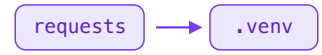
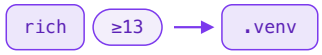
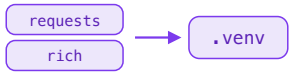
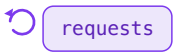
# pip

The pip commands you reach for, from simple installs to editable projects, requirements files, and the cache.



## Installing Packages

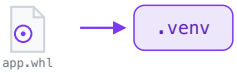
Install, constrain, and upgrade with pip.

| PURPOSE | COMMAND | RESULT |
|---|---|---|
| Install a package. | `pip install requests` |   |
| Constrain the version. | `pip install 'rich>=13'` |   |
| Install several at once. | `pip install requests rich` |   |
| Upgrade to the latest. | `pip install -U requests` |   |
| Force a clean reinstall. | `pip install --force-reinstall requests` |   |

pip installs into the active environment (a virtualenv, ideally)

## Install Sources

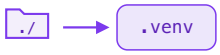
Where pip gets packages from.

| PURPOSE | COMMAND | RESULT |
|---|---|---|
| From a requirements file. | `pip install -r requirements.txt` |   |
| Pin an exact version. | `pip install requests==2.34.2` |   |
| Add optional extras. | `pip install 'httpx[http2]'` |   |
| Straight from Git. | `pip install git+https://github.com/psf/requests` |   |
| From a local wheel. | `pip install ./dist/app.whl` |   |

a requirement can be a name, a URL, a VCS ref, or a local path

## Editable & Project Installs

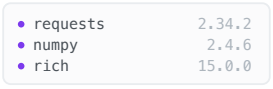
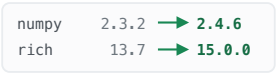
Install the project you are working on.

| PURPOSE | COMMAND | RESULT |
| --- | --- | --- |
| Editable (dev) install. | `pip install -e .` |   |
| Plain local install. | `pip install .` |   |
| Editable with extras. | `pip install -e '.[dev]'` |   |
| Skip dependencies. | `pip install --no-deps .` |   |
| Pin with a constraints file. | `pip install -c constraints.txt .` |   |

`-e` links the project live; edits take effect without reinstalling

## Uninstalling & Inspecting

Remove packages and see what is installed.

| PURPOSE | COMMAND | RESULT |
| --- | --- | --- |
| Uninstall a package. | `pip uninstall requests` |   |
| Show package details. | `pip show requests` |   |
| List its installed files. | `pip show -f requests` |   |
| List installed packages. | `pip list` |   |
| Show what is outdated. | `pip list --outdated` |   |

`pip show` reads metadata; `pip list` walks the environment

## Requirements & Freezing

Capture an environment and reproduce it.

| PURPOSE | COMMAND | RESULT |
| --- | --- | --- |
| Freeze exact versions. | `pip freeze` | `requests==2.34.2`<br>`numpy==2.4.6`<br>`rich==15.0.0` |
| Write a requirements file. | `pip freeze > requirements.txt` | .venv → requirements.txt |
| Reproduce the environment. | `pip install -r requirements.txt` | requirements.txt → .venv |
| Check for conflicts. | `pip check` | No broken requirements |
| List only what you asked for. | `pip list --not-required` | • requests top-level<br>• rich |

pip freeze pins everything; --not-required shows only the roots

## Finding & Downloading

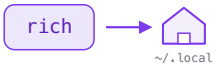
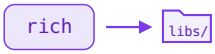
Discover versions and fetch without installing.

| PURPOSE | COMMAND | RESULT |
| --- | --- | --- |
| List available versions. | `pip index versions requests` | requests, available:<br>2.34.2 2.34.1 2.34.0 2.33.0 |
| Download, do not install. | `pip download requests` | requests → .whl |
| Download a whole set. | `pip download -r requirements.txt` | requirements.txt → *.whl |
| Build a wheel from source. | `pip wheel .` | ./ → app.whl |
| Save wheels to a folder. | `pip download -d wheels/ requests` | requests → wheels/ |

download and wheel fetch artifacts without touching the environment

## Indexes & Install Targets

Choose where packages come from and where they go.

| PURPOSE | COMMAND | RESULT |
|---|---|---|
| Install from a specific index. | `pip install -i https://pypi.org/simple rich` |   |
| Add a fallback index. | `pip install --extra-index-url \ https://acme.dev rich` |   |
| Install for your user only. | `pip install --user rich` |   |
| Install into a folder. | `pip install --target ./libs rich` |   |
| Show the active config. | `pip config list` | `global.index-url = '...'`<br>`install.user = true`<br>`global.cache-dir = '...'` |

set these once in pip.conf instead of repeating flags

## The pip Command & Cache

Manage pip itself and its download cache.

| PURPOSE | COMMAND | RESULT |
|---|---|---|
| Upgrade pip itself. | `python -m pip install -U pip` |   |
| Check pip's version. | `pip --version` |   |
| Find the cache directory. | `pip cache dir` |   |
| See the cache size. | `pip cache info` | `Number of files: 180`<br>`Cache size: 412 MB` |
| Clear the cache. | `pip cache purge` |   |

python -m pip avoids "which pip?" ambiguity across environments