

## 7. Benutzeroberflächen II (Menüs, Dialogfelder)

In diesem Kapitel gehen wir auf weitere Elemente zur **Benutzeroberflächen** (GUI = Graphical User Interface) ein.

Wie wird ein Textfeld für die Eingabe verwendet und das Ergebnis der Verarbeitung als Ausgabe im Hauptfenster angezeigt (EVA-Prinzip). Schauen wir uns das nachfolgende Beispielprogramm an:

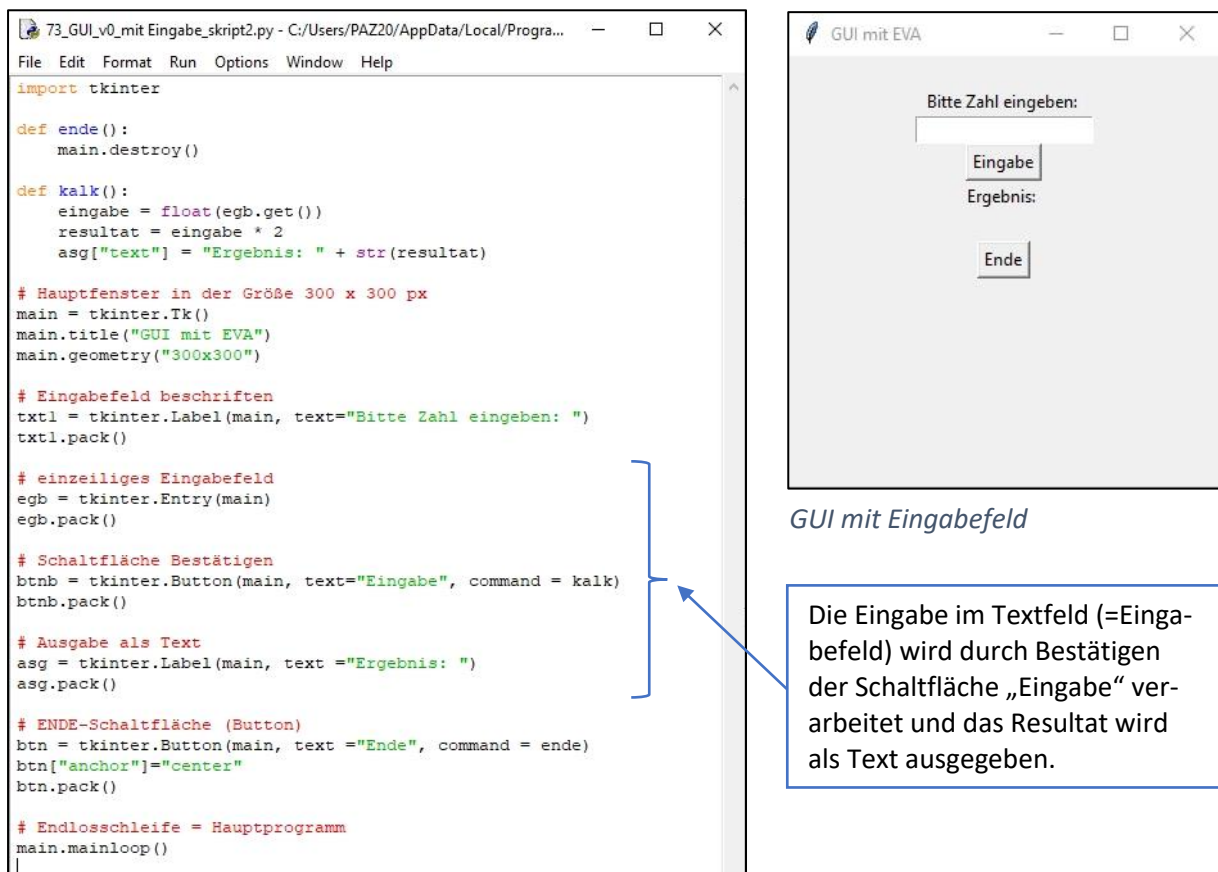


Abb. 29: GUI mit EINGABE-VERARBEITUNG-AUSGABE

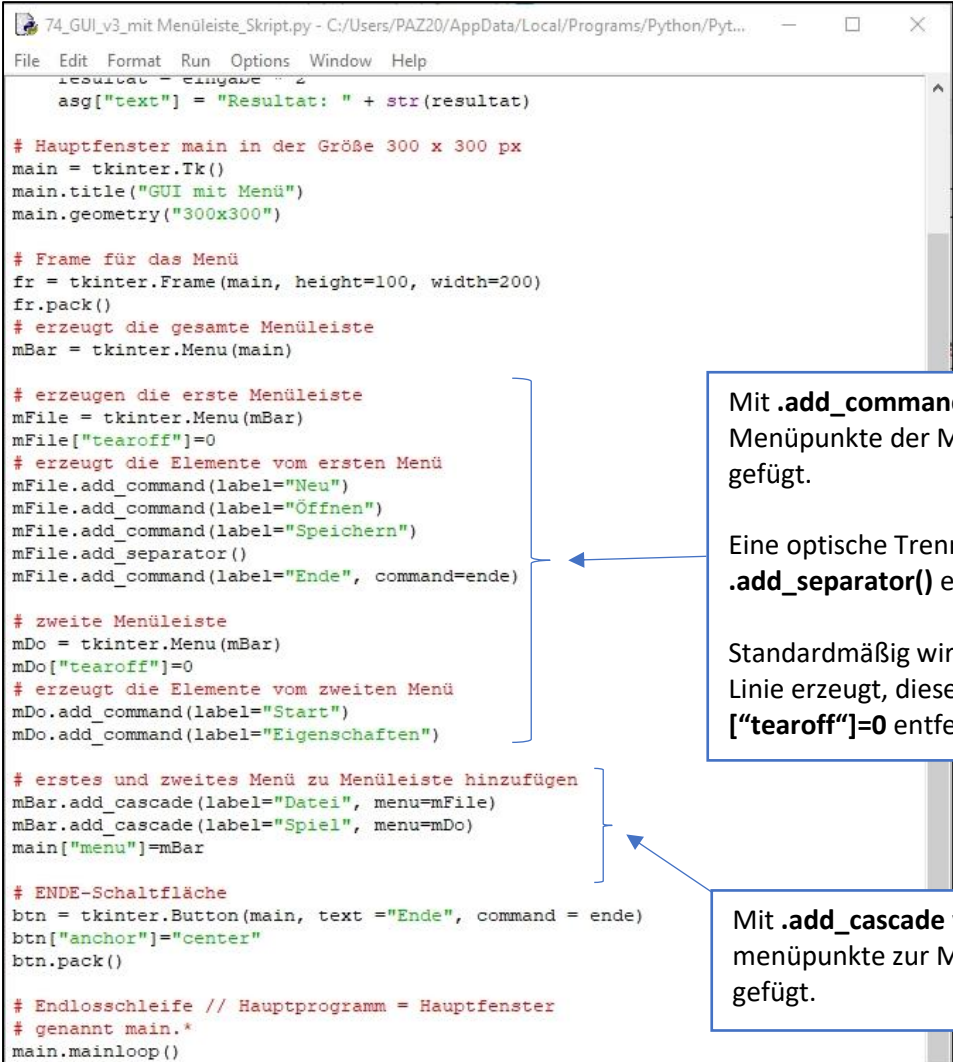
Die Verarbeitung der Eingabe beginnt erst mit dem Klick auf der Schaltfläche „Eingabe“ und wird an die Funktion kalk() übergeben. Das Resultat wird in der Funktion kalk() dem Ausgabertext zugewiesen (Änderung der Beschriftung).

