

Python for Space Science

Snakes on a Spaceship



Python as a Scientific Language

A Brief Introduction

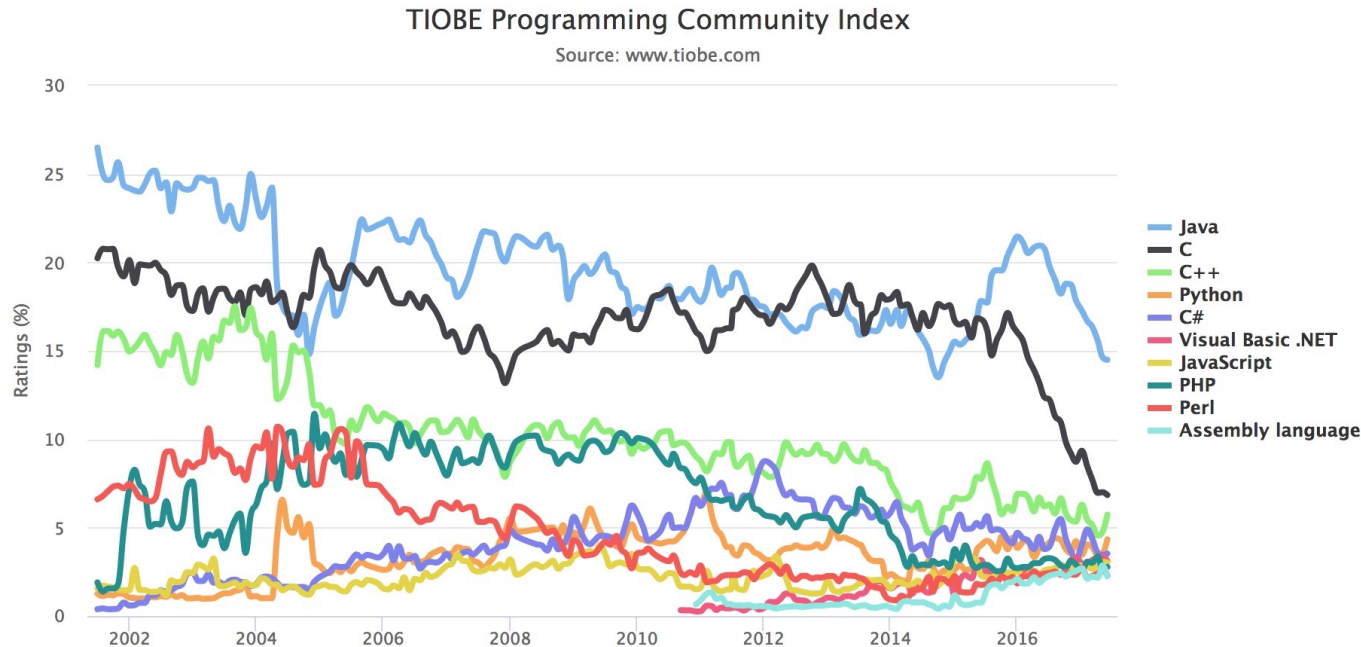
A.G. Burrell, U.T. Dallas Center for Space Science

Based on presentation by D. T. Welling, U. of Michigan CLaSP



Python is...

- ...a multi-paradigm (procedural, **object oriented**, etc.) general purpose language.
- ...named after a Flying Circus.
- ...relatively young: vers. 1.0 in 1994; version 2 in 2000 (IDL->1977; MatLab->1984).
- ...extremely popular (spans disciplines and used outside academia).
- ...Open source and **FREE!**

