

# Einführung in Python/Matplotlib

## AP/FP-Softwarekurse

Markus Ühlein

Fachschaft Physik  
TU Kaiserslautern

19. Februar 2019

# Übersicht

Grundlagen

Plotten

Fitten

Weiteres

# Python starten

## Konsole

- | ● > Konsole öffnen
- | ● > python3 eingeben
- | ● > **help**(<Befehl>)

```
Python 3.6.4 (default, Jan 5 2018, 02:35:40)
[GCC 7.2.1 20171224] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("AP ist toll")
AP ist toll
>>> █
```

## Website und Download

- | ● > Vollständig ausgewachsene Programmiersprache
  - | ● > <https://docs.python.org/>
  - | ● > <https://www.python.org/>
- | ● > Plots
  - | ● > <https://matplotlib.org/>
  - | ● > <https://scipy.org/> (fitten)(enthält matplotlib)

# Skripte

## Skripte und Hilfe

- | • > Textdatei mit Lieblingseditor bearbeiten
- | • > Ausführen mit `python3 <script.py>` aus der Konsole
- | • > Ausführen mit `import <script.py>` aus python

## Hilfe

- | • > `help(<Kommando>)`
- | • > Google ist dein Freund!

# Entwicklungsumgebungen

- Spyder: <https://www.spyder-ide.org/>
- Pycharm: <https://www.jetbrains.com/pycharm/>

# Grundlegendes

- > Rechenoperationen +,-,\*,/,\*\*
- > Einrückung unterscheidet zwischen Leerzeichen und Tabs  
→ Einrückungen sind wichtig!  
Python verzichtet weitesgehend auf Klammern
- > Numpy, Scipy und Matplotlib importieren

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3 import scipy.optimize as fit
```

# Plotten

Befehle sind zu Matlab ähnlich

```
1 import matplotlib.pyplot as plt
2 import numpy as np

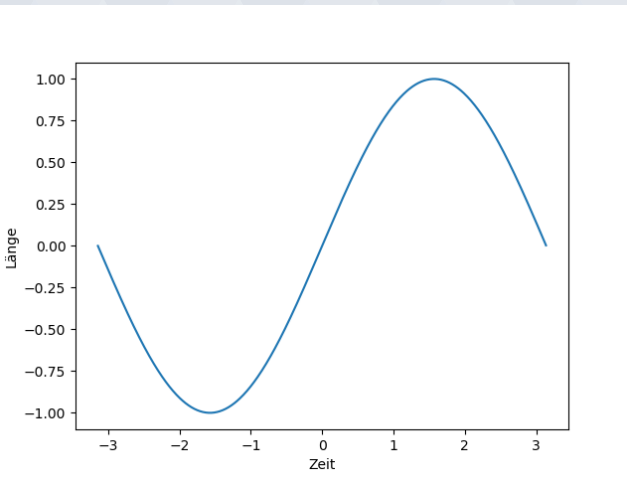
4 #x- und y- Vektor erzeugen
5 x=np.arange(-np.pi,np.pi,0.01)
6 y=np.sin(x)

8 #Plot erstellen
9 plt.plot(x,y)

11 #Beschriftung
12 plt.xlabel='Zeit'
13 plt.ylabel='Länge'

15 #Ausgabedatei
16 plt.savefig("plot.png")
17 plt.show()
```

# Plotten





# Fehlerbalken

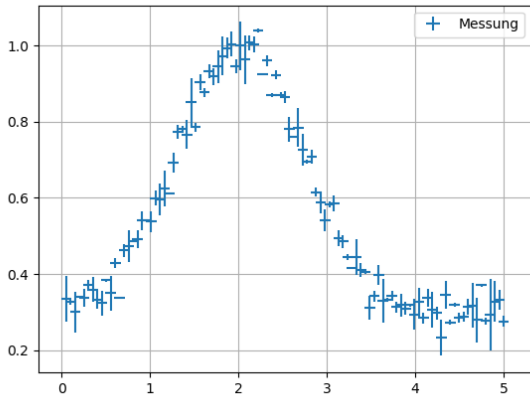
```
1 import matplotlib.pyplot as plt
2 import numpy as np

4 #Daten einlesen
5 x=np.loadtxt("messung.dat", usecols=0, skiprows=1)
6 #... analog: y, xfehler und yfehler

8 #Plot erstellen
9 plt.errorbar(x,y,xerr=xfehler,yerr=yfehler,
10             linestyle='None', label='Messung')
11 plt.legend()
12 plt.grid()

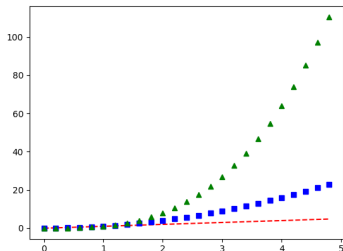
13 #Ausgabedatei
14 plt.savefig("fehler.png")
15 plt.show()
```

# Plotten



# Styling

