

Python Packaging

Jakub Wasielak



<http://blog.pykonik.org/>



<https://pl.pycon.org/2017/>

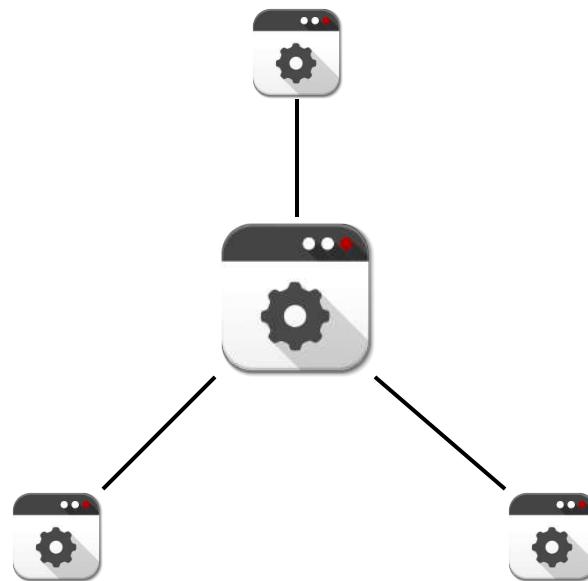


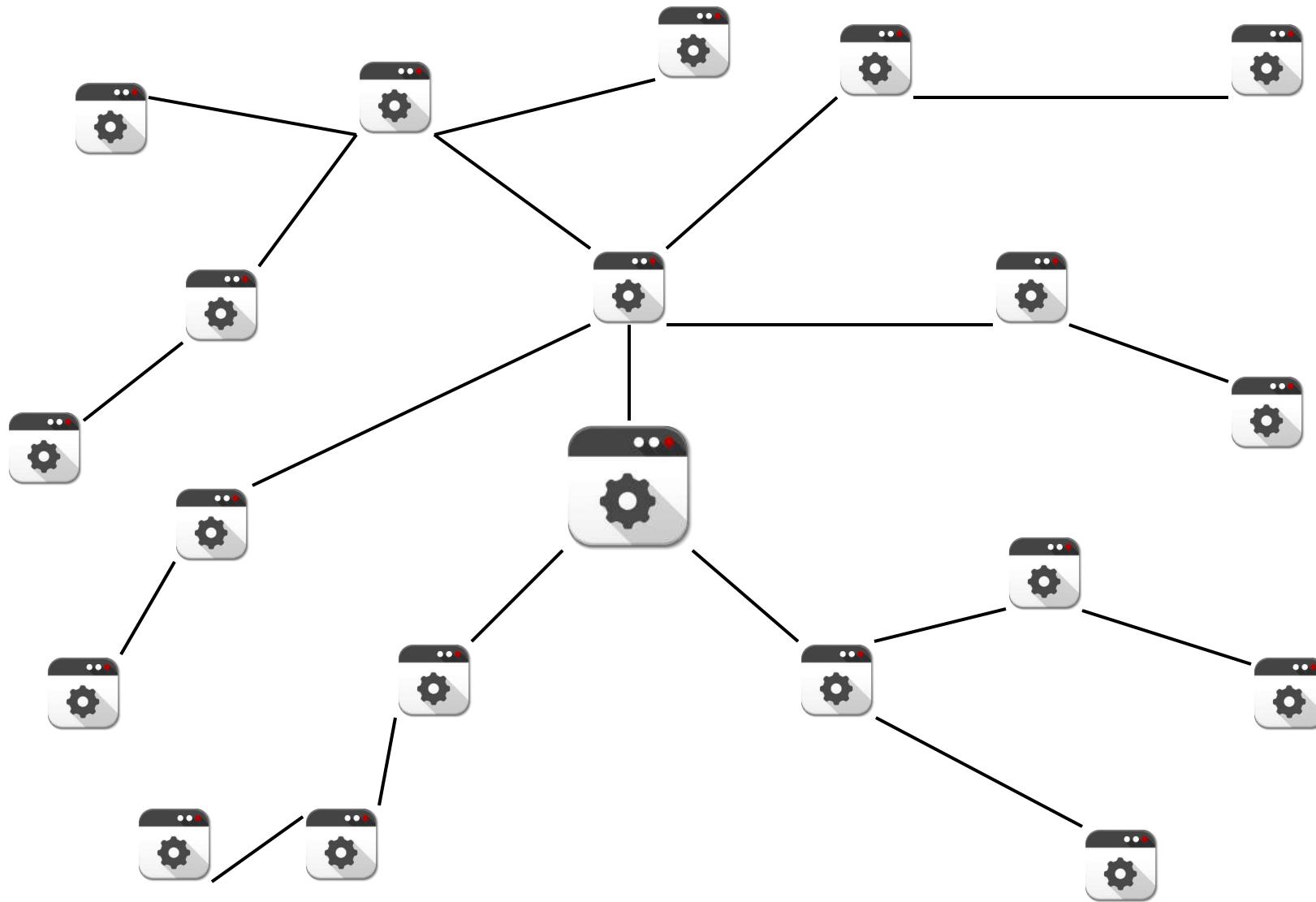
<http://koderek.edu.pl/>
facebook.com/startechkrk



what? why?

Architecture





<https://packaging.python.org/current/>

Installation Tool Recommendations

- pip
- virtualenv

Packaging Tool Recommendations

- setuptools
- bdist_wheel
- twine

Asking for help?

```
$ python setup.py --help-commands
Standard commands:
build           build everything needed to install
build_py        "build" pure Python modules (copy to build directory)
build_ext       build C/C++ extensions (compile/link to build directory)
build_clib      build C/C++ libraries used by Python extensions
build_scripts   "build" scripts (copy and fixup #! line)
clean           clean up temporary files from 'build' command
install         install everything from build directory
install_lib     install all Python modules (extensions and pure Python)
install_headers install C/C++ header files
install_scripts install scripts (Python or otherwise)
install_data    install data files
sdist           create a source distribution (tarball, zip file, etc.)
register        register the distribution with the Python package index
bdist            create a built (binary) distribution
bdist_dumb      create a "dumb" built distribution
bdist_rpm       create an RPM distribution
bdist_wininst   create an executable installer for MS Windows
upload          upload binary package to PyPI
check           perform some checks on the package

Extra commands:
... this one goes on
```

setup.py

```
import os
from setuptools import setup

setup(
    name = "an_example_pypi_project",
    version = "0.0.4",
    author = "Jakub Wasielak",
    author_email = "kuba.wasielak@gmail.com",
    description = ("An demonstration of how to create, document, and publish "
                  "to the cheese shop a5 pypi.org."),
    license = "BSD",
    keywords = "example documentation tutorial",
    url = "http://packages.python.org/an_example_pypi_project",
    packages=['an_example_pypi_project', 'tests'],
    long_description=read('README'),
    classifiers=[
        "Development Status :: 3 - Alpha",
        "Topic :: Utilities",
        "License :: OSI Approved :: BSD License",
    ],
)
```

(source: https://pythonhosted.org/an_example_pypi_project/setuptools.html)

setup.py

```
import os
from setuptools import setup

setup(
    name = "an_example_pypi_project",
    version = "0.0.4",
    author = "Jakub Wasielak",
    author_email = "kuba.wasielak@gmail.com",
    description = ("An demonstration of how to create, document, and publish "
                  "to the cheese shop a5 pypi.org."),
    license = "BSD",
    keywords = "example documentation tutorial",
    url = "http://packages.python.org/an example pypi project",
    packages=['an_example_pypi_project', 'tests'],
    long_description=read('README'),
    classifiers=[
        "Development Status :: 3 - Alpha",
        "Topic :: Utilities",
        "License :: OSI Approved :: BSD License",
    ],
)
```

