An introduction to Sage

Franco Saliola
⟨saliola@gmail.com⟩

Fields Institute

6 November 2009

Outline

What is Sage?

Sage is . . .

a *distribution* of software

Sage is a *distribution* of software

When you install Sage, you get:

ATLAS	Automatically Tuned Linear Algebra Software
BLAS	Basic Fortan 77 linear algebra routines
Bzip2	High-quality data compressor
Cddlib	Double Description Method of Motzkin
Common Lisp	Multi-paradigm and general-purpose programming lang.
CVXOPT	Convex optimization, linear programming, least squares
Cython	C-Extensions for Python
F2c	Converts Fortran 77 to C code
Flint	Fast Library for Number Theory
FpLLL	Euclidian lattice reduction
FreeType	A Free, High-Quality, and Portable Font Engine

Sage is a *distribution* of software

When you install Sage, you get:

G95	Open source Fortran 95 compiler
GAP	Groups, Algorithms, Programming
GD	Dynamic graphics generation tool
Genus2reduction	Curve data computation
Gfan	Gröbner fans and tropical varieties
Givaro	C++ library for arithmetic and algebra
GMP	GNU Multiple Precision Arithmetic Library
GMP-ECM	Elliptic Curve Method for Integer Factorization
GNU TLS	Secure networking
GSL	Gnu Scientific Library
JsMath	JavaScript implementation of LaTeX

Sage is a *distribution* of software

When you install Sage, you get:

IML	Integer Matrix Library
IPython	Interactive Python shell
LAPACK	Fortan 77 linear algebra library
Lcalc	L-functions calculator
Libgcrypt	General purpose cryptographic library
Libgpg-error	Common error values for GnuPG components
Linbox	C++ linear algebra library
Matplotlib	Python plotting library
Maxima	computer algebra system
Mercurial	Revision control system
MoinMoin	Wiki

Sage is a *distribution* of software

When you install Sage, you get:

MPFI	Multiple Precision Floating-point Interval library
MPFR	C library for multiple-precision floating-point computations
ECLib	Cremona's Programs for Elliptic curves
NetworkX	Graph theory
NTL	Number theory C++ library
Numpy	Numerical linear algebra
OpenCDK	Open Crypto Development Kit
PALP	A Package for Analyzing Lattice Polytopes
PARI/GP	Number theory calculator
Pexpect	Pseudo-tty control for Python
PNG	Bitmap image support

Sage is a *distribution* of software

When you install Sage, you get:

PolyBoRi	Polynomials Over Boolean Rings
PyCrypto	Python Cryptography Toolkit
Python	Interpreted language
Qd	Quad-double/Double-double Computation Package
R	Statistical Computing
Readline	Line-editing
Rpy	Python interface to R
Scipy	Python library for scientific computation
Singular	fast commutative and noncommutative algebra
Scons	Software construction tool
SQLite	Relation database

Sage is a *distribution* of software

When you install Sage, you get:

Sympow	L-function calculator
Symmetrica	Representation theory
Sympy	Python library for symbolic computation
Tachyon	lightweight 3d ray tracer
Termcap	for writing portable text mode applications
Twisted	Python networking library
Weave	Tools for including C/C++ code within Python
Zlib	Data compression library
ZODB	Object-oriented database

