

Python for Scientific Computing

freealbert

Blog: <http://dspandlinux.com>

Email: jim2429212@gmail.com

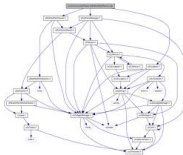
May 1, 2012

New Tasks

gui



<?xml?>



Natural
Language



New Tool



What is Python?

a remarkably powerful dynamic programming language.



Guido van Rossum
Benevolent Dictator For Life

Python's feature

- free and opensource



- runs everywhere

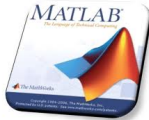


VxWorks



Python's feature

- plays well with others



Python's feature

- very clear,readable syntax

Implementing the basic QuickSort algorithm in Python

```
def qsort(L):  
    if not L: return L # exit recursion if input is empty  
    pivot, rest = L[0], L[1:]  
    less_than = [ x for x in rest if x < pivot ]  
    greater_eq = [ x for x in rest if x >= pivot ]  
    return qsort(less_than) + [pivot] + qsort(greater_eq)
```

- Mandatory indentation
- boosts developer productivity
Python code is typically $\frac{1}{3}$ to $\frac{1}{5}$ the size of equivalent C++ or Java code.

Figure A. Traditional “Hello, World!” program in various languages: Python (a), Perl (b), Ansi C (c), C++ (d), C# (e), Java (f), and Ruby (g).

```
# Hello World in Python
print 'Hello, World'
```

(a)

```
# Hello world in perl
print "Hello World!\n";
```

(b)

```
/* Hello World in C, Ansi-style */
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    puts("Hello World!");
    return EXIT_SUCCESS;
}
```

(c)

```
// Hello World in C++
#include <iostream.h>
main()
{
    cout << "Hello World!" << endl;
    return 0;
}
```

(d)

```
// Hello World in Microsoft C# ("C-Sharp").
using System;
class HelloWorld
{
    public static int Main(String[] args)
    {
        Console.WriteLine("Hello, World!");
        return 0;
    }
}
```

(e)

```
// Hello World in Java
class HelloWorld {
    static public void main( String args[]
    ) {
        System.out.println( "Hello World!" );
    }
}
```

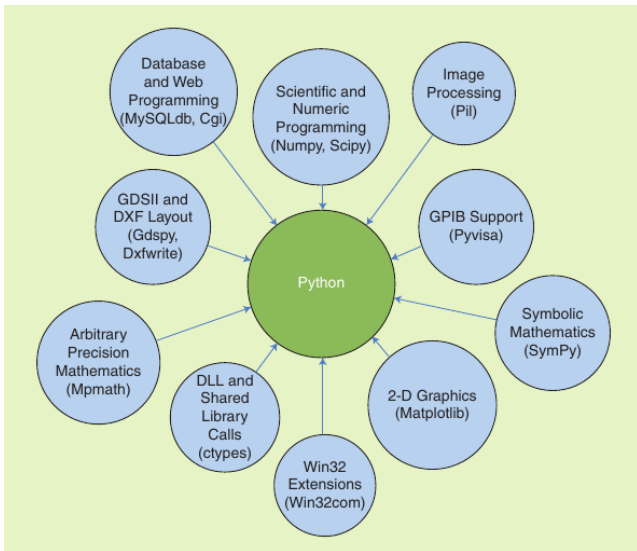
(f)

```
# Hello World in Ruby
STDOUT << "Hello World!"
```

(g)

How to replace Matlab?

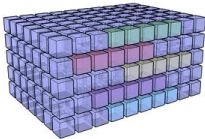
Python: An Ecosystem for Scientific Computing



NumPy

N-dimensional Array manipulations

- N-dimensional array object



- Fourier transforms

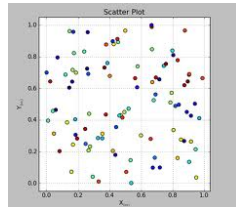


- linear algebra functions

Linear Algebra

```
ax + ay*bx*cy - az*by*cv + aw*b  
ax - az*bu*cx + az*bx*cv - aw*b  
ay ay*bx*cy - ay*bx*cv + aw*b  
ay ay*bx*cy - ay*bx*cv + aw*b  
Cos(t), Sin(t), 0, 0  
-Sin(t), Cos(t), 0, 0  
0, 0, 1, 0  
0, 0, 0, 1  
Cross( e(x), e(y), e(z) ) = (-1)  
Cross( e(x), e(z), e(y) ) =  
Cross( e(y), e(x), e(z) ) =
```

- random number capabilities



SciPy

Scientific tools for Python

a library of scientific tools
depends on the NumPy



SciPy provides modules for

- statistics
- optimization
- numerical integration
- linear algebra
- Fourier transforms
- signal processing
- image processing
- ODE solvers
- special functions
- ...

Image Processing

- PIL



- pyopencv



SymPy

SymPy is a Python library for symbolic mathematics.



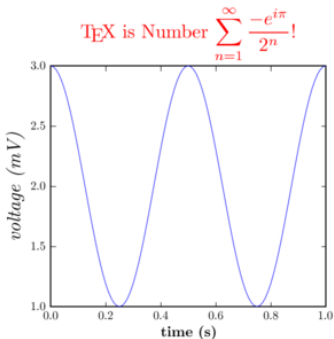
SymPy provides modules for

- Core capabilities
- Polynomials
- Calculus
- Solving equations
- Discrete math
- Matrices
- Geometric Algebra
- Geometry
- Plotting
- Physics
- Statistics
- Printing

matplotlib

a python 2D plotting library

matplotlib is Object-Oriented and its syntax looks alike matlab's.



