

# Symbolix25

In [1]: `using SymPy, Plots;`

In [2]: `x,y=symbols("x,y")`

Out[2]: (x, y)

## Differenzieren und Integrieren

In [3]: `diff(x^2,x)`

Out[3]:  $2x$

In [4]: `y=integrate(x*sin(2*x),x)`

Out[4]:  $-\frac{x \cos(2x)}{2} + \frac{\sin(2x)}{4}$

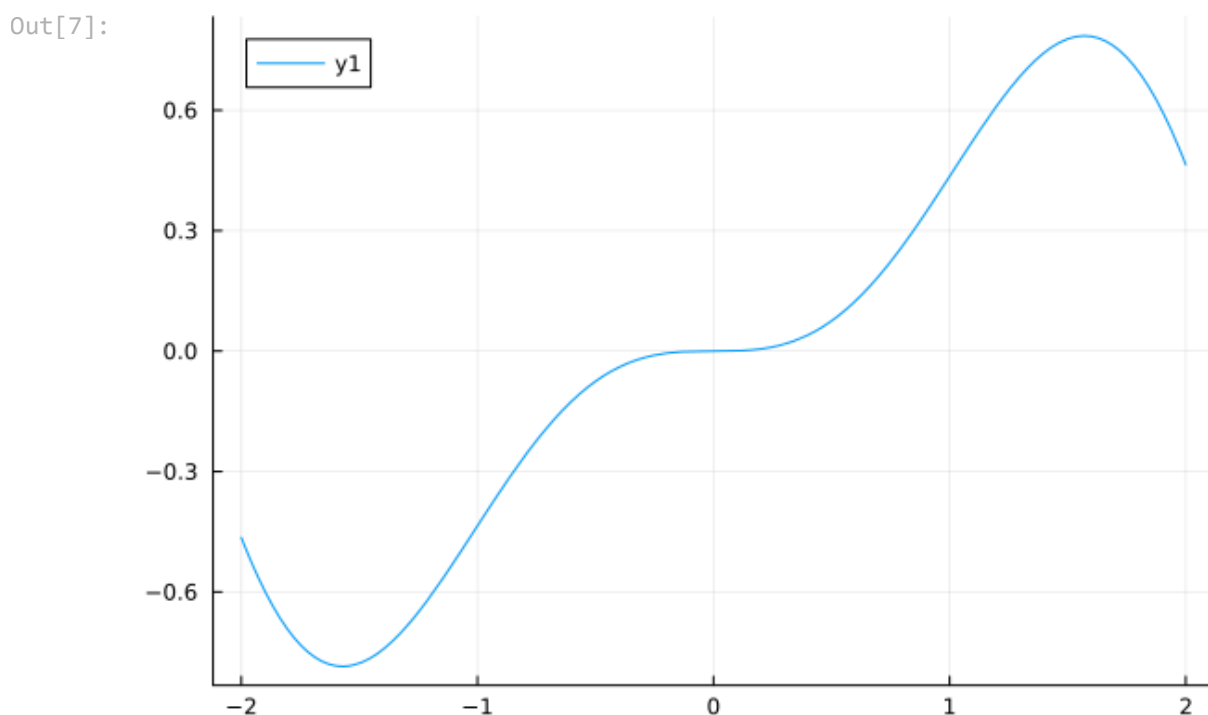
In [5]: `diff(y,x)`

Out[5]:  $x \sin(2x)$

In [6]: `y(2)`

Out[6]:  $\frac{\sin(4)}{4} - \cos(4)$

In [7]: `plot(y,-2,2)`



## Grenzwerte

In [8]: `limit((cos(x)-1)/x^2,x=>0)`

Out[8]:  $-\frac{1}{2}$

In [9]: `limit((1-2/x)^x,x=>oo)`

Out[9]:  $e^{-2}$

$\infty$  wird durch zwei oo eingegeben (Kleinbuchstaben)

die Syntax hat sich hier (wie an diversen anderen Stellen) geändert ...

## Evaluieren

In [10]: `I=integrate(sin(x)/x,(x,0,pi))`

Out[10]:  $\text{Si}(\pi)$

In [11]: `I.evalf()`

Out[11]: 1.85193705198247

In [12]: `I.evalf(72)`

Out[12]: 1.85193705198246617036105337015799136334580972898115490980478378187698189

wenn man einen Ausdruck (Expression) zahlenmäßig auswerten muss ...

## Gleichung

algebraisch lösen

In [13]: `g(x)=sin(x)-cos(x)+0.1  
solve(g(x),x)`

Out[13]:  $\begin{bmatrix} -2.28542475353013 \\ 0.714628426735235 \end{bmatrix}$

In [14]: `solve(g(x),x)[2]`

Out[14]: 0.714628426735235

In [15]: `solve(cos(x)-sin(x),x)`

Out[15]:  $\left[ \frac{\pi}{4} \right]$

an Periodizität muss man selber denken ...

# Differentialgleichung

symbolische Lösung

```
In [17]: @syms t  
u=sympy.Function("u")  
general=dsolve(sympy.Derivative(u(t),t)-2*u(t)/t)
```

Out[17]:  $u(t) = C_1 t^2$

```
In [18]: special=dsolve(sympy.Derivative(u(t),t)-2*u(t)/t,ics=Dict(u(1.0)=>2.0))
```

Out[18]:  $u(t) = 2.0 t^2$

die Syntax von SymPy hat sich an vielen Stellen verändert, siehe ??dsolve für mehr hierzu

```
In [ ]:
```