lektion7

November 12, 2024

1 Einführung in Pyplot

1.1 Pyplot

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
#%matplotlib notebook
%matplotlib inline
#%matplotlib qt5agg
```

Für pyplot gibt es einen *expliziten* Modus, bei dem in ein vorgegebenes Koordinatensystem gezeichnet wird

[2]:
```
x = np.linspace(0, 5*np.pi)
y = np.cos(x)
fig = plt.figure()
fig.show()
```

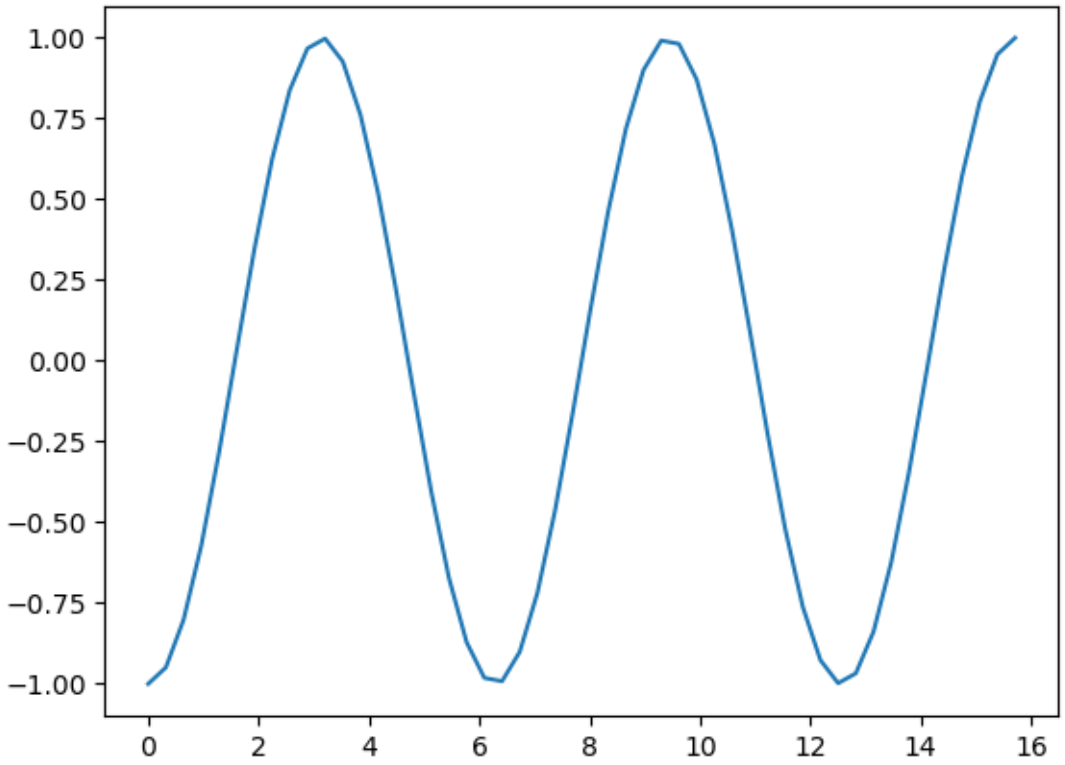
```
/tmp/ipykernel_5569/1506517316.py:4: UserWarning: FigureCanvasAgg is non-
interactive, and thus cannot be shown
  fig.show()
```

```
<Figure size 640x480 with 0 Axes>
```

[3]:
```
ax = fig.add_axes((.1,.1,.8,.8))
ax.plot(x, y); # Polygozug zu den Punkten (x[i], y[i]) i = 0 ...
