

## Aula 17

### Messagebox

O módulo messagebox do tkinter é um conjunto de funções que permitem exibir caixas de diálogo de mensagem em uma aplicação tkinter. Ele fornece uma interface para mostrar mensagens de informação, erro, pergunta e aviso para o usuário.

Algumas das funções mais comuns do módulo messagebox incluem:

- showinfo(title, message) : Exibe uma caixa de diálogo de informação com o título especificado e a mensagem fornecida. O ícone da caixa de diálogo é um sinal de exclamação;

```
import tkinter as tk
from tkinter import messagebox

def mostrar_info():
    ... messagebox.showinfo("Informação", "Este é um exemplo de caixa de diálogo de informação.")

janela = tk.Tk()
botao = tk.Button(janela, text="Mostrar informação", command=mostrar_info)
botao.pack()
janela.mainloop()
```

- showerror(title, message) : Exibe uma caixa de diálogo de erro com o título especificado e a mensagem fornecida. O ícone da caixa de diálogo é um "X" vermelho;

```
import tkinter as tk
from tkinter import messagebox

def mostrar_erro():
    ... messagebox.showerror("Erro", "Este é um exemplo de caixa de diálogo de erro.")

janela = tk.Tk()
botao = tk.Button(janela, text="Mostrar erro", command=mostrar_erro)
botao.pack()
janela.mainloop()
```

- showwarning(title, message) : Exibe uma caixa de diálogo de aviso com o título especificado e a mensagem fornecida. O ícone da caixa de diálogo é um sinal de exclamação amarelo;

```
import tkinter as tk
from tkinter import messagebox

def mostrar_aviso():
    ... messagebox.showwarning("Aviso", "Este é um exemplo de caixa de diálogo de aviso.")

janela = tk.Tk()
botao = tk.Button(janela, text="Mostrar aviso", command=mostrar_aviso)
botao.pack()
janela.mainloop()
```

- askquestion(title, message) : Exibe uma caixa de diálogo de pergunta com o título especificado e a mensagem fornecida. O ícone da caixa de diálogo é um sinal de pergunta. Retorna yes ou no dependendo da escolha do usuário;

```
import tkinter as tk
from tkinter import messagebox

def mostrar_pergunta():
    ... resposta = messagebox.askquestion("Pergunta", "Você gosta de tkinter?")
    ... if resposta == 'yes':
    ...     print("Eu gosto de tkinter também!")
    ... else:
    ...     print("Eu entendo, há muitas outras bibliotecas de interface gráfica para escolher.")

janela = tk.Tk()
botao = tk.Button(janela, text="Mostrar pergunta", command=mostrar_pergunta)
botao.pack()
janela.mainloop()
```

- askokcancel(title, message) : Exibe uma caixa de diálogo de pergunta com o título especificado e a mensagem fornecida. O ícone da caixa de diálogo é um sinal de pergunta. Retorna True se o usuário clicar em "OK" e False se o usuário clicar em "Cancelar";

```
import tkinter as tk
from tkinter import messagebox

def mostrar_ok_cancel():
    ... resposta = messagebox.askokcancel("Confirmação", "Deseja sair da aplicação?")
    ... if resposta:
    ...     janela.destroy()
    ... else:
    ...     print("Ok, continuaremos.")

janela = tk.Tk()
botao = tk.Button(janela, text="Mostrar OK/Cancel", command=mostrar_ok_cancel)
botao.pack()
janela.mainloop()
```

- askyesno(title, message) : Exibe uma caixa de diálogo de pergunta com o título especificado e a mensagem fornecida. O ícone da caixa de diálogo é um sinal de pergunta. Retorna True se o usuário clicar em "Sim" e False se o usuário clicar em "Não";

```
import tkinter as tk
from tkinter import messagebox

def mostrar_sim_nao():
    ... resposta = messagebox.askyesno("Pergunta", "Deseja salvar o arquivo antes de sair?")
    ... if resposta:
    ...     print("Ok, salvando arquivo ... ")
    ... else:
    ...     print("Ok, não salvaremos o arquivo.")

janela = tk.Tk()
botao = tk.Button(janela, text="Mostrar Sim/Não", command=mostrar_sim_nao)
botao.pack()
janela.mainloop()
```

