
将 optparse 代码迁移至 argparse

发行版本 3.13.7

Guido van Rossum and the Python development team

九月 02, 2025

Python Software Foundation
Email: docs@python.org

Contents

argparse 模块提供了一些未获 optparse 模块原生支持的更高层级特性，包括：

- 处理位置参数。
- 支持子命令。
- 允许替代选项前缀例如 + 和 /。
- 处理零个或多个以及一个或多个风格的参数。
- 生成更具信息量的用法消息。
- 提供用于定制 type 和 action 的更为简单的接口。

最初的时候，argparse 模块试图维持与 optparse 的兼容性。不过，在支持声明式命令行选项处理（并将位置参数处理留给应用程序代码）以及在声明式界面中同时支持名称选项和位置参数之间底层设计方面的差异意味着其 API 逐渐与 optparse 的拉开了差距。

正如 `choosing-an-argument-parser` 中所描述的，目前使用 optparse 并且对其效果感到满意的应用程序完全可以继续使用 optparse。

考虑迁移的应用程序开发者也应当先仔细查看在该章节中描述的内在行为差异列表再决定是否需要迁移。

对于确定选择从 optparse 迁移到 argparse 的应用程序，下列建议应该会有帮助：

- 将所有 `optparse.OptionParser.add_option()` 调用替换为 `ArgumentParser.add_argument()` 调用。
- 将 `(options, args) = parser.parse_args()` 替换为 `args = parser.parse_args()` 并为位置参数添加额外的 `ArgumentParser.add_argument()` 调用。请注意之前所谓的 `options` 在 argparse 上下文中被称为 `args`。
- 通过使用 `parse_intermixed_args()` 而非 `parse_args()` 来替换 `optparse.OptionParser.disable_interspersed_args()`。
- 将回调动作和 `callback_*` 关键字参数替换为 `type` 或 `action` 参数。
- 将 `type` 关键字参数字符串名称替换为相应的类型对象（例如 `int`, `float`, `complex` 等）。
- 将 `optparse.Values` 替换为 `Namespace` 并将 `optparse.OptionError` 和 `optparse.OptionValueError` 替换为 `ArgumentError`。

- 将隐式参数字符串例如使用标准 Python 字典语法的 `%default` 或 `%prog` 替换为格式字符串，即 `%(default)s` 和 `%(prog)s`。
- 将 `OptionParser` 构造器 `version` 参数替换为对 `parser.add_argument('--version', action='version', version='<the version>')` 的调用。