```

[4]:
```
fig, ax = plt.subplots() # kompakter
ax.plot(x,-y)
```

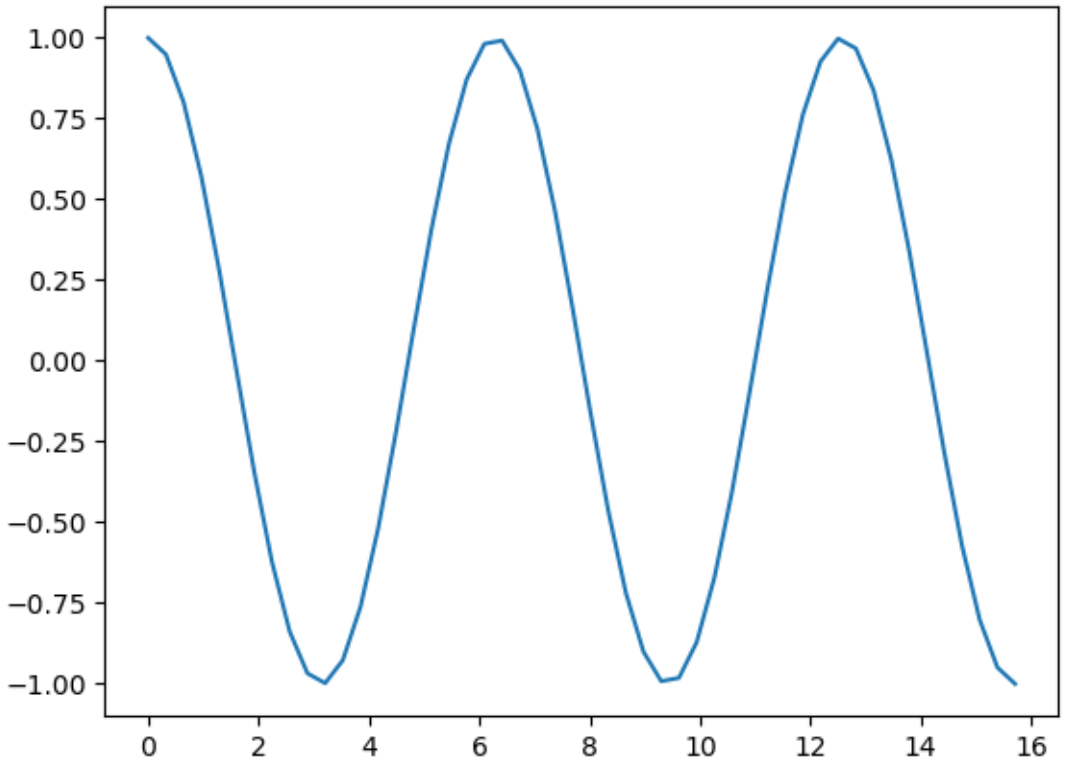
[4]:
```
[<matplotlib.lines.Line2D at 0x7f96e05a9040>]
```



Im *impliziten* Modus sucht sich pyplot das Koordinatensystem in das gezeichnet wird.