Tips: It is necessary to get a handle on its inheritance relationship.

Mayavi Project

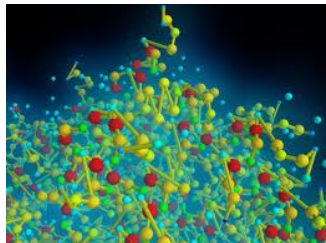
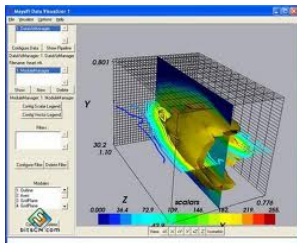
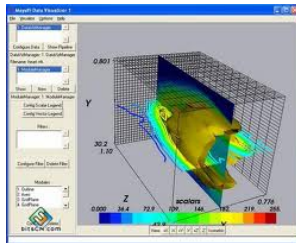
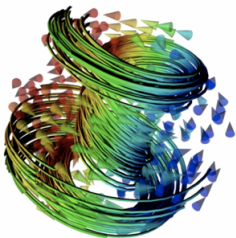
3D Scientific Data Visualization and Plotting

The Mayavi project includes two related packages for 3-dimensional visualization:

- Mayavi: A tool for easy and interactive visualization of data, with seamless integration with Python scientific libraries.
- TVTK: A Traits-based wrapper for the Visualization Toolkit, a popular open-source visualization library.

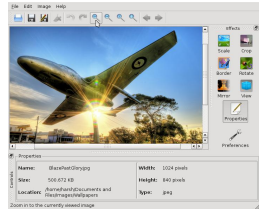


MayaVi Screenshots

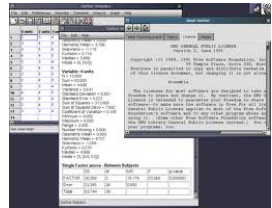
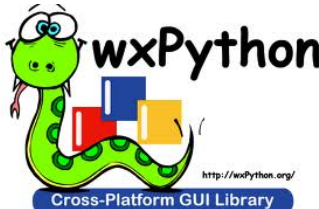


GUI Programming

- PyQt



- wxPython



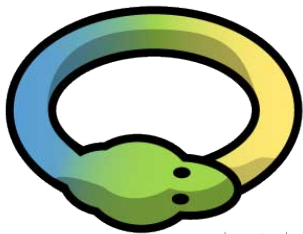
Tkinter PyGtk PyGUI PyKDE ...

PyPy

the coming future

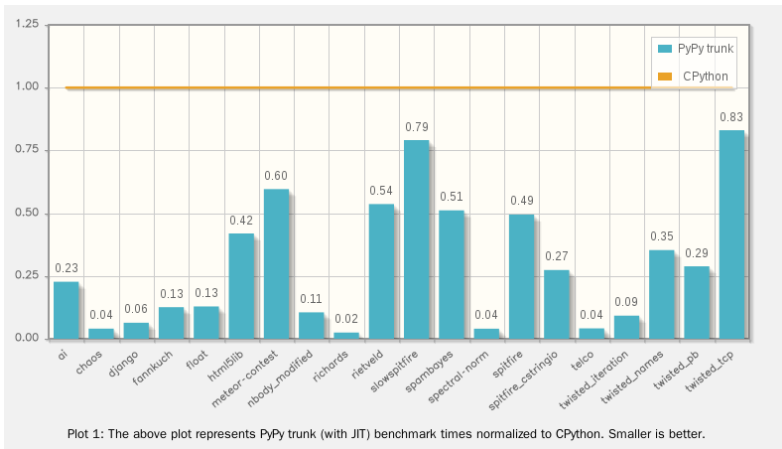
PyPy is a fast, compliant alternative implementation of Python. It has several advantages and distinct features:

- Faster speed much (thanks to JIT)
- Less memory usage
- Highly compatible
- Sandboxing
- Stackless



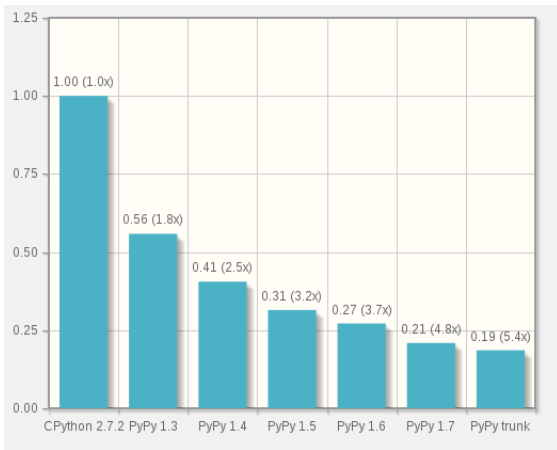
pypy logo artwork
2009/12/07 samuel reis

How fast is PyPY?



It depends greatly on the type of task being performed. The geometric average of all benchmarks is 0.19 or 5.4 times faster than CPython

PyPy is evolving



Summary

an efficient frame for scientific computing

