



Tkinter e CustomTkinter: Interface Gráfica em Python

Introdução ao Tkinter



Tkinter é a biblioteca padrão do Python para a criação de interfaces gráficas (GUIs). Ela fornece uma maneira simples e eficiente de criar janelas, botões, menus e outros elementos de interface. Com Tkinter, você pode desenvolver aplicações desktop que são interativas e visualmente agradáveis.

Principais características do Tkinter

- 
- 1 Facilidade de uso:**
É fácil de aprender e usar, especialmente para iniciantes.
 - 2 Widgets:**
Oferece uma variedade de widgets como botões, rótulos, caixas de texto, menus, etc.
 - 3 Compatibilidade:**
Funciona em várias plataformas, incluindo Windows, macOS e Linux.

Benefícios do Tkinter

1 Simplicidade

Construção rápida
de interfaces básicas
sem complexidade

2 Flexibilidade

Criando aplicações GUI de
diversos tipos

3 Ampla Documentação

Recursos e exemplos
para solucionar
problemas



Limitações do Tkinter

Design Padrão

Aparência básica,
pode parecer
desatualizada

Personalização Limitada

Opções de estilos e temas
restritas

Funcionalidade Básica

Falta de componentes
avançados para
aplicações complexas

Exemplo básico de Tkinter:

```
hoje > ⚡ basico_tkinter.py > ...
1  import tkinter as tk
2
3  def click():
4      print("Botão clicado!")
5
6
7  janela = tk.Tk()
8  janela.title("Exemplo de Tkinter")
9
10
11 botao = tk.Button(janela, text="Clique aqui", command=click)
12 botao.pack()
13
14
15 janela.mainloop()
```

Introdução ao CustomTkinter

CustomTkinter é uma biblioteca que estende o Tkinter, permitindo que você crie interfaces gráficas mais modernas e personalizadas. Ela oferece uma aparência mais estilizada e widgets que podem ser facilmente personalizados.

Principais características do CustomTkinter

1 **Estilo moderno:**
Permite criar interfaces com um design mais contemporâneo.

2 **Personalização:**
Oferece opções para personalizar cores, tamanhos e estilos de widgets.

3 **Facilidade de uso:**
Mantém a simplicidade do Tkinter, mas com mais opções de personalização.

Exemplo básico de CustomTkinter:

```
hoje > ✎ basico_custom_tkinter.py > ...
1  import customtkinter as ctk
2
3  def click():
4      print("Botão clicado!")
5
6
7  janela = ctk.CTk()
8  janela.title("Exemplo de CustomTkinter")
9
10
11 botao = ctk.CTkButton(janela, text="Clique aqui", command=click)
12 botao.pack()
13
14
15 janela.mainloop()
```

Exemplos práticos e melhores práticas

1

Calculadora

Combinando botões
e entradas

2

**Player de
música**

Usando sliders e
botões

3

Editor de texto

Implementando texto
e barras de rolagem

Conclusão



Tkinter: base sólida, CustomTkinter: evolução.

Ambas as bibliotecas são ótimas para criar interfaces gráficas, mas se você deseja um design mais moderno e opções de personalização, o CustomTkinter pode ser a melhor escolha.