Chapter 20 — Using Tkinter to Make Screens

Introduction

The `tkinter` module is the most commonly used graphics library in Python. The turtle and the Zelle graphics library are extensions of tkinter. The tkinter module is based upon the Unix graphics took kit known as "tk". The "tk" was first released in 1991 and has grown to become a cross platform framework for creating Graphical User Interfaces (GUI).

Objectives

Upon completion of this chapter's exercises, you should be able to:

- Define the term Graphical User Interface.
- Describe the concept of the mainloop.
- Create a simple GUI with labels, text entry fields, and buttons.
- Construct functions to handle button click events and to process form data.
- Design forms with the layout methods of pack and grid.
- Display common dialogs.

Prerequisites

The material in this chapter only depends on material from Chapters 1-8.

A Form with Only a Button

`tkinter`	module

`import tkinter`	statement
Tell your Python program that it will be using the `tkinter` module. https://docs.python.org/3/library/tkinter.html#module-tkinter	

`master = tkinter.Tk()`	Class of tkinter
Create a new root window, also known as a master, object of the Tk type. This object will contain all the graphics and fields of your "tk" application.	

Widgets are …

With `tkinter` widgets, you may set their options in one of three ways: 1) defining them using their names when you initially create the object; 2) using the configure method of the widget; or 3) setting the bracketed name of the option. In this chapter you will see the first and third method for configuring widgets.

`tkinter.Button(master)` `tkinter.Button(master, option=value...)`	Widget class in tkinter
Create a new object of the Button type. A button is a clickable object that will execute a command when it is activated. When creating a widget, you must include at least one argument, the controlling window or frame. There are many options that may be set for a Button, you can find them in the documentation page. The most common are: `text` set the text to display `command` definition or method to execute `bg` or `background` set background color `fg` or `foreground` set the text color `bd` or `borderwidth` set the border in pixels https://infohost.nmt.edu/tcc/help/pubs/tkinter/web/button.html	

ADD

pack form layout

master.update()

not explicitly needed in IDLE development but when your program is running as a standalone app you need it to display window
https://docs.python.org/3/library/idle.html#developing-tkinter-applications

master.mainloop()

master.quit
use the name of this method as a command on a button to quit the application.

master.destroy()

```
 1| import tkinter
 2|
 3| app = tkinter.Tk()
 4|
 5| btnQuit = tkinter.Button(app, text='Quit', command=app.quit)
 6| btnQuit.pack()
 7|
 8| app.update()
 9| app.mainloop()
10| app.destroy()
```


Custom Command Definitions


```
 1| import tkinter
 2|
 3| def printAbout():
 4|     print("This program is really cool.")
 5|
 6| def printHello():
 7|     print("I want to say 'hello'.")
 8|
 9| app = tkinter.Tk()
10|
11| btnHello = tkinter.Button(app, text='Say Hello',
     command=printHello)
```

```
12| btnHello.pack()
13|
14| btnQuit = tkinter.Button(app, text='About This Program',
      command=printAbout)
15| btnQuit.pack()
16|
17| btnQuit = tkinter.Button(app, text='Quit', command=app.quit)
18| btnQuit.pack()
19|
20| app.update()
21| app.mainloop()
22| app.destroy()
```

A Form with Fields

tkinter.Label(*master*) **tkinter.Label(*master, option=value...*)**	Widget class in tkinter
Create a new object of the Label type. A label is a single line text message on a screen that typically tells the user what is happening. There are many options that may be set for a Label, you can find them in the documentation page. The most common are: `text` set the text to display `bg` or `background` set background color `fg` or `foreground` set the text color `bd` or `borderwidth` set the border in pixels **https://infohost.nmt.edu/tcc/help/pubs/tkinter/web/label.html**	

tkinter.Entry(master) ***tkinter.Entry(master, option=value...)***	Widget class in tkinter
Create a new object of the Entry type. An entry widget allows the user to enter a single line of text.	