(source: https://pythonhosted.org/an_example_pypi_project/setuptools.html)

setup.py

```
import os
from setuptools import setup

setup(
    name = "an example pypi project",
    version = "0.0.4",
    author = "Jakub Wasielak",
    author_email = "kuba.wasielak@gmail.com",
    description = ("An demonstration of how to create, document, and publish "
                  "to the cheese shop a5 pypi.org."),
    license = "BSD",
    keywords = "example documentation tutorial",
    url = "http://packages.python.org/an_example_pypi_project",
    packages=['an_example_pypi_project', 'tests'],
    long_description=read('README'),
    classifiers=[
        "Development Status :: 3 - Alpha",
        "Topic :: Utilities",
        "License :: OSI Approved :: BSD License",
    ],
)
```

setuptools_scm

```
setup(  
    name = "an_example_pypi_project",  
    use_scm_version=True,  
    setup_requires=[ 'setuptools_scm' ],  
    # ...  
)
```

PEP 440

<https://www.python.org/dev/peps/pep-0440/>

setup.py

```
import os
from setuptools import setup

setup(
    name = "an_example_pypi_project",
    version = "0.0.4",
    author = "Jakub Wasielak",
    author_email = "kuba.wasielak@gmail.com",
    description = ("An demonstration of how to create, document, and publish "
                  "to the cheese shop a5 pypi.org."),
    license = "BSD",
    keywords = "example documentation tutorial",
    url = "http://packages.python.org/an_example_pypi_project",
    packages=['an_example_pypi_project', 'tests'],
    long_description=read('README'),
    classifiers=[
        "Development Status :: 3 - Alpha",
        "Topic :: Utilities",
        "License :: OSI Approved :: BSD License",
    ],
)
```

setup.py

```
import os
from setuptools import setup

setup(
    name = "an_example_pypi_project",
    version = "0.0.4",
    author = "Jakub Wasielak",
    author_email = "kuba.wasielak@gmail.com",
    description = ("An demonstration of how to create, document, and publish "
                  "to the cheese shop a5 pypi.org."),
    license = "BSD",
    keywords = "example documentation tutorial",
    url = "http://packages.python.org/an_example_pypi_project",
    packages=['an example pypi project', 'tests'],
    long_description=read('README'),
    classifiers=[
        "Development Status :: 3 - Alpha",
        "Topic :: Utilities",
        "License :: OSI Approved :: BSD License",
    ],
)
```

(source: https://pythonhosted.org/an_example_pypi_project/setuptools.html)

setup.py

```
import os
from setuptools import setup

setup(
    name = "an_example_pypi_project",
    version = "0.0.4",
    author = "Jakub Wasielak",
    author_email = "kuba.wasielak@gmail.com",
    description = ("An demonstration of how to create, document, and publish "
                  "to the cheese shop a5 pypi.org."),
    license = "BSD",
    keywords = "example documentation tutorial",
    url = "http://packages.python.org/an_example_pypi_project",
    packages=['an_example_pypi_project', 'tests'],
    long_description=read('README'),
    classifiers=[
        "Development Status :: 3 - Alpha",
        "Topic :: Utilities",
        "License :: OSI Approved :: BSD License",
    ],
)
```

(source: https://pythonhosted.org/an_example_pypi_project/setuptools.html)

setup.py

```
import os
from setuptools import setup

setup(
    name = "an_example_pypi_project",
    version = "0.0.4",
    author = "Jakub Wasielak",
    author_email = "kuba.wasielak@gmail.com",
    description = ("An demonstration of how to create, document, and publish "
                  "to the cheese shop a5 pypi.org."),
    license = "BSD",
    keywords = "example documentation tutorial",
    url = "http://packages.python.org/an_example_pypi_project",
    packages=['an_example_pypi_project', 'tests'],
    long_description=read('README'),
    classifiers=[
        "Development Status :: 3 - Alpha",
        "Topic :: Utilities",
        "License :: OSI Approved :: BSD License",
    ],
)
```

(source: https://pythonhosted.org/an_example_pypi_project/setuptools.html)

setup.py

File	Type
setuptools-36.0.1-py2.py3-none-any.whl (md5)	Python
setuptools-36.0.1.zip (md5)	Source

Author: Python Packaging Authority

Home Page: <https://github.com/pypa/setuptools>

Keywords: CPAN PyPI distutils eggs package management

Categories

[Development Status :: 5 - Production/Stable](#)

