

TKInter

- Es gibt APIs = Application Programming Interface
- für GUIs = Graphical Use Interface
- in Python gibt es tkinter (relativ einfach)
- alternativ aber komplizierter sind PyGtk, PyQt, ...
- unser Ziel: Visualisierung unserer geometrischen Objekte.

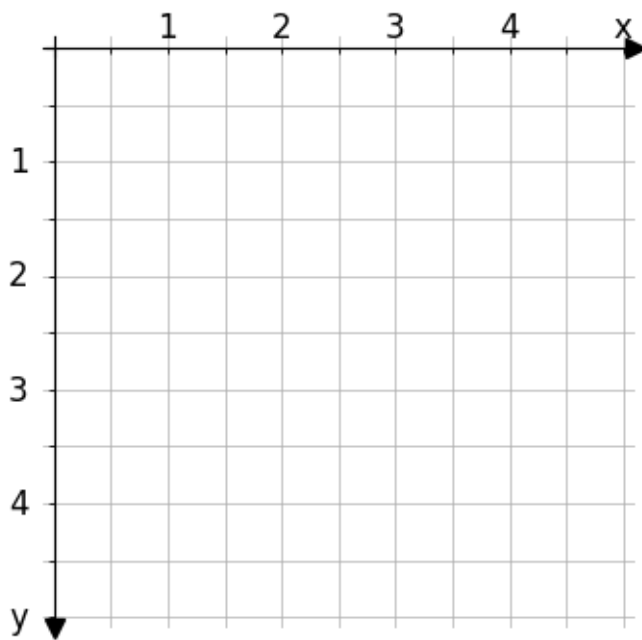
Fenster

1. tkinter wird importiert
2. mit Hilfe des Tk()-Konstruktors wird ein Wurzelobjekt erzeugt.
 - Dieses Wurzelobjekt ist ein Fenster.
 - Zu diesem Objekt können weitere Bestandteile zugefügt werden.
3. `lab` repräsentiert ein Label-Widget.
 - Das Label wird dem Wurzelement hinzugefügt.
 - Widget = rechteckige Fläche auf dem Bildschirm mit einer Funktionalität
 - Label-Widget kann nur Text anzeigen.
4. Anordnung des Widgets im Fenster
 - Vorgefertigte

Canvas

1. Canvas dem Fenster
 - Canvas ist ein Widget
 - Canvas = Leinwand
 - Auf Canvas können geometrische Figuren gemalt werden.
 - Der Konstruktor `tk.Canvas()` übernimmt als Parameter Breite und Höhe in Pixeln
2. Auf dem Canvas malen

Graphikkordinaten



Canvas Methoden

- Linie von (x_1, y_1) nach (x_2, y_2)
`canvas.create_line(x1, y1, x2, y2, **options)`
- Rechteck von obere linke Ecke (x_1, y_1) nach untere rechte Ecke (x_2, y_2)
`canvas.create_rectangle(x1, y1, x2, y2, **options)`
- Oval innerhalb des Rechtecks gebildet durch obere linke Ecke (x_1, y_1) und untere rechte Ecke (x_2, y_2)
`canvas.create_oval(x1, y1, x2, y2, **options)`

=> Alle `create`-Methoden liefern den Index des erzeugten Objekts

= Eindeutige Referenz auf das Objekt. - Damit kann das Objekt noch nachträglich geändert werden.