Astronomical Data Analysis with Python

Lecture 1

Yogesh Wadadekar

NCRA-TIFR

July-August 2010

What this course is about

- prerequisites

What this course is about

- prerequisites
- URL: http://www.ncra.tifr.res.in/∼yogesh/python_course_2010

What this course is about

- prerequisites
- URL: http://www.ncra.tifr.res.in/∼yogesh/python_course_2010
- please interrupt and ask questions at any time.

What this course is about

- prerequisites
- URL: http://www.ncra.tifr.res.in/∼yogesh/python_course_2010
- please interrupt and ask questions at any time.
- programming background of the audience?

Why Python?

- A powerful, general purpose programming language, yet easy to learn. Strong, but optional, Object Oriented Programming support
- Very large user and developer community, very extensive and broad library base
- Very extensible with C, C++, or Fortran, portable distribution mechanisms available
- Free; non-restrictive license; open source
- fast becoming the standard scripting language for astronomy
- very powerful array processing capabilities (numpy)
- extensive documentation - Many books and on-line documentation resources available (for the language and its libraries and modules)

Why python?

- Plotting (matplotlib)
  - framework (matplotlib) more extensible and general
  - Better font support and portability (only one way to do it too!)
  - Usable within many windowing framework (GTK, Tk, WX, Qt...) backends
  - Standard plotting functionality independent of framework used
  - plots are embeddable within other GUIs
  - more powerful image handling (multiple simultaneous LUTS, optional resampling/rescaling, alpha blending, etc)
- Support for many widget systems for GUI development
- superb database interfaces to all popular databases.

Disadvantages of Python

- More items to install separately (eased by yum and apt-get)
- Not as well accepted in astronomical community (but support clearly growing)
- Scientic and numerical libraries not as mature; not as deep in astronomical libraries and utilities
- Array indexing convention backwards
- Small array performance slower
- No standard GUI run/debug tool e.g. like Eclipse for Java
- Support for many widget systems (angst regarding which to choose)

Comparison with FORTRAN/C/C++

- huge amount of legacy code
- compilers highly optimized for excellent runtime performance

but...

- FORTRAN not really general purpose
- relatively primitive datatypes
- manual memory management
- slow edit/compile/test cycle

Comparison with IDL/Matlab

- Extremely popular
- Interactive, great visualization, good libraries

but...

- Not really general purpose
- Vendor lock-in
- fairly expensive, source code of core libraries not changeable.

Comparison with Perl

- quite popular locally - lots of code written
- shares many of Python’s strengths

but...

- Write Once, Read Never
- just think and type, that’s perl
- I think that “Just think and type, that’s python” is more appropriate.

Python usage in optical astronomy

- STScI PyRAF (IRAF) + additional Python only routines
- ESO PyMIDAS (MIDAS)
- STScI PyFITS (access to FITS files)
- Astro-WISE (widefield imaging system)
- Pyephem - solar system ephemeris
- LSST will use Python/C++

Python usage in Radio astronomy

- CasaPy (Casa) - AIPS++, default system for EVLA and ALMA data analysis.
- ParselTongue - call AIPS tasks from Python
- PYGILDAS (GILDAS) - IRAM data analysis software ported to Python
- BoA (Bolometer Analysis Package) for LABOCA on APEX and other bolometers
- APECS (APEX control software)
- KAT-7 CMS is in Python
- Presto - pulsar search and analysis suite; most recent routines in Python

Basic methodology

- prepare a full working prototype of any application in Python, without worrrying about speed

Basic methodology

- prepare a full working prototype of any application in Python, without worrrying about speed
- *profile* the prototype

Basic methodology

- prepare a full working prototype of any application in Python, without worrrying about speed
- *profile* the prototype
- identify the slow parts of the Python code and rewrite them in a compiled language like C/C++ for maximum speed.

Basic methodology

- prepare a full working prototype of any application in Python, without worrrying about speed
- *profile* the prototype
- identify the slow parts of the Python code and rewrite them in a compiled language like C/C++ for maximum speed.
- write wrappers to interface the new compiled code with the Python prototype. Repeat until required optimisation is achieved.

Basic methodology

- prepare a full working prototype of any application in Python, without worrrying about speed
- *profile* the prototype
- identify the slow parts of the Python code and rewrite them in a compiled language like C/C++ for maximum speed.
- write wrappers to interface the new compiled code with the Python prototype. Repeat until required optimisation is achieved.
- If you are talking to devices - write a simulator first.

