

Introduction to using Python to access OOI Data

Sage Lichtenwalner

2018 OOI Data Workshops



Time for a programmers' therapy session



What is Python?

- Flexible, powerful programming language
 - Object oriented
 - Runs everywhere
 - Testing framework
- Easy, clean syntax
- Very readable code
- Balanced high level programming with low level optimization
- Large community of support
 - Modular system, large number of libraries
- Free as in free beer
- Free as in free speech



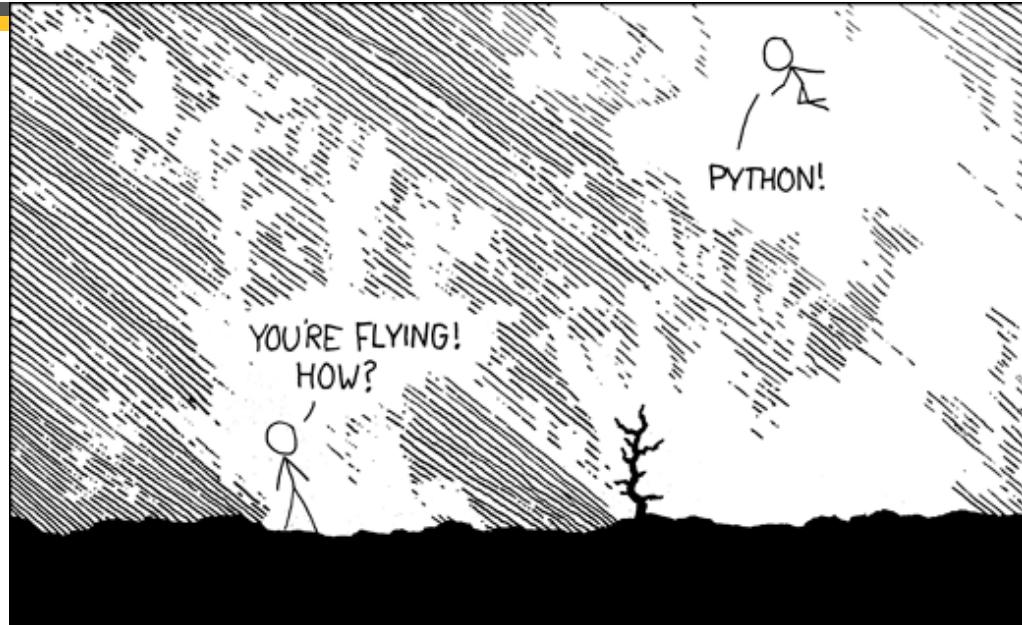
<https://github.com/ResearchComputing/Python-HPC-Spring-2013>



XKCD 353

Python

December 5, 2007



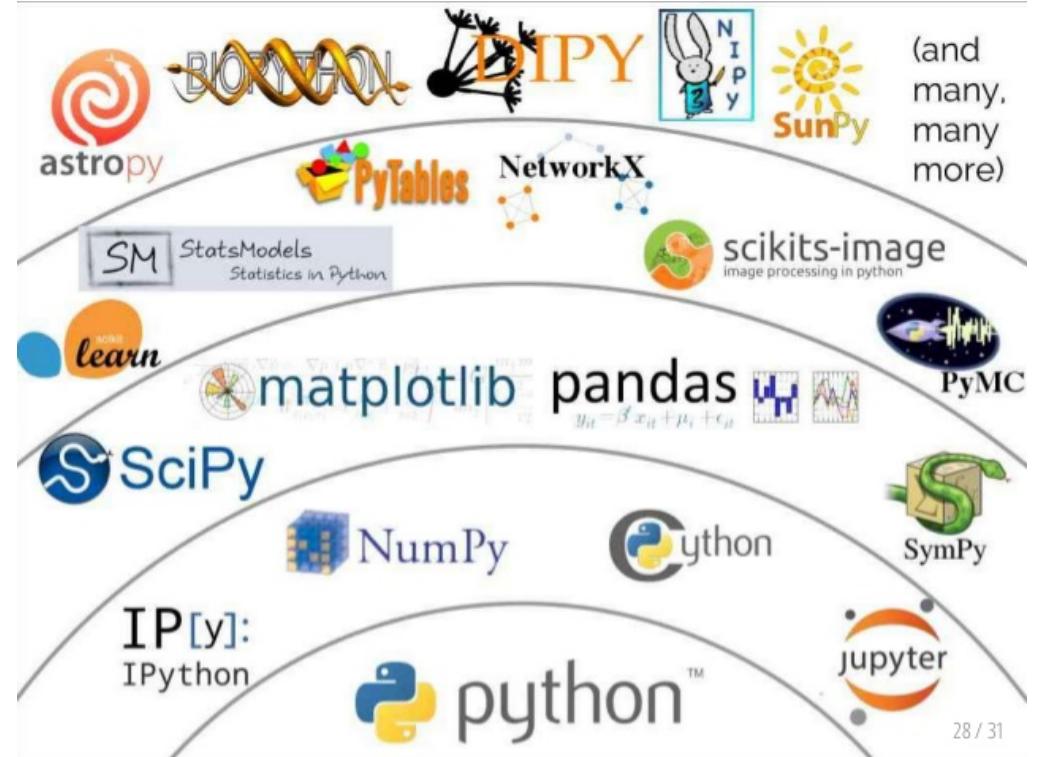
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST
print "Hello, world!"

