

# Managing Python Packages

Concepts, Tooling, History, and Recommendations

2019-12-09

Guilhem Saurel

Available at

`homepages.laas.fr/gsaurel/  
managing-python-packages.pdf`

Under License



<https://creativecommons.org/licenses/by-sa/4.0/>

## Source

`gitlab.laas.fr/gsaurel/talks :`  
`managing-python-packages.md`

## Discussions

<https://im.laas.fr/#/room/#managing-python-packages:laas.fr>

- 1 What are Python Packages ?
- 2 How can they be used ?
- 3 Which tools can help us ?

What are Python Packages ?

- `toto.py`
- `toto.so`
- `toto/__init__.py`

- toto.py
- toto.so
- toto/\_\_init\_\_.py

```
>>> import toto
```

```
from datetime import datetime
from pathlib import Path

def my_log(line: str, filename: Path):
    with filename.open('a') as handle:
        print(f'{datetime.now()}: {line}', file=handle)

if __name__ == '__main__':
    log_file = Path('/tmp/log.txt')
    my_log("test string", log_file)
```

```
toto/  
├── __init__.py  
├── robot.py  
├── settings.py  
├── simulation.py  
└── user_interface.py
```

```
toto/  
├── __init__.py  
├── robot.py  
├── settings.py  
├── simulation.py  
└── user_interface.py  
  
>>> import toto.user_interface
```

```
toto/  
├── __init__.py  
├── robot.py  
├── sensors/  
│   ├── camera.py  
│   ├── imu.py  
│   └── __init__.py  
├── settings.py  
├── simulation.py  
└── user_interface.py
```

```
toto/  
├── __init__.py  
├── robot.py  
├── sensors/  
│   ├── camera.py  
│   ├── imu.py  
│   └── __init__.py  
├── settings.py  
├── simulation.py  
└── user_interface.py  
  
>>> import toto.sensors.imu
```

don't

```
from settings import ROBOT_SIZE
```

don't

```
from settings import ROBOT_SIZE
```

do

```
from toto.settings import ROBOT_SIZE
```

```
# or
```

```
from .settings import ROBOT_SIZE
```

don't

```
from settings import ROBOT_SIZE
```

do

```
from toto.settings import ROBOT_SIZE
```

```
# or
```

```
from .settings import ROBOT_SIZE
```

```
# or
```

```
import toto # or import toto.settings  
toto.settings.ROBOT_SIZE
```

don't

```
from .robot import *
```

don't

```
from .robot import *
```

do

```
from .robot import start, stop, turn, jump
```

```
# or
```

```
import .robot as rb
```

How can they be used ?

```
In [1]: import sys
```

```
In [2]: sys.path
```

```
Out[2]:
```

```
['/usr/lib/python38.zip',  
 '/usr/lib/python3.8',  
 '/usr/lib/python3.8/lib-dynload',  
 '',  
 '/home/user/.local/lib/python3.8/site-packages',  
 '/usr/local/lib/python3.8/dist-packages',  
 '/usr/lib/python3/dist-packages']
```

```
In [1]: import sys
```

```
In [2]: sys.path
```

```
Out[2]:
```

```
['/usr/lib/python38.zip',  
 '/usr/lib/python3.8',  
 '/usr/lib/python3.8/lib-dynload',  
 '',  
 '/home/user/.local/lib/python3.8/site-packages',  
 '/usr/local/lib/python3.8/dist-packages',  
 '/usr/lib/python3/dist-packages']
```

[https://docs.python.org/3/library/site:](https://docs.python.org/3/library/site)

```
sys.prefix + 'lib/pythonX.Y/site-packages'
```

```
$ SP="lib/python3.8/site-packages"  
$ export PYTHONPATH=/opt/openrobots/${SP}  
$ # or  
$ echo /opt/openrobots/${SP} > ~/.local/${SP}/my.pth
```

```
$ sudo apt install python3-numpy
$ dpkg -L python3-numpy
...
/usr/lib/python3/dist-packages/numpy/__init__.py
...
```

```
$ sudo apt install python3-numpy
```

```
$ dpkg -L python3-numpy
```

```
...
```

```
/usr/lib/python3/dist-packages/numpy/__init__.py
```

```
...
```

- Works out of the box
- Intended package for:
  - your system's other applications
  - any other user of your system

don't

```
$ sudo pip install --upgrade numpy
```

```
...
```

```
Installing collected packages: numpy
```

```
Attempting uninstall: numpy
```

```
Found existing installation: numpy 1.17.4
```

```
Not uninstalling numpy at /usr/lib/python3/dist-p
```

```
Can't uninstall 'numpy'. No files were found to u
```

```
Successfully installed numpy-1.21.2
```

```
>>> import numpy
```

```
>>> numpy.__version__
```

```
'1.21.2'
```

```
>>> numpy.__file__
```

```
'/usr/local/lib/python3.8/dist-packages/numpy/...'
```

How can they be used ?

do

```
$ python -m pip install --user --upgrade numpy
```

```
...
```

```
Installing collected packages: numpy
```

```
Successfully installed numpy-1.21.2
```

```
>>> import numpy
```

```
>>> numpy.__version__
```

```
'1.21.2'
```

```
>>> numpy.__file__
```

```
'/home/user/.local/lib/python3.8/site-packages/...'
```

With a `setup.py`:

```
from setuptools import setup
```

```
setup(  
    name="toto",  
    version="1.0.0",  
    packages=["toto"],  
)
```

## Local path

```
$ python -m pip install --user .
```

## Local path

```
$ python -m pip install --user .
```

## Editable path

```
$ python -m pip install --user --editable .
```

## Local path

```
$ python -m pip install --user .
```

## Editable path

```
$ python -m pip install --user --editable .
```

## VCS URL

```
$ python -m pip install --user \  
    'git+https://github.com/psf/black#egg=black'
```

Which tools can help us ?

```
$ cat requirements.txt
```

```
numpy>=1.16
```

```
quadprog==0.1.7
```

```
requests
```

```
tqdm
```

```
python -m pip install --user -r requirements.txt
```

```
$ cat requirements.in  
django<4  
$ python -m pip install --user pip-tools  
$ pip-compile
```

```
$ cat requirements.txt
#
# This file is autogenerated by pip-compile with pytl
# To update, run:
#
#     pip-compile
#
asgiref==3.4.1
    # via django
django==3.2.9
    # via -r requirements.in
pytz==2021.3
    # via django
sqlparse==0.4.2
    # via django
```

```
$ python -m venv ~/my_v_env
```

```
$ python -m venv ~/my_v_env
```

```
~/my_v_env/  
├── bin/  
│   ├── activate  
│   ├── pip  
│   └── python -> /usr/bin/python3  
├── include/  
├── lib/  
│   └── python3.8/  
│       └── site-packages/  
│           ├── pip/  
│           └── setuptools/  
└── pyvenv.cfg
```

Which tools can help us ?

```
$ . ~/my_v_env/bin/activate  
(my_v_env) $ which python  
/home/user/my_v_env/bin/python
```

```
In [1]: import sys
```

```
In [2]: sys.path
```

```
Out[2]:
```

```
['/home/user/my_v_env/bin',  
 '/usr/lib/python38.zip',  
 '/usr/lib/python3.8',  
 '/usr/lib/python3.8/lib-dynload',  
 '',  
 '/home/user/my_v_env/lib/python3.8/site-packages',  
 '/home/user/.ipython']
```

- mkvirtualenv
- workon
- deactivate
- .venv file
- mktmpenv

- mkvirtualenv
- workon
- deactivate
- .venv file
- mktmpenv

virtualfish

*Python Development Workflow for Humans*

— Kenneth Reitz

- pip w/ version pinning
- virtualenv w/ wrapper

- pip w/ version pinning (like Pipenv)
- virtualenv w/ wrapper (like Pipenv)
- with replacement for `setup.py` / `setup.cfg`
- with better handling of versions
- without Kenneth Reitz

*The World's Most Popular Data Science Platform*

— <https://www.anaconda.com/>

- a separate package format
- a separate package repository (condaforge)
- a separate package manager (conda)
- incompatible with everything else you already have

*The World's Most Popular Data Science Platform*

— <https://www.anaconda.com/>

- a separate package format
- a separate package repository (condaforge)
- a separate package manager (conda)
- incompatible with everything else you already have
- with commercial support

- 1 `sudo apt install python3-toto`
- 2 `python -m pip install --user --upgrade toto`

- 1 `sudo apt install python3-toto`
- 2 `python -m pip install --user --upgrade toto`
- 3 `python -m venv venv`

- 1 `sudo apt install python3-toto`
- 2 `python -m pip install --user --upgrade toto`
- 3 `python -m venv venv`
- 4 `poetry`

Thanks for your attention :)