pip Python package installer

CONCEPTS

Use python -m pip so pip always belongs to the interpreter you mean.

Index PyPI or mirror
↓ resolve
Wheel / sdist cached download
↓ build + unpack
site-packages active environment
↓ freeze
requirements.txt pinned list

Install into a virtual environment; on system Pythons marked EXTERNALLY-MANAGED pip refuses.

INSTALL

pip install *<package>*
Install the latest version

pip install *<package>*==*<version>*
Install an exact version

pip install "*<package>*≥1.2,<2"
Install within a range

pip install -r *<file>*
Install from a requirements file

pip install -U *<package>*
Upgrade to the newest version

pip install -e *<path>*
Editable install of a local project

pip install "*<package>*[*<extra>*]"
Install with optional extras

pip install *<path or url>*
Install a wheel, archive or directory

pip install git+*<url>*@*<ref>*
Install from a Git repository

pip install --dry-run *<package>*
Show what would be installed

INSPECT

pip list
List installed packages

INSPECT (cont.)

pip list --outdated
Show packages with newer versions

pip show *<package>*
Show version, location, dependents

pip freeze
Print installed packages as pins

pip list --not-required
Only top-level packages

pip check
Verify dependencies are consistent

pip index versions *<package>*
List versions on the index

REMOVE

pip uninstall *<package>*
Remove a package

pip uninstall -y *<package>*
Remove without confirmation

REQUIREMENTS

pip freeze > requirements.txt
Snapshot the environment

pip install -r requirements.txt
Restore a snapshot

pip install -c constraints.txt -r *<file>*
Cap versions without adding packages

INDEXES AND DOWNLOAD

pip install --no-index -f *<dir>* *<package>*
Install offline from a directory

CONFIG AND CACHE

pip --version
Show pip and Python versions

python -m pip install -U pip
Upgrade pip itself