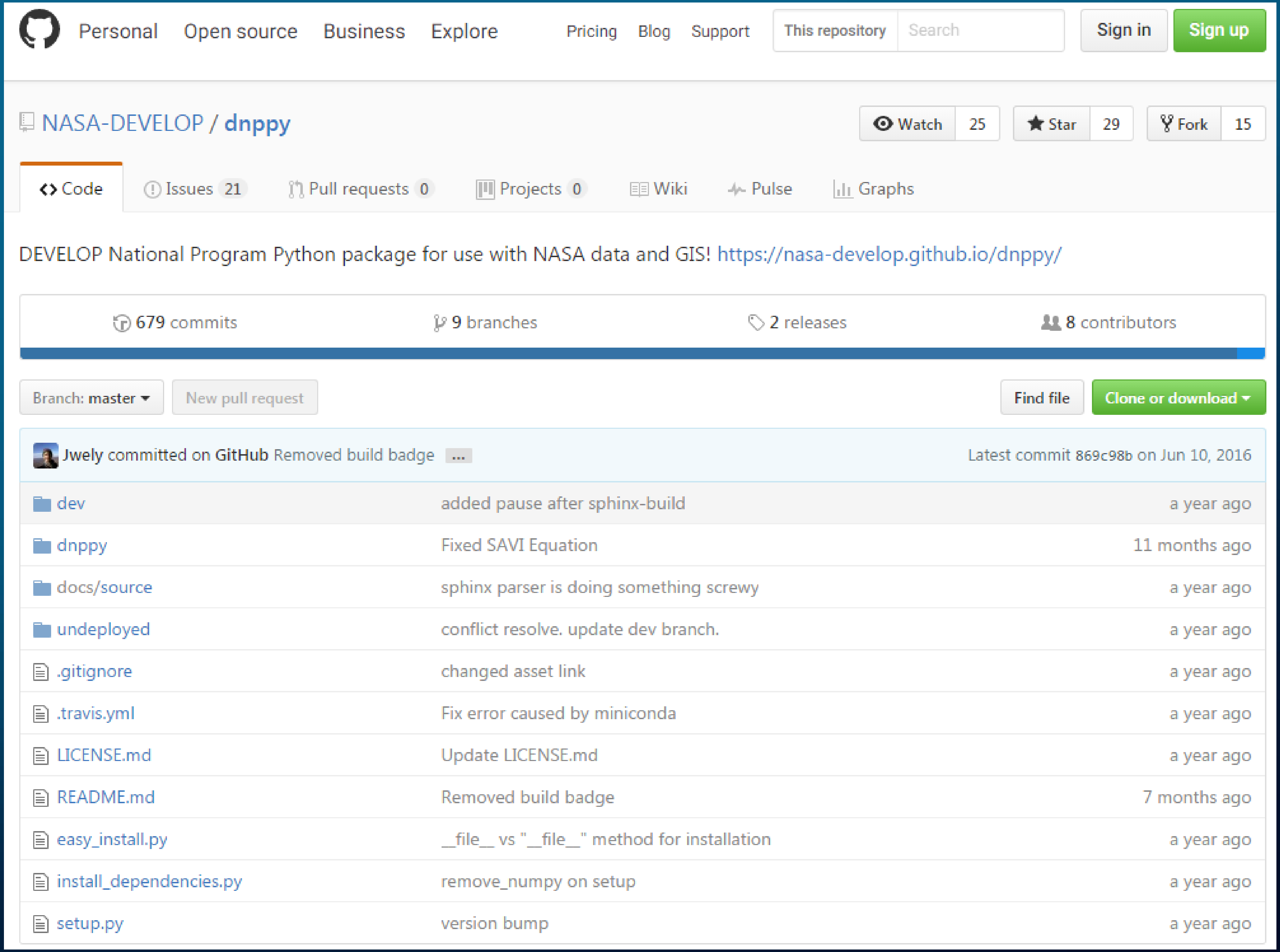
INSTALLING DNPPY

A STEP BY STEP GUIDE

Find NASA DEVELOP dnppy on GitHub

https://github.com/NASA-DEVELOP/dnppy



NASA-DEVELOP / dnppy

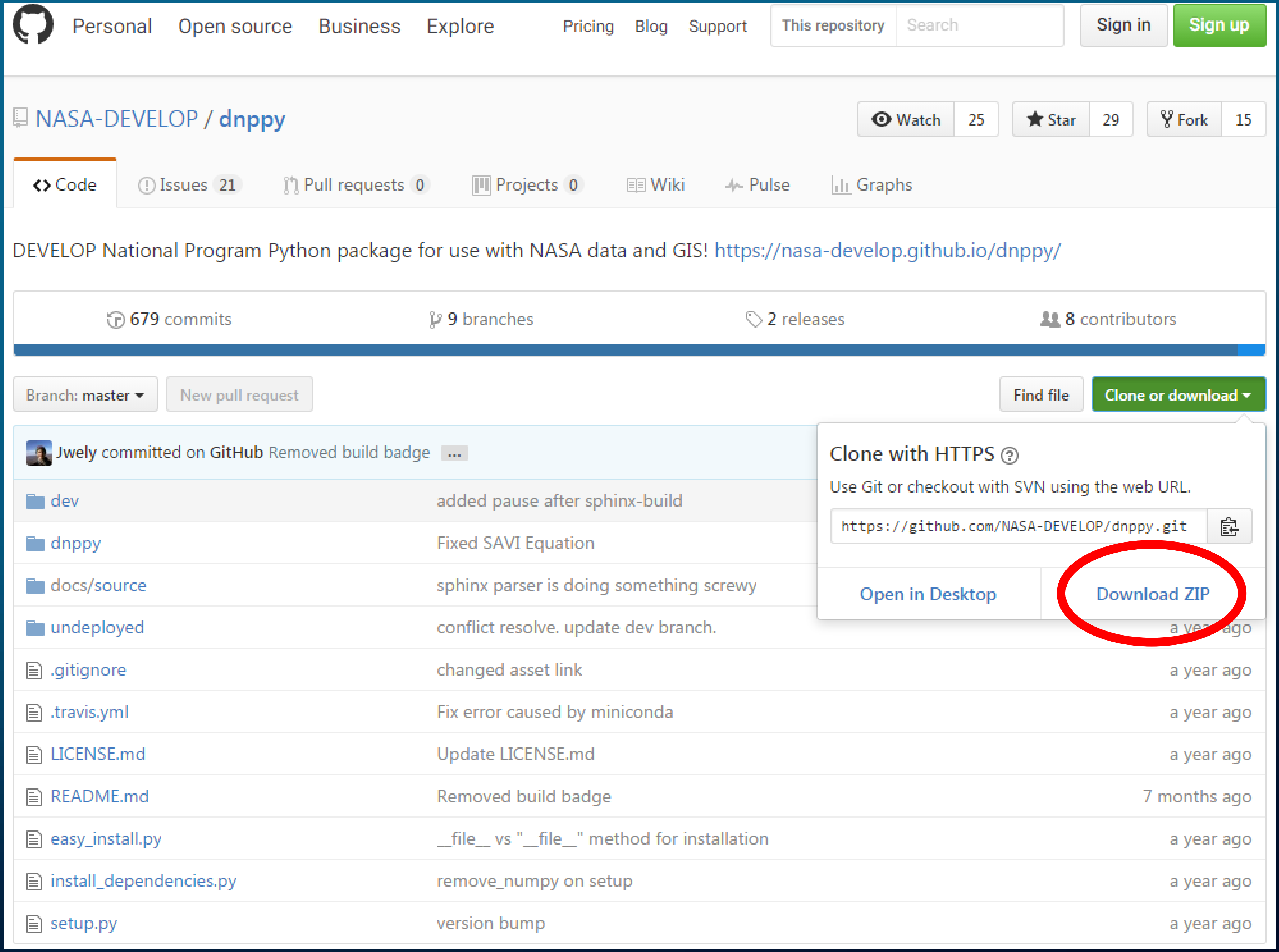
DEVELOP National Program Python package for use with NASA data and GIS! https://nasa-develop.github.io/dnppy/

679 commits · 9 branches · 2 releases · 8 contributors

Jwely committed on GitHub Removed build badge ... Latest commit 869c98b on Jun 10, 2016

dev	added pause after sphinx-build	a year ago
dnppy	Fixed SAVI Equation	11 months ago
docs/source	sphinx parser is doing something screwy	a year ago
undeployed	conflict resolve. update dev branch.	a year ago
.gitignore	changed asset link	a year ago
.travis.yml	Fix error caused by miniconda	a year ago
LICENSE.md	Update LICENSE.md	a year ago
README.md	Removed build badge	7 months ago
easy_install.py	__file__ vs "__file__" method for installation	a year ago
install_dependencies.py	remove_numpy on setup	a year ago
setup.py	version bump	a year ago

- Click on “Download ZIP” to download dnppy-master.zip



NASA-DEVELOP / **dnppy**

DEVELOP National Program Python package for use with NASA data and GIS! https://nasa-develop.github.io/dnppy/

679 commits 9 branches 2 releases 8 contributors

Jwely committed on GitHub Removed build badge ...

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

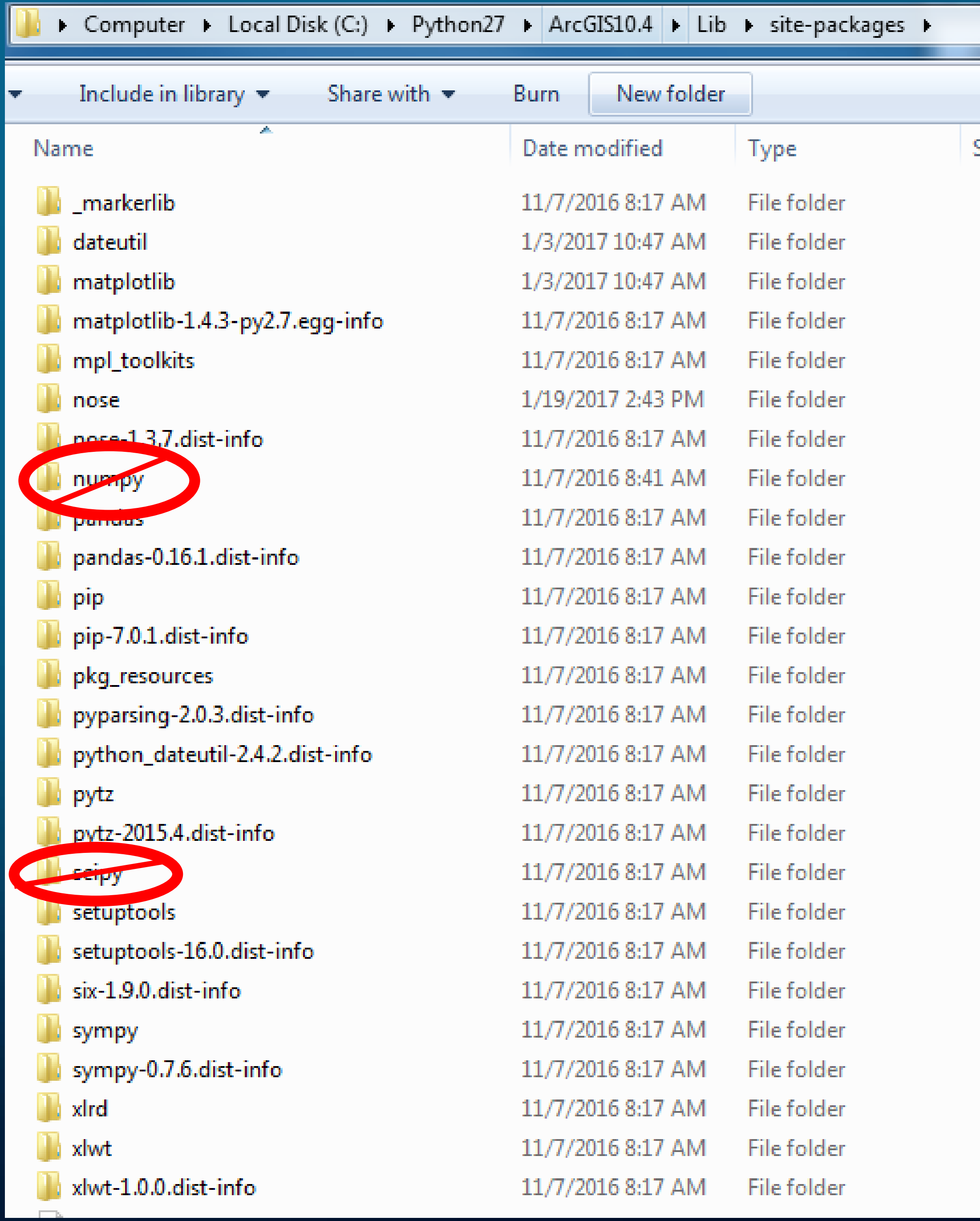
```
https://github.com/NASA-DEVELOP/dnppy.git
```

dev	added pause after sphinx-build	
dnppy	Fixed SAVI Equation	
docs/source	sphinx parser is doing something screwy	
undeployed	conflict resolve. update dev branch.	a year ago
.gitignore	changed asset link	a year ago
.travis.yml	Fix error caused by miniconda	a year ago
LICENSE.md	Update LICENSE.md	a year ago
README.md	Removed build badge	7 months ago
easy_install.py	__file__ vs "__file__" method for installation	a year ago
install_dependencies.py	remove_numpy on setup	a year ago
setup.py	version bump	a year ago

Before Installing dnppy…

- Eliminate redundant files in your local drive where ArcGIS is saved
  - C:\Python27\ArcGIS10.4\Lib\site-packages

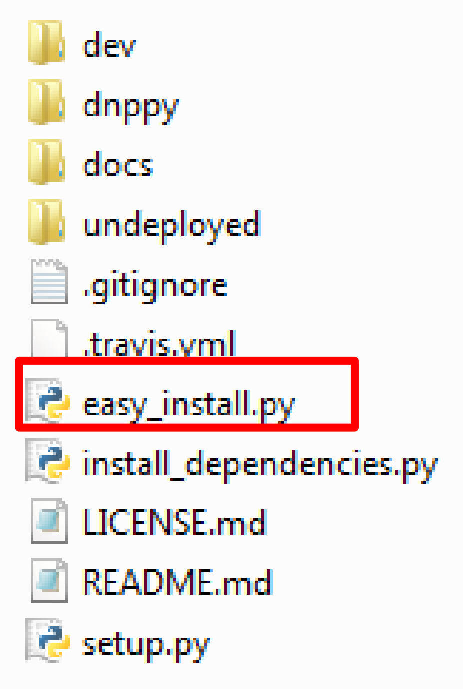
- Delete these files if found in the site-packages folder:
  - Numpy
  - Dnppy
  - dnppy-1.15.3b2.dist-info
  - Scipy
  - scipy-0.15.1.dist-info