[Intended Audience :: Developers](#)

[License :: OSI Approved :: MIT License](#)

[Operating System :: OS Independent](#)

[Programming Language :: Python :: 2](#)

[Programming Language :: Python :: 2.6](#)

[Programming Language :: Python :: 2.7](#)

[Programming Language :: Python :: 3](#)

[Programming Language :: Python :: 3.3](#)

[Programming Language :: Python :: 3.4](#)

[Programming Language :: Python :: 3.5](#)

[Programming Language :: Python :: 3.6](#)

[Topic :: Software Development :: Libraries :: Python Modules](#)

[Topic :: System :: Archiving :: Packaging](#)

[Topic :: System :: Systems Administration](#)

[Topic :: Utilities](#)

Best classifier?

"Private :: Do Not Upload"

setup.py

```
import os
from setuptools import setup

setup(
    name = "an_example_pypi_project",
    version = "0.0.4",
    author = "Jakub Wasielak",
    author_email = "kuba.wasielak@gmail.com",
    description = ("An demonstration of how to create, document, and publish "
                  "to the cheese shop a5 pypi.org."),
    license = "BSD",
    keywords = "example documentation tutorial",
    url = "http://packages.python.org/an_example_pypi_project",
    packages=['an_example_pypi_project', 'tests'],
    long_description=read('README'),
    classifiers=[
        "Development Status :: 3 - Alpha",
        "Topic :: Utilities",
        "License :: OSI Approved :: BSD License",
    ],
)
```

(source: https://pythonhosted.org/an_example_pypi_project/setuptools.html)

setup.py

```
import os
from setuptools import setup, find_packages

PACKAGES = find_packages(where="src")

setup(
    # ...
    packages=PACKAGES,
    # ...
)
```

(source: https://pythonhosted.org/an_example_pypi_project/setuptools.html)

There's more!

```
setup(  
    name = "an_example_pypi_project",  
    # ...  
    install_requires=[  
        "cherrypy==3.5",  
        "lxml",  
        "Pillow>=2.1,<3dev"  
    ],  
)
```

Extras

```
setup(  
    # ...  
    install_requires=[  
        "cherrypy>=3.5,<3.6dev",  
        "lxml",  
        "Pillow>=2.1,<3dev"  
    ],  
    extras_require=dict(  
        doc = [ 'Sphinx>=1.3' ],  
        notebook = [ 'notebook', 'ipywidgets' ],  
        # ... (^ comes from IPython)  
    )  
)
```

```
python setup.py install 'ipython[notebook]'
```

Tests? Why not!

```
setup(  
    # ...  
    install_requires=[  
        "cherrypy>=3.5,<3.6dev",  
        "lxml",  
        "Pillow>=2.1,<3dev"  
    ],  
    tests_require=[  
        'Pyro>=3.16,<4dev',  
        'pytest>=2.3',  
        'selenium'  
    ]  
)
```

python setup.py test

Or...

```
setup(  
    # ...  
    install_requires=[  
        "cherrypy>=3.5,<3.6dev",  
        "lxml",  
        "Pillow>=2.1,<3dev"  
    ],  
    extras_require={  
        'testing': [  
            'Pyro>=3.16,<4dev',  
            'pytest>=2.3',  
            'selenium'  
        ]  
    }  
)
```

And tox to install

Entry Points

```
setup(  
    # ...  
    'entry_points': {  
        'console_scripts': [ 'virtualenv=virtualenv:main' ],  
    },  
)
```

```
$ python virtualenv.py my_venv
```

vs.

```
$ virtualenv my_venv
```

setup.cfg

```
[global]
verbose = 1

[bdist_wheel]
universal = 1

[metadata]
license_file = LICENSE

[easy_install]
index_url = https://devpi.company.net/root/sth/+simple/

[tool:pytest]
norecursedirs = build env services *.egg project/lib/test
```