```
1 plt.plot(t, t, 'r--',  
2 t, t**2, 'bs',  
3 t, t**3, 'g^')
```



## Farben

blue, green, red, cyan, magenta, yellow, black, white, usw.

# Styling

## Linien

- |•〉 : gepunktet
- |•〉 -. Strichpunkte
- |•〉 -- gestrichelt
- |•〉 - durchgezogen

## Punkte

- |•〉 . Punkte
- |•〉 ^ oder v oder < oder > Dreiecke
- |•〉 x oder + Kreuze
- |•〉 square, pentagon, hexagon

# Fitten

- Fitfunktion definieren:  $f = \text{lambda } x, a, b, c : a \cdot \text{np.sin}(b \cdot x) + c$
- $perr, pconv = \text{fit.curve\_fit}(f, x, y)$
- Fitparameter in Array perr gespeichert
- Fehler:  $\text{np.sqrt}(\text{np.diag}(pconv))$
- viele weitere optionale Parameter
- [https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve\\_fit.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html)

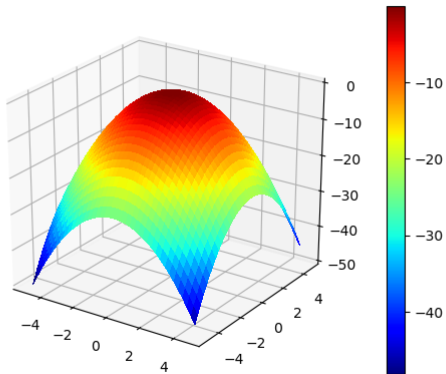
# 3D-Plots

```
1 from mpl_toolkits.mplot3d import Axes3D
2 import matplotlib.pyplot as plt
3 from matplotlib import cm
4 import numpy as np
5 fig = plt.figure()
6 achse = fig.gca(projection='3d')

8 #Daten erzeugen
9 X = np.arange(-5, 5, 0.25)
10 Y = np.arange(-5, 5, 0.25)
11 X, Y = np.meshgrid(X, Y)
12 Z = -(X**2 + Y**2)

14 # Plot definieren
15 surf = achse.plot_surface(X, Y, Z, cmap=cm.jet,
    linewidth=0, antialiased=False)
16 fig.colorbar(surf)
17 plt.show()
```

# 3D-Plots



# Weiteres zur Sprache

```
1  def testfunktion(matrix):
2      output=np.zeros([len(matrix),4])
3      for i in range(len(matrix)):
4          if i%2==0:
5              print("x"+str(i)+"=",matrix[i,0]+
6                  matrix[i,1])
7              output[i,0]=matrix[i,0]
8              output[i,1]=2*matrix[i,0]
9              output[i,2]=matrix[i,1]
10             output[i,3]=2*matrix[i,1]
11         return output

12  beispiel=np.loadtxt("test.txt")
13  testoutput=testfunktion(beispiel)
```



# Lesen und Schreiben von Dateien

```

1  #Messwerte als Matrix von Zahlen lesen
2  #Dezimalzahlen mit Punkten, Zahlen durch
   #Whitespace getrennt
3  data=np.loadtxt("meineMesswerte.txt")

5  #neue Datei zum schreiben öffnen
6  with open("neueDatei.txt", "w") as fobj:
7      for i in range(data.shape[0]):
8          #Die Zahlen werden beim schreiben so
           #modifiziert, dass die 4 Nachkommastellen
           #haben
9          fobj.write("%.4f"%(data[i,0]))
10         for j in range(data.shape[1]-1):
11             fobj.write("\t"+"%.4f"%(data[i,j+1]))
12         fobj.write("\n")

```

# Lesen und Schreiben von Dateien

```
1 #Dateien als string lesen
2 with open("neueDatei.txt", "r") as fobj:
3     zeile1=fobj.readline() #erste Zeile lesen
4     zeile2=fobj.readline() #zweite Zeile lesen
5     alles=fobj.read() #ganzes Dokumentlesen
```

# Bildverarbeitung

```
1  #Bild einlesen
2  bild=plt.imread("meinBild.jpg")

4  #Das Bild wird als Matrix der Pixel gespeichert.
   Jeder Pixel wird durch ein Array mit 3 Einträgen
   in rgb-Darstellung gespeichert und kann
   dementsprechend manipuliert werden

6  #Bild plotten
7  plt.figure()
8  plt.imshow(bild)
9  plt.show()
```

# Übung

## Übung

- | • > Tue das, was du gerne tun möchtest
- | • > Inspiration unter <https://fachschaft.physik.uni-kl.de/softwarekurse.html>  
(inclusive eines von mir erstellten Skripts, dass das plotten erleichtern soll (plotten.py))

## Installparty

- | • > Wenn du Probleme bei der Installation hast, frage!
- | • > Stellt mir gerne auch in Zukunft Fragen an [pythonfrage@muehle.de](mailto:pythonfrage@muehle.de)