lektion3

October 24, 2019

Table of Contents

1 Lektion 3

1.1 Python Funktionen

[1]:
```
pf = lambda x : x**2
pf(2)
```

[1]: 4

das gleiche ausführlicher

[2]:
```
def mysqr2(x):
    """ Berechnet das Quadrat von x"""
    y = x**2
    return y
```

[3]:
```
mysqr2(2)
```

[3]: 4

[4]:
```
?mysqr2
```

[5]:
```python
def mypow(x,n=2):
    """ Berechnet x**n und falls n nicht gegeben ist das Quadrat von x"""
    y = x**n
    return y
```

[6]:
```python
mypow(2), mypow(2,4)
```

[6]: (4, 16)

1.1.1 Achtung

[7]:
```python
def f(a, L=[]):

    L.append(a)
    return L

#Der default Wert wird nur einmal ausgewertet und L ist hier ein veränderbares␣
 ↪Objekt das
# im Kontext der Funktion f erhalten bleibt
print(f(1))
print(f(2))
print(f(3))
print(f(4,[2,3,1]))
```

```
[1]
[1, 2]
[1, 2, 3]
[2, 3, 1, 4]
```

[8]:
```python
def f(a, L=None):

    if L is None:
        L = []
    L.append(a)
    return L


print(f(1))
print(f(2))
print(f(3))
print(f(4,[2,3,1]))
```

```
[1]
[2]
[3]
[2, 3, 1, 4]
```

1.2 Sympy Funktionen

[9]:
```python
from sympy import *
init_printing()
x,y,z = symbols('x y z')
f = Lambda(x,x**2) # vgl. lambda x : expr
f
```

[9]: $(x \mapsto x^2)$

[10]:
```python
f(2)
```

[10]: 4

[11]:
```python
f = Lambda((x,y,z),x*y+y-2*z**2)
f
```

[11]: $((x, \ y, \ z) \mapsto xy + y - 2z^2)$

[12]:
```python
f(1,2,3)
```

[12]: -14

[13]:
```python
param = x,y,z
f = Lambda(param,x+y-z)
type(param), f, f(*param) # * ist hier der "argument unpacking operator"
```

[13]:
```
(tuple, Lambda((x, y, z), x + y - z), x + y - z)
```

[17]:
```python
f = Function('f')
g = Function('g')
```

[18]:
```python
f(g(x))
```

[18]: $f(g(x))$

[19]:
```python
k = f(x)+g(1/x)
```

[20]:
```python
k.diff(x)
```

[20]:
$$\frac{d}{dx}f(x) - \frac{\left.\frac{d}{d\xi_1}g(\xi_1)\right|_{\xi_1=\frac{1}{x}}}{x^2}$$

1.3 Numpy Funktionen

[21]:
```python
import numpy as np
```

[22]:
```
xn = np.linspace(0,1,4)
xn
```

[22]:
```
array([0.        , 0.33333333, 0.66666667, 1.        ])
```

[23]:
```
np.sin(xn)
```

[23]:
```
array([0.        , 0.3271947 , 0.6183698 , 0.84147098])
```

[24]:
```
np.sin(np.pi*xn)
```

[24]:
```
array([0.00000000e+00, 8.66025404e-01, 8.66025404e-01, 1.22464680e-16])
```

1.4 Lamdifizierung (sympy -> numpy/scipy)

[25]:
```
f = x**2 * sin(x)
f
```

[25]: $x^2 \sin(x)$

[26]:
```
fn = lambdify(x,f)
```

[27]:
```
xn = np.linspace(0,11,5)
fn(xn)
```

[27]:
```
array([   0.        ,    2.88631125,  -21.34259485,   62.79474906,
       -120.99881499])
```

[28]:
```
f = Integral(exp(-x**2),(x,1,y))
F = f.doit()
F             # erf: gaußsche Fehlerfunktion (engl. error function)
```

[28]: $\frac{\sqrt{\pi}\operatorname{erf}(y)}{2} - \frac{\sqrt{\pi}\operatorname{erf}(1)}{2}$

[30]:
```
Fn = lambdify(y,F)
Fn(2)
```

[30]: 0.13525725794999466

[31]:
```
F.subs(y,2).n()
```

[31]: 0.135257257949995

[32]:
```
Fn(xn)
```

[32]:
```
array([-0.74682413,  0.13931362,  0.13940279,  0.13940279,  0.13940279])
```

[33]:
```
f = sin(pi/2*x**2)
F = Integral(f,(x,0,y))
F
```

[33]:

$$\int\limits_0^y \sin\left(\frac{\pi x^2}{2}\right)\,dx$$

[34]:
```
F.doit().simplify()
```

[34]:

$$S(y)$$

[36]:
```
Fn = lambdify(y,F.doit().simplify())
Fn(2)
```

```
 ␣
↪---------------------------------------------------------------------------

       NameError                                 Traceback (most recent call last)

       <ipython-input-36-78a2573d54ad> in <module>
         1 Fn = lambdify(y,F.doit().simplify())
   ----> 2 Fn(2)


       <lambdifygenerated-5> in _lambdifygenerated(y)
         1 def _lambdifygenerated(y):
   ----> 2     return (fresnels(y))


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

Mehr zu FresnelS hier: https://en.wikipedia.org/wiki/Fresnel_integral

[37]:
```
?fresnels
```

[38]:
```
def myfresnels(z):
    from scipy.special import fresnel
    sz,cz = fresnel(z)
    return sz
```

[39]:
```
from scipy import special
?special.fresnel
```

[40]:
```
Fn = lambdify(y,F.doit().simplify(),modules=("numpy","scipy", {"fresnels":
 ↪myfresnels}))
```