- `askretrycancel(title, message)` : Exibe uma caixa de diálogo de pergunta com o título especificado e a mensagem fornecida. O ícone da caixa de diálogo é um sinal de pergunta. Retorna True se o usuário clicar em "Tentar Novamente" e False se o usuário clicar em "Cancelar";

```
import tkinter as tk
from tkinter import messagebox

def mostrar_tentar_novamente():
    ... resposta = messagebox.askretrycancel("Erro", "O arquivo não pôde ser aberto. Deseja tentar novamente?")
    ... if resposta:
    ...     print("Tentando novamente ... ")
    ... else:
    ...     print("Ok, cancelado.")

janela = tk.Tk()
botao = tk.Button(janela, text="Mostrar Tentar Novamente/Cancelar", command=mostrar_tentar_novamente)
botao.pack()
janela.mainloop()
```

- `askyesnocancel(title, message)` : Exibe uma caixa de diálogo de pergunta com o título especificado e a mensagem fornecida. O ícone da caixa de diálogo é um sinal de pergunta. Retorna yes se o usuário clicar em "Sim", no se o usuário clicar em "Não" e None se o usuário clicar em "Cancelar";

```
import tkinter as tk
from tkinter import messagebox

def mostrar_sim_nao_cancelar():
    ... resposta = messagebox.askyesnocancel("Pergunta", "Deseja salvar o arquivo antes de sair?")
    ... if resposta == None:
    ...     print("Ok, cancelando.")
    ... elif resposta:
    ...     print("Ok, salvando arquivo ... ")
    ... else:
    ...     print("Ok, não salvaremos o arquivo.")

janela = tk.Tk()
botao = tk.Button(janela, text="Mostrar Sim/Não/Cancelar", command=mostrar_sim_nao_cancelar)
botao.pack()
janela.mainloop()
```

## Entry – part Deux

O widget Entry possui um método para limpar o campo de todo conteúdo nele.

```
from tkinter import Tk, Button, Entry, Label, END

def limpa_entry():
    """Limpa o conteúdo do Entry"""
    ent_nome.delete(0, END)

janela = Tk()
lbl_texto = Label(janela, text="Nome:")
btn_limpa = Button(janela, width=20, text="Limpar")
ent_nome = Entry(janela, width=20)

btn_limpa["command"] = limpa_entry

ent_nome.focus()
ent_nome['justify'] = 'right'

lbl_texto.pack()
ent_nome.pack()
btn_limpa.pack()

janela.mainloop()
```

No exemplo acima, temos um campo de entrada onde que sempre que o botão for clicado, todo o conteúdo será apagado.

O método delete recebe como primeiro valor a partir de qual caractere deve ser apagado e o segundo determina até onde ele deve ser apagado. Se deixarmos apenas um valor, apenas o caractere naquela posição que será apagado.

Já o método focus é usado para que o foco da janela vá para aquele widget ao ser iniciado.

O atributo justify é usado para orientar o texto dentro do widget. Ele pode ser center, left ou right

## Método destroy()

O método destroy está presente em quase todos os widgets do tkinter. Ele serve para removermos algum widget do nosso programa durante a execução do mesmo.

```
from tkinter import Tk, Button

janela = Tk()

btn_1 = Button(janela, text='Apagar', width=20)
btn_2 = Button(janela, text='Sair', width=20)

btn_1['command'] = btn_2.destroy
btn_2['command'] = janela.destroy

btn_1.pack()
btn_2.pack()

janela.mainloop()
```

No código acima, o botão 1 está com o comando para apagar o botão 2 e o botão 2 está com o comando de destruir a janela, isto é, encerrando o programa de forma abrupta.

O widget TK (instanciado na variável janela) tem um método mais elegante para realizar esse encerramento do programa, é o método quit.

```
from tkinter import Tk, Button

janela = Tk()

btn_1 = Button(janela, text='Apagar', width=20)
btn_2 = Button(janela, text='Sair', width=20)

btn_1['command'] = btn_2.destroy
btn_2['command'] = janela.quit

btn_1.pack()
btn_2.pack()

janela.mainloop()
```

A diferença é que o quit realiza uma interrupção no método mainloop, enquanto o destroy pode ser aplicado a outros widgets.

## Variáveis locais e globais

Em Python, as variáveis locais e globais são formas de armazenar e acessar valores.

