

Elementos de Tela no Tkinter - Parte 2

Abordaremos agora outros elementos visuais disponíveis no Tkinter, além dos já discutidos. Também veremos como alterar propriedades como cor, estilo e posicionamento.

1. Labels Avançados (Rótulos)

Os labels são usados para exibir texto na tela, mas você pode personalizá-los para alterar a cor de fundo, a cor do texto, a fonte, etc.

```
import tkinter as tk

# Cria a janela principal
janela = tk.Tk()
janela.title("Exemplo de Label")

# Cria um label com propriedades de cor e fonte
label = tk.Label(janela, text="Olá, Mundo!", bg="yellow", fg="blue",
font=("Helvetica", 16))
label.pack()

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

No exemplo acima:

- `bg="yellow"` altera a cor de fundo do label para amarelo.
- `fg="blue"` altera a cor do texto para azul.
- `font=("Helvetica", 16)` altera a fonte para "Helvetica" com tamanho 16.

2. Entry (Campo de Texto)

O Entry é utilizado para inserir textos de uma única linha. Ele pode ser personalizado para alterar a cor de fundo e a largura.

```
import tkinter as tk

# Cria a janela principal
janela = tk.Tk()
janela.title("Exemplo de Entry")

# Cria um campo de texto (Entry)
campo_texto = tk.Entry(janela, bg="lightgray", width=30)
campo_texto.pack()

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

Neste exemplo:

- `bg="lightgray"` define a cor de fundo do campo de texto.
- `width=30` define a largura do campo (em número de caracteres).

3. Text (Caixa de Texto Multilinhas) NOVO

O Text é semelhante ao Entry, mas permite a inserção de múltiplas linhas de texto.

```
import tkinter as tk

# Cria a janela principal
janela = tk.Tk()
janela.title("Exemplo de Text")

# Cria uma caixa de texto
caixa_texto = tk.Text(janela, height=10, width=40, bg="lightyellow",
fg="black")
caixa_texto.pack()

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

Neste exemplo:

- height=10 define a altura em número de linhas.
- width=40 define a largura em número de caracteres.
- bg="lightyellow" define a cor de fundo.
- fg="black" define a cor do texto.

4. Frames (Quadros)

Os frames são usados como contêineres para organizar outros widgets. Eles ajudam a organizar a interface em seções.

```
import tkinter as tk

# Cria a janela principal
janela = tk.Tk()
janela.title("Exemplo de Frame")

# Cria um frame (quadro) com uma cor de fundo
quadro = tk.Frame(janela, bg="lightblue", width=200, height=100)
quadro.pack_propagate(False) # Impede o redimensionamento do frame
quadro.pack()

# Adiciona um label dentro do frame
label = tk.Label(quadro, text="Estou dentro do Frame", bg="lightblue")
label.pack()

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

No exemplo acima:

- Frame(janela, bg="lightblue", width=200, height=100) cria um frame com cor de fundo azul claro, largura de 200 e altura de 100.
- pack_propagate(False) impede que o frame seja redimensionado automaticamente para se ajustar aos widgets internos.

5. Alterando o Posicionamento de Elementos

No Tkinter, você pode usar diferentes métodos de posicionamento:

- ⇒ `pack()`: Posiciona o widget de forma sequencial (um abaixo do outro ou lado a lado).
- ⇒ `grid()`: Posiciona os widgets em uma grade (com linhas e colunas).
- ⇒ `place()`: Posiciona os widgets em coordenadas absolutas (x, y).

Usando o `pack()`

```
label1 = tk.Label(janela, text="Label 1")
label1.pack(side=tk.LEFT) # Posiciona à esquerda

label2 = tk.Label(janela, text="Label 2")
label2.pack(side=tk.RIGHT) # Posiciona à direita
```

Usando o `grid()`

```
label1 = tk.Label(janela, text="Label 1")
label1.grid(row=0, column=0) # Linha 0, coluna 0

label2 = tk.Label(janela, text="Label 2")
label2.grid(row=0, column=1) # Linha 0, coluna 1
```

Usando o `place()`:

```
label1 = tk.Label(janela, text="Label 1")
label1.place(x=50, y=50) # Posiciona em (x=50, y=50)
```

6. Botões com Imagens

Você pode adicionar imagens a um botão para criar interfaces mais visuais.

```
import tkinter as tk
from tkinter import PhotoImage

# Cria a janela principal
janela = tk.Tk()
janela.title("Exemplo de Botão com Imagem")

# Carrega uma imagem
imagem = PhotoImage(file="caminho_para_imagem.png")

# Cria um botão com imagem
botao_imagem = tk.Button(janela, image=imagem)
botao_imagem.pack()

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

Neste exemplo:

- `PhotoImage(file="caminho_para_imagem.png")` carrega a imagem.
- `Button(janela, image=imagem)` cria um botão que exibe a imagem.

7. Alterando o Estilo de Botões e Widgets

Os botões e outros widgets no Tkinter podem ser estilizados com diferentes propriedades, como a cor de fundo, cor da borda, tamanho da fonte, etc.