[5]:
```
plt.figure() # erzeugt ein neues Bild
plt.plot(x, y) # zeichne in das Koordinatensystem des letzten Bildes
```

[5]: `[<matplotlib.lines.Line2D at 0x7f96d81ca300>]`

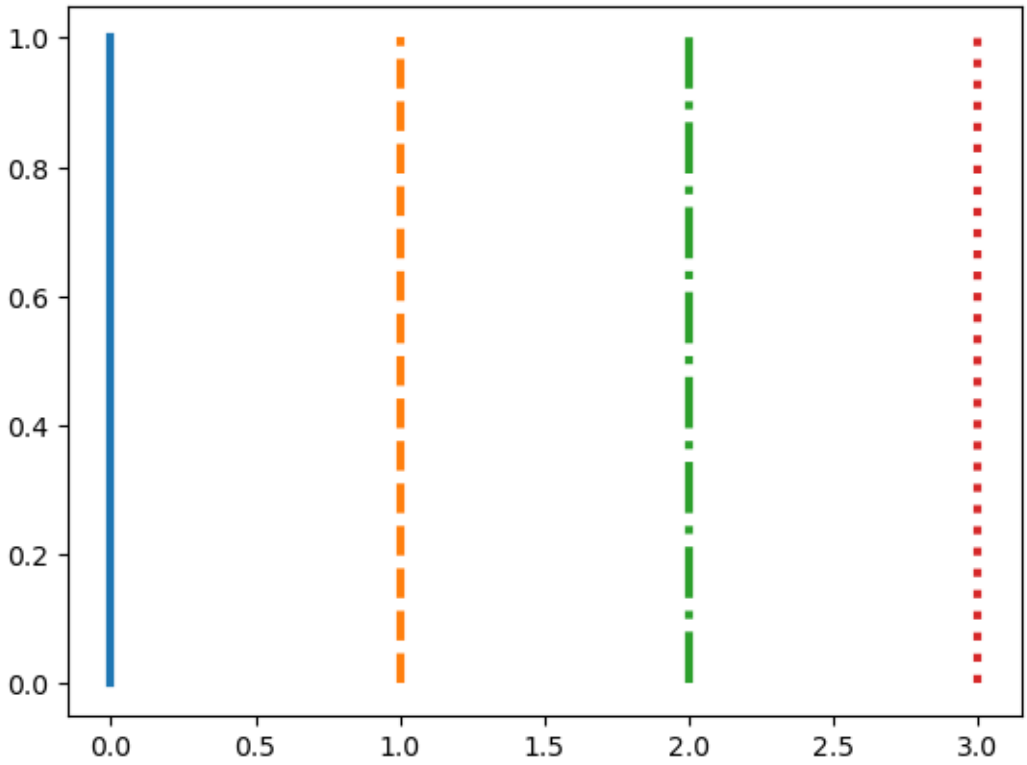


Farben als Kürzel

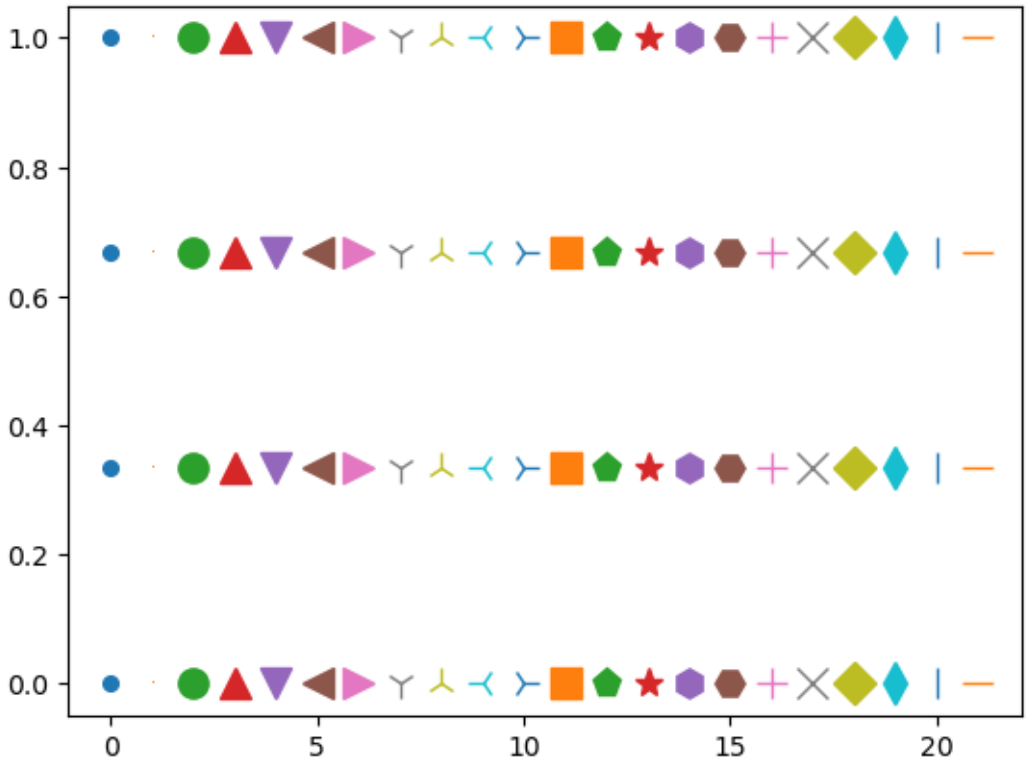
b	g	r	c	m	y	k	w
blue	green	red	cyan	magenta	yellow	black	white

[6]:
```
linestyles = ['-', '--', '-.', ':']
```

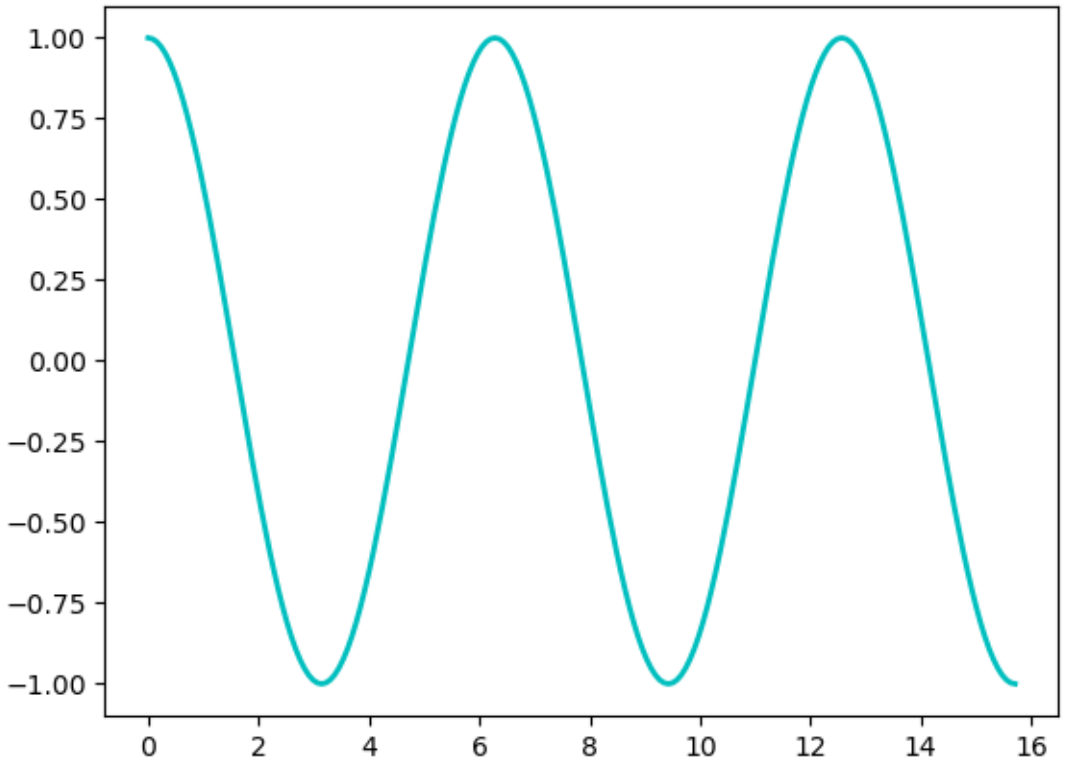
[7]:
```
fig, ax = plt.subplots()
for count, ls in enumerate(linestyles):
    ax.plot([count, count], [0, 1], ls, linewidth=3)
```



[8]:
```
fig, ax = plt.subplots()
markers = ['.', ',', 'o', '^', 'v', '<', '>', '1', '2', '3', '4', 's', 'p',]
markers += ['*', 'h', 'H', '+', 'x', 'D', 'd', '|', '_']
y = np.linspace(0, 1, 4)
x = np.zeros_like(y)
for i, m in enumerate(markers):
    ax.plot(x+i, y, m, markersize=11)
```

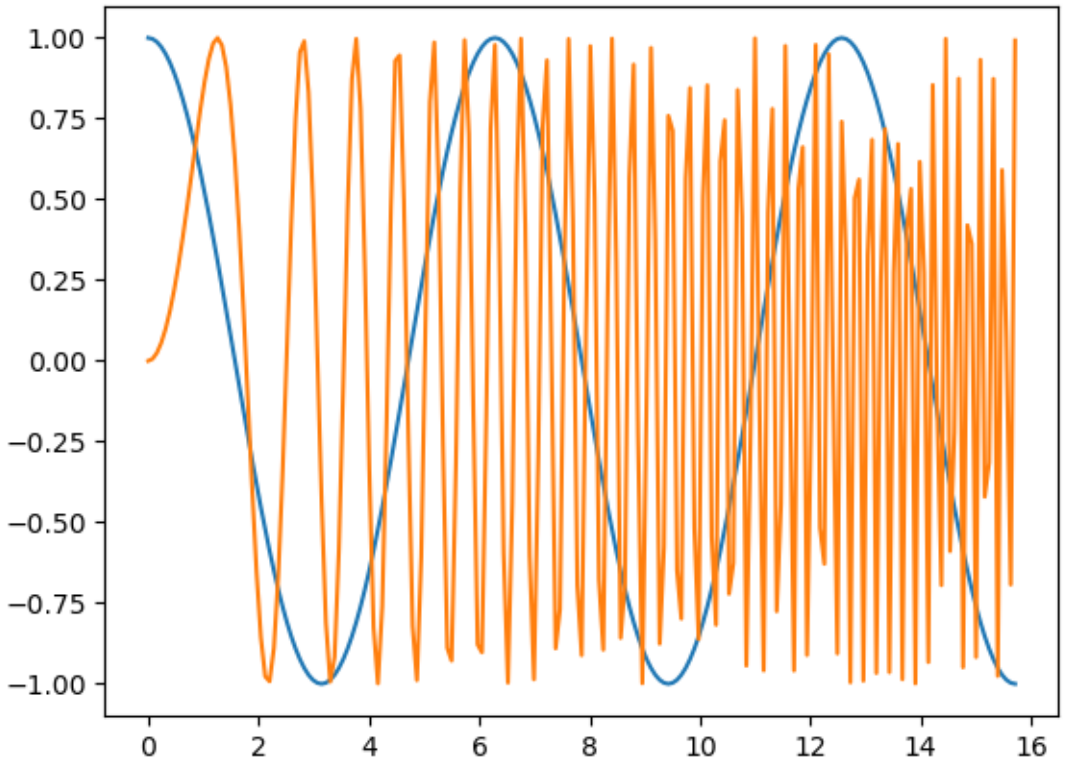


[9]:
```
fig, ax = plt.subplots()