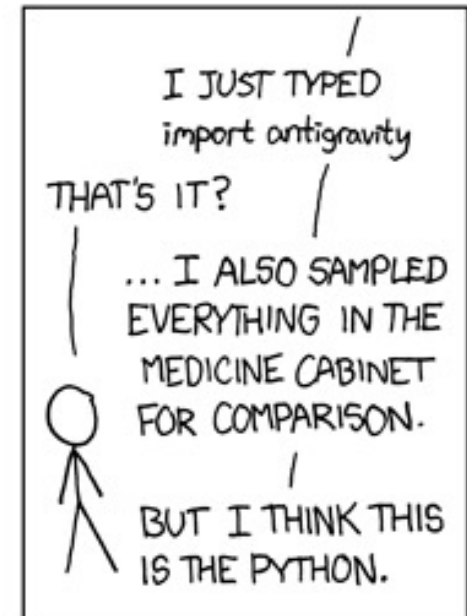
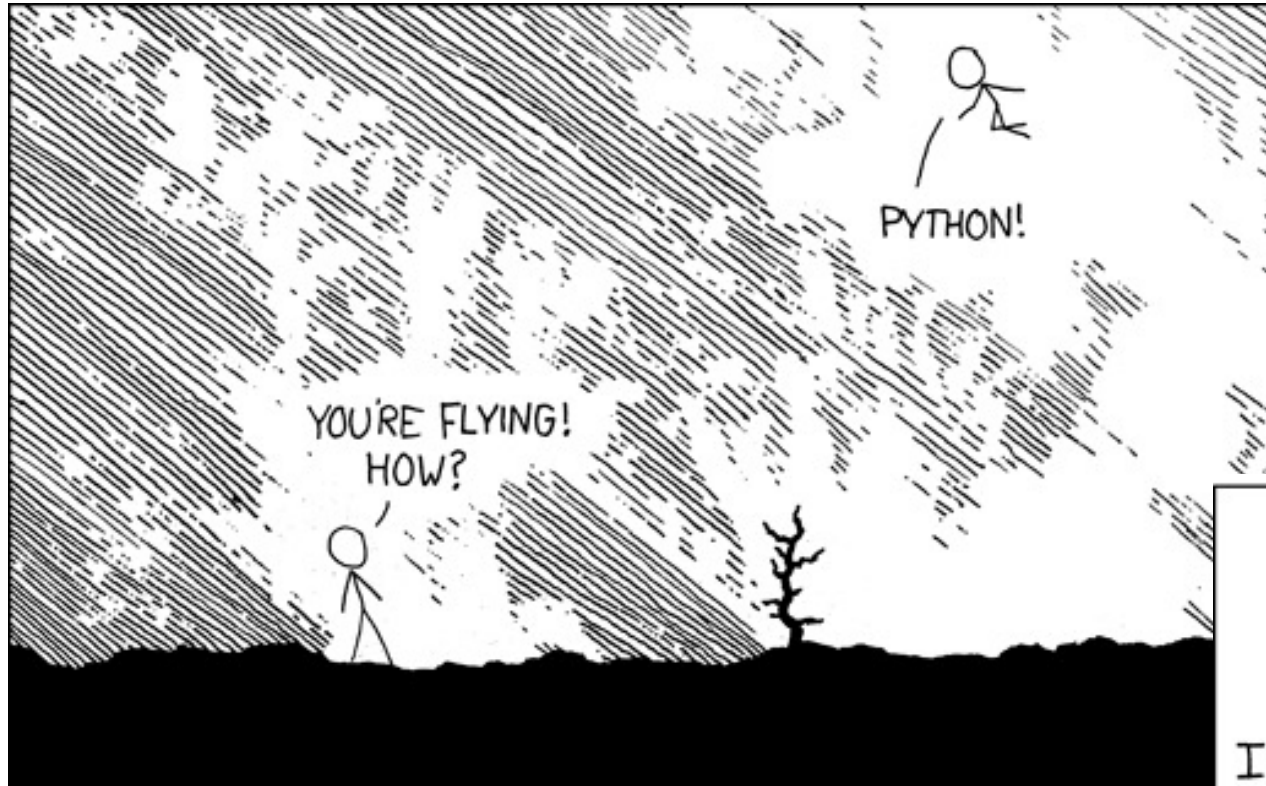


Python:	#4
MatLab:	#17
IDL:	> #50

**Current versions:
2.7.13 and 3.6.1**

Python lets you defy gravity

<https://xkcd.com/353/>



Why Python?

