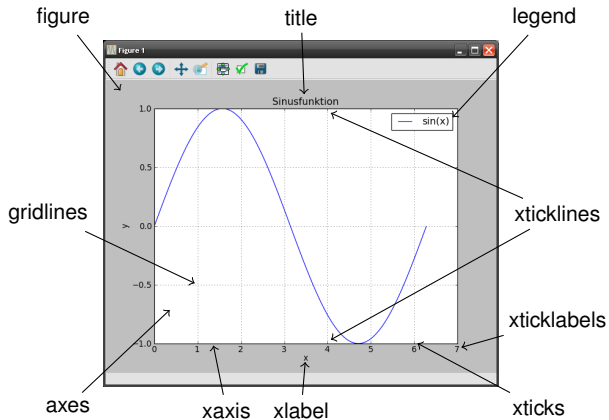


```
# Modul einbinden
import matplotlib.pyplot as plt
# Zeichenfläche erstellen
fig = plt.figure()
# Achsen hinzufügen
ax = fig.add_subplot(111, title="My plot")
# Daten plotten
plt.plot(x, y)
# Fenster mit Plot anzeigen
plt.show()
```



Zwei Möglichkeiten, kurz als Format-String

```
plot(x, y, '-.')
```

Lang als Keyword-Argument (Kurz- und Langform zulässig)

```
plot(x, y, linestyle='dashed') oder plot(x, y, ls='--')
```

Kurzform	Langform	Beschreibung	Output
' ' oder ' '	<code>None</code>	ohne Linie	
'-'	<code>'solid'</code>	durchgehende Linie (Vorgabe)	————
'--'	<code>'dashed'</code>	gestrichelte Linie	-----
'-.'	<code>'dashdot'</code>	Strichpunktlinie	-.-.-.-.
':'	<code>'dotted'</code>	gepunktete Linie	

Wiederum zwei Möglichkeiten, kurz als Format-String

```
plot(x, y, 'o')
```

Lang als Keyword-Argument

```
plot(x, y, marker='h')
```

Größe mit `markersize=10` oder `ms=10` einstellbar

Auswahl:

Kurzform	Beschreibung	Output
' ' oder <code>None</code>	ohne Marker (Vorgabe)	
'.'	Punkt	●
'o'	Kreis	○
'D'	Diamant	◇
'H'	Hexagon	⬡
'+'	Plus	+
's'	Quadrat	■

Wiederum als Kurzform im Format-String, oder als Keyword-Agrument

```
plot(x, y, color='green')
```

Grau lässt sich durch Übergabe einer Zahl (als Zeichenkette) zwischen 0 (schwarz) und 1 (weiß) erreichen: `plot(x, y, color='0.3')`

Für anspruchsvollere Fälle kann man

- die Hexadezimale Notation („HTML“) verwenden: `color='#e3e6f7'`
- ein Tupel für RGB vorgeben: `color=(0.3, 0.8, 0.1)`

Grundlegende Farben sind vordefiniert:

Kurzform	Langform	Farbe
'b'	'blue'	Blau
'g'	'green'	Grün
'r'	'red'	Rot
'c'	'cyan'	Cyan
'm'	'magenta'	Magenta
'y'	'yellow'	Gelb
'k'	'black'	Schwarz
'w'	'white'	Weiß