

Aula 6: Layouts no Tkinter (pack, grid, place)

Nesta aula, vamos aprender como organizar os elementos da interface (widgets) na janela utilizando os principais gerenciadores de layout do Tkinter: `pack`, `grid` e `place`.

Por que usar um gerenciador de layout?

Quando adicionamos widgets em uma interface, precisamos decidir **onde** eles serão posicionados. Em vez de usar coordenadas fixas, o Tkinter oferece formas dinâmicas e flexíveis de organizar os widgets. Isso facilita a criação de interfaces adaptáveis e mais bem estruturadas.

Layout com `pack`

O `pack()` posiciona widgets em relação uns aos outros, empilhando-os por padrão de cima para baixo. Podemos usar o argumento `side` para mudar a direção:

```
import tkinter as tk

janela = tk.Tk()

btn1 = tk.Button(janela, text="Topo")
btn1.pack(side="top")

btn2 = tk.Button(janela, text="Esquerda")
btn2.pack(side="left")

btn3 = tk.Button(janela, text="Direita")
btn3.pack(side="right")

janela.mainloop()
```

Parâmetros comuns:

- **side**: "top", "bottom", "left", "right"
 - **fill**: preenche espaço (ex: **fill**="x" para largura)
 - **expand**: permite ocupar espaço extra
-

Layout com **grid**

O **grid()** permite organizar os widgets em linhas e colunas, como uma tabela.

```
import tkinter as tk
```

```
janela = tk.Tk()
```

```
tk.Label(janela, text="Nome:").grid(row=0, column=0)
```

```
tk.Entry(janela).grid(row=0, column=1)
```

```
tk.Label(janela, text="Idade:").grid(row=1, column=0)
```

```
tk.Entry(janela).grid(row=1, column=1)
```

```
janela.mainloop()
```

Parâmetros comuns:

- **row**: linha
 - **column**: coluna
 - **columnspan**: quantas colunas ocupar
 - **padx, pady**: espaçamento externo
 - **sticky**: alinhar ("n", "s", "e", "w")
-

Layout com `place`

O `place()` posiciona widgets com coordenadas exatas (x, y). É menos flexível e geralmente evitado em interfaces responsivas.

```
import tkinter as tk

janela = tk.Tk()

botao = tk.Button(janela, text="Clique aqui")
botao.place(x=50, y=100)

janela.mainloop()
```

Use `place` apenas quando for absolutamente necessário controlar a posição com precisão.

Exercícios

Exercício 1:

Crie três botões empilhados verticalmente usando o `pack()`.

Exercício 2:

Crie um pequeno formulário com `grid()` contendo dois campos: "Nome" e "Idade", e um botão "Enviar" abaixo deles.

Exercício 3:

Crie um botão posicionado com `place()` no canto inferior direito da janela.

Desafio Final

Monte uma interface com os seguintes elementos:

- Um título centralizado usando `pack()`
 - Um formulário com "Nome", "Email" e "Mensagem" usando `grid()`
 - Um botão de "Enviar" centralizado usando `pack()`
-

Resoluções

Exercício 1:

```
import tkinter as tk
```

```
janela = tk.Tk()
```

```
tk.Button(janela, text="Botão 1").pack()
```

```
tk.Button(janela, text="Botão 2").pack()
```

```
tk.Button(janela, text="Botão 3").pack()
```

```
janela.mainloop()
```

Exercício 2:

```
import tkinter as tk
```

```
janela = tk.Tk()
```

```
tk.Label(janela, text="Nome:").grid(row=0, column=0)
```

```
tk.Entry(janela).grid(row=0, column=1)
```

```
tk.Label(janela, text="Idade:").grid(row=1, column=0)
```

```
tk.Entry(janela).grid(row=1, column=1)
```

```
tk.Button(janela, text="Enviar").grid(row=2, column=0, columnspan=2)
```

```
janela.mainloop()
```

Exercício 3:

```
import tkinter as tk
```

```
janela = tk.Tk()
```

```
janela.geometry("300x200")
```

```
botao = tk.Button(janela, text="OK")
```

```
botao.place(x=240, y=160)
```

```
janela.mainloop()
```

Desafio Final:

```
import tkinter as tk

janela = tk.Tk()

titulo = tk.Label(janela, text="Formulário de Contato", font=("Arial", 14))
titulo.pack(pady=10)

# Formulário
tk.Label(janela, text="Nome:").grid(row=1, column=0, sticky="e")
tk.Entry(janela).grid(row=1, column=1)

tk.Label(janela, text="Email:").grid(row=2, column=0, sticky="e")
tk.Entry(janela).grid(row=2, column=1)

tk.Label(janela, text="Mensagem:").grid(row=3, column=0, sticky="e")
tk.Entry(janela).grid(row=3, column=1)

# Espaço entre grid e botão
tk.Label(janela, text="").grid(row=4)

# Botão
botao = tk.Button(janela, text="Enviar")
botao.pack(pady=10)

janela.mainloop()
```

Conclusão

Com os gerenciadores de layout do Tkinter (**pack**, **grid** e **place**), você pode organizar seus widgets de forma estruturada e responsiva. Na próxima aula, aprenderemos a combinar tudo isso em interfaces mais completas com múltiplas janelas e elementos interativos.