

Aula 5: Eventos e Interatividade (command, bind)

Nesta aula, vamos aprender como tornar nossas interfaces gráficas mais interativas com eventos. O Tkinter permite responder a ações do usuário, como cliques e pressionamentos de tecla, por meio de comandos (`command`) e associações de eventos (`bind`).

O que são eventos?

Eventos são ações realizadas pelo usuário, como:

- Clicar em um botão
- Pressionar uma tecla
- Mover o mouse

Para responder a esses eventos, usamos duas abordagens:

- `command`: vincula uma função a um widget (ex: Button).
 - `bind`: permite capturar eventos diversos (ex: teclas, mouse).
-

Usando **command**

A forma mais simples de interagir com eventos é usando o parâmetro **command** em botões.

```
import tkinter as tk

def dizer_ola():
    print("Olá!")

janela = tk.Tk()

botao = tk.Button(janela, text="Clique aqui", command=dizer_ola)
botao.pack()

janela.mainloop()
```

A função **dizer_ola** é executada toda vez que o botão é clicado.

Usando **bind**

Com **bind**, podemos associar qualquer tipo de evento a um widget. A função associada deve aceitar um parâmetro (o objeto de evento).

```
import tkinter as tk

def tecla_pressionada(evento):
    print(f"Tecla pressionada: {evento.char}")

janela = tk.Tk()
janela.bind("<Key>", tecla_pressionada)

janela.mainloop()
```

Nesse exemplo, toda vez que uma tecla for pressionada dentro da janela, a função será chamada.

Principais eventos com **bind**

Evento	Ação
<Button-1>	Clique com o botão esquerdo do mouse
<Double-1>	Clique duplo com o mouse
<Key>	Qualquer tecla
<Return>	Tecla Enter
<Motion>	Movimento do mouse

Exemplo de clique do mouse:

```
def clicou(evento):  
    print("Clique detectado!")  
  
janela.bind("<Button-1>", clicou)
```

Exercícios

Exercício 1:

Crie um programa com um botão que, ao ser clicado, mude o texto de um rótulo para "Você clicou!".

Exercício 2:

Crie um campo de entrada e um rótulo. Quando o usuário digitar algo e apertar Enter, o texto do rótulo deve ser atualizado com o que foi digitado.

Exercício 3:

Crie uma área onde, ao mover o mouse, a posição x e y seja exibida em tempo real em um rótulo.

Desafio Final

Crie um pequeno aplicativo com:

- Um campo para o usuário digitar o nome
 - Um botão "Saudar"
 - Um rótulo que exiba: "Olá, [nome]!"
 - Um evento para que, ao pressionar Enter, a saudação também seja exibida
-

Resoluções

Exercício 1:

```
import tkinter as tk

def alterar_texto():
    rotulo.config(text="Você clicou!")

janela = tk.Tk()
rotulo = tk.Label(janela, text="Aguardando clique...")
rotulo.pack()

botao = tk.Button(janela, text="Clique aqui", command=alterar_texto)
botao.pack()

janela.mainloop()
```

Exercício 2:

```
import tkinter as tk

def atualizar_label(evento):
    texto = entrada.get()
    rotulo.config(text=texto)

janela = tk.Tk()
entrada = tk.Entry(janela)
entrada.pack()
entrada.bind("<Return>", atualizar_label)

rotulo = tk.Label(janela, text="")
rotulo.pack()

janela.mainloop()
```

Exercício 3:

```
import tkinter as tk

def mover_mouse(evento):
    rotulo.config(text=f"Posição: {evento.x}, {evento.y}")

janela = tk.Tk()
rotulo = tk.Label(janela, text="Mova o mouse na janela")
rotulo.pack()

janela.bind("<Motion>", mover_mouse)

janela.mainloop()
```

Desafio Final:

```
import tkinter as tk

def saudar(evento=None):
    nome = entrada.get()
    rotulo.config(text=f"Olá, {nome}!")

janela = tk.Tk()
entrada = tk.Entry(janela)
entrada.pack()
entrada.bind("<Return>", saudar)

botao = tk.Button(janela, text="Saudar", command=saudar)
botao.pack()

rotulo = tk.Label(janela, text="")
rotulo.pack()

janela.mainloop()
```

Conclusão

Hoje aprendemos a tornar nossas interfaces responsivas e interativas com eventos do Tkinter. Vimos como usar o **command** para executar ações simples e o **bind** para capturar eventos mais variados como teclas e movimento do mouse. Na próxima aula, exploraremos **gerenciadores de layout** para organizar melhor os componentes na tela.