A brief and biased introduction to **python**

Ch. Morisset, IA-UNAM

History

- 1991 : Guido van Rossum started to worked on Python
- 1996 : Numerical python, precursor of Numpy.
- 2003 : Matplotlib, MATLAB-like interface.

How to get python ?

- Most of the OS have some python installed, but not the scientific libraries.
- The best is to use a package that includes everything you need (and more…).
- **Anaconda** is one of them. It includes the former STSCI Ureka project, pyRAF is an option.
- It leaves in a single directory, easy to remove.

Which python version ?

- There are 2 main versions of python currently used :
  - Python 2 : 2.7 latest subversion, maintained till 2020
  - Python 3 : 3.6 most recent.
- PB : no backward compatibility…
- Main painful difference : strings.
- Best to learn python 3, all the major libraries are py3-compliant.

Multiple versions of python

- Using conda, one can create different « environments » : each one being a full python installation.
- Creation of a new environment :
  - `conda create --name py3k python=3 astropy`
- Switching is as easy as (from bash only) :
  - `source activate py3k`

Growing python

- The amount of libraries is huge.
  - Anaconda installing facility:
    - `conda install astropy`
  - Pip installing facility (from PyPI - the Python Package Index) :
    - `pip install pyneb`
  - From github (e.g. devel version):
    - `pip install -U git+https://github.com/Morisset/PyNeb_devel.git`

Running python

- Interactive : ipython
  - Completion
  - History
  - Magic commands
- Running scripts
  - %run test # executing test.py
  - import library
  - test.py ; from Linux prompt

Notebooks

- Esthetic way of gathering script and results, including tables and figures.
- Executable
- Python lectures, PyNeb manual, pyCloudy sample scripts and Cloudy School examples.
- https://github.com/Morisset/Python-lectures-Notebooks

Indispensable libraries

- Numpy : vectorisation
- Matplotlib : plotting facilities
- Scipy : scientific tools
- Astropy : astronomical tools
- Pandas : tables

Object Oriented Programing

- Very powerful way of programing.
- Object = smart container with variables (attributes) and functions (methods)
- Class definition and object instantiations.
- All objects of the same class share the same methods and properties, but can have different values for the attributes.
- As many objects from the same class as one want.

Help

- The python community is very active.
- Somebody already had your problem, and somebody already solved it ; just find the words to describe it
- StackOverflow is THE site where to look for a solution.