2nd ASTERICS-OBELICS International School

4-8 June 2018, Annecy, France.

H2020-Astronomy ESFRI and Research Infrastructure Cluster
(Grant Agreement number: 653477).

Python / Jupyter / IDE

Tammo Jan Dijkema & Zheng Meyer

ASTERICS – Obelics workshop Annecy

June 2018

Some general advice

- Make source code publicly accessible from day one
- Make software easy to discover by providing software metadata via a popular community registry, e.g. ASCL (Astronomical Source Code Library)
- Adopt a licence and comply with the licence of third-party dependencies
- Define clear and transparent contribution, governance and communication processes
    - Tell people how you want to be cited

Jiménez RC, Kuzak M, Alhamdoosh M *et al.* Four simple recommendations to encourage best practices in research software [version 1; referees: 3 approved]. *F1000Research* 2017, **6**:876 (doi: 10.12688/f1000research.11407.1)

ASTRON
Netherlands Institute for Radio Astronomy

Python and paths

- Python is an interpreted language: running a program means running `python` with that program as argument
- There can be multiple `python`'s on your system:

```
which python
```

```
> import sys; sys.executable
```

Package managers

- Installing python packages is easy, and can be done in many ways:
  - Download package, manually copy to right path (please don’t)
  - Download package, python setup.py install (please don’t)
  - `pip install <package>` (please don’t this week)
  - `conda install <package>`
- This week, we will use conda to install packages
- Check where package got installed:

```
> import numpy; numpy.__file__
```

ASTRON
Netherlands Institute for Radio Astronomy

Isolated environments

- Isolate your work from arbitrary changes by using an isolated environment (reproducibility)
- My advice: use a package manager specifically for python
  - Do not use system python
  - Do not use homebrew python
- Two main ways to isolate environments:
  - Virtualenv
  - Conda
- This week, we will use conda

Interactive python

- python
- ipython
- jupyter notebook
- jupyter lab

Developing python scripts

- Any text editor:
  - vim, emacs, notepad.exe, gedit, jupyter lab
- IDE:
  - PyCharm, Eclipse, atom, …

Plan for today

- Check everyone’s environments
- Explore Jupyter, Jupyter notebook, try some new python features
- Learn about good code practice
- Explore some (LOFAR!) data in a jupyter notebook
- Write a script in PyCharm to do something useful to this dataset
    - Follow PEP8 coding guidelines
    - Document code
    - Test code (with high coverage)
- Put this script in an installable package

Demo: PyCharm

- Docstrings and conventions
- Python and data types
- `if __name__=='__main__'`
- Python and data types
- Debugging from pycharm

Demo Jupyter Lab

- Ipywidgets
- Fancy maps
- Terminal from jupyter Lab

- Lab vs Notebook

Acknowledgement

H2020-Astronomy ESFRI and Research Infrastructure Cluster (Grant Agreement number: 653477).