

pip list

Usage

Unix/macOS Windows

```
python -m pip list [options]
```

Description

List installed packages, including editables.

Packages are listed in a case-insensitive sorted order.

Options

-o, --outdated

List outdated packages

(environment variable: `PIP_OUTDATED`)

-u, --uptodate

List uptodate packages

(environment variable: `PIP_UPTODATE`)

-e, --editable

List editable projects.

(environment variable: `PIP_EDITABLE`)

-l, --local

If in a virtualenv that has global access, do not list globally-installed packages.

(environment variable: `PIP_LOCAL`)

[Skip to content](#)

--user

Only output packages installed in user-site.

(environment variable: `PIP_USER`)

--path <path>

Restrict to the specified installation path for listing packages (can be used multiple times).

(environment variable: `PIP_PATH`)

--pre

Include pre-release and development versions. By default, pip only finds stable versions.

(environment variable: `PIP_PRE`)

--format <list_format>

Select the output format among: columns (default), freeze, or json. The 'freeze' format cannot be used with the --outdated option.

(environment variable: `PIP_FORMAT`)

--not-required

List packages that are not dependencies of installed packages.

(environment variable: `PIP_NOT_REQUIRED`)

--exclude-editable

Exclude editable package from output.

(environment variable: `PIP_EXCLUDE_EDITABLE`)

--include-editable

Include editable package from output.

(environment variable: `PIP_INCLUDE_EDITABLE`)

--exclude <package>

Exclude specified package from the output

(environment variable: `PIP_EXCLUDE`)

[Skip to content](#)

-i, --index-url <url>

Base URL of the Python Package Index (default <https://pypi.org/simple>). This should point to a repository compliant with PEP 503 (the simple repository API) or a local directory laid out in the same format.

(environment variable: `PIP_INDEX_URL`, `PIP_PYPI_URL`)

--extra-index-url <url>

Extra URLs of package indexes to use in addition to --index-url. Should follow the same rules as --index-url.

(environment variable: `PIP_EXTRA_INDEX_URL`)

--no-index

Ignore package index (only looking at --find-links URLs instead).

(environment variable: `PIP_NO_INDEX`)

-f, --find-links <url>

If a URL or path to an html file, then parse for links to archives such as sdist (.tar.gz) or wheel (.whl) files. If a local path or [file://](#) URL that's a directory, then look for archives in the directory listing. Links to VCS project URLs are not supported.

(environment variable: `PIP_FIND_LINKS`)

[Skip to content](#)

Examples

1. List installed packages (with the default column formatting).

Unix/macOS Windows

```
$ python -m pip list
Package Version
-----
docopt  0.6.2
idlex   1.13
jedi    0.9.0
```

2. List outdated packages with column formatting.

Unix/macOS Windows

```
$ python -m pip list --outdated --format columns
Package      Version Latest Type
-----
retry        0.8.1   0.9.1  wheel
setuptools  20.6.7  21.0.0 wheel
```

3. List packages that are not dependencies of other packages. Can be combined with other options.

Unix/macOS Windows

```
$ python -m pip list --outdated --not-required
Package Version Latest Type
-----
docutils 0.14    0.17.1 wheel
```

4. Use json formatting

Unix/macOS Windows

```
$ python -m pip list --format=json
[{'name': 'colorama', 'version': '0.3.7'}, {'name': 'docopt', 'version': '0.6.2'}, ...]
```

5. Use freeze formatting

[Skip to content](#) Windows

```
$ python -m pip list --format=freeze
colorama==0.3.7
docopt==0.6.2
idlex==1.13
jedi==0.9.0
```

6. List packages installed in editable mode

When some packages are installed in editable mode, `pip list` outputs an additional column that shows the directory where the editable project is located (i.e. the directory that contains the `pyproject.toml` or `setup.py` file).

Unix/macOS Windows

```
$ python -m pip list
Package            Version  Editable project location
-----
pip                21.2.4
pip-test-package  0.1.1    /home/you/.venv/src/pip-test-package
setuptools         57.4.0
wheel              0.36.2
```

The json format outputs an additional `editable_project_location` field.

[Skip to content](#)

Unix/macOS Windows

```
$ python -m pip list --format=json | python -m json.tool
[
  {
    "name": "pip",
    "version": "21.2.4",
  },
  {
    "name": "pip-test-package",
    "version": "0.1.1",
    "editable_project_location": "/home/you/.venv/src/pip-test-package"
  },
  {
    "name": "setuptools",
    "version": "57.4.0"
  },
  {
    "name": "wheel",
    "version": "0.36.2"
  }
]
```

 Note

Contrary to the `freeze` command, `pip list --format=freeze` will not report editable install information, but the version of the package at the time it was installed.