# Course Outline

- Introduction to Sage
  - overview of the software
- Sage demonstration
  - The Sage notebook
  - Getting help
  - Interfaces
  - Matrices
  - Calculus
- Basic programming in Sage
  - Python lists

# Sage is a *distribution* of open-source software.

# Sage is a *distribution* of open-source software.

Software included with Sage :

| | |
|---|---|
| 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 open-source software.

Software included with Sage :

| | |
|---|---|
| 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 open-source software.

Software included with Sage :

| | |
|---|---|
| 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 open-source software.

Software included with Sage :

| | |
|---|---|
| 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 open-source software.

Software included with Sage :

| | |
|---|---|
| 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 open-source software.

Software included with Sage :

| | |
|---|---|
| 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 open-source software.

Software included with Sage :

| | |
|---|---|
| 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 |

*Plus additional optional packages*

# Sage is a distribution of *mathematics* software.

*Sage's mission: "Creating a viable, free, open-source alternative to Magma$^{TM}$, Maple$^{TM}$, Mathematica$^{TM}$, and Matlab$^{TM}$."*

# Sage is a distribution of *mathematics* software.

*Sage's mission: "Creating a viable, free, open-source alternative to Magma$^{TM}$, Maple$^{TM}$, Mathematica$^{TM}$, and Matlab$^{TM}$."*

| | |
|---:|---|
| Algebra | GAP, Maxima, Singular |
| Algebraic Geometry | Singular, Macaulay2 |
| Arbitrary Precision Arithmetic | GMP, MPFR, MPFI, NTL, . . . |
| Arithmetic Geometry | PARI, NTL, mwrank, ecm, . . . |
| Calculus | Maxima, Sympy |
| Combinatorics | Symmetrica, MuPAD-Combinat* |
| Exact Linear Algebra | Linbox, IML |
| Graph Theory | NetworkX |
| Graphics | MatPlotLib, Tachyon3d |
| Group theory | GAP |
| Numerical Linear Algebra | GSL, Scipy, Numpy |

# Sage is a distribution of *free, 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 . . . .*

# Sage is a distribution of *free, 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."*

# Sage is a distribution of *free, 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 a distribution of *free, 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 a distribution of *free, 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.

Also, you don't have to pay for it.

# The Sage programming language is *Python*

*Python* is a powerful, modern, interpreted programming-language.

# The Sage programming language is *Python*

*Python* is a powerful, modern, interpreted programming-language.

- *Interpreted* means it works like Maple or Mathematica.

```
python: x = 17
python: x
17
python: x**2
289
```

# The Sage programming language is *Python*

*Python* is a powerful, modern, interpreted programming-language.

- *Interpreted* means it works like Maple or Mathematica.

```
python: x = 17
python: x
17
python: x**2
289
```

- It's *easy to learn*. Lots of free documentation.

```
http://diveintopython.org/
http://docs.python.org/tut/
```

# The Sage programming language is *Python*

# The Sage programming language is *Python*

- It's *easy to read and write*.

$$\left\{ 17x \;\middle|\; x \in \{0, 1, \ldots, 9\} \text{ and } x \text{ is odd} \right\}$$

```
python: [17*x for x in range(0,10) if x%2 == 1]
```

# The Sage programming language is *Python*

- It's *easy to read and write*.

$$\left\{ 17x \;\middle|\; x \in \{0,1,\ldots,9\} \text{ and } x \text{ is odd} \right\}$$

```
python: [17*x for x in range(0,10) if x%2 == 1]
```

- Lots of *Python libraries*: databases, graphics, networking, ...

# The Sage programming language is *Python*

- It's *easy to read and write*.

$$\left\{ 17x \;\middle|\; x \in \{0, 1, \ldots, 9\} \text{ and } x \text{ is odd} \right\}$$

```
python: [17*x for x in range(0,10) if x%2 == 1]
```

- Lots of *Python libraries*: databases, graphics, networking, . . .
- It is easy to use *C/C++ libraries* from within Python.

# The Sage programming language is *Python*

- It's *easy to read and write*.

$$\left\{ 17x \;\middle|\; x \in \{0,1,\ldots,9\} \text{ and } x \text{ is odd} \right\}$$

```
python: [17*x for x in range(0,10) if x%2 == 1]
```

- Lots of *Python libraries*: databases, graphics, networking, ...
- It is easy to use *C/C++ libraries* from within Python.
- *Cython*: Python code $\longrightarrow$ compiled C code.

# The Sage programming language is *Python*

*"Google has made no secret of the fact they use Python a lot for a number of internal projects. Even knowing that, once I was an employee, I was amazed at how much Python code there actually is in the Google source code system."*

*— Guido van Rossum*

*(creator of Python)*

# Several ways to use Sage

- A library for Python scripts.

```
#!/usr/bin/env sage -python

import sys
from sage.all import *
```

# Several ways to use Sage

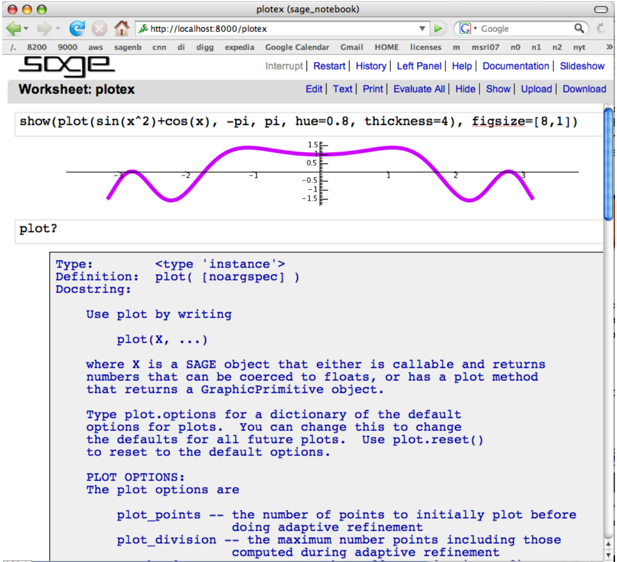
- Command line interface.

```
----------------------------------------------------------------------
| SAGE Version 2.10.1, Release Date: 2008-02-02                      |
| Type notebook() for the GUI, and license() for information.        |
----------------------------------------------------------------------

sage: 17^2
289
sage: |
```

# Several ways to use Sage

- Graphical notebook: try it online at `www.sagenb.org`.



# Sage plays well with LaTeX

LaTeX input:

```
\begin{sagesilent}
  var('s t')
  f = t^2*e^t-sin(t)
\end{sagesilent}

Let $f(t)=\sage{f}$. Then the Laplace tranform
of $f$ is: $\sage{f.laplace(t,s)}$.
```

# Sage plays well with LaTeX

LaTeX input:

```
\begin{sagesilent}
  var('s t')
  f = t^2*e^t-sin(t)
\end{sagesilent}

Let $f(t)=\sage{f}$. Then the Laplace tranform
of $f$ is: $\sage{f.laplace(t,s)}$.
```

LaTeX output:

*Let* $f(t) = t^2e^t - \sin(t)$ *. Then the Laplace tranform of* $f$ *is:* $\frac{2}{(s-1)^3} - \frac{1}{s^2+1}$.

