Sage

Thierry Dumont
ICJ

9 Décembre 2010

`http://www.sagemath.org/`

Mission : Creating a viable free open source alternative to Magma, Maple, Mathematica and Matlab.

Sage is a free open-source mathematics software system licensed under the GPL. It combines the power of many existing open-source packages into a common Python-based interface.

`http://www.sagemath.org/`

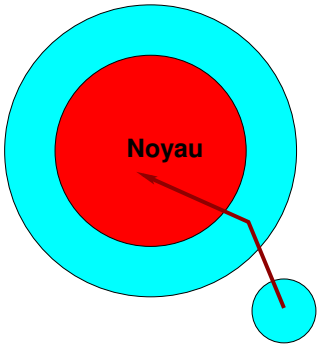
Mission : Creating a viable free open source alternative to Magma, Maple, Mathematica and Matlab.

Sage is a free open-source mathematics software system licensed under the GPL. It combines the power of many existing open-source packages into a common Python-based interface.

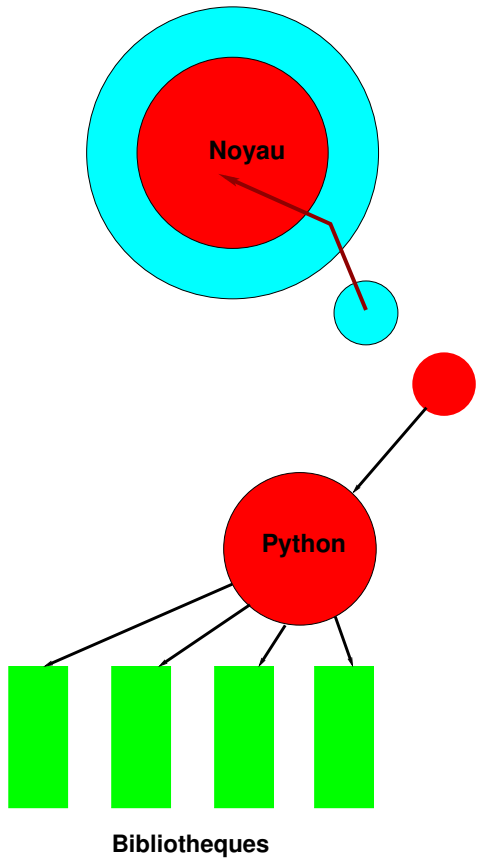
Software for Algebraic and Geometric Experimentation.

Sauge officinale

Conception



Conception



Bibliothèques / Logiciels interfacés

Calcul :

- GAP
- Maxima
- Pynac
- Singular
- Pari
- Linbox
- Scipy
- ...

Bibliothèques / Logiciels interfacés

Calcul :

- GAP
- Maxima
- Pynac
- Singular
- Pari
- Linbox
- Scipy
- ...

Bibliothèques / Logiciels interfacés

Rendu :

- matplotlib
- jmol
- LaTex (non inclus)
- ...

Bibliothèques / Logiciels interfacés

Rendu :

- matplotlib
- jmol
- LaTex (non inclus)
- ...

Web et réseau :

- Jinja
- Twisted
- ...

Tout est libre !

Un système de calcul + un serveur web (notebook)

Utilisation en ligne de commande ou avec le serveur web .
Autres interfaces possibles (Ex. KANTOR sous KDE).

Installation

Binaires disponibles. Mieux : source.

Principe général : le logiciel dépend le moins possible des logiciels de la machine hote.

Sage contient sa propre distribution Python, tous les packages Python utilisés, les bibliothèques comme lapack etc. sont compilés.

Installation

Binaires disponibles. Mieux : source.

Principe général : le logiciel dépend le moins possible des logiciels de la machine hote.

Sage contient sa propre distribution Python, tous les packages Python utilisés, les bibliothèques comme lapack etc. sont compilés.

Fichiers spkg : sources et scripts d'installation.
Distribution de base et spkg optionnels.

`sage -i package.spkg` (téléchargement + installation).

Réalisation des spkg simple.

Extensible *à l'infini.*

Programmation objet et objets mathématiques

- Langage fortement typé ! Aucun objet n'a de type ambigu. Type $=>$ aiguillage vers la bonne méthode et/ou la bonne bibliothèque.

Programmation objet et objets mathématiques

- Langage fortement typé ! Aucun objet n'a de type ambigu. Type => aiguillage vers la bonne méthode et/ou la bonne bibliothèque.
- Classes à formes normales : permet la simplification, la comparaison etc... des objets ;
  exemple type : QQ l'ensemble des nombres rationnels.