Computer ▸ Local Disk (C:) ▸ Python27 ▸ ArcGIS10.4 ▸ Lib ▸ site-packages ▸
Include in library ▾
Share with ▾
Burn
New folder

Name	Date modified	Type
_markerlib	11/7/2016 8:17 AM	File folder
dateutil	1/3/2017 10:47 AM	File folder
matplotlib	1/3/2017 10:47 AM	File folder
matplotlib-1.4.3-py2.7.egg-info	11/7/2016 8:17 AM	File folder
mpl_toolkits	11/7/2016 8:17 AM	File folder
nose	1/19/2017 2:43 PM	File folder
nose-1.3.7.dist-info	11/7/2016 8:17 AM	File folder
numpy	11/7/2016 8:41 AM	File folder
pandas	11/7/2016 8:17 AM	File folder
pandas-0.16.1.dist-info	11/7/2016 8:17 AM	File folder
pip	11/7/2016 8:17 AM	File folder
pip-7.0.1.dist-info	11/7/2016 8:17 AM	File folder
pkg_resources	11/7/2016 8:17 AM	File folder
pyparsing-2.0.3.dist-info	11/7/2016 8:17 AM	File folder
python_dateutil-2.4.2.dist-info	11/7/2016 8:17 AM	File folder
pytz	11/7/2016 8:17 AM	File folder
pytz-2015.4.dist-info	11/7/2016 8:17 AM	File folder
scipy	11/7/2016 8:17 AM	File folder
setuptools	11/7/2016 8:17 AM	File folder
setuptools-16.0.dist-info	11/7/2016 8:17 AM	File folder
six-1.9.0.dist-info	11/7/2016 8:17 AM	File folder
sympy	11/7/2016 8:17 AM	File folder
sympy-0.7.6.dist-info	11/7/2016 8:17 AM	File folder
xlrd	11/7/2016 8:17 AM	File folder
xlwt	11/7/2016 8:17 AM	File folder
xlwt-1.0.0.dist-info	11/7/2016 8:17 AM	File folder

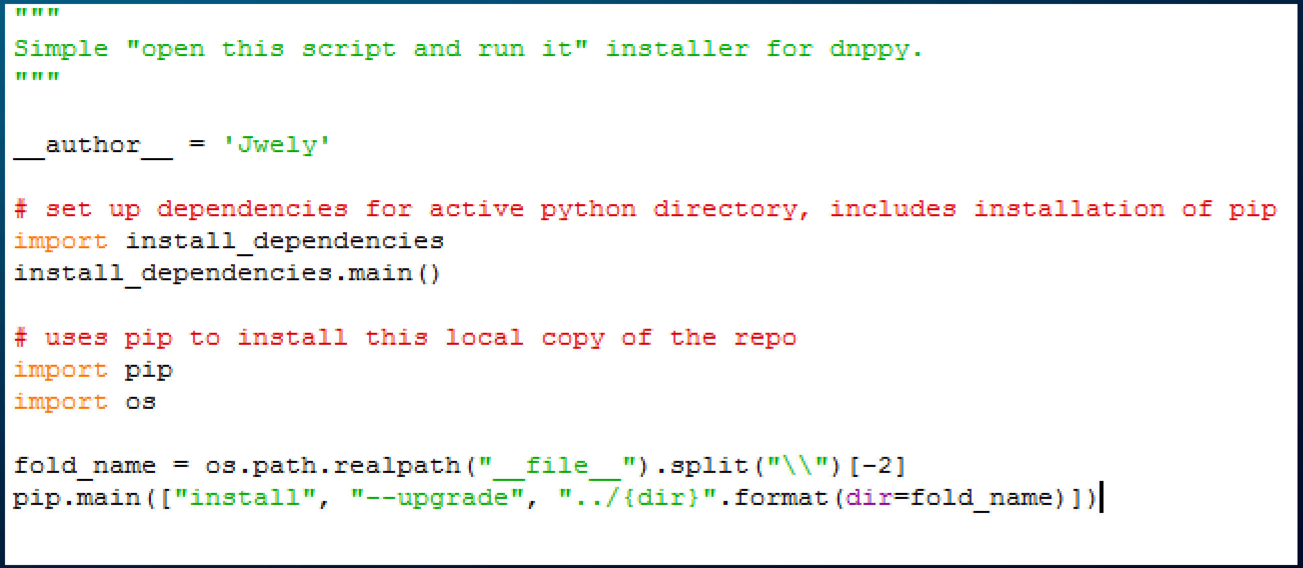
Installing dnppy

- Once the contents in the zip folder are extracted, run “easy_install.py” in an IDLE environment (Python GUI)



dev
dnppy
docs
undeployed
.gitignore
.travis.yml
easy_install.py
install_dependencies.py
LICENSE.md
README.md
setup.py

