

---

# Migrating optparse code to argparse

Version 3.14.0rc3

Guido van Rossum and the Python development team

octobre 01, 2025

Python Software Foundation  
Email : docs@python.org

## Table des matières

---

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

- gère les arguments positionnels ;
- Supporting subcommands.
- permet d'utiliser les alternatives `+` ou `/` comme préfixes d'option ;
- prend en charge la répétition de valeurs (zéro ou plus, un ou plus) ;
- fournit des messages d'aide plus complets ;
- fournit une interface plus simple pour les types et les actions personnalisés.

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 :

- remplacer tous les appels à `optparse.OptionParser.add_option()` par des appels à `ArgumentParser.add_argument()` ;
- remplacer `(options, args) = parser.parse_args()` par `args = parser.parse_args()` et ajouter des appels à `ArgumentParser.add_argument()` pour les arguments positionnels. Prenez note que les valeurs précédemment appelées `options` sont appelées `args` dans le contexte d'`argparse` ;
- remplacer `optparse.OptionParser.disable_interspersed_args()` en appelant `parse_intermixed_args()` plutôt que `parse_args()` ;
- remplacer les actions de rappel (*callback actions* en anglais) et les arguments nommés `callback_*` par des arguments `type` et `actions` ;
- remplacer les chaînes représentant le nom des types pour l'argument nommé `type` par les objets `types` correspondants (par exemple : `int`, `float`, `complex`, etc) ;
- remplacer `optparse.Values` par `Namespace` ; et `optparse.OptionError` et `optparse.OptionValueError` par `ArgumentError` ;
- remplacer les chaînes avec des arguments de formatage implicite (tels que `%default` ou `%prog`) par la syntaxe standard de Python pour l'interpolation d'un dictionnaire dans les chaînes de formatage (c'est-à-dire `%(default)s` et `%(prog)s`) ;
- remplacer l'argument `version` du constructeur d'`OptionParser` par un appel à `parser.add_argument('--version', action='version', version='<la version>')`.