What is NumPy?

IE Python User Group
October 21, 2014
Cameron Goodale @sigep311

Talk Outline

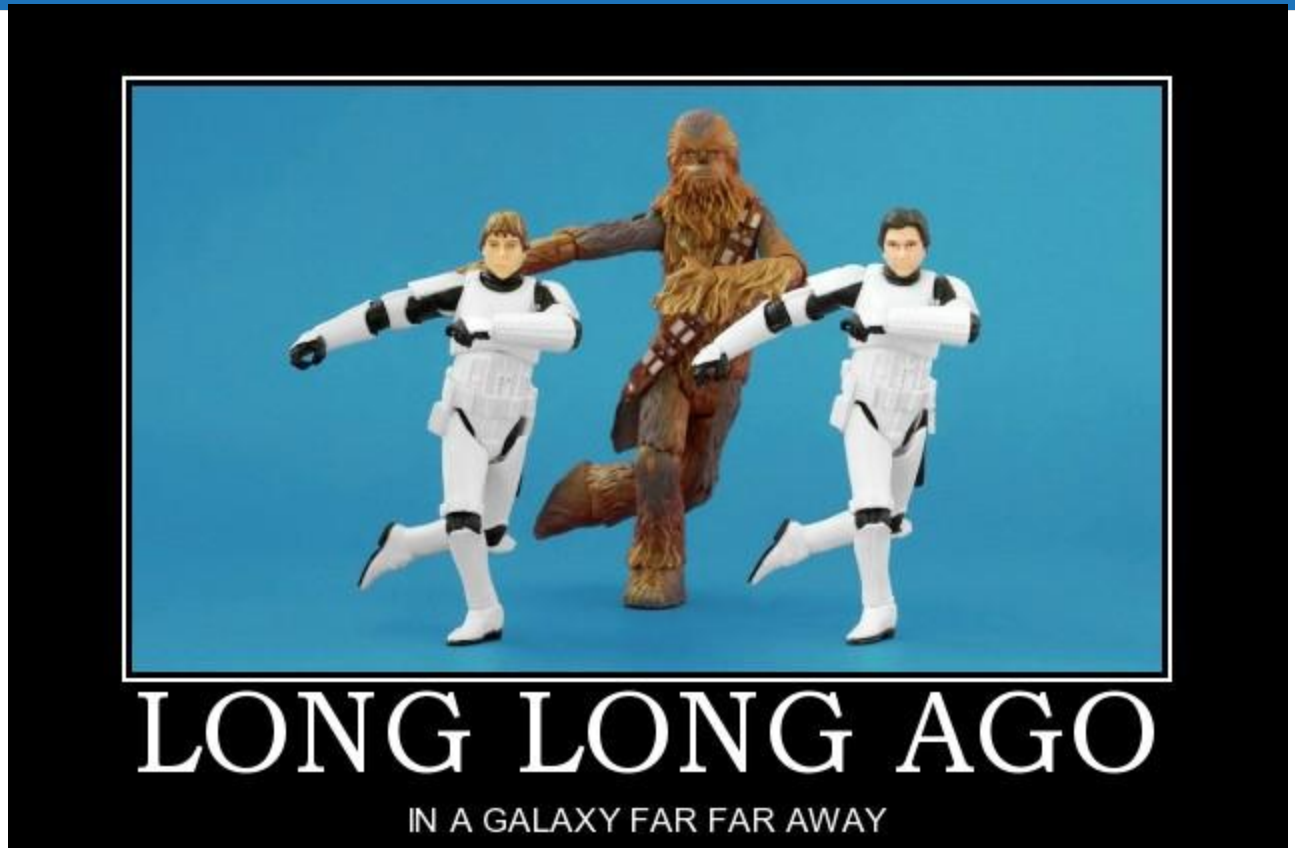
History

NumPy in a Nutshell

Arrays vs. Lists

SciPy

A Long time ago in a ...



Until 1994 - Python wasn’t built for scientific computing
1995 - Jim Hugunin and others created Numeric
???? - Another Python Array module Numarray
2005 - Travis Oliphant created NumPy

*Note: NumPy is part of SciPy

Stolen from: http://en.wikipedia.org/wiki/NumPy

NumPy is...

