lektion3

January 29, 2018

Table of Contents

1 Lektion 3

1.1 Python Funktionen

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

Out[1]: 4

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

In [3]: mysqr2(2)

Out[3]: 4

In [4]: def mypow(x,n=2):
            """ Berechnet x**n und falls n nicht gegeben ist das Quadrat von x"""
            y = x**n
            return y

In [5]: mypow(2), mypow(2,4)

Out[5]: (4, 16)

In [6]: def f(a, L=[]):

            L.append(a)
```

```
            return L

        #Der default Wert wird nur einmal ausgewertet und L ist hier ein veränderbares Objekt
        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]
```

```
In [7]: 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

```
In [8]: from sympy import *
        init_printing()
        x,y,z = symbols('x y z')
        f = Lambda(x,x**2) # vgl. lambda x : expr
```

```
In [9]: f(2)
```

Out[9]:

$$4$$

```
In [10]: f = Lambda((x,y,z),x*y+y-2*z**2)
         f
```

Out[10]:

$$\left((x, \quad y, \quad z) \mapsto xy + y - 2z^2\right)$$

```
In [11]: f(1,2,3)
```

Out[11]:

$$-14$$

```
In [12]: param = x,y,z
         f = Lambda(param,x+y-z)
         f(*param) # * ist hier der "argument unpacking operator"
```

Out[12]:

$$x + y - z$$

```
In [13]: f = Function('f')
         g = Function('g')
         h = Function('h')

In [14]: f(g(x))
```

Out[14]:

$$f(g(x))$$

```
In [15]: f(x)+g(1/x)
```

Out[15]:

$$f(x) + g\left(\frac{1}{x}\right)$$

1.3 Numpy Funktionen

```
In [16]: import numpy as np

In [17]: xn = np.linspace(0,1,4)
         xn

Out[17]: array([0.        , 0.33333333, 0.66666667, 1.        ])

In [18]: np.sin(xn)

Out[18]: array([0.        , 0.3271947 , 0.6183698 , 0.84147098])

In [19]: np.sin(np.pi*xn)

Out[19]: array([0.00000000e+00, 8.66025404e-01, 8.66025404e-01, 1.22464680e-16])
```

1.4 Lamdifizierung (sympy -> numpy)

In [20]:
```
f = x**2 * sin(x)
f
```

Out[20]:

$$x^2 \sin(x)$$

In [21]:
```
fn = lambdify(x,f)
```

In [22]:
```
xn = np.linspace(0,11,5)
fn(xn)
```

Out[22]:
```
array([   0.        ,    2.88631125,  -21.34259485,   62.79474906,
       -120.99881499])
```

In [23]:
```
f = Integral(exp(-x**2),(x,1,y))
ff = f.doit()
ff
```

Out[23]:

$$\frac{\sqrt{\pi}}{2} \operatorname{erf}(y) - \frac{\sqrt{\pi}}{2} \operatorname{erf}(1)$$

In [24]:
```
from scipy import special
special.erf(1)
```

Out[24]:

$$0.8427007929497148$$

In [25]:
```
fn = lambdify(y,f.doit(),modules=("numpy",{"erf":special.erf}))
```

In [26]:
```
f.subs(y,2).evalf()
```

Out[26]:

$$0.135257257949995$$

In [27]:
```
fn(xn)
```

Out[27]:
```
array([-0.74682413,  0.13931362,  0.13940279,  0.13940279,  0.13940279])
```

In [28]:
```
fn(2)
```

Out[28]:

$$0.13525725794999466$$

1.5 Ableitungen

```
In [29]: x = symbols('x'); n = symbols('n')
         f = x**n
         f
```

Out[29]:

$$x^n$$

```
In [30]: f.diff(x)
```

Out[30]:

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

```
In [31]: n.assumptions0
```

```
Out[31]: {'commutative': True}
```

```
In [32]: f.diff(x).powsimp()
```

Out[32]:

$$nx^{n-1}$$

```
In [33]: f.diff(x,3)
```

Out[33]:

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

```
In [34]: f.diff(x,3).factor()
```

Out[34]:

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

```
In [35]: f.diff(x,0) # f.diff()
```

Out[35]:

$$x^n$$

```
In [36]: g = (1-cos(x))/x
```

```
In [37]: g.diff().simplify()
```

Out[37]:

$$\frac{1}{x^{2}} \left(x \sin{\left (x \right)} + \cos{\left (x \right)} - 1\right)$$

```
In [38]: f = Function('f')
         g = Function('g')
         h = Function('h')
```

```
In [39]: diff(f(x),x)
```

Out[39]:

$$\frac{d}{d x} f{\left (x \right)}$$

```
In [40]: f(x).diff(x,2)
```

Out[40]:

$$\frac{d^{2}}{d x^{2}} f{\left (x \right)}$$

```
In [41]: diff(f(x)*g(x),x)
```

