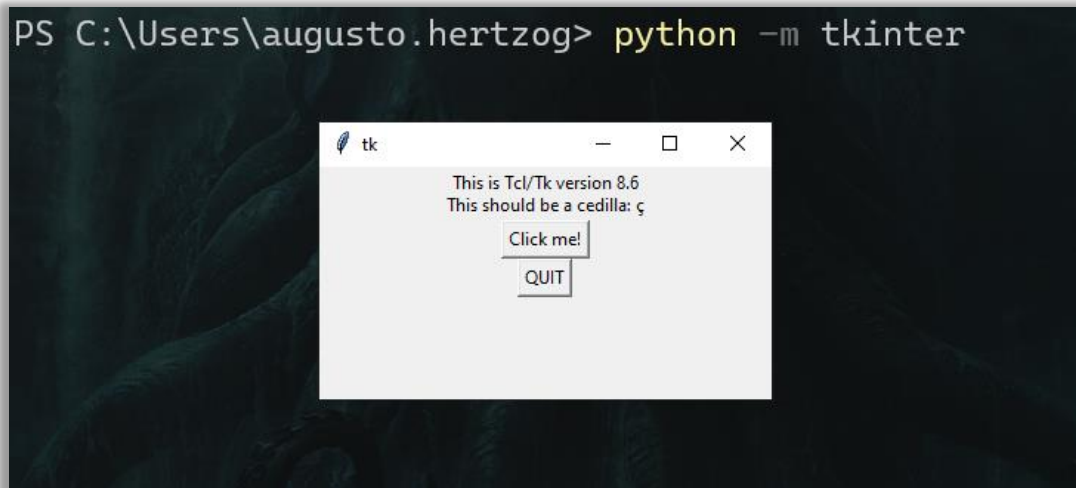


Aula 14

tkinter

O pacote tkinter (Tk interface) é a interface padrão do Python e faz parte do kit de ferramentas Tcl/Tk GUI. Tanto o Tk quanto o tkinter estão disponíveis na maioria das plataformas Unix, incluindo macOS, bem como em sistemas Windows.

Executar `python -m tkinter` na linha de comando deve abrir uma janela demonstrando uma interface Tk simples, informando que tkinter está apropriadamente instalado em seu sistema, e mostrando qual versão do Tcl/Tk está instalado, para que você possa ler a documentação do Tcl/Tk específica para essa versão.



Ele permite que você crie janelas, botões, menus e outros elementos de interface do usuário em seus aplicativos Python.

A maneira mais comum de usar o tkinter é criar uma janela principal, que é uma instância da classe **Tk**. A partir daí, você pode adicionar widgets (como botões, caixas de texto, etc.) a essa janela principal. Cada widget é uma instância de uma classe específica, como **Button**, **Entry**, etc.

Como usar o tkinter

Abaixo, temos uma estrutura simples de uso do tkinter:

```
import tkinter as tk

root = tk.Tk()
button = tk.Button(root, text="Clique Aqui")
button.pack()
root.mainloop()
```

Vamos ver o que cada linha faz:

- Importação do módulo: para começar, precisamos importar o módulo Tkinter no seu script Python. Isso é feito com a seguinte linha de código:

```
import tkinter as tk
```

- Criação de uma janela principal: o Tkinter funciona em torno de uma janela principal (root window), que é a janela principal do seu aplicativo. Você pode criar uma instância dessa janela da seguinte forma:

```
root = tk.Tk()
```

- Widgets: O Tkinter fornece vários widgets que podem ser adicionados à janela principal. Alguns exemplos comuns de widgets incluem botões, rótulos (labels), caixas de texto e listas. Para criar um widget, você precisa usar uma classe específica do Tkinter correspondente a esse widget. Por exemplo, para criar um botão, você pode fazer o seguinte:

```
button = tk.Button(root, text="Clique Aqui")
```

Nesse exemplo, estamos criando um botão e associando-o à janela principal root. O parâmetro text define o texto exibido no botão.

- Posicionamento dos widgets: Depois de criar um widget, você precisa posicioná-lo na janela principal. O Tkinter oferece diferentes gerenciadores de layout, como o pack, o grid e o place. O método pack organiza os widgets em um layout empacotado, o método grid organiza os widgets em uma grade e o método place permite que você coloque os widgets em posições específicas. Aqui está um exemplo usando o método pack:

```
button.pack()
```

- Loop principal: Para que a interface gráfica funcione corretamente, você precisa iniciar o loop principal do Tkinter. Esse loop fica esperando eventos, como cliques de mouse e pressionamentos de tecla, e responde a eles. O loop principal é iniciado chamando o método mainloop na janela principal:

```
root.mainloop()
```

A partir desse ponto, o programa ficará em execução até que a janela seja fechada pelo usuário.

Configurando a janela

A janela principal é uma instância da classe Tk e é a base da sua interface gráfica de usuário. Além de adicionar widgets à janela principal, você também pode personalizar vários aspectos da janela, como o título, tamanho e ícone.

```
import tkinter as tk

janela = tk.Tk()
janela.title("Meu Programa")
janela.geometry('400x300')
janela.iconbitmap('git.ico')
janela.resizable(width=False, height=True)
janela.minsize(width=400, height=250)
janela.maxsize(width=500, height=750)
janela.config(bg='green', cursor='hand2')

janela.mainloop()
```

- para definir o título da janela, usamos o método title;
- para definir o tamanho da janela, usamos o método geometry;
- para definir o ícone da janela, usamos o método iconbitmap;
- para definir se uma janela pode ser redimensionada, usamos o método resizable;
- para definir o tamanho mínimo de uma janela, usamos o método minsize;
- para definir o tamanho máximo de uma janela, usamos o método maxsize;
- para definir algumas outras opções de uma só vez, usamos o método config;

PS : cada objeto do tkinter terá seus próprios argumentos-chave para as configurações.