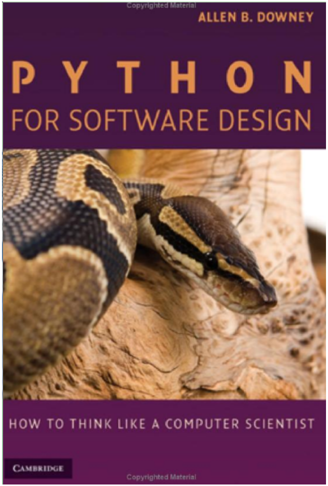
Python for non-astronomers

- full featured, high level programming language
- very easy to learn -National Mission on Education through ICT is sponsoring a large program (crores of rupees) to develop computer education materials in Python for school and college students (http://fossee.in/). Also, OLPC's Sugar environment is written in Python.
- powerful text processing capabilities - many sysadmins are adopting it.
- powerful interfaces to almost any database
- web-friendly language - many frameworks available Django, Zope, CherryPy, Trac for website CMS, wikis etc.
- CERN's INDICO conference management system is all Python.
- good plotting capabilities (see the latest casapy capabilities based on matplotlib)

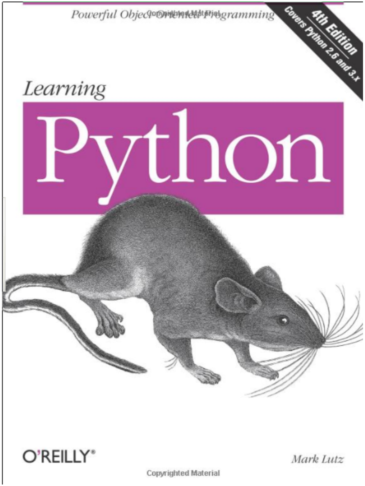
Python extensively used by Google

- The Google build system is written in python. All of Google's corporate code is checked into a repository and the dependency and building of this code is managed by python.
- Packaging. Google has an internal packaging format like RPM. These packages are created using python.
- Binary Data Pusher. This is the area where Alex Martelli is working, on optimizing pushing bits between thousands of servers
- Production servers. All monitoring, restarting and data collection functionality is done with python
- Reporting. Logs are analyzed and reports are generated using Python.
- A few services including code.google.com and google groups run on Python. Most other front ends are in C++ (google.com) and Java (gmail). All web services are built on top of a optimized http server wrapped with SWIG.

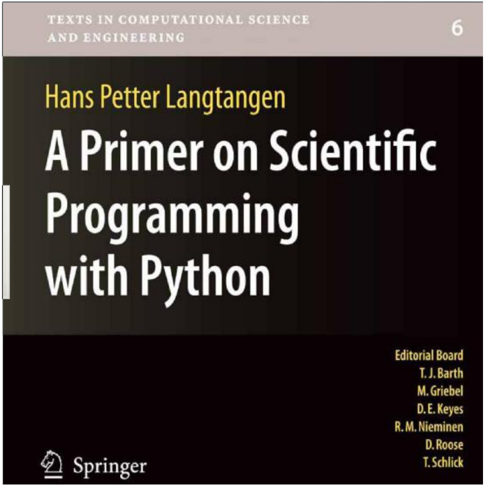
If you are an absolute beginner to programming



Standard book for beginners - Lutz & Ascher



A new primer by Langtangen



TEXTS IN COMPUTATIONAL SCIENCE AND ENGINEERING 6

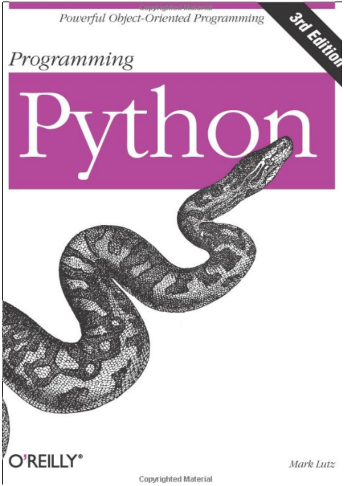
Hans Petter Langtangen

A Primer on Scientific Programming with Python

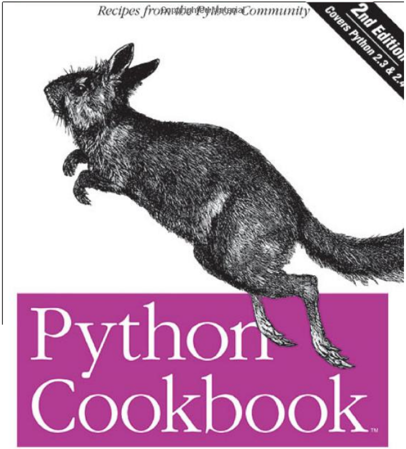
Editorial Board
T. J. Barth
M. Griebel
D. E. Keyes
R. M. Nieminen
D. Roose
T. Schlick

