



最佳做法

使用环境

对我有很大帮助的一点是，我的 Python 2 和 Python 3 具有独立的环境。我使用了 `conda create -n py2 python=2` 和 `conda create -n py3 python=3` 创建两个独立的环境，即 `py2` 和 `py3`。现在，我的每个 Python 版本都有一个通用环境。在所有这些环境中，我都安装了大多数常用的标准数据科学包（numpy、scipy、pandas 等）。

我还发现，为我从事的每个项目创建环境很有用。这对于与数据不相关的项目（例如使用 Flask 开发的 Web 应用）也很有用。例如，我为我的个人博客（使用 [Pelican](#)）创建了一个环境。

共享环境

在 GitHub 上共享代码时，最好同样创建环境文件并将其包括在代码库中。这能让其他人更轻松地安装你的代码的所有依赖项。对于不使用 conda 的用户，我通常还会使用 `pip freeze`（[在此处了解详情](#)）将一个 `requirements.txt` 文件导出并包括在其中。

了解更多信息

要详细了解 conda 以及它如何融入到 Python 生态系统中，请查看这篇由 Jake Vanderplas 撰写的文章：[Conda myths and misconceptions](#)（有关 conda 的迷思和误解）。此外，如果你有多余精力，也可以参考这篇 [conda 文档](#)。

[下一项](#)