### Menüs

Dem Hauptfenster kann eine Menüleiste ergänzt werden, die permanent am oberen Rand sichtbar ist. Dazu wird zuerst das Objekt der Klasse Menu als Menüleiste erzeugt.

tkinter.Menu	Menüleiste erstellen (Menüleiste)
.add_cascade	Menühauptpunkte erstellen
.add_command	Elemente des Menüobjekts erstellen (Menüpunkte)
.add_separator	Optische Trennlinie erzeugen zwischen den Menüpunkten

## Python - Einstieg in die Programmierung



```
resultat = eingabe * 2
asg["text"] = "Resultat: " + str(resultat)

# Hauptfenster main in der Größe 300 x 300 px
main = tkinter.Tk()
main.title("GUI mit Menü")
main.geometry("300x300")

# Frame für das Menü
fr = tkinter.Frame(main, height=100, width=200)
fr.pack()
# erzeugt die gesamte Menüleiste
mBar = tkinter.Menu(main)

# erzeugen die erste Menüleiste
mFile = tkinter.Menu(mBar)
mFile["tearoff"]=0
# erzeugt die Elemente vom ersten Menü
mFile.add_command(label="Neu")
mFile.add_command(label="Öffnen")
mFile.add_command(label="Speichern")
mFile.add_separator()
mFile.add_command(label="Ende", command=ende)

# zweite Menüleiste
mDo = tkinter.Menu(mBar)
mDo["tearoff"]=0
# erzeugt die Elemente vom zweiten Menü
mDo.add_command(label="Start")
mDo.add_command(label="Eigenschaften")

# erstes und zweites Menü zu Menüleiste hinzufügen
mBar.add_cascade(label="Datei", menu=mFile)
mBar.add_cascade(label="Spiel", menu=mDo)
main["menu"]=mBar

# ENDE-Schaltfläche
btn = tkinter.Button(main, text ="Ende", command = ende)
btn["anchor"]="center"
btn.pack()

# Endlosschleife // Hauptprogramm = Hauptfenster
# genannt main.*
main.mainloop()
```

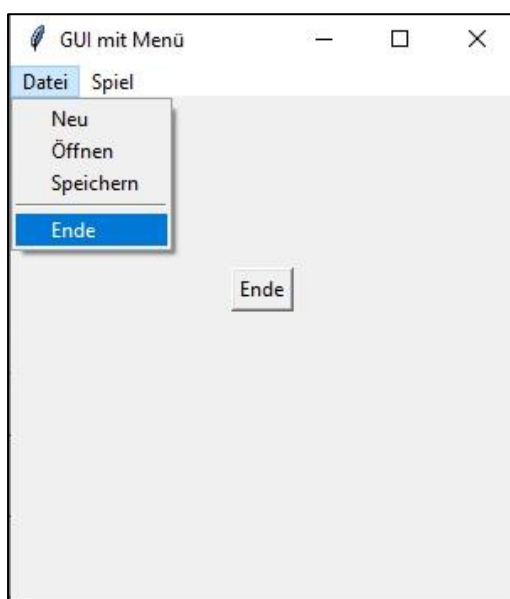
Mit `.add_command()` werden die Menüpunkte der Menüleiste hinzugefügt.

Eine optische Trennlinie kann mit `.add_separator()` erzeugt werden.

Standardmäßig wird eine gestrichelte Linie erzeugt, diese kann mit `["tearoff"]=0` entfernt werden.

Mit `.add_cascade` werden die Hauptmenüpunkte zur Menüleiste hinzugefügt.

Abb. 30: GUI mit Menüleiste



GUI mit Menüleiste

## Dialogfelder (engl. Messageboxen)

Dialogfelder sind vorgefertigte Dialogfenster, die Informationen auf einfache Art und Weise zwischen Benutzer und Programm angezeigt bzw. abgefragt werden können. Die Dialogfenster unterbrechen die Ausführung des Hauptfensters und warten solange bis die Antwort vom Benutzer über das Dialogfenster abgeschlossen ist.

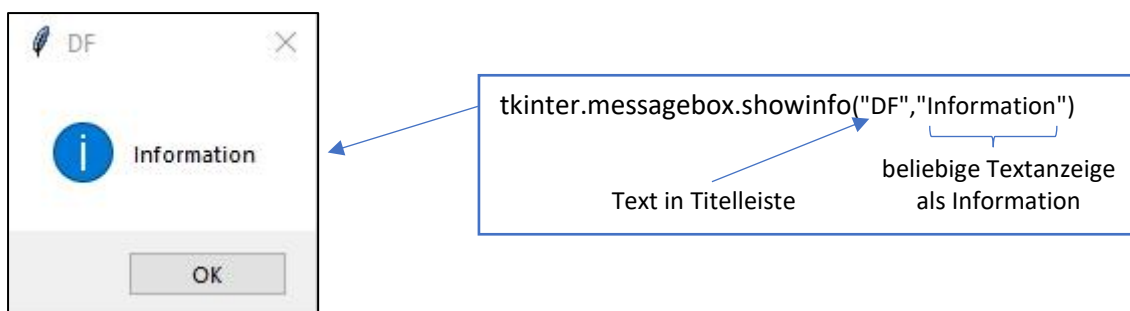
tkinter.messagebox	Anzeige von Dialogfeldern
--------------------	---------------------------