x = np.linspace(0, 5*np.pi, 201)
y = np.cos(x)
ax.plot(x, y, 'c', linewidth=2);
```



Mehrere Graphen in einem Bild

[10]:
```
fig, ax = plt.subplots()
z = np.sin(x**2)
ax.plot(x, y)
ax.plot(x, z);
```

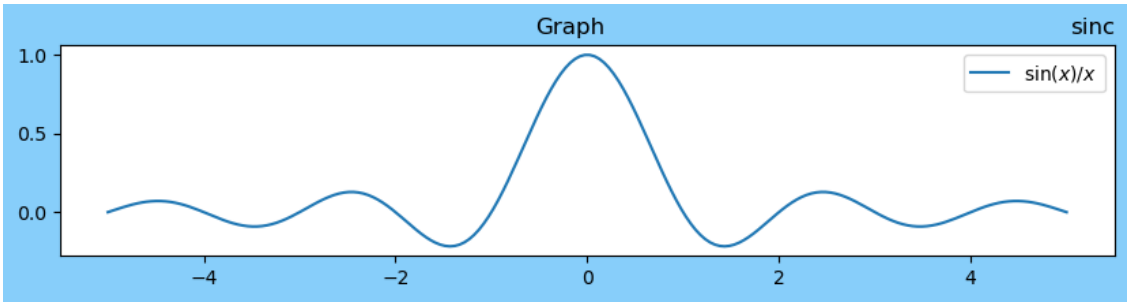


Hier gibt es noch nicht genug Punkte

[11]:
```
x = np.linspace(0, 5*np.pi, 1101)
y = np.cos(x)
z = np.sin(x**2)
ax.clear()
ax.plot(x, y)
ax.plot(x, z);
```

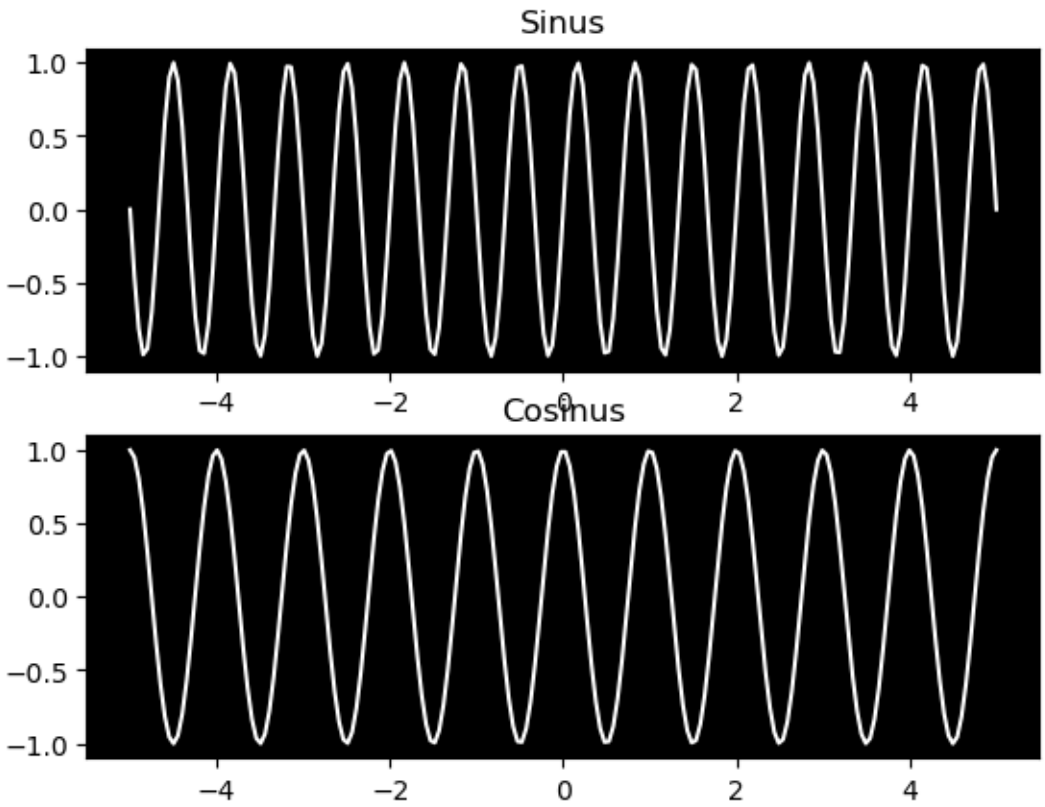
Format ändern, Titel und Legende

