00:26:07 - Saron Yitbarek:

你听说过被称为古代雅典立法者的梭伦的故事吗？^{Solon} 他是个很酷的家伙。^{Athenian democracy} 在梭伦为民主雅典建立宪法之后，他选择了自行流放。因为他

清楚，继续执政只会增加他成为暴君的风险。我觉得 Guido van Rossum 就像是当代梭伦，为我们提供了数十年的标准实践，有点像是一部宪法。他建立起一个出色的编程语言，一个真正由开源社区自己创作的语言。然后他给予他们一个权力转移的时刻，他在那时告诉他们，你们由自己掌控，我不再是你们的独裁者了。

00:26:54 - Saron Yitbarek:

他确保了一定是由社区，而非他本人，来推动 Python 前行。某种意义上，Guido van Rossum 的“移交权利”是开源世界中所有编程语言的共同宣言，因为任何语言随着其社区的发展，终将面临唯有社区才可以解决的挑战。

00:27:19 - Saron Yitbarek:

在《代码英雄》的第三季中，我们会对编程语言的世界进行深入的挖掘。语言影响力的来源，正是它们如何通过强力的新方法去解决新的问题。在本季的剩余时间里，我们会探索 JavaScript、Perl、COBOL、Go，以及更多语言所具备的超能力。在下一集，我们会学习 BASIC 的故事，此外还会谈论到母语究竟教会了我们什么。

00:27:47 - Saron Yitbarek:

如果你想更深入地研究 Python 或你在本集里听到的任何内容，请转至 redhat.com/commandlineheroes。最后，我是 Saron Yitbarek。直到下期，请坚持编程。

什么是 LCTT SIG 和 LCTT LCRH SIG

Special Interest Group
LCTT SIG 是 LCTT 特别兴趣小组，LCTT SIG 是针对特定领域、特定内容的翻译小组，翻译组成员将遵循 LCTT 流程和规范，参与翻译，并获得相应的奖励。LCRH SIG 是 LCTT 联合红帽（Red Hat）发起的 SIG，当前专注任务是《代码英雄》系列播客的脚本汉化，已有数十位贡献者加入。敬请每周三、周五期待经过我们精心翻译、校对和发布的译文。

欢迎[加入 LCRH SIG](#)一同参与贡献，并领取红帽（Red Hat）和我们联合颁发的专属贡献者证书。

via: <https://www.redhat.com/en/command-line-heroes/season-3/pythons-tale>

作者: [Red Hat](#) 选题: [bestony](#) 译者: [caiichenr](#) 校对: [acyanbird](#), [wxy](#)

本文由 [LCRH](#) 原创编译, [Linux中国](#) 荣誉推出

Guido van Rossum, 他是世界上最伟大的编程语言之一, Python 的创造者, 或许说是最伟大的那个也不为过。他面向 Python 社区的全体成员写下这条消息, 我们不难想象出 Python 的粉丝们阅读这条消息时的画面。

****00:00:38 - Multiple voice actors_**

我没想到在我已经如此努力的情况下, 仍有这么多人对我的决策不满。我情愿给自己放永远的长假, 并完全从决策的流程当中退出, 以免我毕生都作为仁慈的独裁者。你们好自为之, 我不会指定继任者。那么你们接下来要如何做呢, 创立民主制度吗? 或者保持无政府的状态?

00:01:00 - Saron Yitbarek:

在这条简短却惊人的消息发布之后, Guido van Rossum, 这个被 Python 社区追随足有数十年的人...算是基本退场了。这条消息被命名为“权力交接”, 它必定会对 Python 带来永久的改变。

00:01:19 - Saron Yitbarek:

不过除此之外, 它又引出一个问题, 那就是我们所有的这些编程语言在未来要怎样衍变与壮大呢? 是否应该由一个仁慈的独裁者把控, 来为语言塑形? 又或者, 在开源世界里, 编程语言的发展与变化应该更像口语, 取决于所有语言使用者的行为? Python 社区, 这个成长速度远超其他语言的社区, 需要找到这个问题的答案。

00:01:56 - Saron Yitbarek:

我是 Saron Yitbarek，这里是代码英雄第三季，一个来自 Red Hat 的原创播客。在上一季的代码英雄里，我们探索了广袤天地，从游戏主题出发，到失败的艺术，再到无服务器开发，最后我们甚至追踪了一台在火星表面，来自 NASA 的漫步者。但是真正抓住每个人想象力的一集，似乎是 Grace Hopper 的故事。她参与开发的编译器创造出了世上第一个高级编程语言，COBOL。我们随后意识到，Grace Hopper 的故事不过是万千个塑造了软件开发与运维的世界的故事之一。新的编程语言使得我们可以连通人类与机器，它们为我们打开了一扇通往无限可能的大门。

00:02:51 - Saron Yitbarek:

因此，第三季将全部围绕这些语言展开。我们会谈论 JavaScript、BASIC、Go 以及 Perl，当然，还有 Python。我们的旅程将从 Python 开始，顺着 Python 的故事，我们会学到一个与所有编程语言未来相关的至关重要的事实。

00:03:16 - Saron Yitbarek:

在 Python 仁慈的独裁者放弃王座之后，整个 Python 社区...呃，有些迷失。你要如何在独裁者退位之后组织工作呢？有人提议采用
Presbyterian church
长老会教堂结构，不过没能奏效。要弄清楚 Python 重新组织的过
程，并了解这对其它语言的未来有什么意义，我们必须回到 Python
故事的源头。

**00:03:46 - Guido van Rossum_

好吧，我用 C 来完成全部代码，这让事情有些枯燥。

00:03:51 - Saron Yitbarek:

本尊驾到，Guido van Rossum，Python 仁慈的独裁者。Van Rossum 先前数年都在为阿姆斯特丹著名的 Centrum Wiskunde & Informatica 工作，在那里他协助 ABC 编程语言的开发。现在，他将描述他使用 C 工作时，发现对一个全新编程语言产生需求的时刻。

**00:04:13 - Guido van Rossum_

仍然感觉它有着大量的 bug，并且发展缓慢。我在想，呃，如果我们现在实现了 ABC，我可以在十五分钟内写出整个注册程序，然后我可以接着开发账户管理之类的功能，但是用 C 进行开发，却要花掉我一整个星期。我开始设想，要提出一种可以在 Amoeba 环境中使用 ABC 语言特性的方法。

00:04:47 - Saron Yitbarek:

在研究编程语言的历史时我们发现，没有什么东西是全新的。它们全都是为了拼凑出解决方案而从旧语言借来的。语言会不断的变化、发展、开枝散叶。当 van Rossum 被种种可能性弄得迷茫时，他设想出一个可以弥合 C 与 Shell 编程两者间隙的编程语言。一些场景下使用 C 语言往往是牛刀杀鸡，与此同时，用 Shell 脚本又常常让人感到繁琐，二者之间的甜点区，正是 Python 的栖身之处。当 van Rossum 在 1991 年首次发布 Python 时，它给大家都带来了启发，尤其是对系统管理员而言。这是一种前无来者，功能全面的脚本语言。

00:05:35 - Emily Morehouse:

首次使用 Python 的那刻，我立即就爱上了它。

00:05:39 - Saron Yitbarek:

这是 Emily Morehouse，目前 Python 五位女性核心开发者之一。

00:05:44 - Emily Morehouse:

我认为当你从接触的首个类 C++ 语言跨越到 Python 时，发现二者之间如此显著的差异，会自然而然地意识到语言与其设计的优美之处。你不必去应付内存管理毛糙的底层实现细节，它可以既快又好的构建一些项目，更不必说它还有着如此广泛的应用领域。

00:06:18 - Saron Yitbarek:

Python 吸引力的关键之处就在于它的可扩展性。举个例子，像 ABC 这样的语言，设计坚若磐石，就无法形成一个能够真正帮助改善语言的社区。反观 Python，van Rossum 在设计之初就希望 Python 是开放，并易于扩展的。

00:06:37 - Emily Morehouse:

进行软件设计时，你往往需要用到一些现有的软件或系统，并且要让它们全都能够协同工作，其实软件设计的核心价值之一就在于确保它的可扩展性。

00:06:58 - Saron Yitbarek:

听起来不怎么费脑子，但并非每一个语言都像 Python 这样，有着与生俱来的强可扩展性。事实上，一门不具备可扩展性的语言，大概率会随着自身的发展而引发崩溃。

00:07:16 - Emily Morehouse:

Python 有一个非常有趣的设计，它允许在其内核上进行扩展。你可以在运行时上对系统的不同片段进行拼接，假如你想要切换如何导入模块的方式，或者你想切换你的字符串类型或整数类型，Python 会允许你用相当容易的方式去实现这些事。

00:07:44 - Emily Morehouse:

Python 可扩展性的核心是所谓的 C 扩展，或者说是 C 模块。因此，Python 实际上被设计出来的意图，是为你提供对其它语言的切入点。本质上来讲，如果你可以写出 C 扩展或 C 模块，可以用它去桥接其它数百种语言，那么你在某种角度上算是破解了 Python。

00:08:16 - Saron Yitbarek:

这完全取决于用户根据自身需求调整语言的能力。所以说 Python，按照 Guido van Rossum 的设想，绝不会局限于独裁者的个人视野。他的“权力交接”停在备忘录上有相当长的一段时间。Van Rossum 明白社区影响力的力量，这是一种能够使所有人都聚集于同一顶“大帐篷”之下的力量。没错，他最终得名独裁者，但却是一名仁慈的独裁者。

00:08:44 - Emily Morehouse:

我认为 Python 社区变得如此多元化，原因之一就是 Guido。Python 现在能够拥有女性核心开发者，正是因为 Guido 希望此事发生，并一

手推动其实现。

00:09:01 - Saron Yitbarek:

Naomi Ceder, Python 软件基金会的主席，她曾在一次演讲中说：“Python。因语言来，为社区留。”这可能正是 Guido van Rossum 留下的最棒的礼物。不仅仅是为 Python 语言，他更是为整个 Python 社区都留下了空间。他不仅在语法意义上为 Python 提供了可扩展性，某种角度来讲，他也在社会意义上提供了扩展性。永远都为新来者留有位置。

00:09:29 - Emily Morehouse:

Python 有着大量不同的应用，而社区又如此多元化。这实在是，充分扩展了社区的边界。

00:09:51 - Saron Yitbarek:

Emily Morehouse 是 Python 核心开发者与 Cuttlesoft 的工程总监。

00:09:59 - Saron Yitbarek:

Python 降世之后，成长便空前绝后。我看到一个 Stack Overflow 上的图表，统计了每门语言被提及的次数，对应着 Python 的那条线仿佛坐上了火箭。2018 年，在 Google 搜索 Python 的人数超过了搜索 Kim Kardashian 的。另一个令人振奋的消息是，Python 开始和 Java, C 以及 C++ 之类的语言争夺最常用语言的头衔。

00:10:26 - Saron Yitbarek:

那么，这些偏爱究竟来源何处呢？为了找到答案，我找到位于 Python zeitgeist 中心的开发者 Michael Kennedy。他主持着两个专注于 Python 的播客：Talk Python to Me 和 Python Bytes。我们会在展示内容里放上对应的链接以便大家查看。Michael 将要和我一起讨论 Python 是如何真正迈出大步的。

00:10:52 - Michael Kennedy:

如果你看过分析报告或总结报告一类的东西，会发现 2012 年是一个重要拐点，而发生在 2012 年左右最具意义的事情莫过于，数据分析社区换掉了 R 之类的东西，开始真正专注于 Python。这件事发生以后，Python 社区就有了更充足的劲头，也有了更多的机器学习库。像许多流行的机器学习库，都会首先考虑使用 Python 来实现，此后才会考虑其他语言。

00:11:22 - Saron Yitbarek:

嗯，我也这么认为。当我想到 Python 时，我知道它可以被用作 web 开发，我也认识许多仍然在使用 Python 来写 web 应用的人。但我感到 Python 如今真正的核心更多是在数据分析领域。你认为是什么导致了这件事情发生？为什么数据分析社区抛弃了那些，或者不应该叫抛弃，替换掉了 R 之类的工具。

00:11:44 - Michael Kennedy:

对，正是如此。

00:11:45 - Saron Yitbarek:

从何而来呢？

00:11:46 - Michael Kennedy:

我认为这种迁移的发生，是有两件事在起作用。其中之一就是 Python 作为一种“真正的”编程语言，它既可以写一些简单的东西，例如构建图形或数据分析工具等等，但是，它也可以用来构建 Instagram 和 YouTube 以及其他类似的复杂应用...

