

SOURCE CODE 1. Symbolisches Rechnen mit sympy.

```
# Das Modul sympy bietet eine Vielzahl an mathematischen Funktionen:  
# Hier bringen wir nur ein paar einfache Beispiele zum Rechnen mit  
# Polynomen; siehe  
#           https://docs.sympy.org  
# für eine detaillierte Dokumentation.  
# Für das "symbolische Rechnen", das sympy zur Verfügung stellt, muß  
# man zunächst die Variablen, die als "Symbole" (also nicht als  
# "normale" Python-Objekte wie Zahlen oder Strings) verwendet werden  
# sollen, deklarieren. Z.B.:  
x,y = sp.symbols('x,y')  
# Dann kann man z.B. ein Polynom in x und y expandieren und im  
# LaTeX-Format ausgeben:  
print(sp.latex(sp.expand((x + y)**6)))  
# Hier als weiteres Beispiel die steigenden Faktoriellen; diesmal  
# ausgegeben als "normaler String" (also nicht im LaTeX-Format):  
print((sp.functions.combinatorial.factorials.RisingFactorial((x + y),6)))  
# Ebenso kann man Gleichungen definieren ...  
left_hand_side = x**2+27*x-91  
right_hand_side = 0  
quadratic_equation = sp.Eq(left_hand_side, right_hand_side)  
print(sp.latex(quadratic_equation))  
# ... und lösen lassen:  
x1,x2 = sp.solvers.solve(quadratic_equation)  
# Faktorisierung des quadratischen Polynoms, wieder im LaTeX-Format:  
print(sp.latex((x-x1)*(x-x2)))  
# Faktorisierung über den ganzen Zahlen funktioniert aber  
# direkt:  
polynomial = x**5 + 3*x**4 - 23*x**3 - 51*x**2 + 94*x + 120  
factorization = sp.factor(polynomial)  
# Ausgabe als String im LaTeX-Format; mit Zeilenumbruch "\n"  
# und Einrückung "\t":  
print(f'\t{sp.latex(polynomial)} =\n\t{sp.latex(factorization)}')
```