Springer

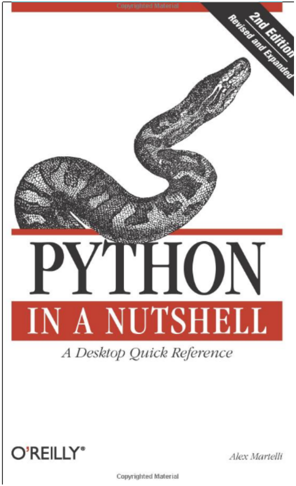
Book for intermediate level



Python Cookbook



A Python quick reference by Martelli



Copyrighted Material

2nd Edition

Revised and Expanded

PYTHON

IN A NUTSHELL

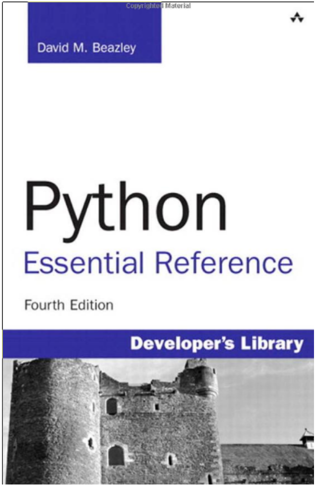
A Desktop Quick Reference

O'REILLY®

Alex Martelli

Copyrighted Material

Another reference book by Beazley



David M. Beazley

Python

Essential Reference

Fourth Edition

Developer's Library

Python for Scientific Computing by Langtangen

Texts in Computational Science and Engineering

3

Hans Petter Langtangen

Python Scripting for Computational Science

Third Edition

Springer

Guide to Numpy by Travis Oliphant, now in public domain

http://www.tramy.us/guidetoscipy.html

Plenty of other books are available

Search for “Python programming” on amazon.com throws up 429 items.
Having said this, I have not bought a Python book yet. Almost all the books referred to above are now in the NCRA and GMRT libraries. Many of them should also be available in the IUCAA library. Also, a lot of documentation is online.

Online Material

- www.python.org

Online Material

- www.python.org
- Start with the Python tutorial - http://docs.python.org/tutorial/ we will cover all of it and more in this course.

Online Material

- www.python.org
- Start with the Python tutorial - http://docs.python.org/tutorial/ we will cover all of it and more in this course.
- Python for astronomers - http://www.scipy.org/wikis/topical_software/Tutorial Excellent tutorial by Greenfield and Jedrzejewski

Online Material

- www.python.org
- Start with the Python tutorial - http://docs.python.org/tutorial/ we will cover all of it and more in this course.
- Python for astronomers - http://www.scipy.org/wikis/topical_software/Tutorial Excellent tutorial by Greenfield and Jedrzejewski
- SciPy conferences - http://conference.scipy.org - lots of interesting talks (some with video versions)

Online Material

- www.python.org
- Start with the Python tutorial - http://docs.python.org/tutorial/ we will cover all of it and more in this course.
- Python for astronomers - http://www.scipy.org/wikis/topical_software/Tutorial Excellent tutorial by Greenfield and Jedrzejewski
- SciPy conferences - http://conference.scipy.org - lots of interesting talks (some with video versions)
- Astropy mailing list - http://mail.scipy.org/mailman/listinfo/astropy

Python on MS Windows

Full distributions provided by Enthought and ActiveState. Enthought distribution is better for scientific/technical computing.

Python on Linux

Usually, Python is already installed. Type `python` in a terminal to check it out.

On Redhat like distributions, the installer – anaconda – is written in Python.

GUIs for python development

IDLE is one that is distributed with Python.

emacs is a very good IDE if you willing to learn how to use it or know it already.

Numerous other free and commercial IDEs are available. A comprehensive list is available at http://wiki.python.org/moin/PythonEditors

Python Version 2 or 3?

Python 3 is newer but Python 2 has more existing third party software. For this reason, in this course, we will use Python 2 only. However, within a year or so it should be possible to move to Python 3.

Hello World program

```
$ python -c ’print "Hello World"’
Hello World
```

Starting Python

simply type *python* at the command prompt

```
$ python
Python 2.6.5 (r265:79063, Apr 16 2010, 13:09:56)
[GCC 4.4.3] on linux2
Type "help", "copyright", "credits" or "license" for
more information.
>>>
```

Ipython - an enhanced python shell

simply type *ipython* at the command prompt. If it is not installed `yum install ipython`

```
IPython 0.9.1 - An enhanced Interactive Python.
?  -> Introduction and overview of IPython's
features.
%quickref -> Quick reference.
help -> Python's own help system.
object?  -> Details about 'object'.  ?object also
works, ??  prints more.
In [1]:
```

Ipython is the shell for casapy. It will also eventually become the default shell for Pyraf.

Numerical types: Integers

```
>>> 2
>>> 0
>>> -4711
>>> 07, 022 # Octal tuple
>>> 0x9, 0xa, 0XF # Hexadecimal tuple
>>> 17 + 4 # Expression
>>> 0xa - 2
>>> 23 ** (2+3) # Power
>>> 7 / 2, 7 / -2 # Int division
>>> from __future__ import division
>>> 7/2
```

Floats

```
>>> 2.3
>>> -4.
>>> 0.1, .1
>>> 2.99E10, 6.62607e-27, -1e10
>>> 1.7 + .4
>>> 17.  + 4
>>> 7./2., 7./2, 7/2.
```