```
Fn(2)
```

[40]:

0.34341567836369824

[41]:
```
F.subs(y,2).evalf()
```

[41]:

0.343415678363698

1.5 Ableitungen

[42]:
```
x = symbols('x'); n = symbols('n')
f = x**n
f
```

[42]:

x^n

[43]:
```
f.diff(x)
```

[43]:

$$\frac{nx^n}{x}$$

[44]:
```
n.assumptions0
```

[44]: `{'commutative': True}`

[45]:
```
f.diff(x).powsimp()
```

[45]:

nx^{n-1}

[46]:
```
f.diff(x,3)
```

[46]:

$$\frac{nx^n\left(n^2-3n+2\right)}{x^3}$$

[47]:
```
f.diff(x,3).factor()
```

[47]:

$$\frac{nx^n\left(n-2\right)\left(n-1\right)}{x^3}$$

[48]:
```
f.diff(x,0) # was macht f.diff() ?
```

[48]:

x^n

[51]:
```
g = (1-cos(x))/x
g
```

[51]:

$$\frac{1-\cos\left(x\right)}{x}$$

[52]:
```
g.diff()
```

[52]:
$$\frac{\sin(x)}{x} - \frac{1-\cos(x)}{x^2}$$

[53]:
```
g.diff().simplify()
```

[53]:
$$\frac{x\sin(x)+\cos(x)-1}{x^2}$$

[54]:
```
f = Function('f')
g = Function('g')
h = Function('h')
```

[55]:
```
diff(f(x),x)
```

[55]:
$$\frac{d}{dx}f(x)$$

[56]:
```
f(x).diff(x,2)
```

[56]:
$$\frac{d^2}{dx^2}f(x)$$

[57]:
```
diff(f(x)/g(x),x).simplify()
```

[57]:
$$\frac{-f(x)\frac{d}{dx}g(x)+g(x)\frac{d}{dx}f(x)}{g^2(x)}$$

[58]:
```
diff(f(g(x)))
```

[58]:
$$\frac{d}{dg(x)}f(g(x))\frac{d}{dx}g(x)$$

[59]:
```
diff(f(x-g(x)),x)
```

[59]:
$$\left(1-\frac{d}{dx}g(x)\right)\left.\frac{d}{d\xi_1}f(\xi_1)\right|_{\xi_1=x-g(x)}$$

1.6 Integration - uneigentliche Integrale

[60]:
```
f = x**n
```

[61]:
```
f.integrate(x)
```

[61]:
$$\begin{cases}\frac{x^{n+1}}{n+1} & \text{for } n \neq -1\\ \log(x) & \text{otherwise}\end{cases}$$

[62]:
```
f.integrate(x,conds='none')
```

[62]:
$$\frac{x^{n+1}}{n+1}$$

[63]:
```
h = sin(x)*cos(x)
h
```

[63]:
$$\sin(x)\cos(x)$$

[64]:
```
Ih = h.integrate()
Ih
```

[64]:
$$\frac{\sin^2(x)}{2}$$

[65]:
```
Eq(Ih.diff(x),h) # equal
```

[65]:
```
True
```

[66]:
```
Ih.diff(x) == h # aber Achtung
```

[66]:
```
True
```

[67]:
```
x**2-y**2 == (x-y)*(x+y)
```

[67]:
```
False
```

[68]:
```
Eq(x**2-y**2, (x-y)*(x+y))
```

[68]:
$$x^2 - y^2 = (x-y)(x+y)$$

[70]:
```
Eq(x**2-y**2, (x-y)*(x+y)).simplify()
```

[70]:
```
True
```

[71]:
```
simplify(x**2-y**2 == (x-y)*(x+y)) # das klappt so nicht
```

[71]:
```
False
```

[72]:
```
f = 1/(1+x**4)
f
```

[72]:
$$\frac{1}{x^4+1}$$

[73]:
```
If = f.integrate()
If
```

[73]:
$$-\frac{\sqrt{2}\log\left(x^2-\sqrt{2}x+1\right)}{8}+\frac{\sqrt{2}\log\left(x^2+\sqrt{2}x+1\right)}{8}+\frac{\sqrt{2}\operatorname{atan}\left(\sqrt{2}x-1\right)}{4}+\frac{\sqrt{2}\operatorname{atan}\left(\sqrt{2}x+1\right)}{4}$$

[74]:
```
If.diff(x)
```

[74]:
$$-\frac{\sqrt{2}\left(2x-\sqrt{2}\right)}{8\left(x^2-\sqrt{2}x+1\right)}+\frac{\sqrt{2}\left(2x+\sqrt{2}\right)}{8\left(x^2+\sqrt{2}x+1\right)}+\frac{1}{2\left(\left(\sqrt{2}x+1\right)^2+1\right)}+\frac{1}{2\left(\left(\sqrt{2}x-1\right)^2+1\right)}$$

[77]:
```
Eq(If.diff(x), f)
```

[77]:

$$-\frac{\sqrt{2}\left(2x-\sqrt{2}\right)}{8\left(x^{2}-\sqrt{2}x+1\right)}+\frac{\sqrt{2}\left(2x+\sqrt{2}\right)}{8\left(x^{2}+\sqrt{2}x+1\right)}+\frac{1}{2\left(\left(\sqrt{2}x+1\right)^{2}+1\right)}+\frac{1}{2\left(\left(\sqrt{2}x-1\right)^{2}+1\right)}=\frac{1}{x^{4}+1}$$

[78]:
```
simplify(Eq(If.diff(x), f))
```

[78]:

True

[82]:
```
a = Symbol('a')
Ig = Integral(x**2*exp(-a*x), (x,0,oo))
Ig, Ig.doit()
```

[82]:

$$\left(\int\limits_{0}^{\infty}x^{2}e^{-ax}\,dx,\ \begin{cases}\frac{2}{a^{3}} & \text{for } \left|\arg\left(a\right)\right|<\frac{\pi}{2}\\ \int\limits_{0}^{\infty}x^{2}e^{-ax}\,dx & \text{otherwise}\end{cases}\right)$$

[84]:
```
a = Symbol('a',positive=True)
```

[85]:
```
Ig2 = Integral(x**2*exp(-a*x), (x,0,oo))
Ig2, Ig2.doit()
```

[85]:

$$\left(\int\limits_{0}^{\infty}x^{2}e^{-ax}\,dx,\ \frac{2}{a^{3}}\right)$$

1.7 Bestimmte Integrale

[86]:
```
g = (1-cos(x))/x
g.simplify()
```

[86]:

$$\frac{1-\cos\left(x\right)}{x}$$

[87]:
```
g.integrate((x,1,2))
```

[87]:

$$-\operatorname{Ci}\left(2\right)+\operatorname{Ci}\left(1\right)+\log\left(2\right)$$

Ci: Cosinus Integral https://en.wikipedia.org/wiki/Trigonometric_integral

[88]:
```
h = sqrt(exp(-x**2)+2)
h
```

[88]:

$$\sqrt{2+e^{-x^{2}}}$$

[89]:
```
Ih = Integral(h,(x,0,1))
Ih
```

[89]:

$$\int\limits_0^1 \sqrt{2 + e^{-x^2}}\, dx$$

[90]:
```
Ih.doit()
```

[90]:

$$\int\limits_0^1 \sqrt{2 + e^{-x^2}}\, dx$$

[91]:
```
Ih.n() # numerische Integration
```

[91]:

1.6562284230502

[92]:
```
# Comp LA / Numerik I Stoff (nicht relevant für Comp Ana)
import numpy as np
from scipy import integrate
```

[93]:
```
res, err = integrate.quad(lambdify(x,h),0,1)
res
```

[93]:

1.6562284230502002