As variáveis locais são definidas dentro de uma função e só podem ser acessadas dentro desse escopo. Elas são úteis para armazenar valores temporários ou intermediários usados apenas dentro da função. Uma vez que a função termina sua execução, as variáveis locais são destruídas e o seu valor é perdido. Isso significa que as variáveis locais são criadas e destruídas sempre que a função é chamada e retorna.

Por outro lado, as variáveis globais são definidas fora de qualquer função e podem ser acessadas de qualquer lugar do programa. Elas são úteis quando você precisa compartilhar valores entre diferentes partes do código. As variáveis globais são criadas quando o programa começa a ser executado e permanecem na memória durante toda a execução do programa. No entanto, é importante tomar cuidado ao modificar variáveis globais dentro de funções, pois isso pode levar a comportamentos inesperados e dificultar a depuração do código.

Veja alguns exemplos:

```
def minha_funcao():  
    nome = "João"  
    print("Olá, ", nome)  
  
minha_funcao()  
print(nome)
```

No exemplo acima, a variável `nome` não existe fora da função, então ela não possui qualquer valor. A execução irá gerar um `NameError`.

```
numero = 10  
  
def dobrar_numero():  
    global numero  
    numero = numero * 2  
  
print(numero)  
dobrar_numero()  
print(numero)
```

Já o código acima, usa o termo `global` para referenciar a variável do lado de fora da função também dentro dela.

## Exercício para Entregar

### Calculadora Tkinter

Crie uma calculadora das operações de soma, subtração, multiplicação, divisão e exponenciação. O usuário deverá digitar toda a operação no campo de Entry.

Por exemplo: o usuário irá digitar 4+4 usando os botões da calculadora e, ao clicar no igual, a expressão será apagada e substituída pela resposta no mesmo campo Entry, bem como funciona uma calculadora.

Não use a função eval do Python. Use apenas o que foi ensinado até a aula do dia 02/06.

### Exercícios para Praticar

1. Escreva uma função que receba um número como parâmetro e retorne o seu quadrado.
2. Crie uma variável global chamada contador e atribua o valor 0 a ela. Escreva uma função que incremente o valor dessa variável em 1 a cada chamada.
3. Escreva uma função que receba uma lista como parâmetro e retorne o maior valor contido nela.
4. Declare uma variável local dentro de uma função e tente acessá-la fora dessa função. O que acontece?
5. Escreva uma função que receba dois números como parâmetros e retorne a soma e a multiplicação desses números.
6. Crie uma variável global chamada pi e atribua o valor 3.1415 a ela. Escreva uma função que calcule a área de um círculo usando essa variável.
7. Escreva uma função que receba uma string como parâmetro e retorne a quantidade de caracteres dessa string.
8. Declare uma variável local e uma variável global com o mesmo nome dentro de uma função. O que acontece quando você tenta acessar essa variável?
9. Escreva uma função que receba uma lista como parâmetro e retorne uma nova lista contendo apenas os números pares da lista original.
10. Crie uma variável global chamada saudacao e atribua a ela a string "Olá, mundo!". Escreva uma função que imprima essa saudação.
11. Escreva uma função que receba um número como parâmetro e retorne o fatorial desse número.
12. Declare uma variável local dentro de uma função e tente acessá-la dentro de outra função. O que acontece?
13. Escreva uma função que receba uma lista como parâmetro e retorne o número de elementos únicos nessa lista.
14. Crie uma variável global chamada mensagem e atribua a ela a string "Bem-vindo!". Escreva uma função que altere o valor dessa variável para "Olá!".
15. Escreva uma função que receba duas strings como parâmetros e retorne a concatenação dessas strings.
16. Crie uma janela com um botão que, quando clicado, exiba uma mensagem simples usando o messagebox.
17. Implemente um messagebox que pergunte ao usuário se ele deseja sair do programa. Se a resposta for positiva, feche a janela; caso contrário, exiba uma mensagem de agradecimento.
18. Crie um messagebox que solicite ao usuário que digite seu nome. Em seguida, exiba uma mensagem de boas-vindas com o nome fornecido.
19. Implemente um messagebox que mostre uma mensagem de erro se o usuário tentar digitar um número no campo destinado a aceitar apenas strings.
20. Crie um messagebox que exiba uma mensagem informativa sobre a funcionalidade de um botão específico.
21. Implemente um messagebox que pergunte ao usuário se ele deseja salvar as alterações antes de fechar o programa. Caso a resposta seja negativa, feche o programa; caso contrário, exiba uma mensagem de confirmação.
22. Crie um messagebox que solicite ao usuário que insira sua idade. Em seguida, exiba uma mensagem informando se ele for maior ou menor de idade.
23. Implemente um messagebox que pergunte ao usuário se ele deseja excluir um determinado item. Dependendo da resposta, exiba uma mensagem de confirmação ou cancelamento.
24. Crie um messagebox que exiba uma mensagem de aviso se o usuário tentar inserir um valor inválido em um campo de entrada.
25. Implemente um messagebox que pergunte ao usuário se ele deseja desfazer uma ação. Dependendo da resposta, exiba uma mensagem de confirmação ou cancelamento.

