Einführung in Tkinter

Was ist Tkinter?

Tkinter ist die Standard-GUI-Bibliothek von Python. Sie ermöglicht die Erstellung von grafischen Benutzeroberflächen (GUIs) für plattformübergreifende Anwendungen. Tkinter basiert auf dem Tk-Toolkit und ist einfach zu verwenden, leistungsstark und flexibel.

Aufbau einer Tkinter-Anwendung

Eine Tkinter-Anwendung besteht aus:

1. **Import der Bibliothek**:

   ```
   import tkinter as tk
   ```

   Alternativ können spezifische Klassen importiert werden, z. B.:

   ```
   from tkinter import Label, Button
   ```

2. **Erstellen des Hauptfensters**: Das Hauptfenster ist die Grundlage jeder Tkinter-App. Es wird mit `Tk()` erstellt:

   ```
   root = tk.Tk()
   ```

3. **Platzierung von Widgets**: Widgets sind grafische Elemente wie Buttons, Labels oder Eingabefelder. Diese müssen dem Hauptfenster hinzugefügt und positioniert werden.

4. **Starten der Hauptschleife**: Die Hauptschleife (`mainloop()`) sorgt dafür, dass das Fenster geöffnet bleibt und Benutzereingaben verarbeitet werden:

   ```
   root.mainloop()
   ```

Hauptschleife (`mainloop()`)

- **Was macht `mainloop()`?** Die Hauptschleife ist der Kern jeder GUI-Anwendung. Sie sorgt dafür, dass das Fenster sichtbar bleibt und Eingaben wie Mausbewegungen oder Tastendrücke verarbeitet werden.
- **Warum ist `mainloop()` notwendig?** Ohne die Hauptschleife würde das Fenster sofort geschlossen werden, da der Python-Interpreter das Ende des Skripts erreicht.
- **Funktionsweise:**
  - Die Hauptschleife wartet auf Ereignisse (z. B. Button-Klicks).
  - Sie führt zugehörige Funktionen (Callbacks) aus, wenn ein Ereignis eintritt.
  - Sie aktualisiert das Fenster, um Änderungen anzuzeigen.

Widgets und Layout-Manager

Widgets

Widgets sind grafische Bausteine einer GUI. Einige wichtige Widgets:

- **Label**: Zeigt statischen Text oder Bilder an.

- **Button**: Interaktives Element, das eine Aktion ausführt.
- **Entry**: Textfeld für Benutzereingaben.

Beispiel:

```
label = tk.Label(root, text="Hallo, Welt!")
button = tk.Button(root, text="Klick mich!")
```

Layout-Manager

Widgets müssen innerhalb des Fensters positioniert werden. Tkinter bietet drei Layout-Manager:

1. `pack()`: Ordnet Widgets relativ zueinander an (von oben nach unten).
2. `grid()`: Rasterbasierte Anordnung in Zeilen und Spalten.
3. `place()`: Absolute Positionierung mit x- und y-Koordinaten.

Warum müssen Layout-Manager verwendet werden? Ohne Layout-Manager wüsste Tkinter nicht, wo Widgets im Fenster angezeigt werden sollen.

Beispiele:

```
label.pack()  # Platziert das Label
button.pack()  # Platziert den Button unterhalb des Labels
```

Unterschiede zwischen `tk` und `ttk`

- `tk`:
  - Standard-Widgets von Tkinter.
  - Schnell und einfach.
- `ttk`:
  - Bietet "Themed Widgets" für moderneres Design.
  - Beispiele: `ttk.Button`, `ttk.Entry`.

Beispiel:

```
from tkinter import ttk
button = ttk.Button(root, text="Moderner Button")
```

Beispiel: Ein einfaches Tkinter-Fenster

```
import tkinter as tk

# Hauptfenster erstellen
root = tk.Tk()
root.title("Mein erstes Tkinter-Fenster")
root.geometry("300x200")

# Label hinzufügen
label = tk.Label(root, text="Willkommen bei Tkinter!")
label.pack()

# Button hinzufügen
def on_button_click():
    print("Button wurde geklickt!")

button = tk.Button(root, text="Klick mich!", command=on_button_click)
```

```
button.pack()

# Hauptschleife starten
root.mainloop()
```