lektion1

October 15, 2025

Inhalt

1 Lektion 1

1.1 Einfache Arithmetik

[1]:
```
2*3
```

[1]: 6

[2]:
```
1+2
```

[2]: 3

[3]:
```
1+2*3
```

[3]: 7

[4]:
```
2*(1+2)
```

[4]: 6

[5]:
```
2**3
```

[5]: 8

[6]:
```
1/3
```

[6]:
```
0.3333333333333333
```

[7]:
```
3*1/3
```

[7]:
```
1.0
```

[8]:
```
(1/3)**3-1/3**3
```

[8]:
```
-6.938893903907228e-18
```

[9]:
```
(1/3)**100 * 3**100
```

[9]:
```
0.9999999999999944
```

[10]:
```
0.000001 - 1e-6
```

[10]:
```
0.0
```

1.2 Variablen

Zuweisungsoperator =

[11]:
```
a = 3
```

[12]:
```
a
```

[12]:
```
3
```

[13]:
```
b
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[13], line 1
----> 1 b

NameError: name 'b' is not defined
```

[14]:
```
b = a
```

[15]:
```
b
```

[15]:
```
3
```

[16]:
```
a
```

[16]:
```
3
```

[17]:
```
b
```

[17]: 3

1.3 Symbolische Arithmetik

[18]:
```
from sympy import *
init_printing()
```

[19]:
```
S(1)/3
```

[19]: $\frac{1}{3}$

[20]:
```
S(3)+3
```

[20]: 6

[21]:
```
(S(3)+3)/9
```

[21]: $\frac{2}{3}$

[22]:
```
Rational(1,3)
```

[22]: $\frac{1}{3}$

[23]:
```
Rational(1/2)
```

[23]: $\frac{1}{2}$

[24]:
```
Rational("1/2")
```

[24]: $\frac{1}{2}$

[25]:
```
(S(1)/3)**100 * 3**100
```

[25]: 1

1.4 Gleitkommazahlen mit vielen Nachkommastellen

[26]:
```
drittel = Rational(1,3)
```

[27]:
```
drittel**100 * 3**100
```

[27]: 1

[28]:
```
(1/3)**1000
```

[28]: 0.0

[29]:
```
(1/3)**1000 * 3**1000
```

```
---------------------------------------------------------------------------
OverflowError                             Traceback (most recent call last)
Cell In[29], line 1
----> 1 (1/3)**1000 * 3**1000

OverflowError: int too large to convert to float
```

[30]:
```
(S(1)/3)**1000 * 3**1000
```

[30]: 1

[31]:
```
pi
```

[31]: π

[32]:
```
N(pi,100)
```

[32]: 3.1415926535897932384626433832795028841971693993751058209749445923078164062862089986280348253421170

[33]:
```
N(1/3,100)
```

[33]: 0.333333333333333314829616256247390992939472198486328125

[34]:
```
N(Rational(1,3),200)
```

[34]: 0.3333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333333

1.5 Sympyfizierung

[35]:
```
S(3)
```

[35]: 3

[36]:
```
3
```

[36]: 3

[37]:
```
type(3)
```

[37]: int

[38]:
```
type(S(3))
```

[38]: sympy.core.numbers.Integer

[39]:
```
type(S(1)/3)
```

[39]:
```
sympy.core.numbers.Rational
```

1.6 Symbole

[40]:
```
x = S('x')
```

[41]:
```
type(x)
```

[41]:
```
sympy.core.symbol.Symbol
```

[42]:
```
y = S('y')
```

[43]:
```
f = (x+y)**2
```

[44]:
```
f
```

[44]: $(x + y)^2$

[45]:
```
type(f)
```

[45]:
```
sympy.core.power.Pow
```

[46]:
```
x = 5
```

[47]:
```
f
```

[47]: $(x + y)^2$

[48]:
```
f.subs(x,5)
```

[48]: $(x + y)^2$

[49]:
```
x = S('x')
```

[50]:
```
f.subs(x,5)
```

[50]: $(y + 5)^2$

1.7 einfache Funktionen

[51]:
```
sqrt(S(81))
```

[51]: 9

[52]:
```
sqrt(S(2))
```

[52]:

$\sqrt{2}$

[53]:
```
sqrt(-81)
```

[53]: $9i$

[54]:
```
sqrt(2.)
```

[54]: 1.4142135623731

[55]:
```
sqrt(9*y**2)
```

[55]: $3\sqrt{y^2}$

[56]:
```
factorial(5)
```

[56]: 120

[57]:
```
factorial(170)
```

[57]: 7257415615307998967396728211129263114716991681296451376543577798900561843401706157852350749242617450

[58]:
```
cos(pi)
```

[58]: -1

[59]:
```
sin(pi)
```

[59]: 0

[60]:
```
tan(pi/2)
```

[60]: $\tilde{\infty}$

[61]:
```
print(tan(pi/2))
```

```
zoo
```

[62]:
```
?zoo
```

```
Type:        ComplexInfinity
String form: zoo
File:        /local/home/schaedle/miniconda3/envs/comp25/lib/python3.12/
 ↪site-packages/sympy/core/numbers.py
Docstring:
Complex infinity.

Explanation
===========

In complex analysis the symbol `\tilde\infty`, called "complex
infinity", represents a quantity with infinite magnitude, but
```

```
undetermined complex phase.

ComplexInfinity is a singleton, and can be accessed by
``S.ComplexInfinity``, or can be imported as ``zoo``.

Examples
========

>>> from sympy import zoo
>>> zoo + 42
zoo
>>> 42/zoo
0
>>> zoo + zoo
nan
>>> zoo*zoo
zoo

See Also
========

Infinity
```

[63]:
```
alpha = S('alpha')
```

[64]:
```
alpha
```

[64]: α

[65]:
```
exp(1)
```

[65]: e

[66]:
```
log(exp(1))
```

[66]: 1

[67]:
```
abs(-1)
```

[67]: 1