

Introdução ao Tkinter

O **Tkinter** é uma biblioteca padrão do Python para criação de interfaces gráficas (GUIs). Com ele, você pode criar janelas, botões, rótulos, caixas de texto e outros elementos visuais de maneira simples. Neste documento, vamos aprender como criar uma tela básica e adicionar componentes a ela.

1. Criando uma janela básica

Para criar uma janela, é necessário importar o módulo tkinter e usar a função Tk() para criar a janela principal. A janela é mantida aberta utilizando o método mainloop().

```
import tkinter as tk

# Cria a janela principal
janela = tk.Tk()

# Define o título da janela
janela.title("Minha primeira tela em Tkinter")

# Define o tamanho da janela
janela.geometry("400x300")

# Roda a aplicação
janela.mainloop()
```

Neste código:

- tk.Tk() cria a janela principal.
- title("Título") define o título da janela.
- geometry("400x300") define o tamanho da janela (400 pixels de largura e 300 pixels de altura).
- mainloop() é um loop que mantém a janela aberta até que o usuário a feche.

2. Adicionando um rótulo (label)

Um **rótulo** é um texto estático exibido na janela. Para adicionar um rótulo, usamos o widget Label().

```
import tkinter as tk

# Cria a janela principal
janela = tk.Tk()

# Define o título da janela
janela.title("Minha primeira tela em Tkinter")

# Define o tamanho da janela
janela.geometry("400x300")

# Adiciona um rótulo
label = tk.Label(janela, text="Bem-vindo à minha primeira interface!")
```

```
label.pack()

# Roda a aplicação
janela.mainloop()
```

Neste código:

- `Label(janela, text="Texto")` cria um rótulo com o texto definido e o coloca na janela.
- `pack()` posiciona o rótulo automaticamente no centro da janela.

3. Adicionando um botão

Os **botões** permitem que o usuário interaja com a interface. Podemos adicionar um botão com o widget `Button()`.

```
import tkinter as tk

# Função chamada quando o botão é clicado
def clique():
    print("Botão clicado!")

# Cria a janela principal
janela = tk.Tk()

# Define o título da janela
janela.title("Minha primeira tela em Tkinter")

# Define o tamanho da janela
janela.geometry("400x300")

# Adiciona um rótulo
label = tk.Label(janela, text="Clique no botão abaixo:")
label.pack()

# Adiciona um botão
botao = tk.Button(janela, text="Clique aqui", command=clique)
botao.pack()

# Roda a aplicação
janela.mainloop()
```

Neste código:

- `Button(janela, text="Texto", command=função)` cria um botão com o texto definido. Quando o botão é clicado, a função `clique()` é chamada.
- `pack()` posiciona o botão na interface.

4. Adicionando um campo de entrada de texto

Para permitir que o usuário digite algum texto, usamos o widget `Entry()`. Podemos pegar o valor digitado com o método `get()`.

```
import tkinter as tk

# Função chamada quando o botão é clicado
def mostrar_texto():
    texto = entrada.get()
    print("Você digitou:", texto)
```

```

# Cria a janela principal
janela = tk.Tk()

# Define o título da janela
janela.title("Minha primeira tela em Tkinter")

# Define o tamanho da janela
janela.geometry("400x300")

# Adiciona um rótulo
label = tk.Label(janela, text="Digite algo:")
label.pack()

# Adiciona um campo de entrada de texto
entrada = tk.Entry(janela)
entrada.pack()

# Adiciona um botão
botao = tk.Button(janela, text="Exibir texto", command=mostrar_texto)
botao.pack()

# Roda a aplicação
janela.mainloop()

```

Neste código:

- `Entry(janela)` cria um campo de entrada de texto.
- `entrada.get()` obtém o valor digitado no campo de texto.
- O botão, ao ser clicado, exibe o valor digitado no terminal.

5. Adicionando e configurando botões

Os **botões** são usados para interagir com o usuário, permitindo que uma ação ocorra ao ser clicado. Para criar um botão, usamos o widget `Button()`.

```

import tkinter as tk

# Função que será executada ao clicar no botão
def acao_botao():
    print("O botão foi clicado!")

# Cria a janela principal
janela = tk.Tk()

# Define o título da janela
janela.title("Botões simples")

# Define o tamanho da janela
janela.geometry("400x300")

# Adiciona um botão com ação
botao = tk.Button(janela, text="Clique aqui", command=acao_botao)
botao.pack()

# Roda a aplicação
janela.mainloop()

```

Neste código:

- `Button(janela, text="Texto", command=função)` cria um botão com o texto "Clique aqui". Quando o botão é clicado, a função `acao_botao()` é chamada.
- `command=acao_botao` faz com que o botão execute a função especificada.

6. Interagindo com o texto de entrada e botões

Além de exibir texto estático, você pode capturar a entrada de texto do usuário utilizando o widget `Entry()`. Com a função `get()`, é possível obter o valor digitado e realizar ações com base nesse valor.

```
import tkinter as tk

# Função que exibe o texto digitado
def mostrar_texto():
    texto = entrada.get() # Obtém o valor do campo de entrada
    print("Você digitou:", texto)

# Cria a janela principal
janela = tk.Tk()

# Define o título da janela
janela.title("Interagindo com texto e botões")

# Define o tamanho da janela
janela.geometry("400x300")

# Adiciona um rótulo explicativo
label = tk.Label(janela, text="Digite algo e clique no botão:")
label.pack()

# Adiciona um campo de entrada de texto
entrada = tk.Entry(janela)
entrada.pack()

# Adiciona um botão que exibe o texto digitado
botao = tk.Button(janela, text="Exibir texto", command=mostrar_texto)
botao.pack()

# Roda a aplicação
janela.mainloop()
```

Neste código:

- `Entry(janela)` cria um campo de entrada para que o usuário possa digitar algo.
- `entrada.get()` captura o valor que foi digitado no campo de texto.
- O botão chama a função `mostrar_texto()`, que exibe o texto digitado no terminal.