MANIFEST.in

```
include CHANGES.txt
include project/handlers/*.html
recursive-include project/static
```

setuptools_scm

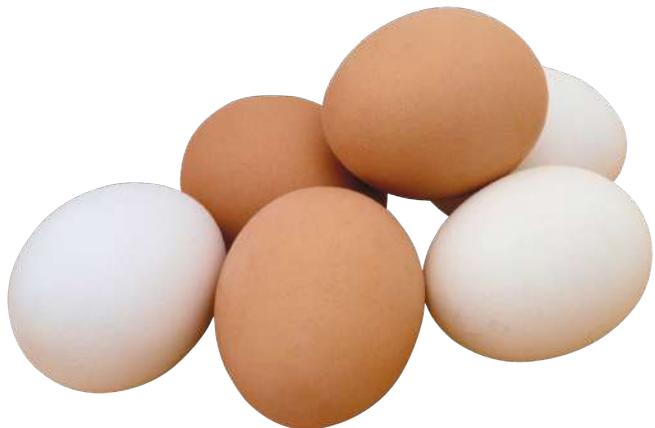
everything not in .gitignore will get used!

install vs. develop

```
$ python setup.py install
...
$ pip freeze | grep project
project==22.2

$ python setup.py develop
...
$ pip freeze | grep project
-e git+ssh://you@your.repo.url/Repositories/Team/project@id_123#egg=project
```

eggs



wheels

vs.

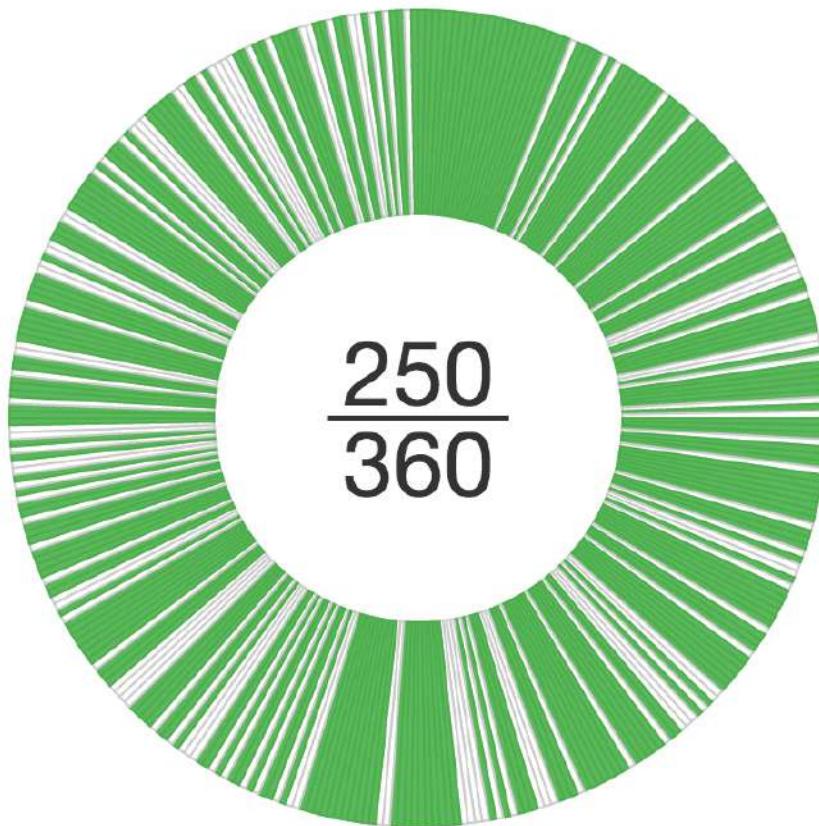


Wheels advantages

distribution-1.0-1-py27-none-any.whl

- official PEP (pep-0427)
- no .pyc files inside (one wheel for both Pythons, .pyc files will be generated upon installation)
- richer file naming conversion
- versioning
- installation of C components does not require compiler

Python Wheels



No pycrypto, SQLAlchemy, MySQL-python, tornado

requirements.txt

```
$ pip install requirements-dev.txt  
Collecting requirements-dev.txt  
Installing collected packages: requirements-dev.txt
```

Please reconsider deleting this from pypi #1

 Closed

piquadrat opened this issue on 27 Jan 2016 · 5 comments



piquadrat commented on 27 Jan 2016



I like a good joke as much as the next guy, but I'm really not interested to have "some nasty hacks" running on my machine due to a typo.

