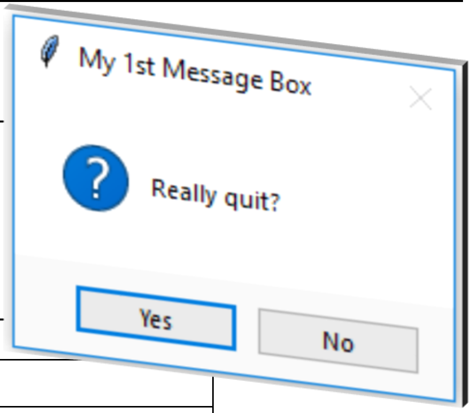
PYTHON GUI: Message Boxes

Computer Science Software Engineering



My 1st Message Box

Really quit?

Learning Objectives

Students will be able to demonstrate mastery of …

1. Python GUI Interface Commands
2. Create Message Boxes with Custom Messages

Procedure

1. Review basic tkinter commands for use in your program.

Basic Tkinter Commands	
Label(#text#)	creates a label
Message(#text#)	creates a message
import messagebox	imports message box features
Button(#text#)	creates a button
.grid(#parameters#)	configures layout management geometry
.config(#parameters#)	configures the widget
.mainloop()	loops the program in tkinter
bg=	sets the background of widget
font=	changes the font properties
fg=	sets the foreground of widget
padx=	provides horizontal space in widget
pady=	provides vertical space in widget
columnspan=	determines the number of columns for the widget
row=	determines vertical placement of widget
column=	determines horizontal placement of widget
text=	provides text within the widget

2. You can now use this basic code to create a rudimentary message using message boxes in a Python GUI.

```
import tkinter as tk
from tkinter import messagebox as mb

root=tk.Tk()

def answer():
    mb.showerror("Answer", "Sorry, no answer available")

def callback():
    if mb.askyesno('My 1st Message Box', 'Really quit?'):
        mb.showwarning('Warning', 'Program Terminated')
        root.quit()
    else:
        mb.showinfo('No', 'Quit has been cancelled')

tk.Button(text='Quit', command=callback).pack(fill=tk.X)
tk.Button(text='Answer', command=answer).pack(fill=tk.X)
root.mainloop()
```

Assignment

Modify the code to practice creating your own message boxes. You must meet the following standards:

1. Creates custom messages
2. Uses custom message type

To exceed the standard (add one or more of the following):

1. Display multiple messages on the screen simultaneously