00:12:08 - Saron Yitbarek:

是的，显而易见。

00:12:09 - Michael Kennedy:

像 R...之类的东西。对，相当的显而易见，它们都是由 Python 来编写的。所以他们之前在使用一些其它的编程语言时，例如 R，这门专精数据科学工作的，科学统计类的编程语言，如果他们想要写一个 web

应用来展示他们的结果，那么，他们要用什么来实现呢？Node 或者 Python 还是...这导致他们没办法坚持使用那些语言。

00:12:31 - Saron Yitbarek:

对，说的很好。

00:12:31 - Michael Kennedy:

所以说 Python 拥有相当棒的能力...基本上，它是一种真正的编程语言，这是其一。第二点是，Python 十分独特，被我称为...全光谱语言。我所谓的全光谱的意思是指，假如我是一名生物学家、天文学家或别的什么，当我想要处理一些数据，想要加载一个 CSV 文件并运行一些指令得到图像时，我无需理解什么是类、静态方法、主函数、编译或者链接。我不必在刚开始就学会这些...其它语言必须要做的一系列事。我只用写几行代码，键入指令，程序就可以运行。

00:13:14 - Michael Kennedy:

但是，你也可以用它去搭建像 Instagram 之类的软件。它可以成长为你所使用的绝对专业的系统，但是你不会在一开始就被逼迫着，去理解大型应用的一切深层抽象，你可以根据自身需要来使用它。这样说的通吗？

00:13:30 - Saron Yitbarek:

对，对。说的非常有道理。刚才我们谈论到 2012 年的拐点时，我搜索了一些关于 Python 的资料，Python 的确是世界上被 Google 次数最多的编程语言。

00:13:42 - Michael Kennedy:

哇。

00:13:43 - Saron Yitbarek:

你感觉它现在确实被大家青睐，并且仍在成长之中吗？

00:13:47 - Michael Kennedy:

我认为它真的被青睐并处于成长之中。在我们谈论到的后几年里，有相当多的企业集团都选择使用 Python。而他们曾经的选择是 .NET，Java，甚至是 C。这就是回答，并且现在 Python 开始走出属于自己的路，我认为它在某些场景下被格外倚重，关于这个，我指的是那些数据科学家们。就像，显然我们用到的 Jupyter Lab 还有那些炫酷笔记本之类的东西。全都是 Python。

00:14:17 - Michael Kennedy:

数据科学没有祖传代码的负担。如果我想开发一个新的项目，用来探索广告活动或科学结果，就不会对一堆陈旧的东西，保持着庞大的依赖关系。模型和数据都会过时，因此对于数据科学世界而言，切换技术栈或保持最新状态要容易得多。

00:14:37 - Saron Yitbarek:

说的不错。

00:14:39 - Michael Kennedy:

嗯，谢谢。

00:14:42 - Saron Yitbarek:

嗯，听上去 Python 不会在近期停止成长，而是保持增长的趋势，甚至这种增长的势头仍在向前发展。你认为对这种增长与前进的势头影响最大的事情是什么呢？

00:14:52 - Michael Kennedy:

我感觉这就像滚雪球。Python 有如此多我们可以使用的库和软件包。现在更是多到难以置信。一两年后，这个数字大概只有 10 万。而现在我们已经有了 17 万包或项目。

00:15:10 - Saron Yitbarek:

哇。

00:15:10 - Michael Kennedy:

你可以仅仅写几行代码，然后说，噢，我要做机器学习。有人在会议上向我们展示了一个例子，她们训练一个机器学习系统，给它提供大量人脸的图像，并告诉系统他们拥有什么类型的眼睛。是圆眼睛？或者椭圆形眼睛，之类的东西。似乎这会由你的妆容或别的什么来驱动。

00:15:30 - Saron Yitbarek:

噢，哇。

00:15:31 - Michael Kennedy:

这位女士的演讲十分精彩，然后她说，这是训练模型的代码，让大家提问。而代码从头到尾只有十五行。

00:15:40 - Saron Yitbarek:

哇。

00:15:40 - Michael Kennedy:

接着你看，就像她说的那样，你提供一张照片，模型就会知道你的眼睛像什么。

00:15:44 - Saron Yitbarek:

噢，天哪。

00:15:45 - Michael Kennedy:

这种类型的项目势头十足。这些简洁，却又极具能量的东西，你可以直接通过一些包来导入，这真是厉害极了。

00:15:53 - Saron Yitbarek:

哇，这好酷。

00:15:54 - Michael Kennedy:

这难道不疯狂吗？

00:15:56 - Saron Yitbarek:

好吧。让我们暂停一下对话。我们会在稍后听到更多 Michael 的观点，但我现在想要回头来强调一些事情。使 Python 这些令人惊叹的特质成为可能的源头，Python 社区。Python 成功的一个决定性因素就是这个庞大的，有影响力的社区。

00:16:21 - Saron Yitbarek:

此时此刻，正如我们在 van Rossum 离开时看到的那样，社区的规模已经过于庞大。想象一下你不得不背着整个语言包袱的样子。某种角度来说，吸引了如此庞大的社区，让保有一个终生独裁者的主意变得站不住脚。Van Rossum 未曾设想他的语言会收到如此巨大的回应，但是，几乎完全自发的，社区成员将 Python 的邮箱列表、新闻组、网站，最后还有通过 Python Enhancement Proposals PEPs 讨论语言变化的流程汇集在一起。这些内容共同组成了 Python 增强建议。

00:17:05 - Saron Yitbarek:

所以，尽管有着独裁者的名号，van Rossum 仍致力于搭建一个用户可以真正传达意见，参与构建的语言。我敢打赌，尽管他在离开的那一刻感到沮丧，但 van Rossum 明白，一个活跃的社区给他的语言带去的好处要远胜他离去而折损的。

00:17:25 - Diane Mueller:

我叫做 Diane Mueller。

00:17:27 - Saron Yitbarek:

Diane 是 Red Hat 云平台的社区开发总监。在过去的 30 年里，她目睹了开源社区的强劲发展，尤其是 Python 社区，给她留下了极为深刻的印象。

00:17:42 - Diane Mueller:

Python 社区做的太棒了... 他们带来了大量行为准则的概念，关于会议、不同奖金、所有这类东西。通过不同的声音以及不同的观点，我们得到了一个更好，更具创新性的项目，它必定能够留存得更久，并有望为更多的人带去更好的体验。

00:18:03 - Diane Mueller:

即便他们犯错了，也会开诚布公的进行处理。在看到这种精神弥漫进硅谷和初创公司的兄弟文化中之后，通过与社区的合作，Python 就像带我回到了我刚起步的地方，围绕着它的社区也像是回到了往日。它真的相当鼓舞人心，非常惊艳。

00:18:33 - Saron Yitbarek:

能够鼓舞如此多的人，缘由是 Python 在最初就重新定义了成为社区一部分的含义。我讲过 Guido van Rossum 即使是在引退之际，仍倡导了在社区里对女性的拥护。此外他也在更宽泛的意义上帮助了社区的拓展。

00:18:50 - Diane Mueller:

个人为集体带来的远远不止是代码贡献。绝大多数社区管理者或是项目主管，都把精力集中在如何促进大家为他们的项目做出贡献。而在 Python 社区里，大家会鼓励你参与文档工作，或是帮助会议的举办，以及对多元性的促进。这里有许多不同类型的事让你参与，都可以使你成为 Python 社区的一份子。

00:19:19 - Diane Mueller:

所以这个想法，即社区贡献不仅仅局限于代码，像参与活动、相互学习和教育，以及大量的文档工作，这些对大多数人而言都是融入 Python 社区的方法。

00:19:35 - Saron Yitbarek:

当然，我们还有许多路要走。

00:19:37 - Diane Mueller:

精英阶层仍然着重于技术。没人会怀疑这个。但我想你也看见社区管理和社区管理者的理念了...我们都是社区重要的一份子，而不是被雇佣来干事的。

00:19:55 - Saron Yitbarek:

对 Diane 而言，van Rossum 正式放弃他独裁者角色的决定，是全球运转的一部分。这代表着编程语言的构建过程开始摆脱陈旧与单一。

00:20:07 - Saron Yitbarek:

我认为我们会从这个模式中前进，尽管在一段时间里，我经常听见人说：“是的，我终生都是这个项目的仁慈独裁者。”但我并不认同。

00:20:21 - Saron Yitbarek:

Diane Mueller 是 Red Hat 社区发展总监。

00:20:28 - Saron Yitbarek:

在 Guido van Rossum 发出那条令人瞠目结舌的消息之后，Python 社区自身便成为了权力中心。随着项目的成熟发展，出现新的管理模式是正常的，并且正如我们所见，这些家伙已经做好准备，要来管理自己的语言了。但我仍然好奇，整件事最终是如何收场的？van Rossum 退场之后究竟发生了什么。让我们回到与 Michael Kennedy 的对话中来寻找一些答案吧。

00:20:59 - Saron Yitbarek:

... 离开了 Python，社区在没有他的情况下过得怎么样呢？

00:21:05 - Michael Kennedy:

嗯，社区的状态其实还不错，但我们陷在一个...制高点，有点像是卡住了。运行时和语言基本上陷停滞不前。有一些有趣的提案，有些比较麻烦，有些又挺简单。比如，嘿，要不然我们每一年发布一次 Python，而不是每 18 个月，这样跟年度会议绑在一起，语言变化会变得更具有可预测性。诸如此类的事情。

00:21:33 - Saron Yitbarek:

噢，嗯。

00:21:33 - Michael Kennedy:

这些都是无法做出定夺的。因为在他引退之后，还没有方法去做决策。他基本上只留下句，我去度假了，全靠你们了，你们必须得弄清楚怎么才能继续运行下去。我不会告诉你们怎样去做决策或怎么去运营。麻烦现在是你们的了。

00:21:48 - Saron Yitbarek:

哇。这听起来非常有戏剧性，但仔细一想。还记得那些

Python Enhancement Proposals

Python 增强建议吗，那些用于接受社区反馈的 PEPs？它们可以拯救大伙。

00:22:02 - Michael Kennedy:

这里有一系列他们试图为 Python 社区确立的新治理模型。大新闻是

他们最终决定选择其中之一，叫做指导委员会，委员会由五人组成——我相信他们都有着平等的投票权——他们最近选举出了五个人。所以，相比于站在一个人的肩膀上，现在取而代之的方法是站在所有人的肩膀上。

00:22:23 - Michael Kennedy:

我感到十分高兴的一件事是，Guido van Rossum 也是他们的一员。他引退后说，我不能作为唯一的发泄口...所有这些人们想要改变和反馈的压力。但是他还在，他没有完全抛弃这门语言。他仍然是一个核心开发者，并且加入了指导委员会。因此他仍然保留了发言权，但无需再独自承受一切，这真是非常棒。

00:22:47 - Saron Yitbarek:

我很好奇，这一切在现实世界里是怎么奏效的？我感觉如果我是指导委员会的一员，和语言的创造者坐在一起，我可能会倾向于赞同他所说的任何意见。

00:22:58 - Michael Kennedy:

对，确实如此。所有事情制衡的情况下，最终取决于 Guido。

00:23:03 - Saron Yitbarek:

嗯，的确。

00:23:03 - Michael Kennedy:

我不确定。但我认识指导委员会的一些人，他们一直以来都是社区的贡献者和开发者，可能超过了...代码层面上，超过了 Guido 十五年。他们也都是非常深入的参与者，并且相当自以为是，所以...

00:23:23 - Saron Yitbarek:

而且相当投入。

00:23:25 - Michael Kennedy:

对，投入巨大。所以我认为这是行之有效的。另外，我觉得 Guido 的态度是，我仍然想参与其中，但不愿把自己的意志强加于人，否则一切都和原先没什么两样...我认为他可能倾向于选择一个更轻松的位置。

00:23:43 - Saron Yitbarek:

好的。我想知道，你是否认为在语言的初创阶段，为了项目能够启动运行，以便语言可以变得激进，从而得到那些突破性的好处，拥有一个 benevolent dictator for life 终生仁慈独裁者（BDFL）的管理模型是必要的？

00:24:00 - Michael Kennedy:

我认同。我有考虑过，大多数由委员会进行的设计，并不惊艳。在早期，有太多决策需要进行，语言如何工作？要不要用分号？怎么做这，怎么做那？所有这些内容都很难由委员会来决定，但是 Python 至今已经有 25 年的历史了。它将太多人卷入其中，我认为现在，它执行的是一个非常好的模型。

00:24:29 - Michael Kennedy:

他们还有过辩论，是否应该换上一个替补的 BDFL，这次我们又要选出谁来做我们的领袖呢？好在他们最后决定反对这个提议。

00:24:37 - Saron Yitbarek:

好。如果 BDFL 的位置是极其重要的，我想知道，社区需要依赖他多久呢？听上去 Guido 是自行做出决定：嘿，太过分了，这不是一个可以延续的状态，我不要再这样做了。但是，假如他没有做出这样的决策，我想知道，是否有一个最佳的时机让这个人退出，让大家走向一个更民主的状态呢？

00:25:01 - Michael Kennedy:

嗯，一定会有的，对吧？我认为这个时机应该存在。一个人难以与社区、技术的脉搏以及新的趋势一直保持紧密联系，说个数，大概 40 年。这是件极其困难的事，因此务必要进行这个转换。我不能确切的说究竟在什么时候，但我认为必须得等到其他人相比 BDFL 做出了更多的贡献。随着核心贡献者和开发者越来越多，然后你就，嗯，我在度假，看着发生的新鲜变化，它们都幸存了下来。这种效果。

00:25:39 - Saron Yitbarek:

嗯。就好像是社区在准备好后会自己告诉你。

00:25:42 - Michael Kennedy:

对，正是如此。

00:25:48 - Saron Yitbarek:

由于 Python 社区仍在自己的生命历程中，因此这里就是我们暂时告一段落的地方。Michael Kennedy 的两个播客会持续追踪 Python 之后的历程。欢迎订阅 Talk Python to Me 和 Python Bytes。

00:26:07 - Saron Yitbarek:

你听说过被称为古代雅典立法者的梭伦的故事吗？他是个很酷的家伙。^{Athenian democracy}在梭伦为民主雅典建立宪法之后，他选择了自行流放。因为他清楚，继续执政只会增加他成为暴君的风险。我觉得 Guido van Rossum 就像是当代梭伦，为我们提供了数十年的标准实践，有点像是一部宪法。他建立起一个出色的编程语言，一个真正由开源社区自己创作的语言。然后他给予他们一个权力转移的时刻，他在那时告诉他们，你们由自己掌控，我不再是你们的独裁者了。

00:26:54 - Saron Yitbarek:

他确保了一定是由社区，而非他本人，来推动 Python 前行。某种意义上，Guido van Rossum 的“权力转移”是开源世界中所有编程语言的共同宣言，因为任何语言随着其社区的发展，终将面临唯有社区才可以解决的挑战。

00:27:19 - Saron Yitbarek:

在代码英雄的第三季中，我们会对编程语言的世界进行深入的挖掘。语言影响力的来源，正是它们如何通过强力的新方法去解决新的问题。在本季的剩余时间里，我们会探索 JavaScript, Perl, COBOL, Go, 以及更多语言所具备的超能力。在下一集，我们会学习 BASIC 的故事，此外还会谈论到母语究竟教会了我们什么。

00:27:47 - Saron Yitbarek:

如果你想更深入地研究 Python 或你在本集里听到的任何内容，请转至 redhat.com/commandlineheroes。最后，我是 Saron Yitbarek。直到下期，请坚持编程。

什么是 LCTT SIG 和 LCTT LCRH SIG

^{Special Interest Group}
LCTT SIG 是 LCTT 特别兴趣小组，LCTT SIG 是针对特定领域、特定内容的翻译小组，翻译组成员将遵循 LCTT 流程和规范，参与翻译，并获得相应的奖励。LCRH SIG 是 LCTT 联合红帽（Red Hat）发起的 SIG，当前专注任务是《代码英雄》系列播客的脚本汉化，已有数十位贡献者加入。敬请每周三、周五期待经过我们精心翻译、校对和发布的译文。

欢迎[加入 LCRH SIG](#) 一同参与贡献，并领取红帽（Red Hat）和我们联合颁发的专属贡献者证书。

via: <https://www.redhat.com/en/command-line-heroes/season-3/pythons-tale>

作者: [Red Hat](#) 选题: [bestony](#) 译者: [caiichenr](#) 校对: [acyanbird](#), [wxy](#)

本文由 [LCRH](#) 原创编译, [Linux中国](#) 荣誉推出