

# □ BmswGraphLib Cheat Sheet

Die Library für das Schweizer Koordinatensystem in Python.

## 1. Basis-Setup & System

```
from bmsw_graph_lib import b

# Erstellt das Koordinatensystem (x_min, x_max, y_min, y_max)
b.draw_system(-5, 5, -2, 8)

# hier folgen Funktionen, Boxplots, Barcharts, Histogramme, dots und texte
# ...

# Legende und Anzeige
b.legend(loc='upper right')
b.show()

# Speichern (erkennt Dateiname automatisch vom Skriptnamen)
b.save_system("png") # "png", "svg", "eps" ebenfalls möglich
```

## 2. Zeichnen & Beschriften

Methode	Parameter	Beschreibung
draw_function_into_system	func, x_range, label, color	Zeichnet Funktionen (nutze b.log, b.sin, etc.).
draw_polygon	points, color, label	Zeichnet Flächen (20% Opazität).
dot	x, y, style	Einfacher Punkt (z.B. 'ko' für Schwarz, 'rx' für rotes Kreuz).
labeled_dot	x, y, label, color	Setzt Punkt und Text-Label gleichzeitig.
text	x, y, label, color	Schreibt Text an eine Koordinate.

## 3. Diagramm-Typen

- **Säulendiagramm:** b.draw\_bar\_chart(x\_values, y\_values, width=0.8)
- **Histogramm:** b.draw\_histogram(data, bin\_width, start\_value)
- **Boxplot:** b.draw\_boxplot(data, y\_position, height=1.5)  
*(Tipp: Nutze show\_y\_axis=False bei Boxplots)*
- **Kreisdiagramm:** b.draw\_pie\_chart(captions, values, mode="relativ")  
*(Modi: "none", "relativ", "absolute")*

## 4. Spezielle Achsen-Beschriftungen

Um die x-Achse für Kategorien oder Trigonometrie umzubauen:

- **Trigonometrie:** `b.set_trig_labels()` (Setzt  $\pi$ -Schritte automatisch)
- **Eigene Texte:** `b.set_custom_labels([1, 2, 3], ['A', 'B', 'C'])`

### □ Mathe-Profi-Tipp

Wenn du innerhalb deiner Funktionen (Lambda oder def) mathematische Operationen nutzt, verwende immer die Methoden des Objekts `b`, um Fehler bei negativen Zahlen zu vermeiden:

- `b.sqrt(x)`, `b.log(x)`, `b.exp(x)`
- `b.sin(x)`, `b.cos(x)`, `b.tan(x)`
- `b.pi`