

# pip Cheatsheet

By Dejan Panovski • Updated on Jun 20, 2026 • [Download PDF](#)

Quick reference for pip commands covering package installation, version constraints, requirements files, upgrades, virtual environments, and configuration.

pip is the package installer for Python. This cheatsheet covers common pip commands for installing and removing packages, pinning versions, working with requirements files, inspecting installed packages, managing virtual environments, and configuring pip.

## Install Packages

Add packages from PyPI and other sources.

<code>pip install package</code>	Install the latest version
<code>pip install pkg1 pkg2</code>	Install several packages
<code>pip install --user package</code>	Install into the user site
<code>pip install -e .</code>	Install local project in editable mode
<code>pip install 'package[extra]'</code>	Install with optional extras
<code>pip install ./dist/pkg.whl</code>	Install from a wheel file
<code>python -m pip install package</code>	Install through the python launcher

## Versions and Sources

Pin versions and install from other indexes.

<code>pip install package==1.2.3</code>	Install an exact version
<code>pip install 'package&gt;=1.0,&lt;2.0'</code>	Install within a version range
<code>pip install --upgrade package</code>	Upgrade to the latest version
<code>pip install --pre package</code>	Allow pre-release versions
<code>pip install 'git+https://host/repo.git'</code>	Install from a Git repository
<code>pip install -i https://mirror/simple package</code>	Use an alternate index URL
<code>pip index versions package</code>	List available versions

## Requirements Files

Record and reproduce a project environment.

<code>pip install -r requirements.txt</code>	Install from a requirements file
<code>pip freeze</code>	Print installed packages with versions
<code>pip freeze &gt; requirements.txt</code>	Write current packages to a file
<code>pip download -r requirements.txt -d ./pkgs</code>	Download packages without installing
<code>pip install --no-deps -r requirements.txt</code>	Install without dependencies
<code>pip uninstall -r requirements.txt</code>	Remove packages listed in a file

## Remove Packages

Uninstall packages and verify the tree.

<code>pip uninstall package</code>	Remove a package
<code>pip uninstall -y package</code>	Remove without confirmation
<code>pip uninstall pkg1 pkg2</code>	Remove several packages
<code>pip check</code>	Verify installed dependencies

## Inspect Packages

List installed packages and read metadata.

<code>pip list</code>	List installed packages
<code>pip list --outdated</code>	Show packages with newer versions
<code>pip list --not-required</code>	List packages no other package needs
<code>pip show package</code>	Show package details
<code>pip show -f package</code>	List files installed by a package
<code>pip --version</code>	Show the pip and Python version

## Virtual Environments

Isolate project dependencies with venv.

<code><a href="#">python -m venv myenv</a></code>	Create a virtual environment
<code>source myenv/bin/activate</code>	Activate the environment (Linux/macOS)
<code>deactivate</code>	Leave the active environment
<code>pip install --upgrade pip</code>	Upgrade pip inside the environment
<code>pip install --break-system-packages package</code>	Override PEP 668 (use sparingly)

## Configuration and Cache

Adjust pip settings and manage the cache.

<code>pip config list</code>	Show the current configuration
<code>pip config set global.index-url URL</code>	Set a default index URL
<code>pip install --no-cache-dir package</code>	Install without using the cache
<code>pip cache dir</code>	Show the cache location
<code>pip cache info</code>	Show cache size and counts
<code>pip cache purge</code>	Clear the wheel cache

## Related Guides

Use these references for pip and Python workflows.

<a href="#"><u>Install pip on Debian</u></a>	Install pip3 on Debian 11, 12, and 13
<a href="#"><u>Install pip on Ubuntu</u></a>	Install pip3 on Ubuntu
<a href="#"><u>Python Virtual Environments</u></a>	Create and use isolated environments
<a href="#"><u>pip vs apt</u></a>	When to use pip or the system package manager