



# Tag 10

## GUI with tkinter – Layouten und Events

28. April 2023

# Ablauf

- Evt. Prüfung besprechen
- Ausblick Modul
- Rückblick und Rekapitulation GUI
- Layouts
  - Übung
- Funktionalität / Handler (Events)
  - Übung
- Projektarbeit

# Rekapitulation GUI mit tkinter

# Layouts

- **pack**
  - Von oben nach unten
  - Standardmässig vertikal oben & horizontal mittig ausgerichtet
- **grid**
  - Regelmässiges Gitter mit <auto> oder \* Spalten und Zeilen
- **place**
  - Absolutes positionieren
  - Selten verwendet

# pack layout

## Tutorial

### Wichtigste Optionen:

- ipadx, ipady
- padx, pady
- fill & expand
- anchor

internes padding (aka margin)  
padding

# grid layout

## Tutorial

### Wichtigste Optionen:

- column, row
- rowspan/columnspan
- sticky
- ipadx, ipady
- padx, pady

Zelle X/Y, obligatorisch

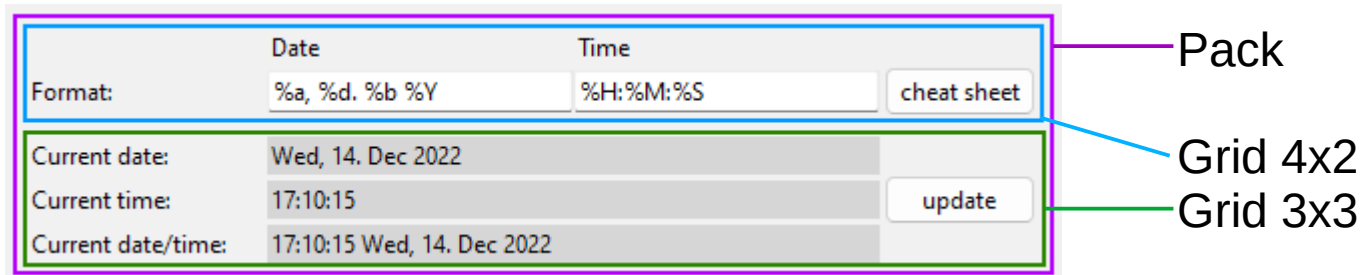
vergleichbar mit *anchor*  
internes padding (aka margin)  
padding

# Demo & Hands-on

- pack layout manager
- grid layout manager

# Verschachtelung

Reale Screens bestehen aus verschachtelten Frames



The diagram illustrates a GUI window with nested frames and layout annotations:

- Pack:** A purple line points to the outermost frame, which contains the top section (Format, Date, Time, cheat sheet) and the bottom section (Current date, Current time, Current date/time, update).
- Grid 4x2:** A blue line points to the top section, which is a 4x2 grid containing the Format, Date, Time, and cheat sheet elements.
- Grid 3x3:** A green line points to the bottom section, which is a 3x3 grid containing the Current date, Current time, Current date/time, and update elements.

	Date	Time	
Format:	%a, %d. %b %Y	%H:%M:%S	cheat sheet
Current date:	Wed, 14. Dec 2022		
Current time:	17:10:15		update
Current date/time:	17:10:15 Wed, 14. Dec 2022		



# Events

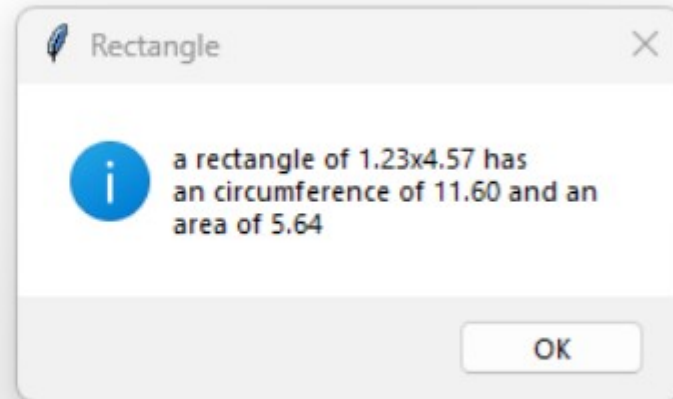
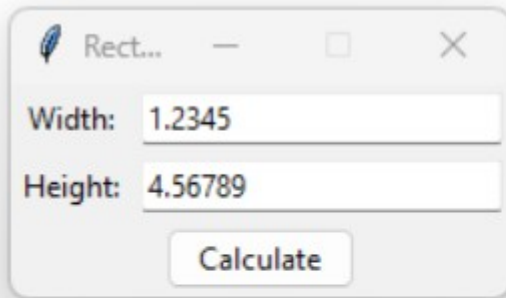
- Einfach – command binding  
`widgets.Button(root, text="Click Me", command=button_clicked1)`
- Erweitert – event binding  
`btn = widgets.Button(root, text="Click Me")`  
`btn.bind("<Button>", button_clicked2)`
- Erweitertes Binding notwendig wenn kein Kommando zur Verfügung steht.
  - Listbox selection changes
  - Doppelklicks
  - Uvm.
- `command=x` und `bind(x)` erhalten die Funktion selber, nicht den Aufruf  
=> keine Klammern() am Schluss
- handler (Funktion) für command binding (hier: `button_clicked1`) hat kein Argument.  
Der handler für das erweiterte Binding (`button_clicked2`) hat ein Argument «event\_argument»

# Variablen

- z. B. ***StringVar***
- Verwaltet Werte, die vom Benutzer eingegeben werden
- Sie «vermitteln» zwischen UI und Model
- 
- Mit callbacks um informiert zu werden, wenn der Wert geändert wird.  
Siehe [Tutorial](#).

# Übung

Erstelle eine Benutzeroberfläche mit tkinter, welches die Fläche und den Umfang eines Rechtecks berechnet.



- Erste Spalte mit den fixen Texten soll so weit sein wie notwendig
- Zweite Spalte so die restliche Weite einnehmen
- Button soll zentriert sein
- Alle Elemente sollen etwas Abstand zueinander haben
- Es sollen nur Widgets verwendet werden
- Zahlen in der Nachrichtenbox sollen mit zwei Nachkommastellen dargestellt werden
- Fenster soll 200 x 85 Pixel gross sein

# Übung

- Demo-GUI (b1\_main.py und Abhängigkeiten (bX\_\_\_\_.py) durchgehen
- Debuggen, verstehen und testweise Änderungen vornehmen
- 3ter Tab einführen: Computerinformationen mit Modul «platform» darstellen