

Basic Concepts

`uv` is an extremely fast Python package and project manager written in Rust. It serves as a rapid replacement for `pip`, `pip-tools`, and partially `poetry`.

Package Management (pip replacement)

- `uv pip install [package]`: Install packages (significantly faster than standard pip).
- `uv pip list`: List installed packages.
- `uv pip uninstall [pkg]`: Remove packages.
- `uv pip freeze`: Output installed versions in requirements format.

Virtual Environments

- `uv venv`: Create a virtual environment (`.venv` by default).
- `uv venv --python 3.12`: Create with a specific Python version.
- Activation: `source .venv/bin/activate` (Standard shell activation).

Dependency Locking (pip-tools replacement)

- `uv pip compile requirements.in -o requirements.txt`: Resolve and lock dependencies.
- `uv pip sync requirements.txt`: Synchronize the environment to match the lock file exactly.

Project Management (Modern uv)

Recent versions of `uv` manage projects similar to Poetry/Cargo.

- `uv init`: Initialize a new project.
- `uv add [package]`: Add and install a dependency to `pyproject.toml`.
- `uv run [command]`: Run a command in the project environment.
- `uv lock`: Update the lock file.
- `uv sync`: Install all project dependencies.

Script Execution and Tools

- `uv run script.py`: Run a Python script. If dependencies are specified in headers (PEP 723), `uv` creates a temporary environment automatically.
- `uv tool install [package]`: Install a command-line tool (like `ruff` or `black`) in an isolated environment.

Python Version Management

- `uv python list`: List available/installed Python versions.
- `uv python install 3.11`: Download and install a specific Python version locally.

Why uv? (Pro Tips)

Unmatched Speed

Written in Rust, `uv` is designed for performance. It uses a global cache and hard-links to minimize disk I/O and network requests.

Built-in Resolver

Includes a highly optimized dependency resolver, solving complex version constraints much faster than legacy tools.

Single Binary

No Python required to run `uv` itself. It comes as a standalone binary, making it ideal for CI/CD pipelines.

Cache Management

Clean up the global cache with `uv cache clean` or view its status with `uv cache info`.