Обзор популярных менеджеров пакетов Python: pip, uv и Poetry

Автор: Мокобиа Джоан Чидиеберe

Группа: 24.Б83_мм

Курс: Программирование на Python

Дата: 18 Октябрь 2025

Введение

При работе с Python разработчики используют *менеджеры пакетов* для установки, обновления и управления внешними библиотеками.
Менеджеры пакетов упрощают распространение программного обеспечения, управляют зависимостями и помогают поддерживать согласованные окружения в разных проектах.

В этом отчёте мы рассмотрим три популярных менеджера пакетов Python:

1. pip — стандартный установщик пакетов для Python.
2. uv — новый, высокопроизводительный менеджер пакетов, созданный для скорости.
3. Poetry — менеджер зависимостей и окружений, предоставляющий полный контроль над проектом.

Мы обсудим их особенности, преимущества и ограничения, а также сравним, как они работают.

1. pip — стандартный менеджер пакетов Python

pip (Pip Installs Packages) — это официальный и самый широко используемый менеджер пакетов Python.
Он подключается к Python Package Index (PyPI) — хранилищу сотен тысяч открытых библиотек с исходным кодом.

Основные особенности

- Устанавливается вместе с Python (начиная с версии 3.4)
- Устанавливает и удаляет пакеты из PyPI

- Работает с форматами .whl, .tar.gz и репозиториями GitHub
- Поддерживает использование virtualenv или venv для изолированных окружений

Основные команды

Задача	Команда
Установить пакет	pip install requests
Показать список установленных пакетов	pip list
Показать информацию о пакете	pip show numpy
Удалить пакет	pip uninstall pandas
Сохранить зависимости в файл	pip freeze > requirements.txt
Установить зависимости из файла	pip install -r requirements.txt

Преимущества

- Устанавливается по умолчанию вместе с Python
- Прост в использовании и достаточно быстрый
- Работает на всех платформах, где есть Python

Ограничения

- Плохо справляется с конфликтами зависимостей
- Требует ручного управления виртуальными окружениями
- Не управляет метаданными проекта

2. uv — современный быстрый менеджер пакетов

uv — это новый высокопроизводительный менеджер пакетов Python, написанный на Rust командой Astral (создатели Ruff).
Он разработан, чтобы быть чрезвычайно быстрым, эффективным и полностью заменять pip.

Основные особенности

- Написан на Rust — значительно быстрее, чем pip
- Автоматически управляет виртуальными окружениями
- Выполняет продвинутое разрешение зависимостей
- Может заменить pip и venv
- Совместим с большинством команд pip

Основные команды

Задача	Команда
Установить пакет	uv pip install requests
Создать новое виртуальное окружение	uv venv
Запустить Python-скрипт в окружении	uv run script.py
Синхронизировать зависимости	uv pip sync requirements.txt

Преимущества

- Очень высокая скорость (до 10× быстрее pip)
- Встроенное управление окружениями и зависимостями
- Использует современное кэширование для повышения эффективности
- Совместим с существующими workflow pip

Ограничения

- Всё ещё относительно новый инструмент
- Пока не поддерживается всеми IDE и серверами

3. Poetry — менеджер зависимостей и проектов

Poetry — это инструмент, который управляет зависимостями, виртуальными окружениями и даже публикацией Python-пакетов.
Он ориентирован на простоту и воспроизводимость, используя один конфигурационный файл: pyproject.toml.

Основные особенности

- Управляет зависимостями и окружениями одновременно
- Использует pyproject.toml для конфигурации
- Автоматически создаёт виртуальные окружения
- Генерирует poetry.lock для воспроизводимых сборок
- Может публиковать пакеты на PyPI

Основные команды

Задача	Команда
Создать новый проект	poetry new myproject
Добавить зависимость	poetry add requests

Задача	Команда
Установить зависимости	poetry install
Запустить скрипт	poetry run python script.py
Собрать пакет	poetry build
Опубликовать на PyPI	poetry publish

Преимущества

- Легко справляется с конфликтами зависимостей
- Автоматически создаёт изолированные окружения
- Отлично подходит для командных проектов
- Идеален для создания и распространения библиотек

Ограничения

- Немного медленнее, чем pip
- Требуется изучение формата TOML
- Может быть сложнее для новичков

Сравнение pip, uv и Poetry

Особенность	pip	uv	Poetry
Входит в состав Python по умолчанию	✅	❌	❌
Скорость	Средняя	⚡ Очень высокая	🐢 Умеренная
Написан на	Python	Rust	Python
Поддержка виртуальных окружений	Ручная	Встроенная	Встроенная
Разрешение зависимостей	Базовое	Продвинутое	Продвинутое
Использует pyproject.toml	❌	✅ (опционально)	✅
Идеальное применение	Простые скрипты	Быстрые, современные проекты	Полное управление проектом

Заключение

В Python существует несколько менеджеров пакетов, каждый из которых выполняет разные функции:

- pip прост и предустановлен — идеально подходит для небольших проектов.
- uv современный, быстрый и автоматизирует работу с окружениями.
- Poetry функционально богатый, управляет полными проектами и пакетированием.

Как выбрать подходящий:

Ситуация	Лучший выбор
Быстрый скрипт или обучение Python	pip
Нужна скорость и автоматизация	uv
Командный проект или пакетирование библиотек	Poetry

Ссылки

- Руководство пользователя по упаковке Python
- Документация pip
- Документация uv
- Документация Poetry