I DUNNO...
DYNAMIC TYPING?
WHITESPACE?
COME JOIN US!
PROGRAMMING IS FUN AGAIN!
IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?

I JUST TYPED
import antigravity
THAT'S IT?
... I ALSO SAMPLED
EVERYTHING IN THE
MEDICINE CABINET
FOR COMPARISON.
BUT I THINK THIS
IS THE PYTHON.

The Python Ecosystem

- Python
 - the base language
- Numpy
 - arrays, fast operations on arrays
- Scipy
 - higher level computational routines
- Matplotlib
 - plotting
- Ipython
 - notebooks, flexible shell, and parallel
- Pandas
 - data analysis



Python vs. Matlab

Python

```
>>> import numpy as np

# Create row vector
>>> row = np.array([1, 2, 3])
>>> row
array([1, 2, 3])

# Transpose
>>> col = row.T

# Compute inner product
>>> inner = np.dot(row,col)
>>> inner
14

# Compute outer product
>>> outer = np.dot(col,row)
>>> outer
14
```

MATLAB

```
% Create row vector
>> row = [1 2 3]
row =
    1      2      3

% Transpose
>> col = row';

% Compute inner product
>> inner = row*col
inner =
    14

% Compute outer product
>> outer = col*row
outer =
    1      2      3
    2      4      6
    3      6      9
```

<https://www.mathworks.com/products/matlab/matlab-vs-python.html>

MATLAB lets you express math directly



Why Python?

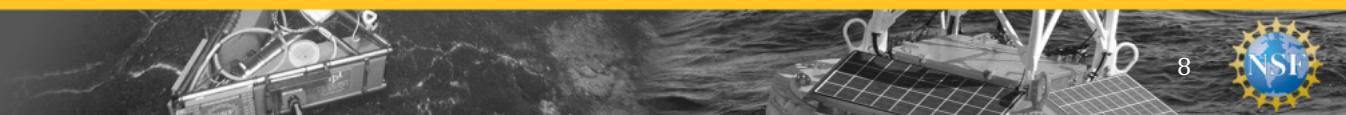
http://researchcomputing.github.io/meetup_fall_2014/pdfs/fall2014_meetup13_python_matlab.pdf



NumPy for Matlab Users

Matlab	NumPy / Python
Basic data type is a multidimensional array of double precision floating point numbers. Most expressions take such arrays and return such arrays. Operations on the 2-D instances of these arrays are designed to act more or less like matrix operations in linear algebra.	Basic type is a multidimensional array. Operations on these arrays in all dimensionalities including 2D are element-wise operations. However, there is a special matrix type for doing linear algebra, which is just a subclass of the array class. Operations on matrix-class arrays are linear algebra operations.
Uses 1 (one) based indexing. a(1)	Uses 0 (zero) based indexing. a[0]
Created for doing linear algebra. The syntax for basic matrix operations is nice and clean, but the API for adding GUIs and making full-fledged applications is more or less an afterthought.	Designed to be an excellent general-purpose programming language. While Matlab's syntax for some array manipulations is more compact than NumPy's, NumPy (by virtue of being an add-on to Python) can do many things that Matlab just cannot, for instance subclassing the main array type to do both array and matrix math cleanly.
Arrays have pass-by-value semantics, with a lazy copy-on-write scheme to prevent actually creating copies until they are actually needed. Slice operations copy parts of the array.	In NumPy arrays have pass-by-reference semantics. Slice operations are views into an array.

<http://mathesaurus.sourceforge.net/matlab-numpy.html>



Matlab vs. Python Cheatsheet

	Matlab	Python
Power	$a.^b$	$a^{**} b$ or <code>power(a,b)</code>
Remainder	<code>rem(a,b)</code>	$a \% b$ or <code>remainder(a,b)</code>
Not Equal	$a \sim= b$	$a != b$ or <code>not_equal(a,b)</code>
Short-circuit logical AND	<code>a && b</code>	<code>a and b</code>
Short-circuit logical OR	<code>a b</code>	<code>a or b</code>
pi	<code>pi</code>	<code>math.pi</code>
Row Vector	<code>a=[2 3 4 5];</code>	<code>a=array([2,3,4,5])</code>
Sequence	<code>1:10</code>	<code>range(1,11)</code>
First Row	<code>a(1,:)</code>	<code>a[0,:]</code>
All, except first row	<code>a(2:end,:)</code>	<code>a[1:,:]</code>
Last two rows	<code>a(end-1:end,:)</code>	<code>a[-2:,:]</code>