26. Implemente um messagebox que exiba uma mensagem de conclusão após a execução bem-sucedida de uma tarefa.
27. Crie um messagebox que solicite ao usuário que digite um número. Em seguida, exiba uma mensagem informando se o número é par ou ímpar.
28. Implemente um messagebox que pergunte ao usuário se ele deseja substituir um arquivo existente. Dependendo da resposta, exiba uma mensagem de confirmação ou cancelamento.
29. Implemente um messagebox que solicite ao usuário que digite seu endereço de e-mail. Em seguida, exiba uma mensagem informando se o endereço é válido ou não.
30. Crie um messagebox que exiba uma mensagem de parabéns quando o usuário atingir uma determinada pontuação em um jogo.
31. Implemente um messagebox que pergunte ao usuário se ele deseja limpar todos os dados do aplicativo. Dependendo da resposta, exiba uma mensagem de confirmação ou cancelamento.
32. Implemente um messagebox que pergunte ao usuário se ele deseja imprimir um documento. Dependendo da resposta, exiba uma mensagem de confirmação ou cancelamento.
33. Crie um messagebox que solicite ao usuário que insira um número de telefone. Em seguida, exiba uma mensagem informando se o número é válido ou não.
34. Implemente um messagebox que pergunte ao usuário se ele deseja abrir um arquivo. Dependendo da resposta, exiba uma mensagem de confirmação ou cancelamento.
35. Crie um messagebox que exiba uma mensagem de lembrete para o usuário em um horário especificado.
36. Implemente um messagebox que pergunte ao usuário se ele deseja copiar um determinado arquivo. Dependendo da resposta, exiba uma mensagem de confirmação ou cancelamento.
37. Crie um messagebox que solicite ao usuário que digite um número inteiro. Em seguida, exiba uma mensagem informando se o número é positivo, negativo ou zero.
38. Implemente um messagebox que pergunte ao usuário se ele deseja adicionar um novo contato à lista de contatos. Dependendo da resposta, exiba uma mensagem de confirmação ou cancelamento.
39. Crie um messagebox que exiba uma mensagem de erro se o usuário tentar fazer login com credenciais inválidas.
40. Implemente um messagebox que pergunte ao usuário se ele deseja desativar uma determinada configuração. Dependendo da resposta, exiba uma mensagem de confirmação ou cancelamento.
41. Crie um messagebox que exiba uma mensagem de boas festas em uma data específica.
42. Implemente um messagebox que pergunte ao usuário se ele deseja salvar um documento antes de fechar. Dependendo da resposta, exiba uma mensagem de confirmação ou cancelamento.
43. Crie uma janela com um Entry e um Button. Ao clicar no Button, exiba em um Label o texto inserido no Entry.
44. Implemente uma função que, ao pressionar um botão, limpe o conteúdo de um Entry.
45. Crie um Entry e um Button que, ao serem clicados, mostrem uma mensagem de boas-vindas personalizada no Label.
46. Implemente uma função que defina o foco no Entry assim que a janela for aberta.
47. Crie um Entry e um Button que, ao ser clicado, exiba em um Label o número de caracteres presentes no Entry.
48. Crie dois Entries e um Button. Ao clicar no Button, concatene o texto dos dois Entries e exiba o resultado em um Label.
49. Crie um Entry e um Button. Ao pressionar o Button, exiba em um Label se o texto inserido no Entry é um número ou não.
50. Crie uma janela com um Entry e um Button. Ao clicar no Button, exiba em um Label o texto inserido no Entry.