Congrats on the idea though!



1

devpi

<http://doc.devpi.net/latest/>

The screenshot shows a search interface with the YouGov logo. The search bar contains the query 'devpi'. Below the search bar, the URL 'http://doc.devpi.net/latest/' is displayed. To the right of the search bar are buttons for 'Search', 'server status ok', and 'How to search'. The main content area displays a list of search results for 'cherrypy' under the heading 'root/pypi/: cherrypy versions'. The results are listed in descending order of version number.

YouGov devpi

Search

server status ok How to search

devpi » root/pypi » cherrypy

root/pypi/: cherrypy versions

Index	Version
root/pypi	8.1.2
root/pypi	8.1.1
root/pypi	8.1.0
root/pypi	8.0.1
root/pypi	8.0.0
root/pypi	7.1.0
root/pypi	7.0.0
root/pypi	6.2.1
root/pypi	6.2.0
root/pypi	6.1.1
root/pypi	6.1.0
root/pypi	6.0.2
root/pypi	6.0.1
root/pypi	6.0.0
root/pypi	5.6.0

root/pypi/: cherrypy versions

Index Version

root/pypi [8.1.2](#)

root/pypi [8.1.1](#)

root/pypi [8.1.0](#)

root/pypi [8.0.1](#)

root/pypi [8.0.0](#)

root/pypi [7.1.0](#)

root/pypi [7.0.0](#)

root/pypi [6.2.1](#)

root/pypi [6.2.0](#)

root/pypi [6.1.1](#)

root/pypi [6.1.0](#)

root/pypi [6.0.2](#)

root/pypi [6.0.1](#)

root/pypi [6.0.0](#)

root/pypi [5.6.0](#)

Your projects, your packages

.pypirc

```
[distutils]
index-servers =
    my_devpi

[my_devpi]
repository: https://devpi-master.company.net/root/my_devpi/
username: {your username}
password: {your password}
```

Register

```
python setup.py sdist register -r my_devpi upload -r my_devpi
```

Upload

```
python setup.py sdist upload -r my_devpi
```

Or twine

Create

```
python setup.py sdist bdist_wheel
```

Upload

```
twine upload dist/*
```

Better Upload

```
export TWINE_USERNAME=foo  
export TWINE_PASSWORD=bar  
twine upload dist/*
```

Test your package

```
pip install dist/package-1.0.0.tar.gz  
pip install dist/package-1.0.0-py2.py3-none-any.whl
```

.pypirc

```
[distutils]  
index-servers =  
    test  
  
[test]  
repository: https://testpypi.python.org/pypi  
# repository: https://test.pypi.org/legacy/  
username: {your username}
```

```
pip install -i https://testpypi.python.org/pypi <package_name>
```

PEP 20, last line

Namespaces are one honking great idea
-- let's do more of those!

Namespaces

```
sound/                                Top-level package
    __init__.py                         Initialize the sound package
    formats/                            Subpackage for file format conversions
        __init__.py
        wavread.py
        wavwrite.py
        aiffread.py
        aifffwrite.py
        auread.py
        auwrite.py
        ...
    effects/                            Subpackage for sound effects
        __init__.py
        echo.py
        surround.py
        reverse.py
        ...
    filters/                            Subpackage for filters
        __init__.py
        equalizer.py
        vocoder.py
        karaoke.py
        ...
```

Namespaces

```
import sound.effects.echo  
import sound.effects.surround  
from sound.effects import *
```

pip cool features

(that easy_install doesn't have)

- easy_install can finish up with a partially completed installation
- better console output
- reasons for actions are kept
- native support for git, mercurial, bazaar
- uninstallation of packages
- pip freeze
- pip install -r requirements.txt

What's next?