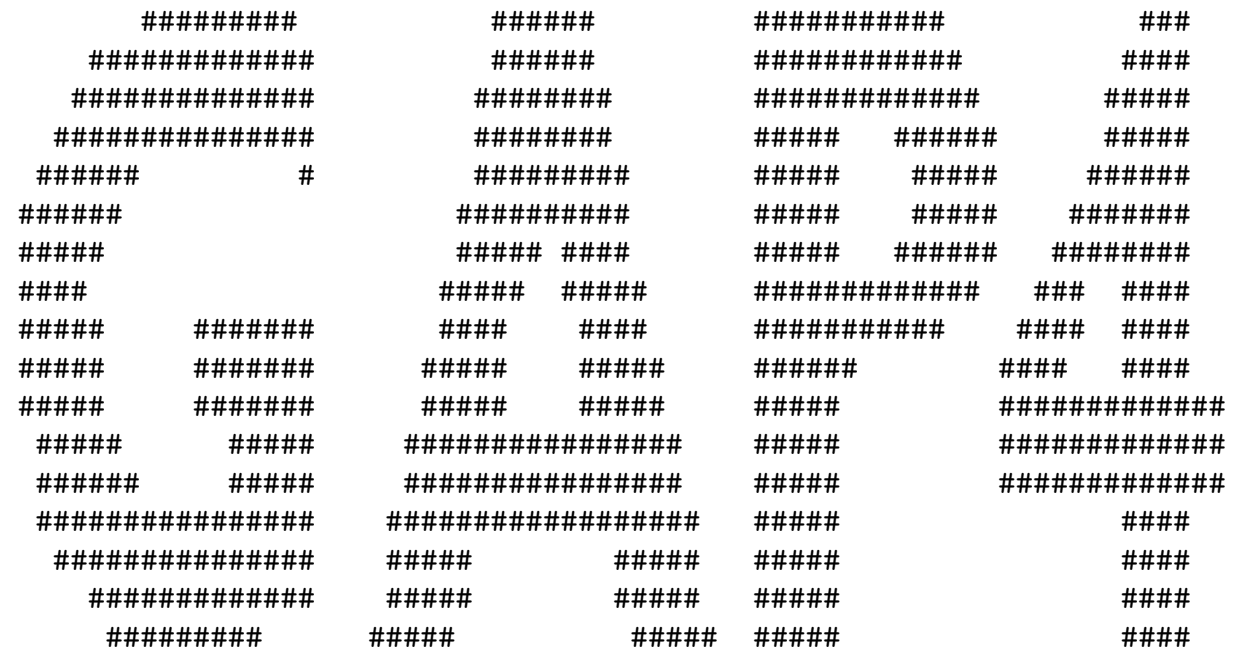
Sage is a *distribution* of software

When you install Sage, you get:

Sympow	L-function calculator
Symmetrica	Representation theory
Sympy	Python library for symbolic computation
Tachyon	lightweight 3d ray tracer
Termcap	for writing portable text mode applications
Twisted	Python networking library
Weave	Tools for including C/C++ code within Python
Zlib	Data compression library
ZODB	Object-oriented database

. . . and more!

karkwa: sage -gap



```
   Information at:  http://www.gap-system.org
   Try '?help' for help. See also  '?copyright' and  '?authors'

  Loading the library. Please be patient, this may take a while.
GAP4, Version: 4.4.10 of 02-Oct-2007, x86_64-unknown-linux-gnu-gcc

gap>
```

```
karkwa: sage -singular

                     SINGULAR                                 /  Development
 A Computer Algebra System for Polynomial Computations       /   version 3-1-0
                                                            0<
     by: G.-M. Greuel, G. Pfister, H. Schoenemann             \   Mar 2009
FB Mathematik der Universitaet, D-67653 Kaiserslautern         \
>
```

```
karkwa: sage -maxima

Maxima 5.16.3 http://maxima.sourceforge.net
Using Lisp ECL 9.4.1
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1)
```

karkwa: sage -gp

```
                GP/PARI CALCULATOR Version 2.3.3 (released)
          amd64 running linux (x86-64/GMP-4.2.1 kernel) 64-bit version
           compiled: Jul 10 2009, gcc-4.3.2 (Ubuntu 4.3.2-1ubuntu12)
               (readline v5.2 enabled, extended help available)

                        Copyright (C) 2000-2006 The PARI Group

PARI/GP is free software, covered by the GNU General Public License, and
comes WITHOUT ANY WARRANTY WHATSOEVER.

Type ? for help, \q to quit.
Type ?12 for how to get moral (and possibly technical) support.

parisize = 8000000, primelimit = 500000
?
```

karkwa: sage -R

```
R version 2.6.1 (2007-11-26)
Copyright (C) 2007 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Sage is . . .

a distribution of software

for *mathematics research*

Sage is software for *mathematics research*

Type of mathematics	Included Software
Algebra	GAP, Maxima, Singular, ...
Exact linear algebra	Linbox, IML, ...
Numerical linear algebra	GSL, Scipy, Numpy, ...
Arbitrary precision arithmetic	GMP, MPFR, MPFI, NTL, ...
Calculus	Maxima, Sympy, ...
Combinatorics	Symmetrica, *-combinat, ...
Algebraic geometry	Singular, ...
Arithmetic geometry	PARI, NTL, mwrank, ecm, ...
Graph theory	NetworkX, ...
Group theory	GAP, ...
⋮	⋮

Sage *combines* the power of many existing programs.

Sage *combines* software

[This example is from a talk by William Stein]

Sage *combines* software

[This example is from a talk by William Stein]

Construct an elliptic curve using *John Cremona's table*:

```
sage: E = EllipticCurve('389a')
```

Sage *combines* software

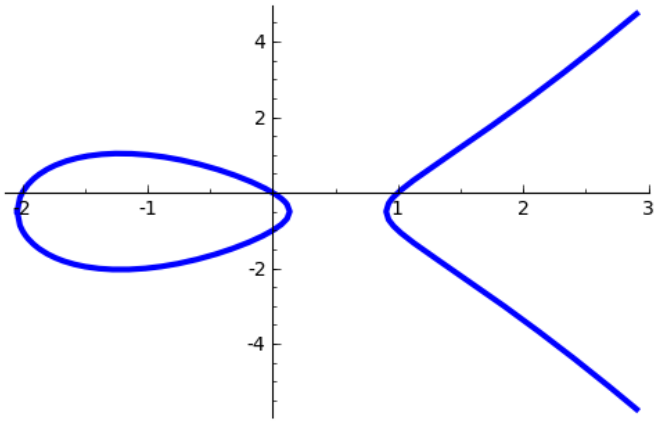
[This example is from a talk by William Stein]

Construct an elliptic curve using *John Cremona's table*:

```
sage: E = EllipticCurve('389a')
```

Use *matplotlib* to plot it:

```
sage: plot(E,thickness=3)
```



Sage *combines* software

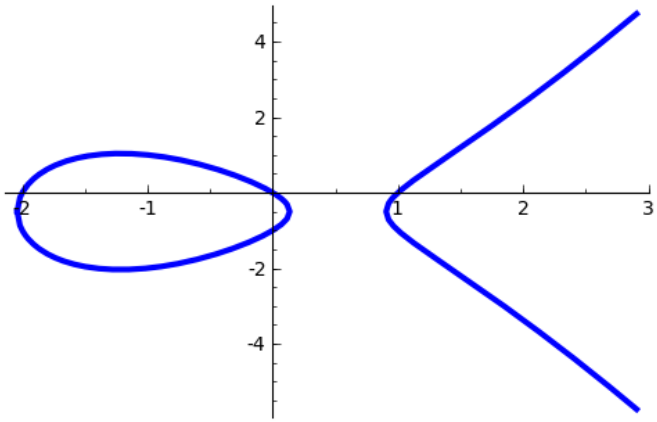
[This example is from a talk by William Stein]

Construct an elliptic curve using *John Cremona's table*:

```
sage: E = EllipticCurve('389a')
```

Use *matplotlib* to plot it:

```
sage: plot(E,thickness=3)
```



mwrank to do a 2-descent:

```
sage: E.mwrank()
Curve [0,1,1,-2,0] :     Rank = 2
```

Sage *combines* software

PARI to compute Fourier coefficients a_n:

```
sage: E.anlist(15)
[0, 1, -2, -2, 2, -3, 4, -5, 0, 1, 6, -4, -4, -3, 10, 6
```

Sage *combines* software

PARI to compute Fourier coefficients a_n:

```
sage: E.anlist(15)
[0, 1, -2, -2, 2, -3, 4, -5, 0, 1, 6, -4, -4, -3, 10, 6
```

lcalc to compute zeros in the critical strip of the L-series:

```
sage: E.lseries().zeros(5)
[0.000000000, 0.000000000, 2.87609907, 4.41689608, 5.79
```

Sage *combines* software

PARI to compute Fourier coefficients a_n:

```
sage: E.anlist(15)
[0, 1, -2, -2, 2, -3, 4, -5, 0, 1, 6, -4, -4, -3, 10, 6
```

lcalc to compute zeros in the critical strip of the L-series:

```
sage: E.lseries().zeros(5)
[0.000000000, 0.000000000, 2.87609907, 4.41689608, 5.79
```

sympow to compute the modular degree:

```
sage: E.modular_degree()
40
```

Sage *combines* software

PARI to compute Fourier coefficients a_n:

```
sage: E.anlist(15)
[0, 1, -2, -2, 2, -3, 4, -5, 0, 1, 6, -4, -4, -3, 10, 6
```

lcalc to compute zeros in the critical strip of the L-series:

```
sage: E.lseries().zeros(5)
[0.000000000, 0.000000000, 2.87609907, 4.41689608, 5.79
```

sympow to compute the modular degree:

```
sage: E.modular_degree()
40
```

Magma to compute the rank of the 3-selmer group:

```
sage: magma(E).ThreeSelmerGroup()
```

Sage uses *Python* as its programming language.

Sage uses *Python*

Sage uses *Python*, which is an interpreted, modern and powerful programming language.

Sage uses *Python*

Sage uses *Python*, which is an interpreted, modern and powerful programming language.

Interpreted means that it works like MuPAD, Maple, Mathematica, ...

```
sage: 2 + 3
5
```

Sage uses *Python*

- Sage = Python + a big Python library
- Python is one of top 5 most used programming languages, with millions of users.
- Tens of thousands of third party packages are immediately available to you.
- Sage may be the first successful math software system to not invent its own new language just for mathematics.

Sage uses *Python*

Python is easy to read:

mathematics:

$$\left\{ 8A \;\middle|\; A \in \{0, 1, \ldots, 10\} \text{ if } A \text{ is odd} \right\}$$

python:

```
[8*A for A in range(10) if A%2 == 1]
```

Sage uses *Python*

Python is easy to learn:

- *Python Tutorial*

  `docs.python.org`

- *Dive Into Python*

  `www.diveintopython.org`

- *Sage Tutorial*

  `sagemath.org/doc/tutorial`

Sage is . . .

a distribution of software

for mathematics research

licensed under the GPL

Sage is *open-source software*

You have the freedom:

- to run the program, for any purpose.
- to study how the program works, and adapt it to your needs.
- to redistribute copies so you can help your neighbour.
- to improve the program, and release your improvements to the public, so that the whole community benefits.

Sage is *open-source software*

"You can read Sylow's Theorem and its proof in Huppert's book in the library . . . then you can use Sylow's Theorem for the rest of your life free of charge, but for many computer algebra systems license fees have to be paid regularly . . . .

With this situation two of the most basic rules of conduct in mathematics are violated: In mathematics information is passed on free of charge and everything is laid open for checking."

— *J. Neubüser (1993)*

(started GAP in 1986)

Sage is *open-source software*

"I think, fundamentally, open source does tend to be more stable software. It's the right way to do things. I compare it to science versus witchcraft.

In science, the whole system builds on people looking at other people's results and building on top of them.

In witchcraft, somebody had a small secret and guarded it – but never allowed others to really understand it and build on it.

Traditional software is like witchcraft. In history, witchcraft just died out. The same will happen in software. When problems get serious enough, you can't have one person or one company guarding their secrets. You have to have everybody share in knowledge."

— Linus Torvalds

Sage is *open-source software*

"No closed-source developer can match the pool of talent the Linux community can bring to bear on a problem. Perhaps in the end the open-source culture will triumph not because cooperation is morally right or software "hoarding" is morally wrong (assuming you believe the latter, which neither Linus nor I do), but simply because the closed-source world cannot win an evolutionary arms race with open-source communities that can put orders of magnitude more skilled time into a problem."

— *Eric S. Raymond*

Mission

To create a viable free open-source alternative to Magma, Maple, Mathematica, Matlab [and MuPAD].

Some history of the Sage project

- *1999-2005.* William Stein wrote over 25,000 lines of Magma code for his research. Decided that Magma was a bad long term investment since he couldn't see of modify the internals.
- *Jan. 2005.* William Stein started Sage.
- *Feb. 2005.* SAGE versione 0.1: a Python library gluing together PARI, Maxima, Python, Singular e GAP.
- *Feb. 2006.* SAGE versione 1.0 released; and the "first annual" Sage Days workshop.

Some history of the Sage project

- *Nov. 2007.* Sage won first prize in Les Trophées du Libre (the competition honours the best existing free software)
- *$$$:* Univ. of Washington, NSF, DoD, Google, Sun, Research Centres, Universities, private donations, etc.
- *Current version:* Sage-4.2

Sage Days!

- Intensive workshop to develop and implement new features.
- Developed software is made freely available as part of Sage.
- Sufficiently novel algorithms are submitted for publication.

There have been over 20 Sage Days workshops so far!

Upcoming Sage Days

- Sage Days 18: Cambridge, MA (December 2009)
- Sage Days 19: Seattle, WA (January 2010)
- Sage Days 20: Marseille (February 2010)
- Sage Days 20.5 (?): Fields Institute (May 2010)
- Sage Days 21: Seattle, WA (June 2010)
- Sage Days 22: Berkeley, CA (July 2010)
- Sage Days 23: Leiden, Netherlands (July 2010)
- Sage Days 24: RISC, Linz, Austria (July 2010)
- Sage Days 25: Mumbai, India (August 2010)

Useful features

LaTeX

In this LaTeX file, I typed:

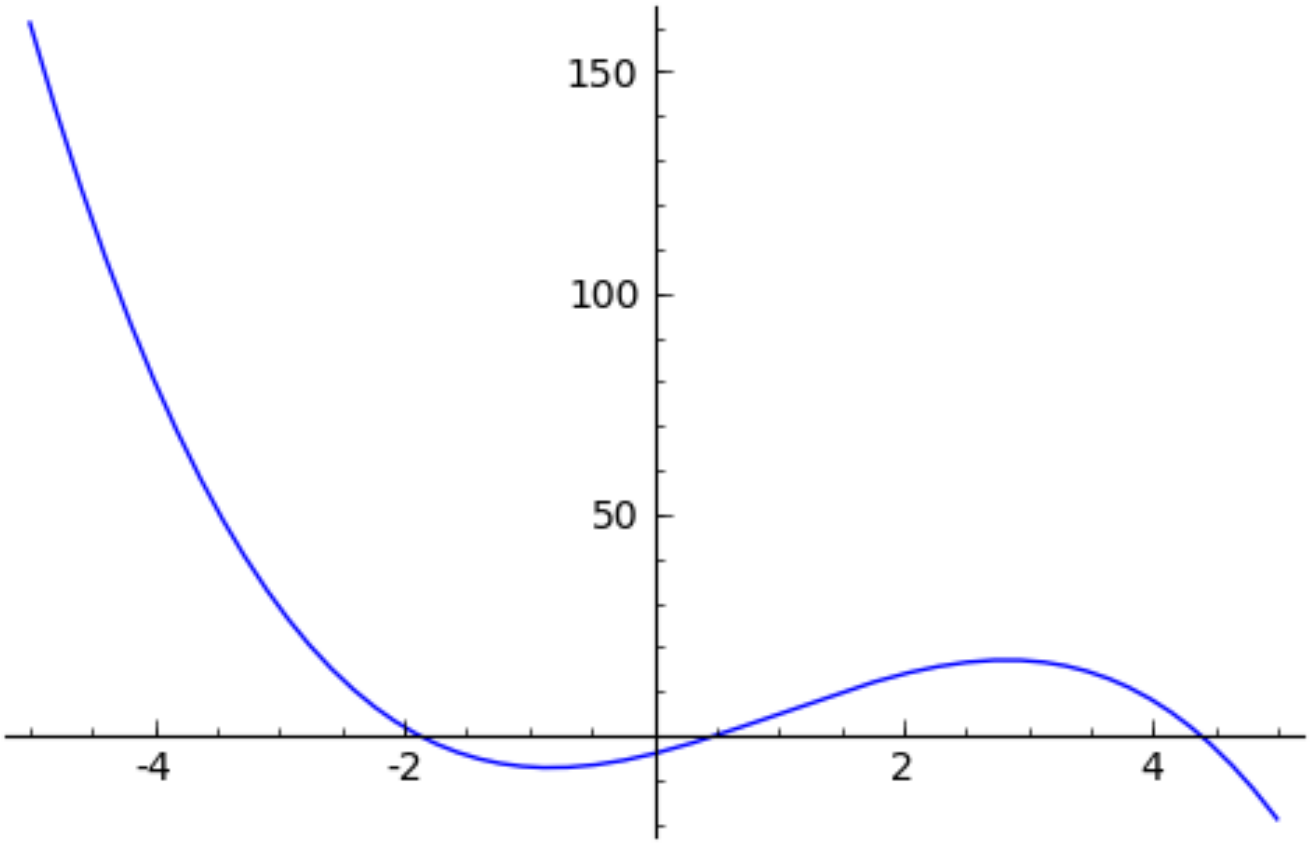
```
\sageplot{plot(-x^3+3*x^2+7*x-4,-5,5)}
```

LaTeX

In this LaTeX file, I typed:

```
\sageplot{plot(-x^3+3*x^2+7*x-4,-5,5)}
```

and it got replaced by:



LaTeX

In this LaTeX file:

```
\begin{sagesilent}
  sigma = Permutation([7,3,1,5,2,6,8,4])
  P, Q = sigma.robinson_schensted()
\end{sagesilent}

Let $\sigma = \sage{sigma}$. The
RSK algorithm produces
the tableaux:
    \[\sage{P} \quad \sage{Q}\]
```

LaTeX

It got replaced with:

Let $\sigma = [7, 3, 1, 5, 2, 6, 8, 4]$. The RSK algorithm produces the tableaux:

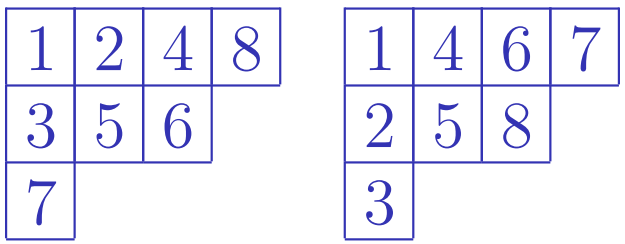
1	2	4	8
3	5	6	
7			

1	4	6	7
2	5	8	
3			

LaTeX

It got replaced with:

Let $\sigma = [7, 3, 1, 5, 2, 6, 8, 4]$. The RSK algorithm produces the tableaux:



This is done with the *sagetex* package, written by Dan Drake. Of course, the package is included with Sage.

Notebook interface

plotex (sage_notebook)

http://localhost:8000/plotex

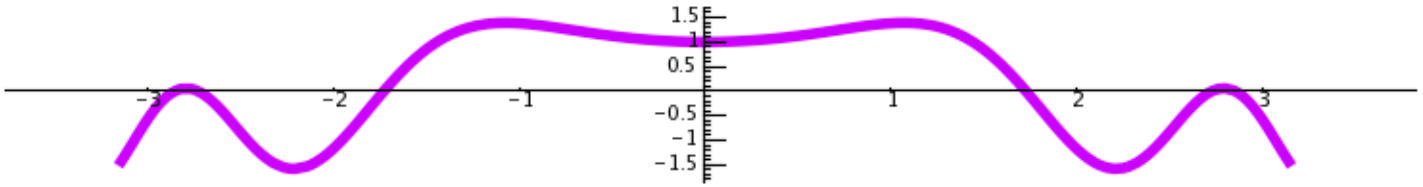
/. 8200 9000 aws sagenb cnn di digg expedia Google Calendar Gmail HOME licenses m msri07 n0 n1 n2 nyt

Interrupt | Restart | History | Left Panel | Help | Documentation | Slideshow

Worksheet: plotex

Edit | Text | Print | Evaluate All | Hide | Show | Upload | Download

```
show(plot(sin(x^2)+cos(x), -pi, pi, hue=0.8, thickness=4), figsize=[8,1])
```




```
plot?
```

```
Type:        <type 'instance'>
Definition:  plot( [noargspec] )
Docstring:

    Use plot by writing

        plot(X, ...)

    where X is a SAGE object that either is callable and returns
    numbers that can be coerced to floats, or has a plot method
    that returns a GraphicPrimitive object.

    Type plot.options for a dictionary of the default
    options for plots.  You can change this to change
    the defaults for all future plots.  Use plot.reset()
    to reset to the default options.
```

Includes extensive and beautiful documentation

Yoda!

@interact!

Cython