Widget Button

A classe Button do Tkinter é utilizada para criar botões na interface gráfica. Esta classe é um widget do Tkinter que herda as características básicas da classe Tk e adiciona a funcionalidade de um botão.

Para criar um botão, é preciso criar uma instância da classe Button, passando como argumento o widget pai (geralmente a janela principal) e as opções desejadas. Alguns dos argumentos-chave da classe Button incluem:

1. master : especifica o widget pai do botão. O padrão é o widget raiz (Tk);
2. text : especifica o texto exibido no botão;
3. command : especifica a função ou método chamado quando o botão é pressionado;
4. bg : especifica a cor de fundo do botão;
5. fg : especifica a cor do texto do botão;
6. activebackground : especifica a cor de fundo quando o mouse estiver sobre o botão;
7. activeforeground : especifica a cor do texto quando o mouse estiver sobre o botão;
8. state : especifica o estado do botão, pode ser normal (ativado) ou disabled (desativado);
9. width : especifica a largura do botão;
10. height : especifica a altura do botão;
11. font : especifica a fonte usada para exibir o texto do botão;

Abaixo temos o exemplo de algumas aplicações:

```
import tkinter as tk

janela = tk.Tk()
janela.geometry('600x500')

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

botao = tk.Button(janela, text="Sou Colorido", bg="blue", fg="white",
activebackground="green", activeforeground="orange")
botao.pack()

botao = tk.Button(janela, text="Estou Desativado", state="disabled",
bg="gray")
botao.pack()

botao = tk.Button(janela, text="Botão com Fonte", font=("Arial", 20))
botao.pack()

janela.mainloop()
```

Widget Label

A classe Label do Tkinter é utilizada para criar labels, ou rótulos, na interface gráfica. Esta classe é um widget do Tkinter que herda as características básicas da classe Tk e adiciona a funcionalidade de exibir texto ou imagens.

Para criar um label, é preciso criar uma instância da classe Label, passando como argumento o widget pai (geralmente a janela principal) e as opções desejadas. Alguns dos argumentos-chave da classe Label incluem:

- master : especifica o widget pai do label. O padrão é o widget raiz (Tk);
- text : especifica o texto exibido no label;
- image : especifica a imagem exibida no label;
- bg : especifica a cor de fundo do label;
- fg : especifica a cor do texto do label;
- font : especifica a fonte usada para exibir o texto do label;
- width : especifica a largura do label;
- height : especifica a altura do label;
- anchor : especifica a posição do texto ou imagem dentro do label;

Abaixo temos um exemplo de algumas opções:

```
import tkinter as tk

janela = tk.Tk()
janela.geometry('300x200')

label = tk.Label(janela, text="Texto do Label")
label.pack()

label = tk.Label(janela, text="Texto do Label", bg="green", fg="white", font=
("Arial", 20))
label.pack()

label = tk.Label(janela, text="Label grande", width=50, height=10)
label.pack()

label = tk.Label(janela, text="Label alinhado", anchor="e")
label.pack()

janela.mainloop()
```

Exercícios para Praticar

1. Crie uma janela com um botão e um Label.
2. Posicione o botão à esquerda e o Label à direita na janela.
3. Defina o texto do Label como "Olá, Mundo!".
4. Altere a cor de fundo do botão para vermelho.
5. Defina a fonte do texto do Label como "Arial" e tamanho 12.
6. Altere a largura do botão para 100 pixels.
7. Defina o texto do botão como "Clique Aqui".
8. Mude a cor do texto do Label para azul.
9. Coloque o Label em negrito.
10. Crie uma segunda janela com outro botão e Label.
11. Posicione o segundo botão abaixo do primeiro botão na primeira janela.
12. Defina o texto do segundo Label como "Exemplo".
13. Altere a cor de fundo do segundo botão para verde.
14. Defina a fonte do texto do segundo Label como "Helvetica" e tamanho 14.
15. Altere a largura do segundo botão para 150 pixels.
16. Defina o texto do segundo botão como "OK".
17. Mude a cor do texto do segundo Label para vermelho.
18. Coloque o segundo Label em itálico.
19. Crie uma terceira janela com um botão e um Label.
20. Posicione o terceiro botão à direita do segundo botão na primeira janela.
21. Defina o texto do terceiro Label como "Python".
22. Altere a cor de fundo do terceiro botão para amarelo.
23. Defina a fonte do texto do terceiro Label como "Times New Roman" e tamanho 16.
24. Altere a largura do terceiro botão para 200 pixels.
25. Defina o texto do terceiro botão como "Fechar".
26. Mude a cor do texto do terceiro Label para verde.
27. Coloque o terceiro Label sublinhado.
28. Adicione um quarto botão e um quarto Label à primeira janela.
29. Posicione o quarto botão abaixo do terceiro botão na primeira janela.
30. Defina o texto do quarto Label como "Interface".
31. Altere a cor de fundo do quarto botão para cinza.
32. Defina a fonte do texto do quarto Label como "Courier" e tamanho 10.
33. Altere a largura do quarto botão para 120 pixels.
34. Defina o texto do quarto botão como "Enviar".
35. Mude a cor do texto do quarto Label para laranja.
36. Coloque o quarto Label com efeito de sombra.
37. Adicione um quinto botão e um quinto Label à segunda janela.
38. Posicione o quinto botão à direita do segundo Label na segunda janela.
39. Defina o texto do quinto Label como "Widget".
40. Altere a cor de fundo do quinto botão para roxo.