Should look like this

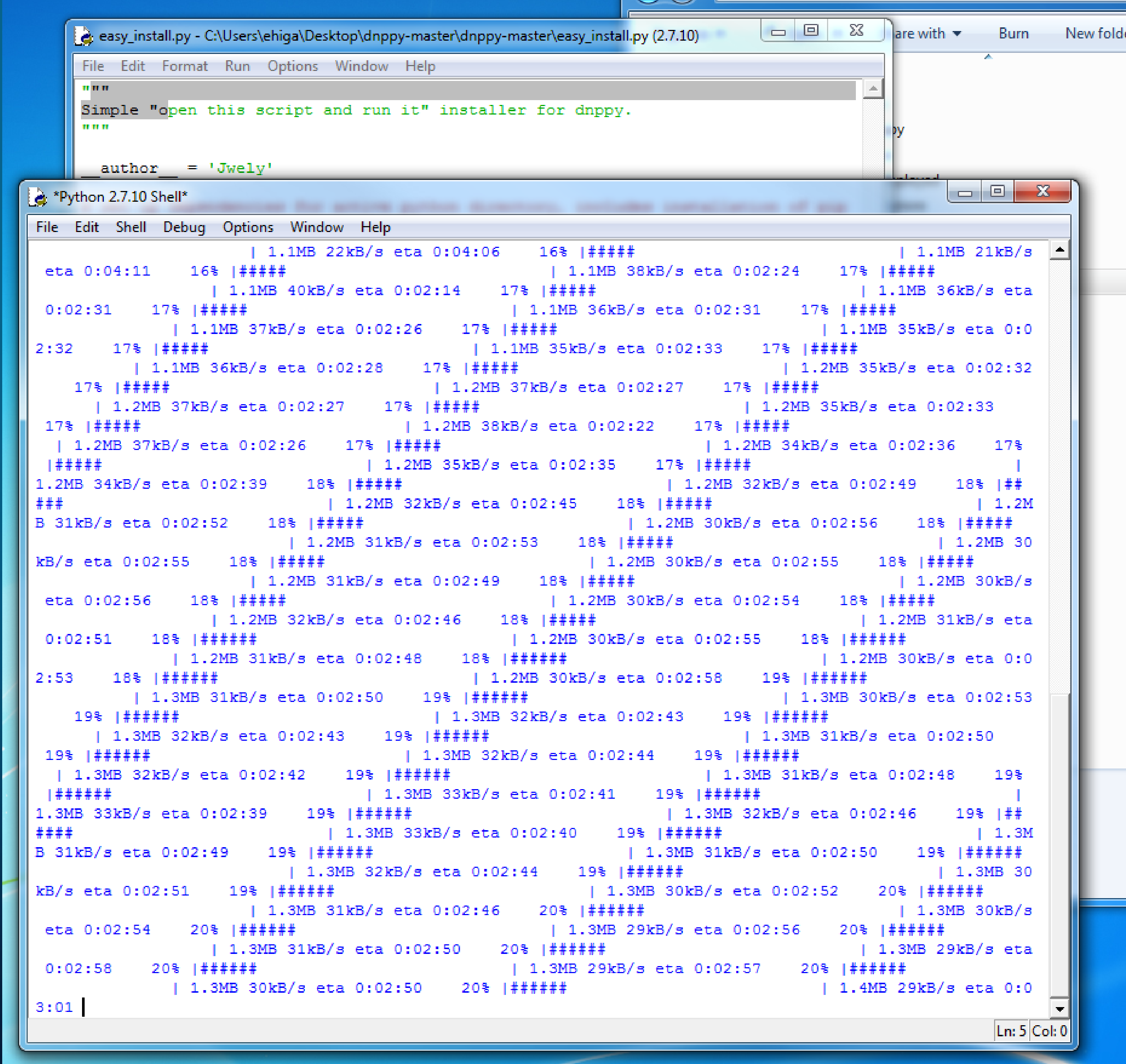


```
"""
Simple "open this script and run it" installer for dnppy.
"""

__author__ = 'Jwely'

# set up dependencies for active python directory, includes installation of pip
import install_dependencies
install_dependencies.main()

# uses pip to install this local copy of the repo
import pip
import os

fold_name = os.path.realpath("__file__").split("\\")[-2]
pip.main(["install", "--upgrade", "../{dir}".format(dir=fold_name)])
```

Installing dnppy

- This will begin the process!
- Other dependencies will be installed on your computer at this time
    - It may take a while based on your connection speed