Programmation objet et objets mathématiques

- Langage fortement typé ! Aucun objet n'a de type ambigu. Type => aiguillage vers la bonne méthode et/ou la bonne bibliothèque.
- Classes à formes normales : permet la simplification, la comparaison etc... des objets ;
  exemple type : QQ l'ensemble des nombres rationnels.

Utilisation de *tout* Python ! Classes, Méta-Classes, décorateurs...

On ne peut pas calculer sans savoir dans quel ensemble on calcule !

Programmation objet et objets mathématiques

- Langage fortement typé ! Aucun objet n'a de type ambigu. Type => aiguillage vers la bonne méthode et/ou la bonne bibliothèque.
- Classes à formes normales : permet la simplification, la comparaison etc... des objets ;
  exemple type : QQ l'ensemble des nombres rationnels.

Utilisation de *tout* Python ! Classes, Méta-Classes, décorateurs...

On ne peut pas calculer sans savoir dans quel ensemble on calcule !

Exemples :

- Matrices : dans quel ensemble sont les coefficients ?
- Polynômes ; exemple $\mathbb{Q}[x]$.
- ...

Ensembles de nombres

- entiers relatifs `ZZ`.
- rationnels : `QQ`.

Ensembles de nombres

- entiers relatifs `ZZ`.
- rationnels : `QQ`.
- flottants doubles `RDF`.
- flottants de précision quelconque : instanciation de `RealField`.
- `CDF` et `ComplexField`.

Ensembles de nombres

- entiers relatifs `ZZ`.
- rationnels : `QQ`.
- flottants doubles `RDF`.
- flottants de précision quelconque : instanciation de `RealField`.
- `CDF` et `ComplexField`.
- anneau des entiers modulo n : `IntegerModRing(97)`.
- nombre $p-$adiques.

Exemple : matrices et vecteurs

```
MS=MatrixSpace(ZZ,2,3).
q=MS([[1,2,3],[3,2,1]]).
```

Exemple : matrices et vecteurs

```
MS=MatrixSpace(ZZ,2,3).
q=MS([[1,2,3],[3,2,1]]).
```

ou bien :

```
q=matrix(ZZ,2,3,[[1,2,3],[3,2,1]]).
```

Exemple : matrices et vecteurs

```
MS=MatrixSpace(ZZ,2,3).
q=MS([[1,2,3],[3,2,1]]).
```

ou bien :

```
q=matrix(ZZ,2,3,[[1,2,3],[3,2,1]]).
```

- `q=matrix(QQ,2,3,[[1,2,3],[3,2,1]])` => coefficients rationnels => systèmes résolus avec LinBox .
- `q=matrix(RDF,2,3,[[1,2,3],[3,2,1]])` => coefficients rationnels => systèmes résolus avec Lapack (scipy).

Exemple : matrices et vecteurs

```
MS=MatrixSpace(ZZ,2,3).
q=MS([[1,2,3],[3,2,1]]).
```

ou bien :

```
q=matrix(ZZ,2,3,[[1,2,3],[3,2,1]]).
```

- `q=matrix(QQ,2,3,[[1,2,3],[3,2,1]])` => coefficients rationnels => systèmes résolus avec LinBox .
- `q=matrix(RDF,2,3,[[1,2,3],[3,2,1]])` => coefficients rationnels => systèmes résolus avec Lapack (scipy).

De même :

`VS=VectorSpace(QQ,2)` et `q=vector(ZZ,2,[0,1])`.

Z ! les ensembles de matrices ont leurs coefficients dans des anneaux, les ensembles de vecteurs dans des corps.

Exemple : polynômes

```
R.<x>=PolynomialRing(GF(5),'x')
```

Exemple : polynômes

```
R.<x>=PolynomialRing(GF(5),'x')

R.random_element()
```

Exemple : polynômes

```
R.<x>=PolynomialRing(GF(5),'x')

R.random_element()

A=random_matrix(R,2,2)
```

Exemple : polynômes

```
R.<x>=PolynomialRing(GF(5),'x')

R.random_element()

A=random_matrix(R,2,2)

B=random_matrix(R,2,1)
```

Exemple : polynômes

```
R.<x>=PolynomialRing(GF(5),'x')

R.random_element()

A=random_matrix(R,2,2)

B=random_matrix(R,2,1)

A.solve_right(B)
```

Calcul formel

Tout est typé !

```
sage: var('x')
u=sin(20*cos(x))
```