Die Funktion des Moduls **tkinter.messagebox.fktname()** ist mit verschiedenen bekannten Dialogfenstern ausführbar:

z.B. Dialogfenster mit einer Information und OK-Schaltfläche

1	import tkinter	← import des Moduls tkinter
2	import tkinter.messagebox	← import Modul tkinter.messagebox
3	def ende():	
4	main.destroy()	
5	def msginfo():	
6	tkinter.messagebox.showinfo("DF","Information")	← Dialogfenster mit OK-Button
7	main = tkinter.Tk()	Hauptfenster
8	binfo = tkinter.Button(main, text="Info", command=msginfo)	Schaltfläche Info
9	binfo.pack()	
10	b = tkinter.Button(main, text = "Ende", command = ende)	Schaltfläche Ende
11	b.pack()	
12	main.mainloop()	Endlosschleife

Aus dem Hauptfenster wird per Info-Schaltfläche das Dialogfenster mit einer Information aufgerufen. Das Dialogfenster wird mit dem **i-Symbol** und der **Information** angezeigt.



Das „i“-Symbol wird automatisch durch die Funktion **showinfo()** in dem Dialogfenster angezeigt. Es lassen sich beliebige Texte als Information anzeigen. Sobald der Benutzer auf die OK-Schaltfläche klickt, kann er weitere Aktionen im Hauptfenster fortsetzen.

Das Dialogfenster kann für Warnungen mit dem Ausrufezeichen „!“-Symbol und ebenso kann es als Fehlermeldung mit dem „✖“-Symbol angezeigt werden. Auf der nächsten Seite sind die zugehörigen Funktionsnamen aufgelistet.

→ Dialogfelder mit OK-Schalfläche

Funktionsname	Schalfläche	Funktion
showinfo()	OK	Benutzer <b>informieren</b> mit Bestätigung.
showwarning()	OK	Benutzer <b>warnen</b> mit Bestätigung.
showerror()	OK	Benutzer <b>Fehler melden</b> mit Bestätigung.



Dialogbox für Informationen

Dialogbox für Warnungen

Dialogbox für Fehlermeldungen

Für einfache Entscheidungsfragen (Ja/Nein, OK/Abbrechen und Wiederholen/Abbrechen) gibt es die spezielleren Dialogfelder.