It will look something similar to this



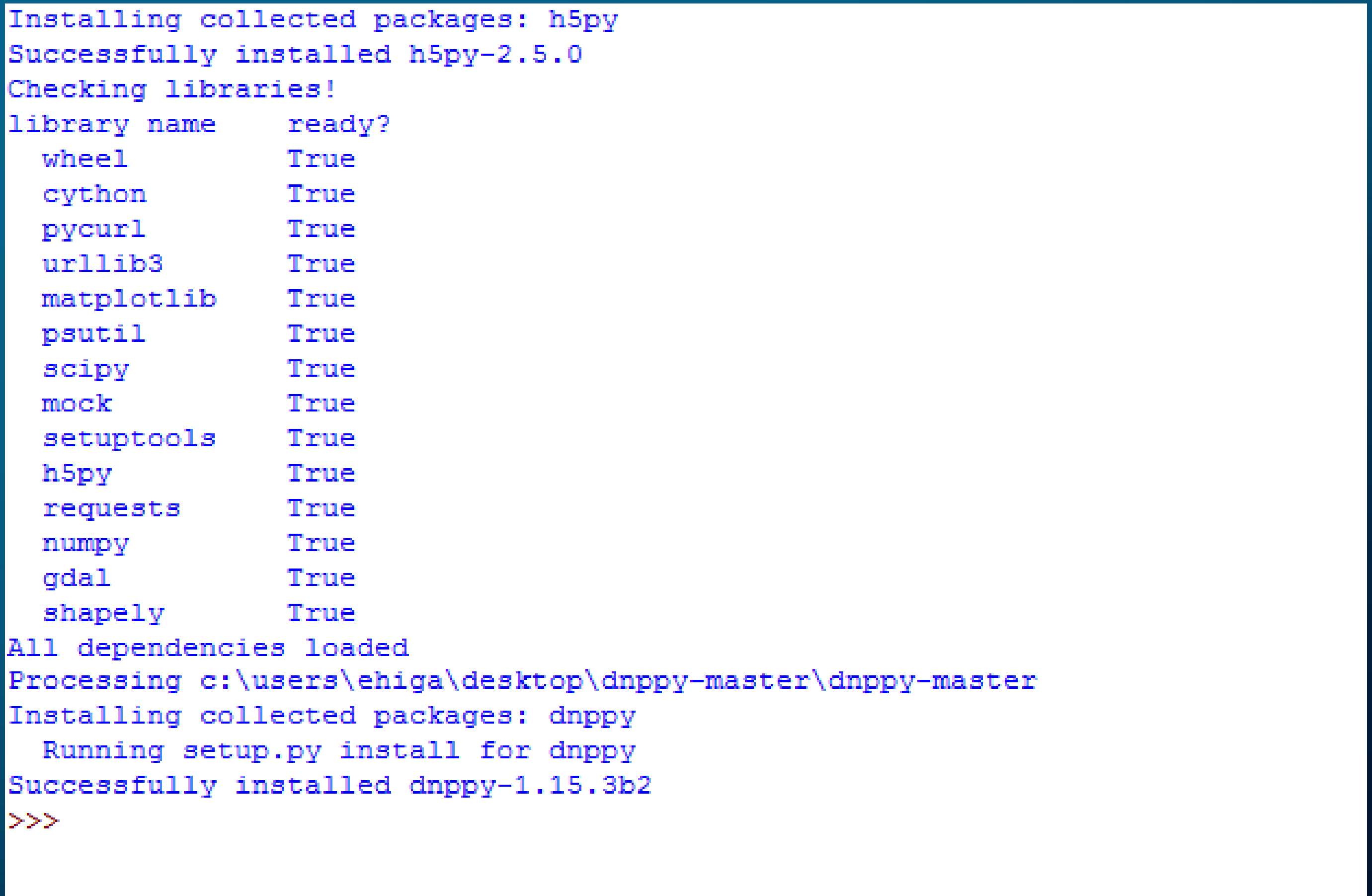
easy_install.py - C:\Users\ehiga\Desktop\dnppy-master\dnppy-master\easy_install.py (2.7.10)

```
"""
Simple "open this script and run it" installer for dnppy.
"""

__author__ = 'Jwely'
```

```
                      | 1.1MB 22kB/s eta 0:04:06    16% |#####                           | 1.1MB 21kB/s
 eta 0:04:11    16% |#####                           | 1.1MB 38kB/s eta 0:02:24    17% |#####
                  | 1.1MB 40kB/s eta 0:02:14    17% |#####                           | 1.1MB 36kB/s eta
 0:02:31    17% |#####                           | 1.1MB 36kB/s eta 0:02:31    17% |#####
              | 1.1MB 37kB/s eta 0:02:26    17% |#####                           | 1.1MB 35kB/s eta 0:0
2:32    17% |#####                           | 1.1MB 35kB/s eta 0:02:33    17% |#####
          | 1.1MB 36kB/s eta 0:02:28    17% |#####                           | 1.2MB 35kB/s eta 0:02:32
    17% |#####                           | 1.2MB 37kB/s eta 0:02:27    17% |#####
      | 1.2MB 37kB/s eta 0:02:27    17% |#####                           | 1.2MB 35kB/s eta 0:02:33
 17% |#####                           | 1.2MB 38kB/s eta 0:02:22    17% |#####
 | 1.2MB 37kB/s eta 0:02:26    17% |#####                           | 1.2MB 34kB/s eta 0:02:36    17%
 |#####                           | 1.2MB 35kB/s eta 0:02:35    17% |#####                           |
1.2MB 34kB/s eta 0:02:39    18% |#####                           | 1.2MB 32kB/s eta 0:02:49    18% |##
###                           | 1.2MB 32kB/s eta 0:02:45    18% |#####                           | 1.2M
B 31kB/s eta 0:02:52    18% |#####                           | 1.2MB 30kB/s eta 0:02:56    18% |#####
                          | 1.2MB 31kB/s eta 0:02:53    18% |#####                           | 1.2MB 30
kB/s eta 0:02:55    18% |#####                           | 1.2MB 30kB/s eta 0:02:55    18% |#####
                      | 1.2MB 31kB/s eta 0:02:49    18% |#####                           | 1.2MB 30kB/s