# Sage plays well with LaTeX

LaTeX input:

```
Here is an example of a tree:
\sageplot{Graph({0:[1,2,3], 2:[5]}).plot()}
```

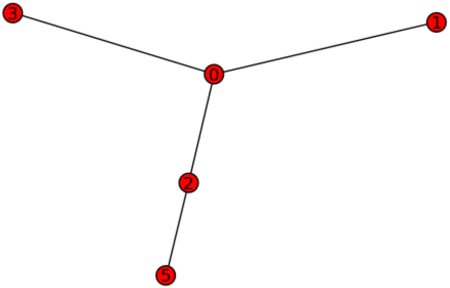
# Sage plays well with LaTeX

LaTeX input:

```
Here is an example of a tree:
\sageplot{Graph({0:[1,2,3], 2:[5]}).plot()}
```

LaTeX output:

Here is an example of a tree:



# Sage plays well with $\LaTeX$

$\LaTeX$ input:

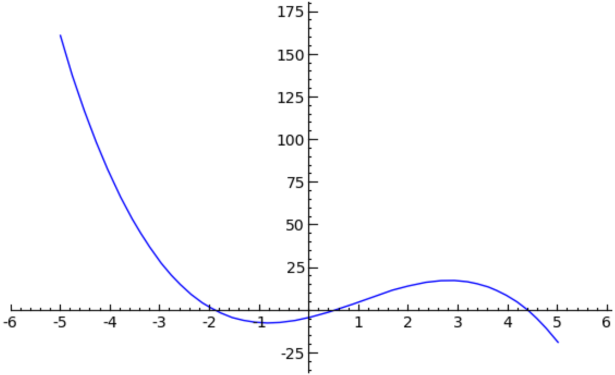
```
\sageplot{plot(-x^3+3*x^2+7*x-4,-5,5)}
```

# Sage plays well with LaTeX

LaTeX input:

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

LaTeX output:



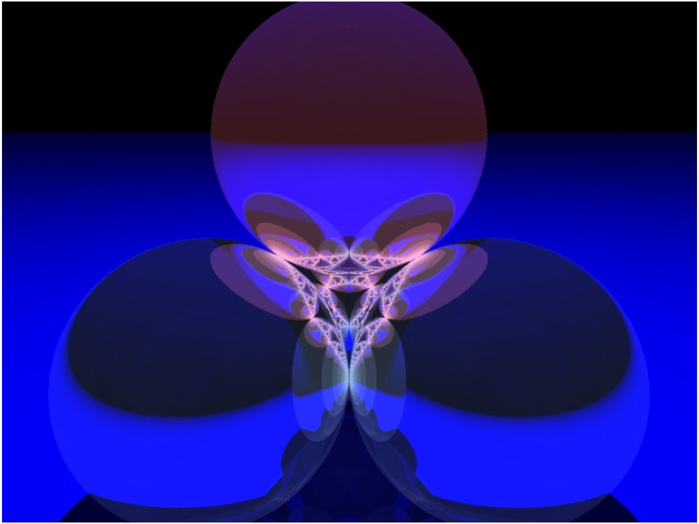
# Sage plays well with LaTeX

LaTeX input:

```
\begin{sagesilent}
  t6 = Tachyon(camera_center=(0,-4,1), xres = 800, yres = 600, \
               raydepth = 12, aspectratio=.75, antialiasing = True)
  t6.light((0.02,0.012,0.001), 0.01, (1,0,0))
  t6.light((0,0,10), 0.01, (0,0,1))
  t6.texture('s', color = (.8,1,1), opacity = .9, specular = .95, \
             diffuse = .3, ambient = 0.05)
  t6.texture('p', color = (0,0,1), opacity = 1, specular = .2)
  t6.sphere((-1,-.57735,-0.7071),1,'s')
  t6.sphere((1,-.57735,-0.7071),1,'s')
  t6.sphere((0,1.15465,-0.7071),1,'s')
  t6.sphere((0,0,0.9259),1,'s')
  t6.plane((0,0,-1.9259),(0,0,1),'p')
\end{sagesilent}
\sageplot{t6}
```

# Sage plays well with $\LaTeX$

$\LaTeX$ output:



# The Sage community

- Many people have contributed to Sage (directly & indirectly).
- There are several mailing lists.

  `http://www.sagemath.org`
- IRC: `#sage-devel` on `freenode.org`.
- Developers are very friendly and helpful.

Let's use Sage