Powerful *n*-Dimensional Array

Sophisticated Broadcasting Functions

Integrate with C/C++/Fortran

Useful for Linear Algebra, Fourier Transforms, and random number capabilities

BSD Licensed

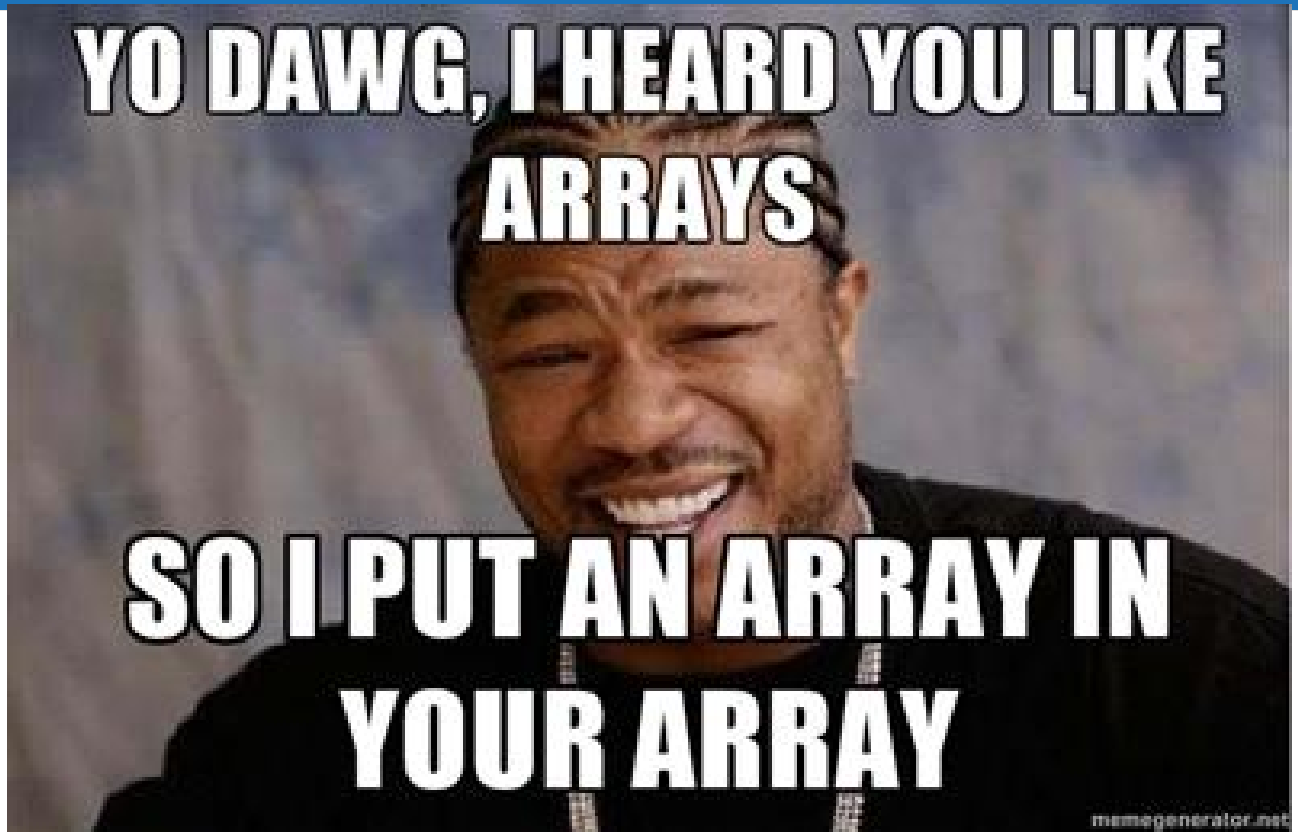
n-Dimensionality

1D - Just like a Python list array([1,2,3,4])

2D - Think of an Image (x,y)

```
array([[1, 0, 0, 1],
       [1, 1, 0, 1],
       [1, 0, 1, 1],
       [1, 0, 0, 1]])
```

Arrays in Arrays?



