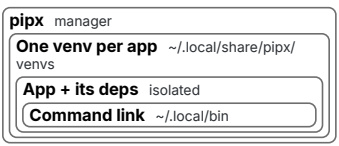
# pipx Python app installer

## CONCEPTS

pipx is for applications (black, httpie), not libraries: each gets its own venv, only its commands land on PATH.




Use pip or uv to add libraries to a project; use pipx for command-line tools you run everywhere.

## INSTALL

`pipx install <package>`
Install an app in its own venv

`pipx install <package> --python <python>`
Pick the Python version

`pipx install "<package>[<extra>]"`
Install with an extra

`pipx install git+<url>`
Install from a git repository

`pipx install ./<path>`
Install a local project

`pipx install --editable ./<path>`
Editable local install

`pipx install --force <package>`
Overwrite an existing install

## RUN WITHOUT INSTALLING

`pipx run <package>`
Run an app in a temporary venv

`pipx run <package> <args>`
Pass arguments to the app

`pipx run --spec "<package>==<version>" <command>`
Run a specific version

## RUN WITHOUT INSTALLING (cont.)

`pipx run --python <python> <package>`
Run on a chosen Python

`pipx run <script>.py`
Run a script with its inline dependencies

## MANAGE

`pipx list`
Show installed apps and versions

`pipx list --short`
One line per app

`pipx upgrade <package>`
Upgrade one app

`pipx upgrade-all`
Upgrade every app

`pipx uninstall <package>`
Remove an app and its venv

`pipx reinstall <package>`
Rebuild the venv

`pipx reinstall-all`
Rebuild every venv

## INJECT AND PIN

`pipx inject <package> <dep>`
Add a library to an app's venv

`pipx uninject <package> <dep>`
Remove an injected library

`pipx runpip <package> <pip-args>`
Run pip inside an app's venv

## SETUP AND ENVIRONMENT

`pipx ensurepath`
Put the app directory on PATH

`python3 -m pip install --user pipx`
Install pipx itself

`pipx environment`
Show paths and settings

On Debian/Ubuntu 23.04+ and Fedora, pip install --user is blocked; install pipx from the system package manager.