[12]:
```
fig, ax = plt.subplots(figsize=(10, 2), facecolor='lightskyblue')
fig.suptitle("Graph")
x = np.linspace(-5, 5, 200)
ax.plot(x, np.sinc(x), label='$\\sin(x)/x$')
ax.set_title('sinc',loc='right')
ax.legend(loc="upper right");
```



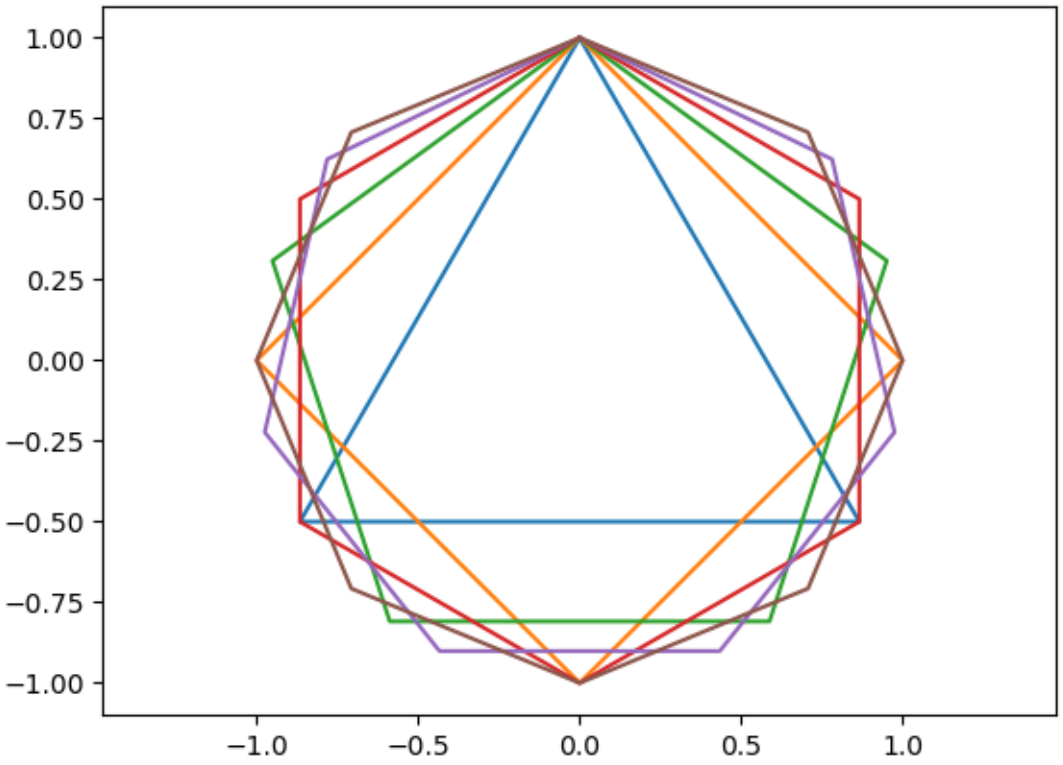
[13]:
```
ax.plot(x, np.sin(3*np.pi*x), 'w')
ax.set_title("Sinus",loc='left')
ax.set_facecolor('k')
```

[14]:
```
fig, axs = plt.subplots(2, 1)    # 2 x 1 Koordinatensysteme
axs[0].plot(x, np.sin(3*np.pi*x), 'w')
axs[1].plot(x, np.cos(2*np.pi*x), 'w')
axs[0].set_title('Sinus')
axs[1].set_title('Cosinus')
for ax in axs:
    ax.set_facecolor('k')
```



[15]:
```
def n_eck(n):
    ''' Eckpunkte eines n-Ecks'''
    x = [np.sin(2*np.pi*j/n) for j in range(n+1)]
    y = [np.cos(2*np.pi*j/n) for j in range(n+1)]
    return x, y
```

[16]:
```
fig, ax = plt.subplots()
for n in range(3, 9):
    ax.plot(*n_eck(n))  # <- Auspacken
ax.axis('equal');
```



[ ]: