Introduction to python
Why Python?

Highly expressive:
```
friends = ['john', 'pat', 'gary', 'michael']
for i, name in enumerate(friends):
    print("iteration {iteration} is {name}".format(iteration=i, name=name))
```

Object oriented

Ease:
Easy to learn, change from IDL or Matlab.

Very active community

Extendability:
libraries for numerics, data analysis, plotting, financing, ...

Interoperability:
Import IDL and Matlab code. “But all my routines are written in IDL.”
Why Python?

“Python is everywhere, it is all around us, even now in this very room.”

3ds Max
Maya
Blender
Cinema 4D
...

Paraview, Visit, Vapor, ...

Civilization IV, Battlefield 2, World of Tanks, ...
Why Python?
Why Python?
History

1991, Guido van Rossum

Namesake:

Python 2.0 2000

Python 3.0 2008

Python 3.52 2016

Python Course
Ways of Using Python

python/ipython

https://www.python.org/
Ways of Using Python

```
276    return ['R', R, 'T', T]
277
284    vw = fresnel_data['w_list'][layer]
285    k2 = fresnel_data['k2_list'][layer]
286    n = fresnel_data['n_list'][layer]
287    n_0 = fresnel_data['n_list'][0]
288
289    # amplitude of forward-moving wave is Ef, backwards is Eb
290    Ef = vw(0) * exp(1j * k2 * dist)
291    Eb = vw(1) * exp(-1j * k2 * dist)

# Poynting vector
296    if pol == 's':
297        Ey = ((cos(th_0) * abs(Ef+Eb)) * real) / (n_0 * cos(th_0)).real
298        E0 = ((cos(th_0) * abs(Ef-Eb)) * real) / (n_0 * cos(th_0)).real
299        return (Ey, E0)
300
301    # absorbed energy density
302    if pol == 'p':
303        aborp = (n_0 * cos(th_0) * abs(Ef+Eb)) * real)
304        aborp = (n_0 * cos(th_0) * abs(Ef-Eb)) * real)
305        return (aborp, aborp)
306
307    def find_in_structure(d_list, dist):
308        d_list is list of thicknesses of layers, all of which are finite.
309        dist is the distance from the front of the whole multilayer structure
310        (i.e., from the start of layer 0).
```
Matplotlib Gallery

- broken_barh
- centered_ticklabels
- cohere_demo
- color_by_yvalue

- color_demo
- colorbar_tick_labelling_demo
- colorbar_tick_labelling_demo

- contour_corner_mask
- contour_demo
- contour_demo
- contour_demo

Scipy Functionalities

Tutorial

Tutorials with worked examples and background information for most SciPy submodules:

- SciPy Tutorial
  - Introduction
  - Basic functions
  - Special functions (scipy.special)
  - Integration (scipy.integrate)
  - Optimization (scipy.optimize)
  - Interpolation (scipy.interpolate)
  - Fourier Transforms (scipy.fftpack)
  - Signal Processing (scipy.signal)
  - Linear Algebra (scipy.linalg)
  - Sparse Eigenvalue Problems with ARPACK
  - Compressed Sparse Graph Routines (scipy.sparse.csgraph)
  - Spatial data structures and algorithms (scipy.spatial)
  - Statistics (scipy.stats)
  - Multidimensional image processing (scipy.ndimage)
  - File IO (scipy.io)
  - Weave (scipy.weave)
Outlook

Python Data Analysis Library

pandas

\[ y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it} \]

http://pandas.pydata.org

mpmath

floating-point arithmetic with arbitrary precision

http://mpmath.org

Hierarchical Data Format

vtk Data Format
Outlook

SunPy

http://sunpy.org

astropy

http://www.astropy.org
# Python Documentation

```python
help(plt.plot)
plt.plot?
source(plt.plot)
plt.plot??
```

[Google Search Result]

**update U V data for matplotlib streamplot**

After plotting streamlines using `matplotlib.streamplot` I need to change the U V data and update the plot. For `imshow` and `quiver` there are the functions `set_data` and `set_UVC`, respectively. There does not seem to be any similar function for streamlines. Is there any way to still updateget similar functionality?

```
3 share edit delete flag
```

1. **I suspect the answer is no, because if you change the vectors, it would need to re-compute the stream lines. The objects returned by `streamline` are a line and patch collections, which know nothing about the streamlines. To get this functionality would require writing a new class to wrap everything up and finding a sensible way to re-use the existing objects.** – tacaswell Dec 24 ’12 at 17:31

1. **A dirty workaround would be setting the visibility of the arrows and lines to 0 and then plotting the new streamlines. Will try if that is fast enough, since speed is an issue.** – loman Dec 23 ’12 at 0:08
Practice!