```
import tkinter as tk

# Cria a janela principal
janela = tk.Tk()
janela.title("Estilo de Botões")

# Cria um botão com estilo personalizado
botao = tk.Button(janela, text="Clique Aqui", bg="green", fg="white",
font=("Arial", 14, "bold"))
botao.pack()

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

Aqui:

- `bg="green"` define a cor de fundo do botão.
- `fg="white"` define a cor do texto do botão.
- `font=("Arial", 14, "bold")` define a fonte, tamanho e estilo (negrito).

8. Personalizando Janela Principal

Você pode alterar algumas propriedades da janela principal, como o tamanho inicial e ícones.

Alterando o tamanho da janela

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

Alterando o ícone da janela:

janela.iconbitmap("caminho_para_icone.ico") # Define o ícone da janela
```

9. Exercícios

Esta lista de exercícios foi criada para ajudar você a praticar a criação de interfaces gráficas em Python usando o Tkinter.

Cada exercício aborda conceitos importantes que já vimos, como botões, labels, caixas de texto, radio buttons, frames e posicionamento de widgets.

Tente criar as telas de acordo com as especificações fornecidas e personalizá-las utilizando cores, estilos e fontes.

1. Crie uma interface gráfica que simule um formulário de cadastro com os seguintes campos:

- Nome (Entry)
- E-mail (Entry)
- Gênero (Radio Buttons: Masculino/Feminino)
- Botão “Enviar” que, ao ser clicado, exibe os dados inseridos em um label.

2. Crie uma calculadora com dois campos para entrada de números (Entry) e quatro botões para as operações (Soma, Subtração, Multiplicação, Divisão).
3. Crie uma tela de login com os seguintes elementos:
 - Campo de texto para o nome de usuário (Entry)
 - Campo de texto para senha (Entry, use show='*' para esconder os caracteres)
 - Botão “Login” que, ao ser clicado, exibe uma mensagem de boas-vindas – utilize print()
 - Use o método grid() para posicionar os elementos na tela.
4. Crie uma tela com um botão que exibe uma imagem.
 - Ao clicar no botão, exiba uma mensagem de “Imagem Clicada!” – utilize print().
 - Utilize o método pack() para organizar os elementos.
5. Crie uma interface que permita ao usuário calcular o Índice de Massa Corporal (IMC).
 - Insira dois campos para que o usuário informe o peso (kg) e a altura (m).
 - Adicione um botão “Calcular IMC”.
 - Ao clicar no botão, o IMC deve ser exibido – utilize print().
 - Utilize place() para posicionar os widgets nas coordenadas desejadas.
6. Crie uma tela com os seguintes elementos:
 - Três radio buttons com opções de cores: “Vermelho”, “Verde”, “Azul”.
 - Um botão que, ao ser clicado, altera a cor de fundo da janela para a cor selecionada.
 - Exiba a cor selecionada – utilize print().
7. Crie uma interface com um campo de texto multilinhas (Text).
 - Adicione um botão “Salvar” que, ao ser clicado, exibe o texto digitado em um label.
 - Personalize o campo de texto com uma cor de fundo e cor da fonte.
 - Use o método pack() para organizar os elementos.
8. Crie uma interface que utilize dois frames (quadros) para organizar os elementos:
 - No primeiro frame, adicione dois labels com textos explicativos.
 - No segundo frame, adicione três botões (Botão 1, Botão 2, Botão 3) e faça com que cada um exiba uma mensagem diferente em um label quando clicado.
 - Personalize a cor de fundo de cada frame e posicione os frames utilizando pack().
9. Crie uma interface com três opções que o usuário pode selecionar usando checkboxes:
 - “Receber Novidades”, “Aceitar Termos”, “Ativar Notificações”.
 - Adicione um botão “Confirmar” que, ao ser clicado, exibe quais opções foram selecionadas em um label.

10. Crie uma interface com uma lista de seleção (Listbox) contendo os nomes de cinco frutas (ex.: “Maçã”, “Banana”, “Laranja”, etc.).

- Adicione um botão “Selecionar” que, ao ser clicado, exibe a fruta selecionada em um label.
- Personalize a cor de fundo do listbox e o tamanho da fonte.

11. Crie uma tela que organize três labels e três botões usando o método grid().

- Cada botão, quando clicado, deve exibir uma mensagem no label correspondente (ex.: “Botão 1 foi clicado!”).
- Personalize o tamanho e a cor dos botões.

12. Crie um formulário com os seguintes campos:

- Nome (Entry)
- E-mail (Entry)
- Um listbox com três opções de curso (ex.: “Python”, “Kotlin”, “Java”).
- Um botão “Enviar” que, ao ser clicado, exibe todas as informações selecionadas (nome, e-mail e curso) em um label.

Dicas

- ⇒ Utilize os métodos pack(), grid() ou place() para posicionar os elementos conforme necessário.
- ⇒ Personalize os elementos com cores (bg, fg), estilos de fonte (font) e tamanhos para deixar sua interface mais visual.
- ⇒ Teste diferentes combinações de widgets para entender como cada elemento interage na tela.