pipx Commands Cheatsheet

Subcommand	Example	Description
`run`	`$ pipx run --spec httpie http --body ifconfig.co/country`	Install a Python package into a temporary virtual environment and run the specified entry point with the given arguments.
`install`	`$ pipx install ipython`	Install a Python package into a virtual environment managed by pipx and register symlinks in the shell for each runnable entry point.
`upgrade`	`$ pipx upgrade ipython`	Install the most recent version of the specified Python package in its corresponding virtual environment.
`upgrade-all`	`$ pipx upgrade-all`	Install the most recent versions of all installed Python packages in their respective virtual environments.
`uninstall`	`$ pipx uninstall ipython`	Remove symlinks along with the virtual environment associated with the given Python package that was installed with pipx.
`uninstall-all`	`$ pipx uninstall-all`	Remove all symlinks and virtual environments created by pipx so far.
`reinstall`	`$ pipx reinstall --python=python3.12 isort`	Recreate the virtual environment of a package using the specified Python interpreter.
`reinstall-all`	`$ pipx reinstall-all --skip black flake8 --python=python3.12`	Recreate the virtual environments of all packages except for the listed ones using the specified Python interpreter.
`inject`	`$ pipx inject mypy setuptools`	Install an extra dependency into an existing virtual environment of a given package.
`uninject`	`$ pipx uninject poetry poetry-plugin-export`	Remove a dependency along with its indirect or transitive dependencies from a virtual environment of the specified package.
`runpip`	`$ pipx runpip poetry freeze > requirements.txt`	Run an arbitrary pip command in a virtual environment managed by pipx, for example, to pin the dependencies in a requirements file.
`list`	`$ pipx list --json`	List the Python packages installed with pipx using the JSON format.
`environment`	`$ pipx environment`	Display environment variables and directory paths used by pipx on this computer.
`completions`	`$ pipx completions`	Print instructions on enabling shell completions for pipx.
`ensurepath`	`$ pipx ensurepath`	Add the necessary folder paths to the `PATH` variable in your shell.

For more details on using these commands, check out Real Python's tutorial Install and Execute Python Applications Using pipx.