n-Dimensionality

3D - Now it’s (x,y,z) or (x,y,t) - Image over Time

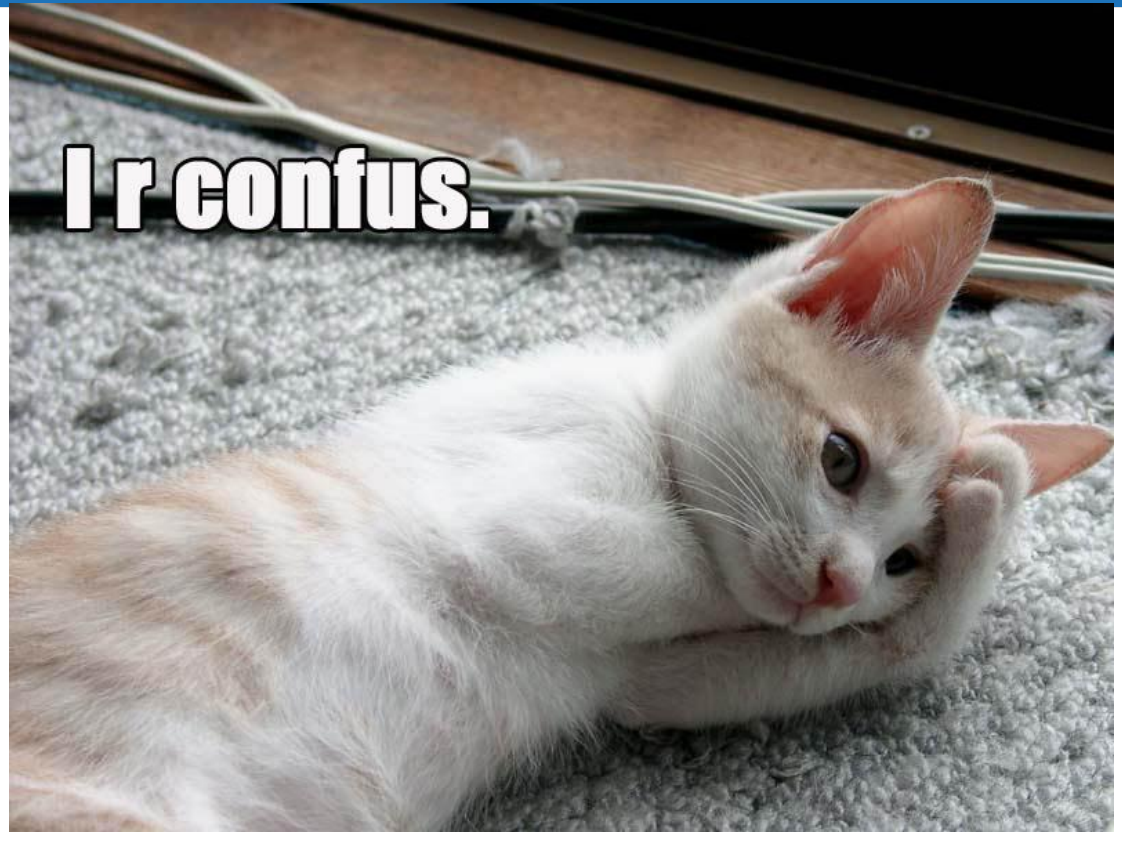
4D - 3D Model through Time (x,y,z,t)

Common in Scientific Data Formats

(NetCDF, HDF, Grib, etc…)

Lat, Lon, Height, Time

n-Dimensional WAT?



Arrays vs. Lists

NumPy Array	**Python List**
Static Typed	Dynamic Typing
Fixed Length	Can Append or Extend
Maps to Contiguous Block of Memory	
Nearly C Speed	Much Slower

Arrays vs. Lists

```
>>> import numpy as np
>>> x = np.array(range(9))
array([0, 1, 2, 3, 4, 5, 6, 7, 8])
vs.
>>> x = range(9)
[0, 1, 2, 3, 4, 5, 6, 7, 8]
```

Python List API

```
>>> dir(y)
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__',
'__delslice__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__',
'__getitem__', '__getslice__', '__gt__', '__hash__', '__iadd__', '__imul__',
'__init__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__',
'__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__',
'__setattr__', '__setitem__', '__setslice__', '__sizeof__', '__str__',
'__subclasshook__', 'append', 'count', 'extend', 'index', 'insert', 'pop',
'remove', 'reverse', 'sort']
```

