
Mettre à jour du code optparse

Version 3.12.9

Guido van Rossum and the Python development team

mars 10, 2025

Python Software Foundation
Email : docs@python.org

Table des matières

Originally, the `argparse` module had attempted to maintain compatibility with `optparse`. However, `optparse` was difficult to extend transparently, particularly with the changes required to support `nargs=` specifiers and better usage messages. When most everything in `optparse` had either been copy-pasted over or monkey-patched, it no longer seemed practical to try to maintain the backwards compatibility.

The `argparse` module improves on the `optparse` module in a number of ways 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.

Le portage partiel d'`optparse` à `argparse` :

- 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>')`.