- Powerful scripting rivaling Perl, Bash, etc.
- “Batteries included” (GUI, Regular expressions, datetime, etc. out of the box.)
- Ubiquitous across platforms (Win, *nix, etc.)
- Extensible with C/C++
- Emphasis on readability
- Natural, easy to learn

IT'S FREE

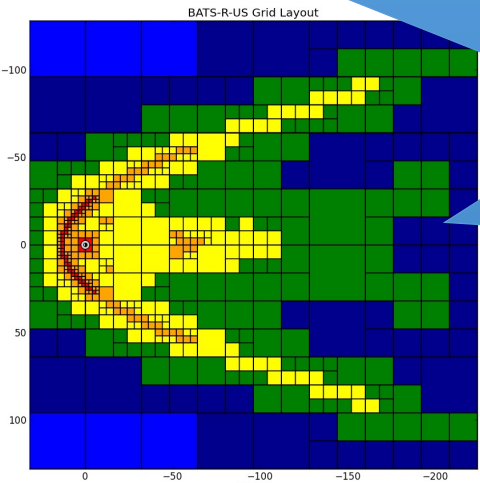
and I'm cheap.

Why Python for Science?

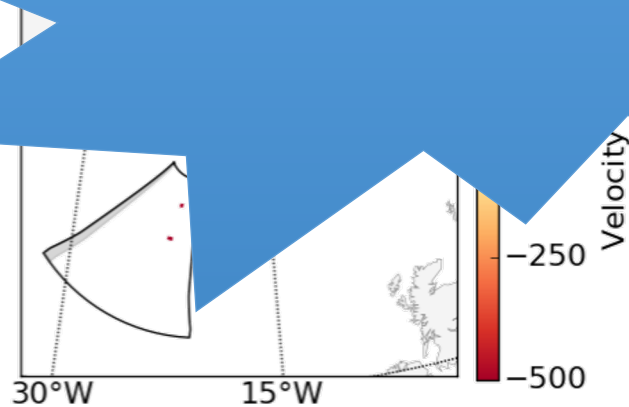
- Numpy (array algebra), SciPy (common scientific functions), and Matplotlib (plotting based on MatLab) combine to rival IDL / MatLab.
- Easy to drop into C, F90 for “heavy lifting”
- Modules available to read / write common file formats (e.g., hdf5, cdf, idl .sav, .m, etc.)
- Combines scripting with a complete scientific solution.

IT'S FREE

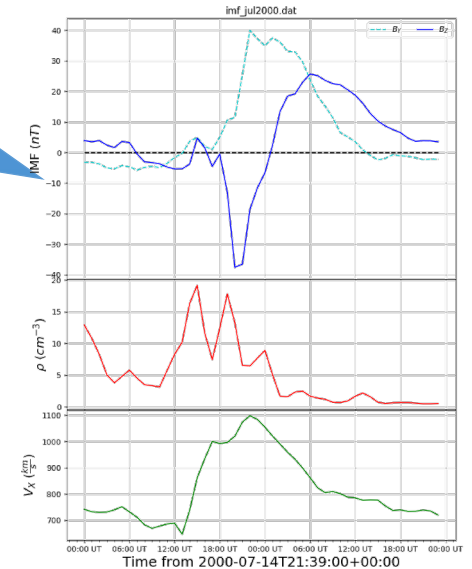
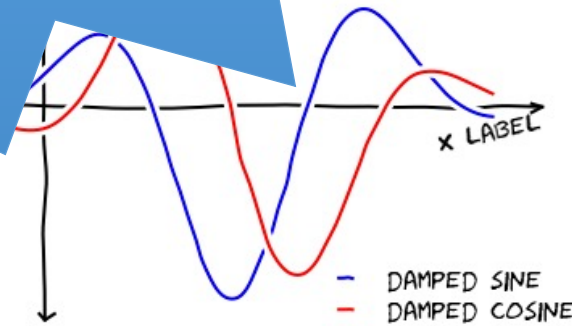
and who knows if we'll have funding next year.