<http://mathesaurus.sourceforge.net/matlab-numpy.html>



2018 Data Workshops



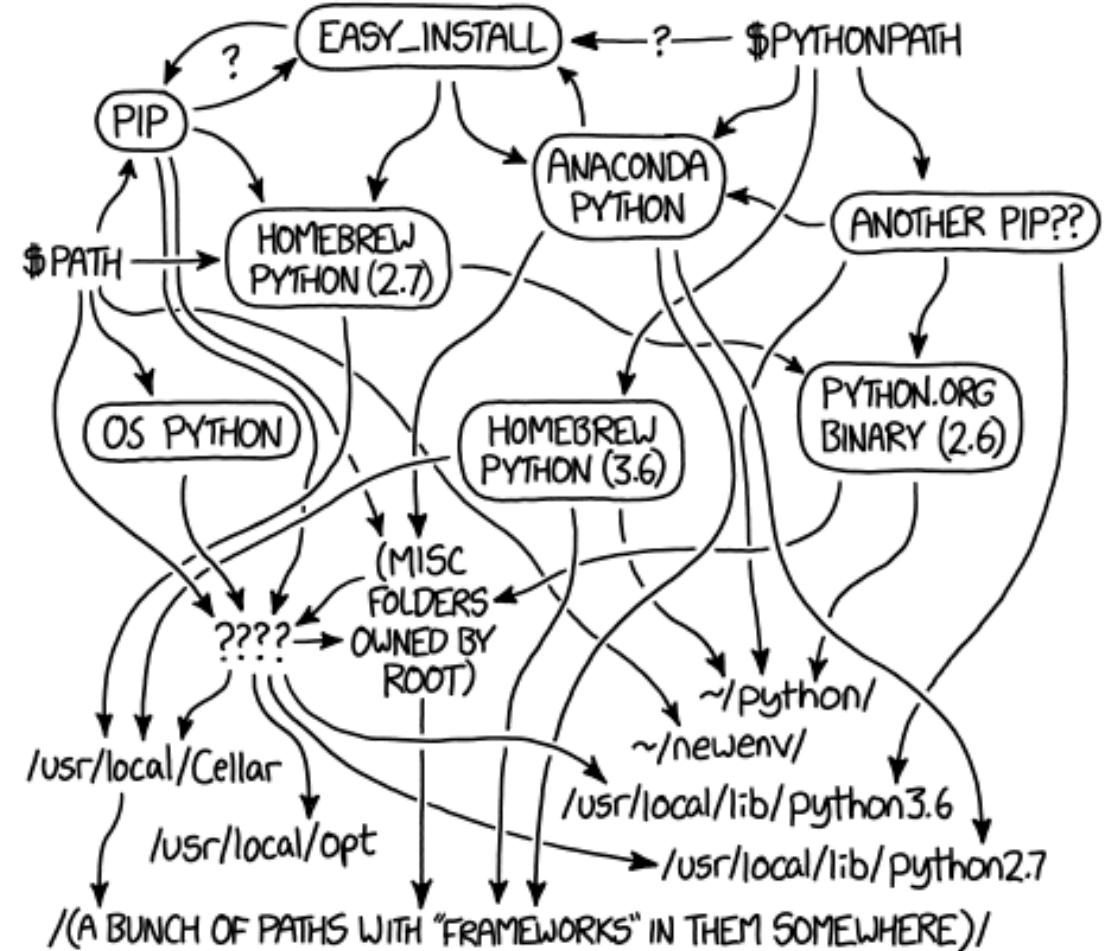
How do you get it?

- If you have a Mac or a Linux system, you might already have it
 - Type “python” or “which python” at the command line to see if you do
 - But, it’s probably only the basics and may not be the version you want
- Pip and virtual environments
 - Best for DIY and simple needs
- Anaconda – The full kit and kaboodle
 - <https://www.anaconda.com>
- Miniconda – Conda “Lite”
 - <https://conda.io/miniconda.html>



XKCD 1987

Python Environment
April 30, 2018



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Our Python Adventure

Session 1 (3 hours)

- Python Basics for Matlab Wizards
- Using the OOI API with Python
- Synchronous Requests & Basic Plotting
 - Time series
 - TS diagram
 - Profiles
 - Subplots
- Asynchronous Requests & Plotting NetCDF files



Session 2 (90 minutes)

- Multi-instrument quick plots (time series)
- Glider Data via ERDDAP (mapping and profile time series)
- Basic Data Processing (statistics, averaging, filtering, gridding, interpolation)
- Aggregation and Resampling (hexbins)
- Realtime Plotting

Data Validation Report (30 minutes)

Session 3 (30 minutes)

- Cruise Data Comparison
- Deciphering OOI Data Quality Flags