NumPy Array API

```
>>>dir(x)
['T', '__abs__', '__add__', '__and__', '__array__', '__array_finalize__',
'__array_interface__', '__array_prepare__', '__array_priority__',
'__array_struct__', '__array_wrap__', '__class__', '__contains__', '__copy__',
'__deepcopy__', '__delattr__', '__delitem__', '__delslice__', '__div__',
'__divmod__', '__doc__', '__eq__', '__float__', '__floordiv__', '__format__',
'__ge__', '__getattribute__', '__getitem__', '__getslice__', '__gt__', '__hash__',
'__hex__', '__iadd__', '__iand__', '__idiv__', '__ifloordiv__', '__ilshift__',
'__imod__', '__imul__', '__index__', '__init__', '__int__', '__invert__', '__ior__',
'__ipow__', '__irshift__', '__isub__', '__iter__', '__itruediv__', '__ixor__', '__le__',
'__len__', '__long__', '__lshift__', '__lt__', '__mod__', '__mul__', '__ne__',
'__neg__', '__new__', '__nonzero__', '__oct__', '__or__', '__pos__', '__pow__',
```

NumPy Array API Continued...

'__radd__', '__rand__', '__rdiv__', '__rdivmod__', '__reduce__',
'__reduce_ex__', '__repr__', '__rfloordiv__', '__rlshift__', '__rmod__',
'__rmul__', '__ror__', '__rpow__', '__rrshift__', '__rshift__', '__rsub__',
'__rtruediv__', '__rxor__', '__setattr__', '__setitem__', '__setslice__',
'__setstate__', '__sizeof__', '__str__', '__sub__', '__subclasshook__',
'__truediv__', '__xor__', ***'all', 'any', 'argmax', 'argmin', 'argsort', 'astype', 'base', 'byteswap', 'choose', 'clip', 'compress', 'conj', 'conjugate', 'copy', 'ctypes', 'cumprod', 'cumsum', 'data', 'diagonal', 'dot', 'dtype', 'dump', 'dumps', 'fill', 'flags', 'flat', 'flatten', 'getfield', 'imag', 'item', 'itemset', 'itemsize', 'max', 'mean', 'min', 'nbytes', 'ndim', 'newbyteorder', 'nonzero', 'prod', 'ptp', 'put', 'ravel', 'real', 'repeat', 'reshape', 'resize', 'round', 'searchsorted', 'setasflat', 'setfield', 'setflags', 'shape', 'size', 'sort', 'squeeze', 'std', 'strides', 'sum', 'swapaxes', 'take', 'tofile', 'tolist', 'tostring', 'trace', 'transpose', 'var', 'view']***

Arrays vs. Lists

CAN I HAZ MOAR SPEEDZ?

Arrays vs. Lists

Speed Test Code Review

speed_run.py

Using timeit module for accurate timings

Test Case:

Trivial 100 x 100 2D Dataset where each element in the Dataset must be raise by a power of 2.

Speed Test Code Review

speed_run.py

We will use the following 5 functions:

Python Lists and FOR LOOPING

Python Lists and using map() function

Python Lists and List Comprehensions

Numpy Array and FOR LOOPING

Numpy Array and Vectorization

Speed Test Code Review

$ python speed_run.py

NumPy Routines

Array creation routines

Array manipulation routines

Binary operations

String operations

C-Types Foreign Function Interface (numpy.ctypeslib)

Datetime Support Functions

Data type routines

Optionally Scipy-accelerated routines (numpy.dual)

Mathematical functions with automatic domain (numpy.emath)

Floating point error handling

Discrete Fourier Transform (numpy.fft)

Financial functions

Functional programming

Numpy-specific help functions

Indexing routines

Input and output

Taken from: http://docs.scipy.org/doc/numpy/reference/routines.html

NumPy Routines

Linear algebra (numpy.linalg)

Logic functions

Masked array operations

Mathematical functions

Matrix library (numpy.matlib)

Numarray compatibility (numpy.numarray)

Old Numeric compatibility (numpy.oldnumeric)

Miscellaneous routines

Padding Arrays

Polynomials

Random sampling (numpy.random)

Set routines

Sorting, searching, and counting

Statistics

Test Support (numpy.testing)

Asserts

Window functions

Taken from: http://docs.scipy.org/doc/numpy/reference/routines.html

So...what about SciPy?

SciPy

http://scipy.org/

Collection of python libraries for Math, Science and Engineering

- Numpy (N-Dimensional Arrays)
- SciPy Library (Fundamentals for Sci-Computing)
- IPython (Interactive Python Console)
- Sympy (Symbolic Mathematics)
- Matplotlib (2D Plotting and close to MATLAB)
- Pandas (Data Structures & Analysis and close to R)

Thank You

Questions?