
Migrating optparse code to argparse

Wydanie 3.14.0rc2

Guido van Rossum and the Python development team

września 01, 2025

Python Software Foundation
Email: docs@python.org

Spis treści

The `argparse` module offers several higher level features not natively provided by the `optparse` module, including:

- Обробка позиційних аргументів.
- Supporting subcommands.
- Дозволяє альтернативні префікси параметрів, наприклад `+i /`.
- Обробка нуля або більше та одного або кількох аргументів стилю.
- Створення більш інформативних повідомлень про використання.
- Забезпечення набагато простішого інтерфейсу для користувацького типу дії.

Originally, the `argparse` module attempted to maintain compatibility with `optparse`. However, the fundamental design differences between supporting declarative command line option processing (while leaving positional argument processing to application code), and supporting both named options and positional arguments in the declarative interface mean that the API has diverged from that of `optparse` over time.

As described in `choosing-an-argument-parser`, applications that are currently using `optparse` and are happy with the way it works can just continue to use `optparse`.

Application developers that are considering migrating should also review the list of intrinsic behavioural differences described in that section before deciding whether or not migration is desirable.

For applications that do choose to migrate from `optparse` to `argparse`, the following suggestions should be helpful:

- Замініть усі виклики `optparse.OptionParser.add_option()` на виклики `ArgumentParser.add_argument()`.
- Замініть `(options, args) = parser.parse_args()` на `args = parser.parse_args()` і додайте додаткові виклики `ArgumentParser.add_argument()` для позиційних аргументів. Майте на увазі, що те, що раніше називалося `options`, тепер у контексті `argparse` називається `args`.
- Замініть `optparse.OptionParser.disable_interspersed_args()` на використання `parse_intermixed_args()` замість `parse_args()`.
- Замініть дії зворотного виклику та аргументи ключового слова `callback_*` на аргументи `type` або `action`.
- Замініть назви рядків для ключових аргументів `type` відповідними об'єктами типу (наприклад, `int`, `float`, `complex` тощо).

- Замініть `optparse.Values` на `Namespace` і `optparse.OptionError` і `optparse.OptionValueError` на `ArgumentError`.
- Замініть рядки неявними аргументами, такими як `%default` або `%prog` стандартним синтаксисом Python, щоб використовувати словники для форматування рядків, тобто `%(default)s і %(prog)s`.
- Замініть аргумент `version` конструктора `OptionParser` на виклик `parser.add_argument('--version', action='version', version=' <the version> ')`.