<https://www.pypa.io/en/latest/roadmap/>

The screenshot shows the PyPA Roadmap page with the following sections:

- Table of Contents**: A sidebar with links to "PyPA Home", "PyPA News", "PyPA GitHub", "PyPA GitHub Issues", "PyPA GitHub Pull Requests", "PyPA GitHub Milestones", "PyPA GitHub Labels", and "PyPA GitHub Activity".
- PyPA Roadmap**:
 - Last Updated:** 2024-01-18
 - Description:** A central page for the major Python hub teams that will determine the direction Python is taking.
 - Core Standards**:
 - Milestone 2.0**:
 - Summary:** The maintainers requested the maintainers handle incoming pull requests (PRs). The review process, T.R., is specified to include "Build", "Review", and "Test". However, any PRs that do not fit the review process being used by the maintainers can be rejected.
 - Issue #114**: [pyproject.toml](#) is a standard file
 - Issue #115**: [pyproject.toml](#) is a standard file
 - Enhancement Maturity Update**:
 - Summary:** The maintainers requested the maintainers handle incoming pull requests (PRs). The review process, T.R., is specified to include "Build", "Review", and "Test". However, any PRs that do not fit the review process being used by the maintainers can be rejected.
 - Issue #116**: [pyproject.toml](#) is a standard file
 - Issue #117**: [pyproject.toml](#) is a standard file
 - Core Dependencies ("Python")**:
 - Summary:** An attempt to formalize the notion of "Python" introduced by proposal 2027123 was originally declined to include the basic dependencies in the Python distribution. This is now being reconsidered.
 - Issue #118**: [pyproject.toml](#) is a standard file
 - Issue #119**: [pyproject.toml](#) is a standard file
 - Core Utilities Enhancements**:
 - Summary:** A proposal to improve dependency resolution with custom metadata. 2027123 was originally rejected because the utilities it introduced were not yet finalized.
 - Issue #120**: [pyproject.toml](#) is a standard file
 - Issue #121**: [pyproject.toml](#) is a standard file
 - Standards Pathway Progression**:
 - Summary:** An attempt to standardize the progression of standards. The CTR documents and standard enforcement for Python packages. This is also intended to standardize the progression of standards for other Python components.
 - Issue #122**: [pyproject.toml](#) is a standard file
 - Issue #123**: [pyproject.toml](#) is a standard file
 - Issue #124**: [pyproject.toml](#) is a standard file
 - Build Infrastructure**:
 - Summary:** An attempt to specify a standard build API. Not yet final, after the Python Devs' Survey supported other build systems, this was eventually renamed to "Build Infrastructure".
 - Issue #125**: [pyproject.toml](#) is a standard file
 - Issue #126**: [pyproject.toml](#) is a standard file
 - Source Distribution 2.0**:
 - Summary:** A proposal for a new source format. The current proposal has been declined but is still under consideration. It is an attempt to standardize the source distribution format to something like ZIP or ZIP64.
 - Issue #127**: [pyproject.toml](#) is a standard file
 - Issue #128**: [pyproject.toml](#) is a standard file
 - Installation On-Demand Updates**:
 - Summary:** An update to the "Install" installation subcommand to install/uninstall PyPI packages.
 - Issue #129**: [pyproject.toml](#) is a standard file
 - Issue #130**: [pyproject.toml](#) is a standard file
 - Wheel Upgrades**:
 - Summary:** An update to the "wheel" command to remove a tagging scheme for installing wheels, amongst others, via pip. Other releases that people want to use.
 - Issue #131**: [pyproject.toml](#) is a standard file
 - Issue #132**: [pyproject.toml](#) is a standard file
 - Common Features Scheme**:
 - Summary:** This is another implementation for 2027123 that increases usage in other parts of the existing infrastructure while also addressing some of the concerns of the original proposal, but with different solutions.
 - Issue #133**: [pyproject.toml](#) is a standard file
 - Issue #134**: [pyproject.toml](#) is a standard file
 - Tots & SDG Initiatives**:
 - Summary:** An attempt to unify various "Tots" (Tooling, Outreach, Training) and "SDG" (Sustainability, Diversity, Growth) initiatives under one umbrella.
 - Issue #135**: [pyproject.toml](#) is a standard file
 - Issue #136**: [pyproject.toml](#) is a standard file
 - PEP 599 [WIP]**:
 - Summary:** Many organizations apply PEP 599 to their codebase, but currently there is no consistency, and many organizations have different ways of doing so.
 - Issue #137**: [pyproject.toml](#) is a standard file
 - Issue #138**: [pyproject.toml](#) is a standard file
 - Vendor drift fix for sdists**:
 - Summary:** Python "wheels" keep track of dependencies, so that packages at the G既然 take advantage of the shared library space and wheels.
 - Issue #139**: [pyproject.toml](#) is a standard file
 - Issue #140**: [pyproject.toml](#) is a standard file
 - The New & Improved PyPi**:
 - Summary:** An attempt to improve the Python PyPi interface to maintainers and users. Involves both the search and the rest of the site.
 - Issue #141**: [pyproject.toml](#) is a standard file
 - Issue #142**: [pyproject.toml](#) is a standard file
 - PyPi Integration w/ TUF**:
 - Summary:** An attempt to integrate PyPi with the "The Update Framework" (TUF). This is specified in PEP 553.
 - Issue #143**: [pyproject.toml](#) is a standard file
 - Issue #144**: [pyproject.toml](#) is a standard file
 - Doc migration**:
 - Summary:** An attempt to improve the Python PyPi documentation to maintainers and users. Involves both the search and the rest of the site.
 - Issue #145**: [pyproject.toml](#) is a standard file
 - Issue #146**: [pyproject.toml](#) is a standard file
 - New PyPi Tutorials**:
 - Summary:** An attempt to improve the Python PyPi tutorial for maintainers and users. Involves both the search and the rest of the site.
 - Issue #147**: [pyproject.toml](#) is a standard file
 - Issue #148**: [pyproject.toml](#) is a standard file
 - Specs vs PCDs**:
 - Summary:** An attempt to formalize the Python PyPi specification. Specifically how to register things in the PyPi. All this is to be better than what is currently implemented in the PyPi.
 - Issue #149**: [pyproject.toml](#) is a standard file
 - Issue #150**: [pyproject.toml](#) is a standard file
 - PyPi PEP Process**:
 - Summary:** An attempt to formalize the Python PyPi process. Not sound the entire thing is a single workflow that must be documented with the goal being to increase maintainability. The process only changes if the Python PyPi version requires it.
 - Issue #151**: [pyproject.toml](#) is a standard file
 - Issue #152**: [pyproject.toml](#) is a standard file