13 August 2021



Snakes on a Spaceship: 2 Fast, 2 Furions - CEDAR 2017



www.python.org	Source, documentation, other resources.
.../dev/peps/pep-0008/	Style guide (suggested coding conventions)
enthought.com	Python think-tank; Canopy Python Distribution
<u>Dive Into Python</u> (Mark Pilgrim)	Open-source introduction (.net for website)
<u>Core Python Programming</u> (Wesley Chun)	Excellent introduction and reference; very thorough.
https://imgtfy.com/	Let Me Google That For You...
Dan Welling¹	www-personal.umich.edu/~dwelling/python/

¹ He was the one who got me into this mess...

Getting Python

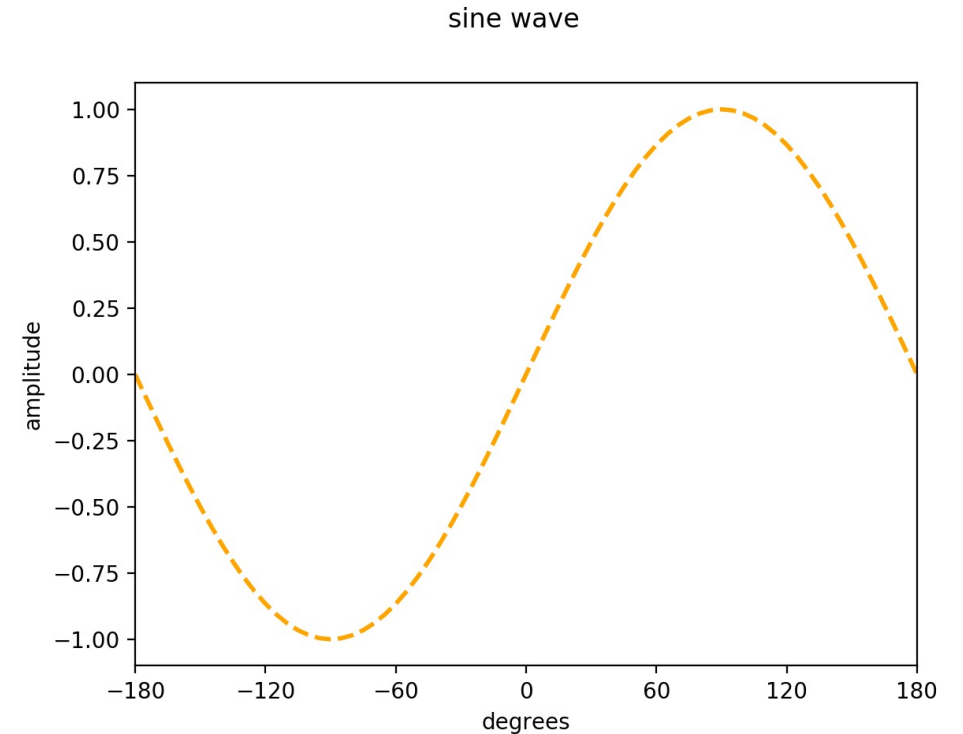
*nix	Already done! Use your package manager to get additional modules.
OS X	Use internal installation (WARNING: non-standard!)
	Use package manager (Fink, MacPorts, Homebrew).
	Get software from python.org and install.
	Use Enthought Canopy Python distribution.
Windows	Get software from python.org and install.
	Use Enthought Canopy Python distribution.

Scientists will want Python, Numpy, SciPy, Matplotlib, and iPython.