 eta 0:02:56    18% |#####                           | 1.2MB 30kB/s eta 0:02:54    18% |#####
                  | 1.2MB 32kB/s eta 0:02:46    18% |#####                           | 1.2MB 31kB/s eta
 0:02:51    18% |######                          | 1.2MB 30kB/s eta 0:02:55    18% |######
              | 1.2MB 31kB/s eta 0:02:48    18% |######                          | 1.2MB 30kB/s eta 0:0
2:53    18% |######                          | 1.2MB 30kB/s eta 0:02:58    19% |######
          | 1.3MB 31kB/s eta 0:02:50    19% |######                          | 1.3MB 30kB/s eta 0:02:53
    19% |######                          | 1.3MB 32kB/s eta 0:02:43    19% |######
      | 1.3MB 32kB/s eta 0:02:43    19% |######                          | 1.3MB 31kB/s eta 0:02:50
 19% |######                          | 1.3MB 32kB/s eta 0:02:44    19% |######
 | 1.3MB 32kB/s eta 0:02:42    19% |######                          | 1.3MB 31kB/s eta 0:02:48    19%
 |######                          | 1.3MB 33kB/s eta 0:02:41    19% |######                          |
1.3MB 33kB/s eta 0:02:39    19% |######                          | 1.3MB 32kB/s eta 0:02:46    19% |##
####                          | 1.3MB 33kB/s eta 0:02:40    19% |######                          | 1.3M
B 31kB/s eta 0:02:49    19% |######                          | 1.3MB 31kB/s eta 0:02:50    19% |######
                          | 1.3MB 32kB/s eta 0:02:44    19% |######                          | 1.3MB 30
kB/s eta 0:02:51    19% |######                          | 1.3MB 30kB/s eta 0:02:52    20% |######
                      | 1.3MB 31kB/s eta 0:02:46    20% |######                          | 1.3MB 30kB/s
 eta 0:02:54    20% |######                          | 1.3MB 29kB/s eta 0:02:56    20% |######
                  | 1.3MB 31kB/s eta 0:02:50    20% |######                          | 1.3MB 29kB/s eta
 0:02:58    20% |######                          | 1.3MB 29kB/s eta 0:02:57    20% |######
              | 1.3MB 30kB/s eta 0:02:50    20% |######                          | 1.4MB 29kB/s eta 0:0
3:01 
```

