Python: pip

pip

[Docs](https://pip.pypa.io/) | [Issues](https://github.com/pypa/pip/issues) | [GitHub](https://github.com/pypa/pip) | [PyPI](https://pypi.org/project/pip/)

The most popular tool for installing Python packages, and the one included with modern versions of Python.

It provides the essential core features for finding, downloading, and installing packages from PyPI and other Python package indexes, and can be incorporated into a wide range of development workflows via its command-line interface (CLI).

<https://pip.pypa.io/en/stable/>

Afbeelding met tekst

Automatisch gegenereerde beschrijving

<https://pip.pypa.io/en/stable/getting-started/>

# **Getting Started**

**To get started with using pip, you should**[**install Python**](https://realpython.com/installing-python/)**on your system.**

## Ensure you have a working pip

As a first step, you should check that you have a working Python with pip installed. This can be done by running the following commands and making sure that the output looks similar.

Linux

$ **python --version**

*Python 3.N.N*

$ **python -m pip --version**

*pip X.Y.Z from ... (python 3.N.N)*

MacOSWindows

$ **py --version**

*Python 3.N.N*

$ **py -m pip --version**

If that worked, congratulations! You have a working pip in your environment.

If you got output that does not look like the sample above, please read the [Installation](https://pip.pypa.io/en/stable/installation/) page. It provides guidance on how to install pip within a Python environment that doesn’t have it.

# **Installation**

Usually, pip is automatically installed if you are:

* working in a [virtual environment](https://packaging.python.org/en/latest/tutorials/installing-packages/#creating-and-using-virtual-environments)
* using Python downloaded from [python.org](https://www.python.org/)
* using Python that has not been modified by a redistributor to remove [ensurepip](https://docs.python.org/3/library/ensurepip.html#module-ensurepip)

## Supported Methods

If your Python environment does not have pip installed, there are 2 mechanisms to install pip supported directly by pip’s maintainers:

* ensurepip
* get-pip.py

### ensurepip

Python comes with an [ensurepip](https://docs.python.org/3/library/ensurepip.html#module-ensurepip) module[1](https://pip.pypa.io/en/stable/installation/#python), which can install pip in a Python environment.

Linux

$ **python -m ensurepip --upgrade**

MacOSWindows

C:\ **py -m ensurepip --upgrade**

Afbeelding met tekst

Automatisch gegenereerde beschrijving

Path: c:\users\peter.schepens\appdata\local\programs\python\python310\lib\site-packages\

More details about how [ensurepip](https://docs.python.org/3/library/ensurepip.html#module-ensurepip) works and how it can be used, is available in the standard library documentation.

### get-pip.py

This is a Python script that uses some bootstrapping logic to install pip.

* Download the script, from <https://bootstrap.pypa.io/get-pip.py>.
* Open a terminal/command prompt, cd to the folder containing the get-pip.py file and run:

Linux

$ **python get-pip.py**

MacOSWindows

C:\**py get-pip.py**

Afbeelding met tekst

Automatisch gegenereerde beschrijving  
path: C:\Users\peter.schepens\AppData\Local\Programs\Python\Python310\Scripts

More details about this script can be found in [pypa/get-pip](https://github.com/pypa/get-pip)’s README.

Afbeelding met tekst

Automatisch gegenereerde beschrijving

## Alternative Methods

Depending on how you installed Python, there might be other mechanisms available to you for installing pip such as [using Linux package managers](https://packaging.python.org/en/latest/guides/installing-using-linux-tools/#installing-pip-setuptools-wheel-with-linux-package-managers).

These mechanisms are provided by redistributors of pip, who may have modified pip to change its behaviour. This has been a frequent source of user confusion, since it causes a mismatch between documented behaviour in this documentation and how pip works after those modifications.

If you face issues when using Python and pip installed using these mechanisms, it is recommended to request for support from the relevant provider (eg: Linux distro community, cloud provider support channels, etc).

## Upgrading pip

Upgrading your pip by running:

Linux

$ **python -m pip install --upgrade pip**

MacOSWindows

C:\**py -m pip install --upgrade pip**

**(let op: het kan zijn dat je in de sub-directory: lib\site-packages moet gaan staan voor uitvoeren van commando!!)**

## Compatibility

The current version of pip works on:

* Windows, Linux and MacOS.
* CPython 3.7, 3.8, 3.9, 3.10 and latest PyPy3.

pip is tested to work on the latest patch version of the Python interpreter, for each of the minor versions listed above. Previous patch versions are supported on a best effort approach.

pip’s maintainers do not provide support for users on older versions of Python, and these users should request for support from the relevant provider (eg: Linux distro community, cloud provider support channels, etc).

## Common tasks

### Install a package

Linux

$ python -m pip install sampleproject

*[...]*

*Successfully installed sampleproject*

MacOSWindows

By default, pip will fetch packages from [Python Package Index](https://pypi.org/), a repository of software for the Python programming language where anyone can upload packages.

### Install a package from GitHub

Linux

$ python -m pip install git+https://github.com/pypa/sampleproject.git@main

*[...]*

*Successfully installed sampleproject*

MacOSWindows

See [VCS Support](https://pip.pypa.io/en/stable/topics/vcs-support/) for more information about this syntax.

### Install a package from a distribution file

pip can install directly from distribution files as well. They come in 2 forms:

* [source distribution](https://packaging.python.org/en/latest/glossary/#term-Source-Distribution-or-sdist) (usually shortened to “sdist”)
* [wheel distribution](https://packaging.python.org/en/latest/glossary/#term-Wheel) (usually shortened to “wheel”)

Linux

$ python -m pip install sampleproject-1.0.tar.gz

*[...]*

*Successfully installed sampleproject*

$ python -m pip install sampleproject-1.0-py3-none-any.whl

*[...]*

*Successfully installed sampleproject*

MacOSWindows

### Install multiple packages using a requirements file

Many Python projects use requirements.txt files, to specify the list of packages that need to be installed for the project to run. To install the packages listed in that file, you can run:

Linux

$ python -m pip install -r requirements.txt

*[...]*

*Successfully installed sampleproject*

MacOSWindows

### Upgrade a package

Linux

$ python -m pip install --upgrade sampleproject

*Uninstalling sampleproject:*

*[...]*

*Proceed (y/n)? y*

*Successfully uninstalled sampleproject*

MacOSWindows

### Uninstall a package

Linux

$ python -m pip uninstall sampleproject

*Uninstalling sampleproject:*

*[...]*

*Proceed (y/n)? y*

*Successfully uninstalled sampleproject*

MacOSWindows

## Next Steps

It is recommended to learn about what virtual environments are and how to use them. This is covered in the [“Installing Packages”](https://packaging.python.org/en/latest/tutorials/installing-packages/) tutorial on packaging.python.org.