There are many, many ways to work with Python

- Scripting and executing from the system shell.
- Command line interfaces through a Python shell, such as the default shell or iPython.
- Interactive Development Environments that combine text editors with shell prompts, such as Spyder.
- Jupyter Notebook, a web-based interactive session that combines mark-up and code.

```
$ ipython
In [1]: import numpy as np
In [2]: import matplotlib as mpl
In [3]: import matplotlib.pyplot as plt
In [4]: plt.ion()
In [5]: f = plt.figure()
In [6]: ax = f.add_subplot(1,1,1)
In [7]: x = np.arange(-180, 181, 5.0)
In [8]: ax.plot(x, np.sin(np.radians(x)), "--", color="orange", linewidth=2)
In [9]: ax.set_xlim(-180, 180)
In [10]: ax.xaxis.set_major_locator(mpl.ticker.MultipleLocator(60))
In [11]: ax.set_xlabel("degrees")
In [12]: ax.set_ylabel("amplitude")
In [13]: f.suptitle("sine wave")
```



- **ipython** begins interactive shell
- **import** command loads modules or python files into interactive namespace
- **ion()** command turns on interactive plotting

www-personal.umich.edu/~dwellings/python/notebooks/example_imf.zip

```
#!/usr/bin/env python
'''
An example module [...]
'''

import numpy as np

def format_ax(ax, ylabel=None):
    '''
    Format an axes object [...]
    '''
    <commands>
# Comments start with hashtags!
```

- “shebang” – tells shell how to execute.
 - “docstring” – long form comments/documentation
 - Imports – include code from other python files into namespace
 - Function and variable definitions (note docstrings associated with definitions)
- # Don't forget to comment!**

```
class ImfData(dict):
    """
    A class for handling Imf [...]
    """
    def __init__(self, filename):
        self.file = filename
        <more commands>

    def calc_v(self):
        """
        Calculate [...]
        """
        <commands>
```

Class definitions:

- Class-level docstring
- Class attributes save instances that describe the object
 - Can be any data type: integers, Boolean, lists, other classes...
- Class methods are functions that leverage attributes
- Special methods define basic object behavior
 - *e.g.* `__init__` initializes the class object

NOTE NESTED TABBING

```
if __name__ == '__main__':  
    <commands>  
    <commands>  
    <commands>
```

```
$ ./example_imf.py
```

```
usage: example_imf.py [-h] file
```

```
example_imf.py: error: too few arguments
```

Optional code that runs in the “__main__” namespace (when file is executed, not imported).

```
$ ./example_imf.py imf_jul2000.dat
```

