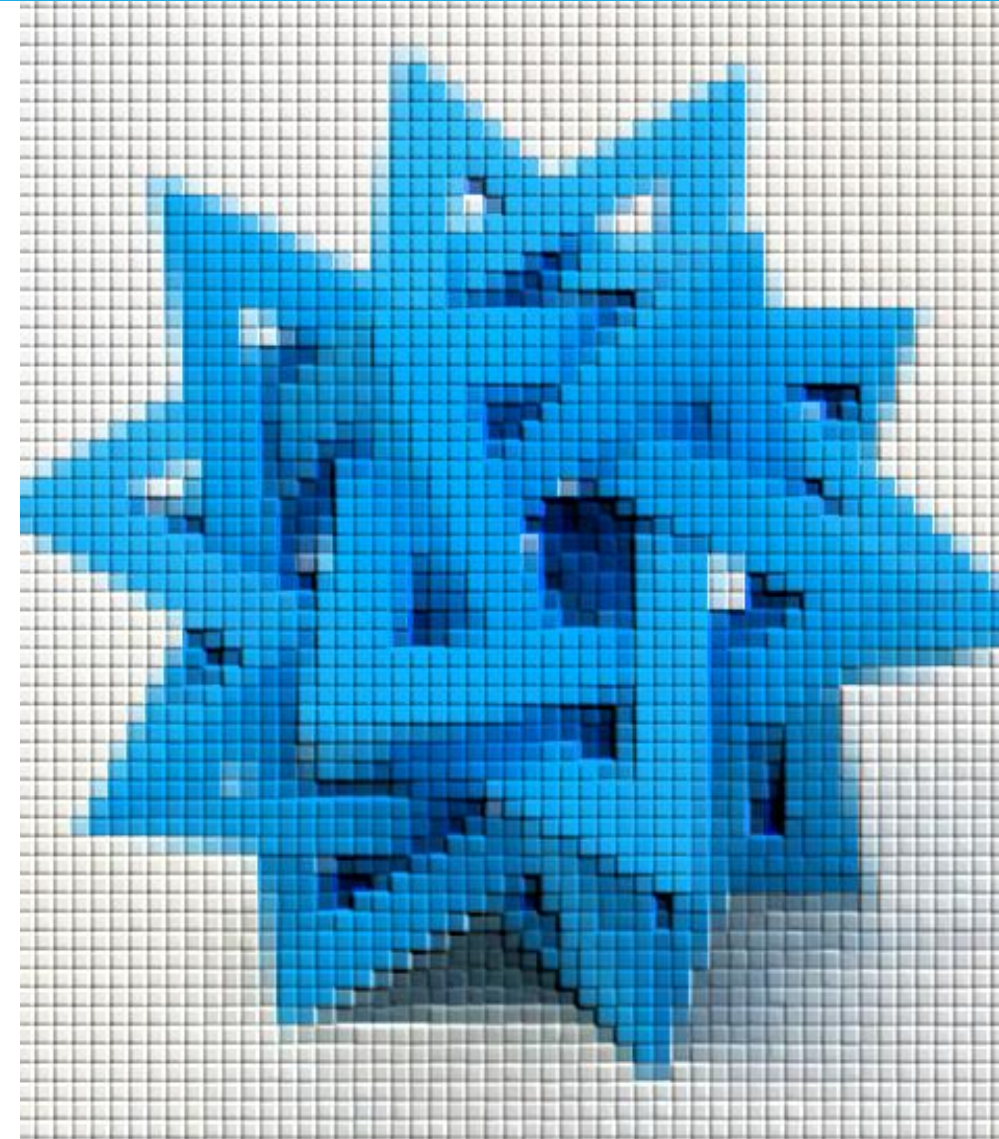


from python import ...more

Calcolo Simbolico

ESPLORAZIONI IN PYTHON PER I PIÙ CURIOSI



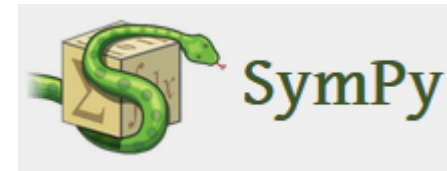
Obiettivo

COSA VOGLIAMO FARE?

- Risolvere espressioni algebriche
- Risolvere limiti, integrali, derivate, ...
- In modo «simbolico», manipolando direttamente le espressioni

QUALI STRUMENTI USIAMO?

- Libreria SymPy
- <https://www.sympy.org/>

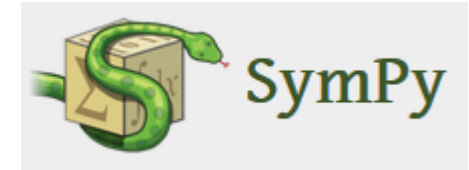


SymPy



- «SymPy is a Python library for symbolic mathematics. It aims to become a full-featured computer algebra system (CAS) while keeping the code as simple as possible in order to be comprehensible and easily extensible.
- SymPy is written entirely in Python. »
- Installare il modulo **sympy**

SymPy



The screenshot shows the SymPy website homepage. At the top left is the SymPy logo. A navigation bar contains links for Main Page, Features, Download, Documentation, Support, Development, Roadmap, Donate, and Online Shell. The main content area is divided into several sections: 'About' with a description of SymPy as a Python library for symbolic mathematics; 'Why SymPy' with bullet points about its license, Python-based nature, and lightweight dependency; 'Projects using SymPy' with a list of projects like Cadabra, ChemPy, EinsteinPy, galgebra, and Lcapy; 'Compute with Gamma' with a text input field containing a mathematical expression and a 'Compute' button; 'Download Now' with links for the latest and development versions; 'Quick Links' with a list of links to documentation, source code, and other resources; and 'Follow Us' with a social media follow button.

<https://www.sympy.org/>

The screenshot shows the SymPy documentation website. The header includes the SymPy logo and the text 'Welcome to SymPy's documentation!'. Below the header, there is a navigation bar with links for 'next', 'modules', and 'index'. The main content area is divided into several sections: 'Documentation version' showing the current version (SymPy 1.6.2) and the development version (SymPy 1.8.dev); 'Next topic' with a link to 'Installation'; 'This Page' with a link to 'Show Source'; 'Quick search' with a search input field and a 'Go' button; and a 'Contents' section with a list of links to various documentation topics, including Installation, SymPy Tutorial, Gotchas and Pitfalls, and SymPy Modules Reference.

<https://docs.sympy.org/>

Variabili simboliche

```
from sympy import symbols  
  
x = symbols('x')  
a, b, c = symbols('a b c')
```



Espressioni

```
from sympy import sqrt, pprint

delta = b**2 - 4*a*c
sol1 = (-b + sqrt(delta))/(2*a)

pprint(delta)
pprint(sol1)
```



Equazioni

```
from sympy import Eq

eq = Eq(x**2+1, x**3-3)

result = solve(eq, x)
```



Funzioni

```
from sympy import sin, exp, sqrt
```



```
f = sin(x) / x
```

```
g = exp(sqrt(x))
```


Limiti

```
from sympy import limit, oo
```

```
lim1 = limit(f, x, 0)
```

```
# funzione, variabile, punto di limite
```

```
lim2 = limit(f, x, +oo)
```



Derivate

```
from sympy import diff
```

```
f1 = diff(f, x)
```

```
# funzione, variabile indipendente
```

```
g1 = diff(g, x)
```



Integrali indefiniti

```
from sympy import integrate
```



```
F = integrate(f, x)
```

```
# funzione, variabile
```

```
G = integrate(g, x)
```

Integrali definiti

```
from sympy import integrate, oo

area_f = integrate(f, (x, -oo, +oo))
# tupla: (variabile, inizio, fine)

area_g = integrate(g, (x, 0, 1))
```



Espansione in serie di Taylor

```
from sympy import series, pi
```



```
esp_0 = series(f, x, 0, 3)
```

```
# funzione, variabile, punto di espansione, ordine
```

```
esp_pi = series(f, x, pi, 3)
```