7. Alterando o texto de um rótulo com o botão

Você também pode modificar o conteúdo de um rótulo após o clique de um botão. O método `config()` permite alterar o texto exibido no rótulo.

```
import tkinter as tk

# Função que altera o texto do rótulo
def alterar_texto():
    label.config(text="O botão foi clicado!")

# Cria a janela principal
janela = tk.Tk()
```

```

# Define o título da janela
janela.title("Alterando texto com o botão")

# Define o tamanho da janela
janela.geometry("400x300")

# Adiciona um rótulo
label = tk.Label(janela, text="Clique no botão para mudar este texto.")
label.pack()

# Adiciona um botão que altera o texto do rótulo
botao = tk.Button(janela, text="Clique aqui", command=alterar_texto)
botao.pack()

# Roda a aplicação
janela.mainloop()

```

Neste código:

- `label.config(text="Novo texto")` altera o texto do rótulo quando o botão é clicado.
- O botão chama a função `alterar_texto()`, que atualiza o texto do rótulo.

8. Radio Buttons (Botões de Opção)

Os radio buttons são usados quando você deseja que o usuário selecione apenas uma opção de um grupo. Eles são agrupados por uma variável comum que registra qual botão foi selecionado.

```

import tkinter as tk

# Função chamada ao selecionar um botão
def selecionar_opcao():
    print("Opção selecionada:", opcao.get())

# Cria a janela principal
janela = tk.Tk()

# Define o título da janela
janela.title("Exemplo de Radio Buttons")

# Variável que armazena o valor do botão selecionado
opcao = tk.StringVar()

# Cria os radio buttons
radio1 = tk.Radiobutton(janela, text="Opção 1", variable=opcao,
                        value="1", command=selecionar_opcao)
radio1.pack()

radio2 = tk.Radiobutton(janela, text="Opção 2", variable=opcao,
                        value="2", command=selecionar_opcao)
radio2.pack()

radio3 = tk.Radiobutton(janela, text="Opção 3", variable=opcao,
                        value="3", command=selecionar_opcao)
radio3.pack()

# Roda a aplicação

```

```
janela.mainloop()
```

Neste código:

- Radiobutton(janela, text="Texto", variable=opcao, value="Valor", command=função) cria um radio button.
- opcao = tk.StringVar() armazena o valor do botão selecionado.
- Quando o botão é clicado, a função selecionar_opcao() é chamada, exibindo o valor selecionado.

9. Checkboxes (Caixas de Seleção)

As checkboxes permitem que o usuário selecione uma ou mais opções de um conjunto. Cada caixa tem sua própria variável para registrar se está marcada ou não.

```
import tkinter as tk

# Função chamada ao clicar nas caixas de seleção
def mostrar_selecao():
    print("Caixa 1:", opcao1.get())
    print("Caixa 2:", opcao2.get())

# Cria a janela principal
janela = tk.Tk()

# Define o título da janela
janela.title("Exemplo de Checkboxes")

# Variáveis para armazenar o estado das caixas de seleção
opcao1 = tk.IntVar()
opcao2 = tk.IntVar()

# Cria as checkboxes
checkboxox1 = tk.Checkbutton(janela, text="Opção 1", variable=opcao1,
                           command=mostrar_selecao)
checkboxox1.pack()

checkboxox2 = tk.Checkbutton(janela, text="Opção 2", variable=opcao2,
                           command=mostrar_selecao)
checkboxox2.pack()

# Roda a aplicação
janela.mainloop()
```

Neste código:

- Checkbutton(janela, text="Texto", variable=opcao, command=função) cria uma caixa de seleção.
- opcao1 = tk.IntVar() e opcao2 = tk.IntVar() armazenam o estado das caixas (0 para desmarcada e 1 para marcada).
- A função mostrar_selecao() exibe se as caixas estão marcadas ou não.

10. Listbox (Lista)

O Listbox permite exibir uma lista de opções para o usuário selecionar. Ele pode ser configurado para permitir seleção única ou múltipla.

```
import tkinter as tk

# Função chamada ao selecionar um item da lista
def selecionar_item():
    selecao = listBox.curselection()
    for i in selecao:
        print("Item selecionado:", listBox.get(i))

# Cria a janela principal
janela = tk.Tk()

# Define o título da janela
janela.title("Exemplo de Listbox")

# Cria o listbox
listBox = tk.Listbox(janela, selectmode=tk.SINGLE) # Modo de seleção única
listBox.pack()

# Adiciona itens à lista
itens = ["Item 1", "Item 2", "Item 3", "Item 4"]
for item in itens:
    listBox.insert(tk.END, item)

# Botão para selecionar o item
botao = tk.Button(janela, text="Selecionar", command=selecionar_item)
botao.pack()

# Roda a aplicação
janela.mainloop()
```

Neste código:

- `Listbox(janela, selectmode=tk.SINGLE)` cria um listbox com seleção única. Para permitir seleção múltipla, use `selectmode=tk.MULTIPLE`.
- `listBox.insert(tk.END, "Item")` insere itens na lista.
- `listBox.curselection()` retorna os índices dos itens selecionados.
- `listBox.get(indice)` obtém o valor do item selecionado pelo índice.