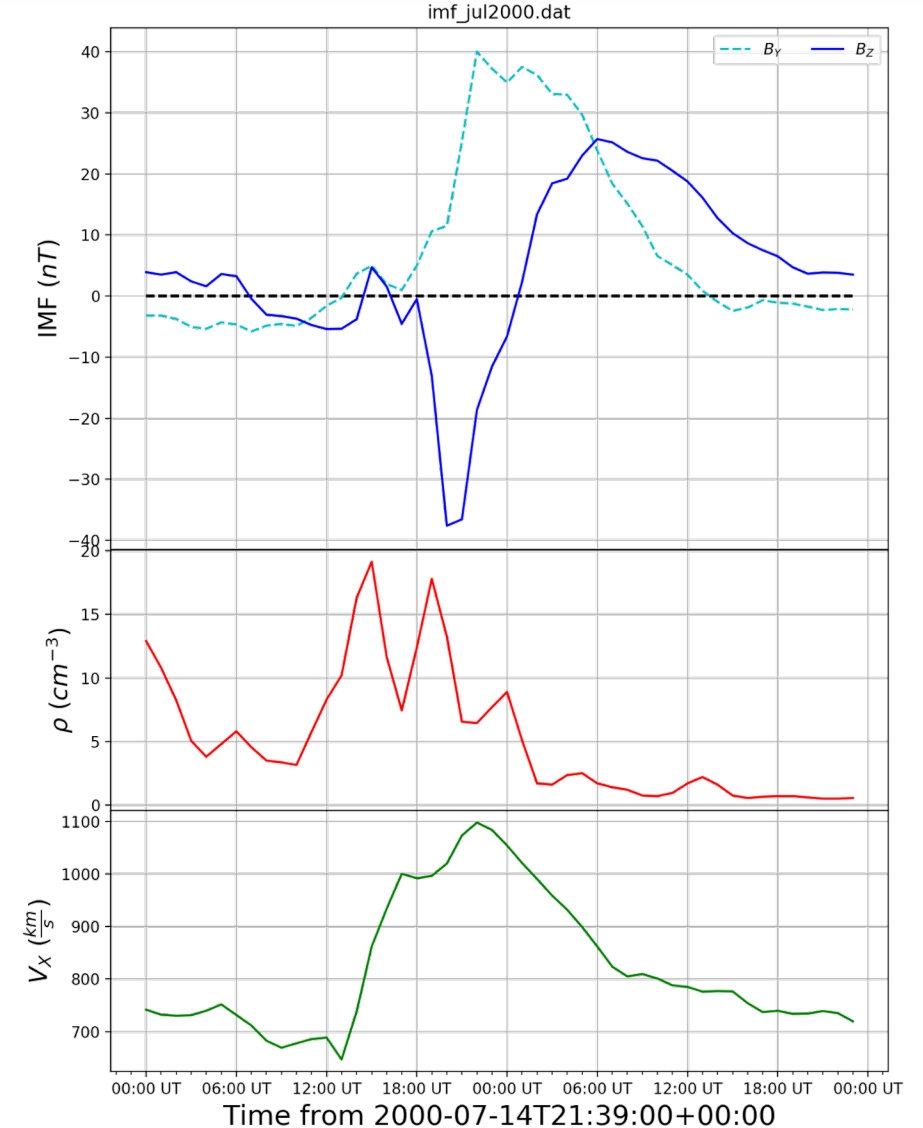
```
$ ls imf_jul2000.png
```

Or from ipython

```
In [1]: import example_imf
```

```
In [2]: imf = example_imf.ImfData('imf_jul2000.dat')
```

```
In [3]: imf.plot_imf('imf_jul2000.jpg')
```



- SpacePy – Angeline Burrell
- Madrigal – Bill Rideout
- GeoData – John Swoboda
- Davitpy – Ashton Reimer

Lunch

- OvationPyme – Liam Kilcommons
- Signal Chain – Alexander Valdez
- PyGlow – Tim Duly
- PySat – Russell Stoneback
- Panel Discussion

Snakes on a Spaceship





A software library for space science data analysis, modelling and space weather forecasting

SpacePy team: Steve Morley¹; Dan Welling²; Jon Niehof³; Brian Larsen¹; et al.⁴

1 – Los Alamos National Laboratory

2 – University of Michigan

3 – University of New Hampshire

4 - Additional contributions to SpacePy from more people than I have space here...

Tools Include (very incomplete selection)

- Tracing
 - Field lines
 - Drift shells
- Superposed epoch analysis (1D, 2D)
- Bootstrap CI
- Association analysis
- Windowing mean (time based, points based)
- Time & Coordinate conversions
- Quaternion Math
- Full interface to NASA CDF library
- Interface to IRBEM library
- Plot "helper" routines
 - add logo
 - automated time tick formatting
 - rebinning/spectrograms
 - plot styles

Versions, repositories, and resources

We support Linux, Mac and Windows
v 0.1.6: Mac (on MacPorts), Linux
v 0.1.5: Windows

Code repositories:

- sourceforge.net/projects/spacepy
- github.com/spacepy/

Some Python/SpacePy/PyBATS resources:
www-personal.umich.edu/~dwelling/python/