Pipfile

```
[ [source] ]
url = 'https://pypi.python.org/simple'
verify_ssl = true

[requires]
python_version = '2.7'

[packages]
requests = { extras = [ 'socks' ] }
Django = '>1.10'
pinax = { git = 'git://github.com/pinax/pinax.git', ref = '1.4', editable = true }

[dev-packages]
nose = '*'
```

Using TOML (Tom's Obvious, Minimal Language)

PyPI?

<https://pypi.python.org/pypi>

<https://pypi.org/>

⚠ This is a pre-production deployment of [Warehouse](#). Changes made here affect the production instance of PyPI (pypi.python.org).

Help us improve Python packaging - [Donate today!](#)

 Help Donate Login Register

Find, install and publish Python packages
with the Python Package Index

[Search](#)

Or browse projects.

111,816 Projects 737,204 Releases 940,329 Files 221,740 Users

Warehouse

<https://github.com/pypa/warehouse>



Recommended reading

- **Tool Recommendations:**

<https://packaging.python.org/current/>

- **Wheel vs. Egg:**

https://packaging.python.org/wheel_egg/

- **Getting Started With setuptools and setup.py:**

https://pythonhosted.org/an_example_pypi_project/setuptools.html

- **Sharing Your Labor of Love: PyPI Quick and Dirty:**

<https://hynek.me/articles/sharing-your-labor-of-love-pypi-quick-and-dirty/>

- **PyPA**

<https://www.pypa.io>

THANKS!

QUESTIONS?

<https://about.me/jakub.wasielak>