z.B. Dialogfenster mit Ja/Nein-Schalflächen

```

import tkinter
import tkinter.messagebox

def ende():
    main.destroy()

def msgyesno():
    antwort = tkinter.messagebox.askyesno("Ja/Nein-DF", "Sind Sie Nichtraucher?")
    if antwort == 1:
        lbant["text"]="Sie leben gesund."
    else:
        lbant["text"]="Hören Sie auf zu rauchen! Es ist ungesund."

# Hauptfenster
main = tkinter.Tk()
main.title("Hauptfenster")
main.geometry("300x200")

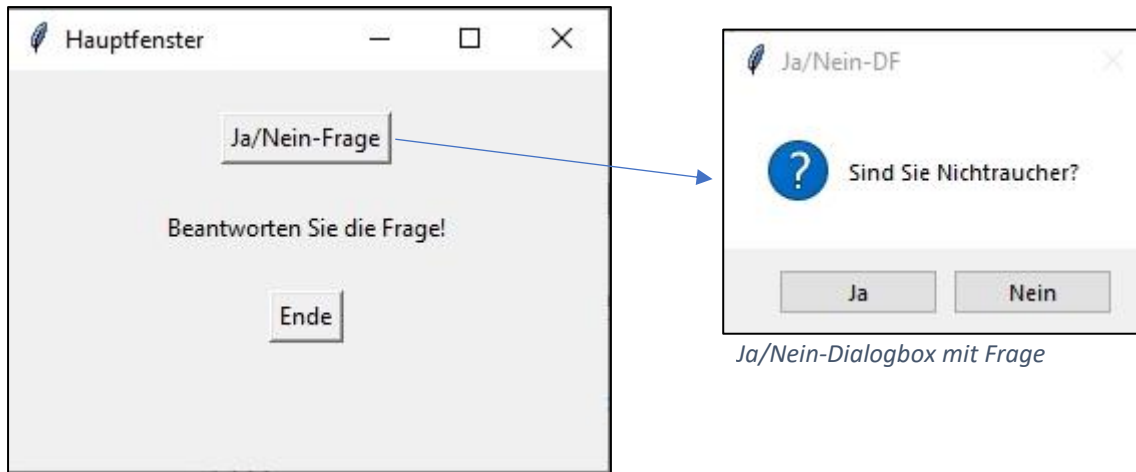
byesno = tkinter.Button(main, text="Ja/Nein-Frage", command=msgyesno)
byesno.pack()

# Meldung im Hauptfenster
lbant = tkinter.Label(main, text="Beantworten Sie die Frage!")
lbant.pack()

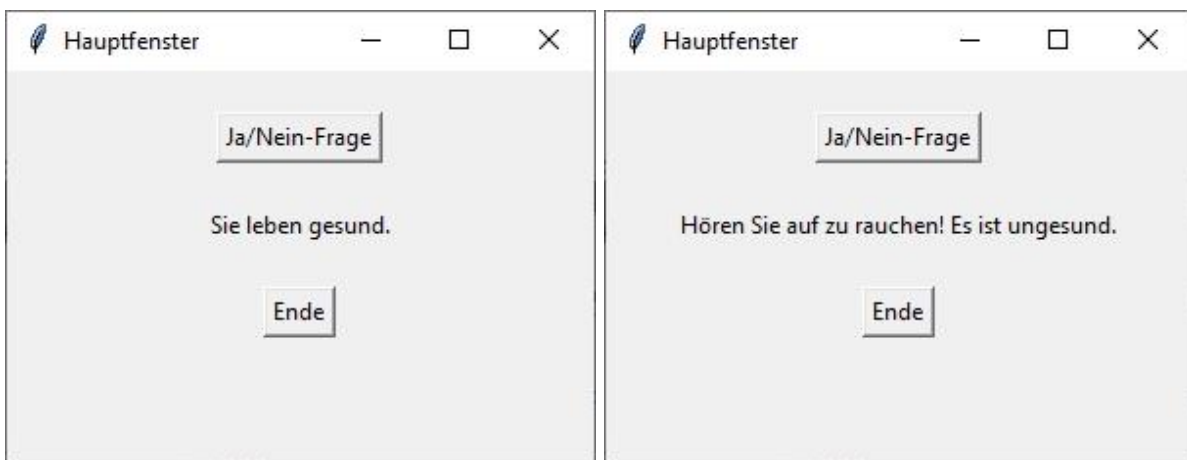
b = tkinter.Button(main, text = "Ende", command = ende)
b.pack()

main.mainloop()
    
```

Abb. 29: GUI mit der Ja/Nein-Frage „Sind Sie Nichtraucher?“



GUI-Hauptfenster mit Ja/Nein-Dialogbox Aufruf



GUI-Hauptfenster = **Ja** → „Sie leben gesund.“

GUI-Hauptfenster = **Nein** → „Hören Sie auf zu rauchen! (...)“

→ Dialogfelder mit zwei Antworten

Funktionsname	Schaltflächen	Funktion
askyesno()	Ja, Nein	Benutzer mit <b>Ja/Nein</b> -Antwort.
askokcancel()	OK, Abbrechen	Benutzer mit <b>OK/Abbrechen</b> -Antwort.
askretrycancel()	Wiederholen, Abbrechen	Benutzer mit <b>Wiederholen/Abbrechen</b> -Antwort.



Dialogbox für OK/Abbrechen

Dialogbox Wiederholen/Abbrechen