Installing dnppy

- Success!
- You now have access to dnppy, gdal, scipy and more

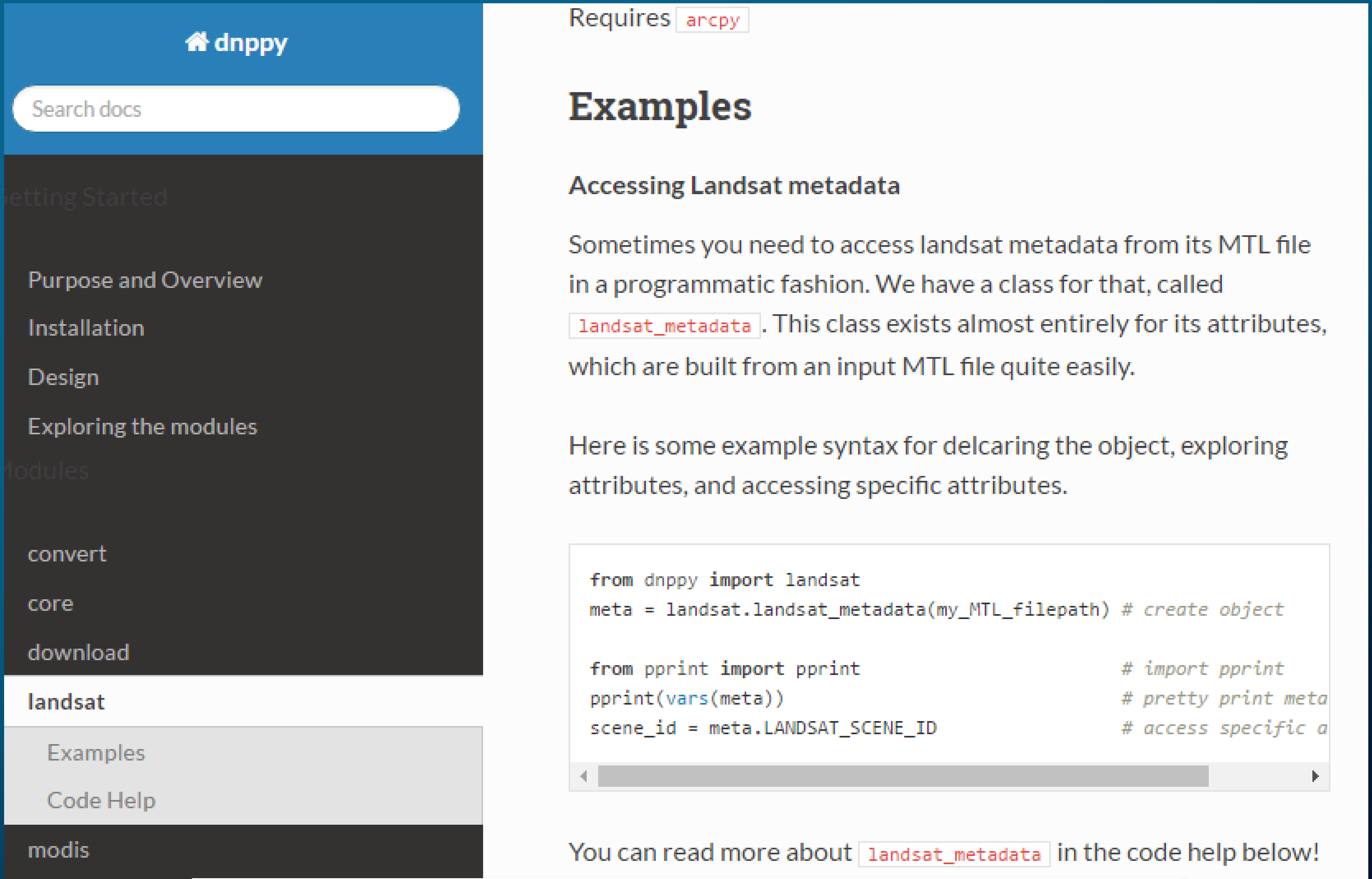


```
Installing collected packages: h5py
Successfully installed h5py-2.5.0
Checking libraries!
library name    ready?
  wheel         True
  cython        True
  pycurl        True
  urllib3       True
  matplotlib    True
  psutil        True
  scipy         True
  mock          True
  setuptools    True
  h5py          True
  requests      True
  numpy         True
  gdal          True
  shapely       True
All dependencies loaded
Processing c:\users\ehiga\desktop\dnppy-master\dnppy-master
Installing collected packages: dnppy
  Running setup.py install for dnppy
Successfully installed dnppy-1.15.3b2
>>>
```

Note: if the dependencies were not successfully installed, try running “install_dependencies.py” first, then rerun “easy_install.py”

Using dnppy

- Explore dnppy modules at https://nasa-develop.github.io/dnppy/
- Explains parameters needed to run each module
- Provides examples of code



Requires `arcpy`

Examples

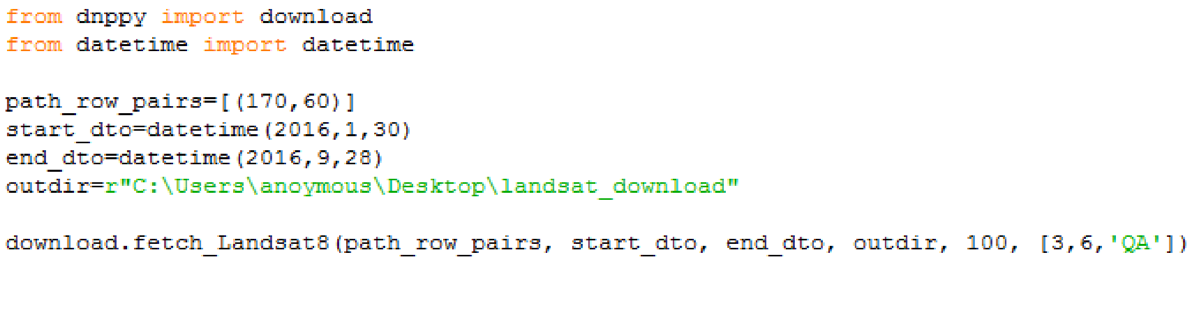
Accessing Landsat metadata

Sometimes you need to access landsat metadata from its MTL file in a programmatic fashion. We have a class for that, called `landsat_metadata`. This class exists almost entirely for its attributes, which are built from an input MTL file quite easily.

Here is some example syntax for delcaring the object, exploring attributes, and accessing specific attributes.

```
from dnppy import landsat
meta = landsat.landsat_metadata(my_MTL_filepath) # create object

from pprint import pprint                        # import pprint
pprint(vars(meta))                               # pretty print meta
scene_id = meta.LANDSAT_SCENE_ID                 # access specific a
```

You can read more about `landsat_metadata` in the code help below!



```
from dnppy import download
from datetime import datetime

path_row_pairs=[(170,60)]
start_dto=datetime(2016,1,30)
end_dto=datetime(2016,9,28)
outdir=r"C:\Users\anoymous\Desktop\landsat_download"

download.fetch_Landsat8(path_row_pairs, start_dto, end_dto, outdir, 100, [3,6,'QA'])
```

Example code to fetch Landsat data →