

Einführung in Python/Matplotlib

AP/FP-Seminarreihe

Markus Ühlein

Fachschaft Physik
TU Kaiserslautern

15. Februar 2018

Übersicht

Grundlagen

Plotten

Fitten

Ausprobieren

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/>

Installation

```
python3 -m pip install --user numpy matplotlib
```

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>)`

Grundlegendes

- > Rechenoperationen +,-,*,/,**
- > Einrückung unterscheidet zwischen Leerzeichen und Tabs
- > Numpy und Matplotlib importieren

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

Plotten

Befehle sind zu Matlab ähnlich

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

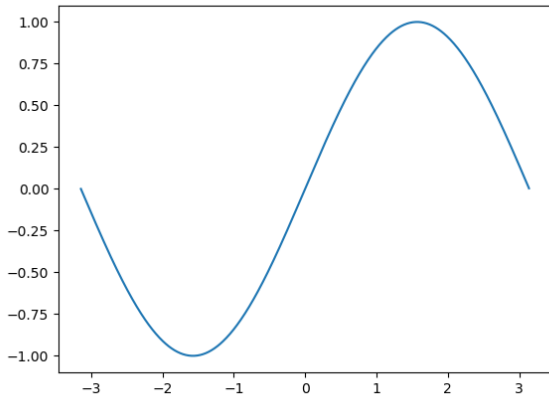
4 #x- und y- Vektor erzeugen
5 x=np.arange(-np.pi,np.pi,0.01)
6 y=np.sin(x)

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

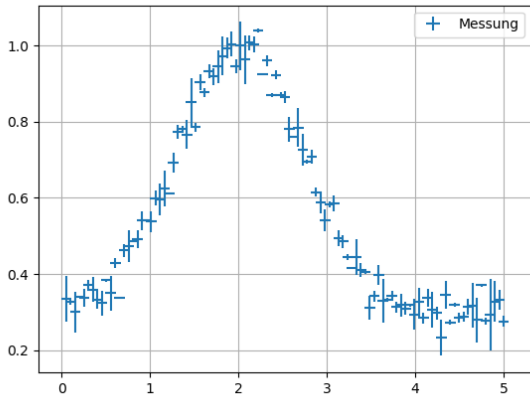
```
1 import matplotlib.pyplot as plt
2 import numpy as np

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()

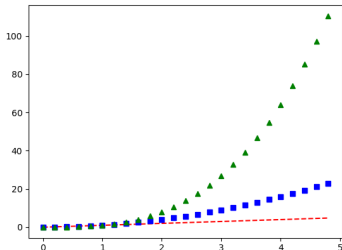
13 #Ausgabedatei
14 plt.savefig("fehler.png")
15 plt.show()
```


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

Polynome

- Polynom von Grad n anfiten: `fit=np.polyfit(x,y,n)`
- Liefert Koeffizienten
- In Werte umwandeln `wert=np.poly1d(fit)`

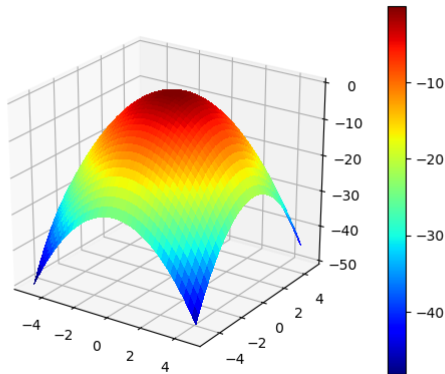
3D-Plots

```
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



Übung

Übung

- |•> Tue das, was du gerne tun möchtest
- |•> Inspiration unter
<http://www.fs.physik.uni-kl.de/vortrag.html>

Installparty

- |•> Wenn du Probleme bei der Installation hast, frage!