

04c_matplotlib_Miniuebungen_Loesungen

0.1 Mini-Aufgaben zur Überprüfung des Verständnis: Grundlagen von matplotlib

Plotten Sie die Funktionen $\sin(x)$, $\cos(x)$ und $\tan(x)$ auf dem Intervall $[-2\pi, 2\pi]$ in einer Darstellung. Experimentieren Sie mit Beschriftungen, Labels und dem dargestellten Bereich, um ein möglichst aussagekräftiges Ergebnis zu erzielen.

Speichern Sie den Plot anschließend in eine PDF-Datei, öffnen Sie diese in einem PDF-Betrachter, und verifizieren Sie, dass die Grafikdatei bei Größenänderungen in Word etc. tatsächlich skaliert, und nicht "pixelig" wird.

```
[29]: import numpy as np
import matplotlib.pyplot as plt

a = -2*np.pi
b = 2*np.pi

x = np.linspace(a,b)

plt.plot(x, np.sin(x), label="sin(x)")
plt.plot(x, np.cos(x), label="cos(x)")
#plt.plot(x, np.tan(x), label="tan(x)")

# Ergebnis sieht blöd aus wegen Polstellen des Tangens
# Wir probieren es damit, die Wertebereiche zu beschränken
plt.xlim(a-0.3,b+0.3)
plt.ylim(-1.5,1.5)

# Ergebnis sieht immer noch blöd aus, deshalb tan oben auskommentiert
# und stattdessen einfach bei den Polstellen nichts plotten :)
# Aus der Dokumentation:
# * nan erscheint nie im Plot
# * np.diff() testet so, ob bei zwei aufeinanderfolgenden Auswertungspunkten
#   ein Vorzeichenwechsel vorliegt

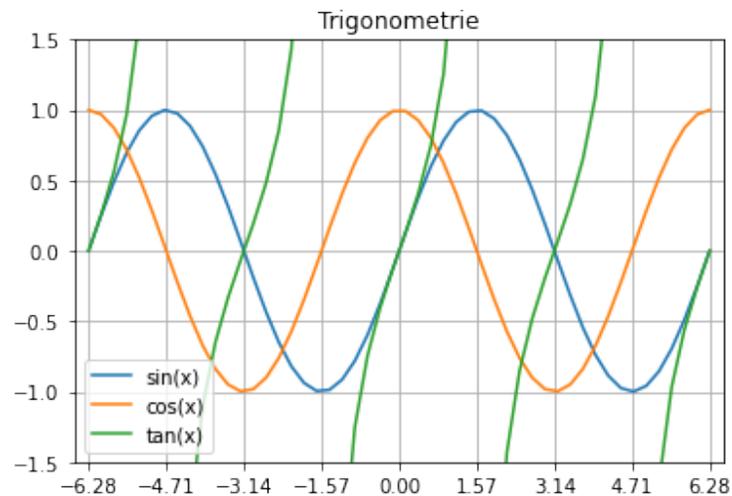
z = np.cos(x)
y[:-1][np.diff(y) < 0] = np.nan
plt.plot(x, y, label="tan(x)")
```

```

# Jetzt noch aufhübschen
plt.legend()
plt.xticks(np.arange(a,b+0.1,np.pi/2))
plt.grid()
plt.title("Trigonometrie")

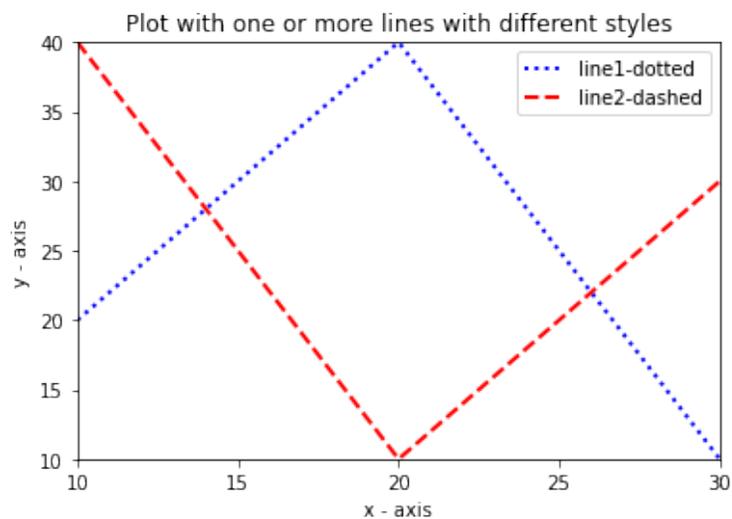
plt.show()

```



Der folgende Plot ist mit Matplotlib erstellt worden. Leider ist der zugehörige Programmcode verschwunden. Rekonstruieren Sie den Plot durch Ablesen der Funktionswerte und Angabe der korrekten Linienstile.

Hinweis: `np.interp()` kann hilfreich sein, wenn Sie keine Geradengleichungen ausrechnen wollen.



```
[1]: import numpy as np
import matplotlib.pyplot as plt

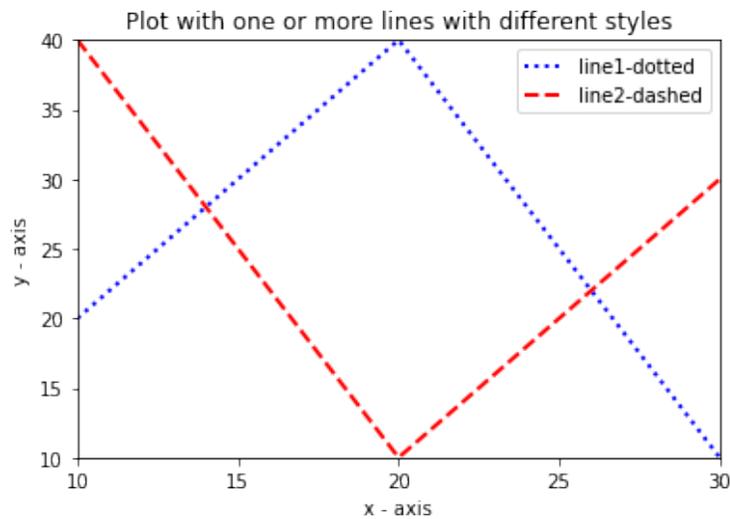
x = np.linspace(10,30,1000)

x1 = [10,20,30]
f1 = [20,40,10]
f2 = [40,10,30]

data1 = np.interp(x, x1, f1)
data2 = np.interp(x, x1, f2)

plt.plot(x, data1, ':b', lw=2, label="line1-dotted")
plt.plot(x, data2, '--r', lw=2, label="line2-dashed")
plt.xlabel("x - axis")
plt.ylabel("y - axis")
plt.title("Plot with one or more lines with different styles")
plt.xlim(10,30)
plt.ylim(10,40)
plt.xticks([10,15,20,25,30])
plt.legend()

plt.show()
```



0.2 Impressum

0.2.1 Programmierkurs Python, Dominik Göddeke <https://www.ians.uni-stuttgart.de>,
Universität Stuttgart

Version vom April 2023

Lizenziert unter der Creative Commons Namensnennung 4.0 International Lizenz



Veröffentlicht auf <https://zoerr.de>, (alle Rechte am Logo vorbehalten)



Gefördert durch die Stiftung Innovation in der Hochschullehre. (alle Rechte am Logo vorbehalten)



Gefördert mit Mitteln der Deutschen Forschungsgemeinschaft (EXC 2075 - 390740016) im Rahmen der Exzellenzstrategie.

[]: