Python Packaging/ Affiliated Package Template Tutorial

Terminology

Terminology

- **“package”**: the biggest thing. E.g., *astropy*, *numpy*, *sunpy.* A directory with an “__init__.py”

Terminology

- **“package”**: the biggest thing. E.g., *astropy*, *numpy*, *sunpy.* A directory with an “__init__.py”
- **“module”:** a single “something.py” file - the module is “something”

Terminology

- **“package”**: the biggest thing. E.g., *astropy*, *numpy*, *sunpy.* A directory with an “__init__.py”
- **“module”:** a single “something.py” file - the module is “something”
- **“subpackage”**: a package within a package

Terminology

- **“package”**: the biggest thing. E.g., *astropy*, *numpy*, *sunpy.* A directory with an “__init__.py”
- **“module”:** a single “something.py” file - the module is “something”
- **“subpackage”**: a package within a package
- **“source directory/folder”**: the directory/folder with all of a codes “stuff”

Terminology

- **“package”**: the biggest thing. E.g., *astropy*, *numpy*, *sunpy.* A directory with an “__init__.py”
- **“module”:** a single “something.py” file - the module is “something”
- **“subpackage”**: a package within a package
- **“source directory/folder”**: the directory/folder with all of a codes “stuff”
- **“repository”/“repo”**: the source directory *in version control*

Terminology

- **“package”**: the biggest thing. E.g., *astropy*, *numpy*, *sunpy.* A directory with an “__init__.py”
- **“module”:** a single “something.py” file - the module is “something”
- **“subpackage”**: a package within a package
- **“source directory/folder”**: the directory/folder with all of a codes “stuff”
- **“repository”/“repo”**: the source directory *in version control*
- **“submodule”**: a git repo embedded in *another* git repo

Terminology

- **“package”**: the biggest thing. E.g., *astropy*, *numpy*, *sunpy.* A directory with an “__init__.py”
- **“module”:** a single “something.py” file - the module is “something”
- **“subpackage”**: a package within a package
- **“source directory/folder”**: the directory/folder with all of a codes “stuff”
- **“repository”/“repo”**: the source directory *in version control*
- **“submodule”**: a git repo embedded in *another* git repo
- **“astropy-helpers”:** a particular submodule that is used by the package template

Example Layout

Starting from “/Users/erik/src/mypackage”

README
LICENSE
setup.py
mypackage/__init__.py
mypackage/mymodule.py
mypackage/secondmodule.py
mypackage/subpackage/__init__.py
mypackage/subpackage/anothermodule.py

```
import mypackage
from mypackage import my module
from mypackage import secondmodule
from mypackage import subpackage
from mypackage.subpackage import anothermodule
```

The goal of packaging and installing is basically to make that work anywhere

The goal of packaging and installing is basically to make that work anywhere

And also manage releases and stuff…

So how do you actually do this?

So how do you actually do this?

- Astropy Affiliated Package Template
  - Comes pre-populated with all the layout
  - Includes the testing, documentation, and Cython building machinery.
    - All part of “astropy-helpers”
- (No commitment to be an affiliated package or even use Astropy)

So how do you actually do this?

- Astropy Affiliated Package Template
  - Comes pre-populated with all the layout
  - Includes the testing, documentation, and Cython building machinery.
    - All part of “astropy-helpers”
  - (No commitment to be an affiliated package or even use Astropy)
- Roll your own
  - https://python-packaging-user-guide.readthedocs.org

OK, I’ve done all that, and my code and docs are awesome. What now?

OK, I’ve done all that, and my code and docs are awesome. What now?

OK, I’ve done all that, and my code and docs are awesome. What now?

- You should make a release. Don’t let the word scare you.

OK, I've done all that, and my code and docs are awesome. What now?

- You should make a release. Don't let the word scare you.
- Decide on a version numbering scheme.
  - 0.1 -> 0.2 -> 0.3 -> 1.0 -> 1.1
    - You just release when ready
  - 0.1 -> 0.1.1 -> 0.1.2 -> 0.2 -> 0.2.1
    - A lot more overhead because multiple branches needed, but you can release fixes without new features.

OK, I’ve done all that, and my code and docs are awesome. What now?

OK, I’ve done all that, and my code and docs are awesome. What now?

OK, I’ve done all that, and my code and docs are awesome. What now?

- Put your code on PyPI

OK, I've done all that, and my code and docs are awesome. What now?

- Put your code on PyPI
    - If you're an affiliated package: http://astropy.readthedocs.org/en/latest/development/affiliated-packages.html#releasing-an-affiliated-package

OK, I've done all that, and my code and docs are awesome. What now?

- Put your code on PyPI
  - If you're an affiliated package: http://astropy.readthedocs.org/en/latest/development/affiliated-packages.html#releasing-an-affiliated-package
  - In the end the key is: python setup.py register build sdist upload

OK, I've done all that, and my code and docs are awesome. What now?

- Put your code on PyPI
  - If you're an affiliated package: http://astropy.readthedocs.org/en/latest/development/affiliated-packages.html#releasing-an-affiliated-package
    - In the end the key is: python setup.py register build sdist upload
- Start maintaining a changelog!

Yay packaging! What now?

Yay packaging! What now?

Yay packaging! What now?

- Think about how you want to accept feedback
- Github PRs

Yay packaging! What now?

- Think about how you want to accept feedback
    - Github PRs
- Maybe get on conda?
    - If you’re Astropy Affiliated, Matt/@mwcraig has got your back

Let’s have a look at the package template…