Out[41]:

$$f{\left (x \right)} \frac{d}{d x} g{\left (x \right)} + g{\left (x \right)} \frac{d}{d x} f{\left (x \right)}$$

```
In [42]: diff(f(x)*g(x)*h(x),x)
```

Out[42]:

$$f{\left (x \right)} g{\left (x \right)} \frac{d}{d x} h{\left (x \right)} + f{\left (x \right)} h{\left (x \right)} \frac{d}{d x} g{\left (x \right)} + g{\left (x \right)} h{\left (x \right)} \frac{d}{d x} f{\left (x \right)}$$

```
In [43]: diff(f(g(x)))
```

Out[43]:

$$\frac{d}{d x} g{\left (x \right)} \left.\frac{d}{d \xi_{1}} f{\left (\xi_{1} \right)} \right|_{\substack{ \xi_{1}=g{\left (x \right)} }}$$

```
In [44]: diff(f(x,g(x)),x)
```

Out[44]:

$$\frac{d}{d x} g{\left (x \right)} \left.\frac{\partial}{\partial \xi_{2}} f{\left (x,\xi_{2} \right)} \right|_{\substack{ \xi_{2}=g{\left (x \right)} }} + \left.\frac{\partial}{\partial \xi_{1}} f{\left (\xi_{1},g{\left (x \right)} \right)} \right|_{\substack{ \xi_{1}=x }}$$

1.6 Integration - uneigentliche Integrale

```
In [45]: f = x**n
```

```
In [46]: f.integrate(x)
```

Out[46]:

$$\begin{cases} \log(x) & \text{for } n = -1 \\ \frac{x^{n+1}}{n+1} & \text{otherwise} \end{cases}$$

```
In [47]: f.integrate(x,conds='none')
```

Out[47]:

$$\frac{x^{n+1}}{n+1}$$

```
In [48]: h = sin(x)*cos(x)
         h
```

Out[48]:

$$\sin(x)\cos(x)$$

```
In [49]: Ih = h.integrate()
         Ih
```

Out[49]:

$$\frac{1}{2}\sin^2(x)$$

```
In [50]: Eq(Ih.diff(x),h) # equal
```

Out[50]:

True

```
In [51]: f = 1/(1+x**4)
         f
```

Out[51]:

$$\frac{1}{x^4+1}$$

```
In [52]: If = f.integrate()
         If
```

Out[52]:

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

```
In [53]: If.diff(x) == f # equal
```

```
Out[53]: False
```

```
In [54]: If.diff(x).simplify() == f # geht doch
```

```
Out[54]: True
```

```
In [55]: a = Symbol('a')
         Ig = Integral(x**2*exp(-a*x), (x,0,oo))
         Ig.doit()
```

Out[55]:

$$\begin{cases}\frac{2}{a^3} & \text{for } \left|\text{periodic}_{\text{argument}}\left(a,\infty\right)\right| < \frac{\pi}{2}\\ \int_0^\infty x^2e^{-ax}\,dx & \text{otherwise}\end{cases}$$

```
In [56]: a = Symbol('a',positive=True)
```

```
In [57]: Ig2 = Integral(x**2*exp(-a*x), (x,0,oo))
         Ig2.doit()
```

Out[57]:

$$\frac{2}{a^3}$$

1.7 Bestimmte Integrale

```
In [70]: g = (1-cos(x))/x
         g.simplify()
```

Out[70]:

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

```
In [71]: g.integrate((x,1,2))
```

Out[71]:

$$-\frac{1}{2}\log\left(4\right)-\operatorname{Ci}\left(2\right)+\operatorname{Ci}\left(1\right)+2\log\left(2\right)$$

```
In [60]: from sympy import *
         x = symbols('x')
         init_printing()
         h = sqrt(exp(-x**2)+2)
         h
```

Out[60]:

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

In [61]:
```
Ih = Integral(h,(x,0,1))
Ih
```

Out[61]:

$$\int_0^1 \sqrt{2 + e^{-x^2}}\, dx$$

In [62]:
```
Ih.doit()
```

Out[62]:

$$\int_0^1 \sqrt{2 + e^{-x^2}}\, dx$$

In [63]:
```
Ih.n() # numerische Integration
```

Out[63]:

1.6562284230502

In [64]:
```
# Comp LA / Numerik I Stoff (nicht relevant für Comp Ana)
import numpy as np
from scipy import integrate
```

In [65]:
```
res, err = integrate.quad(lambdify(x,h),0,1)
res
```

Out[65]:

1.6562284230502002

In [66]:
```
f = Function('f')
y = symbols('y')
```

In [67]:
```
Integral(f(x),(x,0,y))
```

Out[67]:

$$\int_0^y f(x)\, dx$$