The most common options are:

`width`	set the width in characters
`bg` or `background`	set background color
`fg` or `foreground`	set the text color
`bd` or `borderwidth`	set the border in pixels

Some of Entry's methods:

get()	get the text value
.delete(first, last)	delete a range of characters [[.delete(0, tkinter.END) clears field]]
.insert(index, string)	insert string at given index

https://infohost.nmt.edu/tcc/help/pubs/tkinter/web/label.html

ADD GRID form layout


```
 1| import tkinter
 2|
 3| def doAdd():
 4|     ## event handler
 5|     try:
 6|         n1 = float(txtN1.get())
 7|         n2 = float(txtN2.get())
 8|         a = n1 + n2
 9|         print(a)
10|         lblAnsNumber["text"] = str(a)
11|         lblAnsNumber['fg'] = 'black'
12|     except:
13|         lblAnsNumber["text"] = "Error."
14|         lblAnsNumber['fg'] = 'red'
15|
16| app = tkinter.Tk()
17|
18| # add fields to master
19| lblN1 = tkinter.Label(app, text="Number1:")
20| lblN1.grid(row=0, column=0)
21| txtN1 = tkinter.Entry(app)
22| txtN1.grid(row=0, column=1)
```

```
23|
24| lblN2 = tkinter.Label(app)
25| lblN2['text'] = "Number2:"
26| lblN2.grid(row=1, column=0)
27| txtN2 = tkinter.Entry(app)
28| txtN2.grid(row=1, column=1)
29|
30| btnAdd = tkinter.Button(app, text="+")
31| btnAdd.grid(row=2, column=1)
32| btnAdd['command'] = doAdd
33|
34| lblAns = tkinter.Label(app, text="Answer:")
35| lblAns.grid(row=3, column=0)
36| lblAnsNumber = tkinter.Label(app)
37| lblAnsNumber.grid(row=3, column=1)
38|
39| app.update()
40| app.mainloop()
41| app.destroy()
```


Message Boxes

`tkinter.messagebox`	Module
The messagebox module, within tkinter, contains a group of dialogs to display messages, warnings, and errors. The messagebox module also includes dialogs to ask simple questions.	

`import tkinter.messagebox`	Statement
Tell your Python program that it will be using the `tkinter` message box module. https://docs.python.org/3/library/tkinter.html#module-tkinter	

tkinter.messagebox.showinfo(title, message)

tkinter.messagebox.showwarning(title, message)
tkinter.messagebox.showerror(title, message)

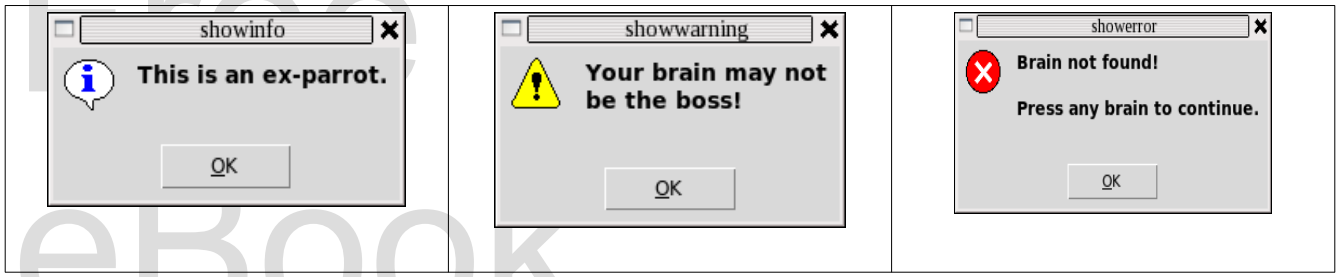


Table 11: Simple OK Message Boxes - Images from https://infohost.nmt.edu/tcc/help/pubs/tkinter/web/tkMessageBox.html

tkinter.messagebox.askokcancel(title, message)
tkinter.messagebox.askretrycancel(title, message)
tkinter.messagebox.askyesno(title, message)

returns True for OK, retry, or Yes
otherwise False

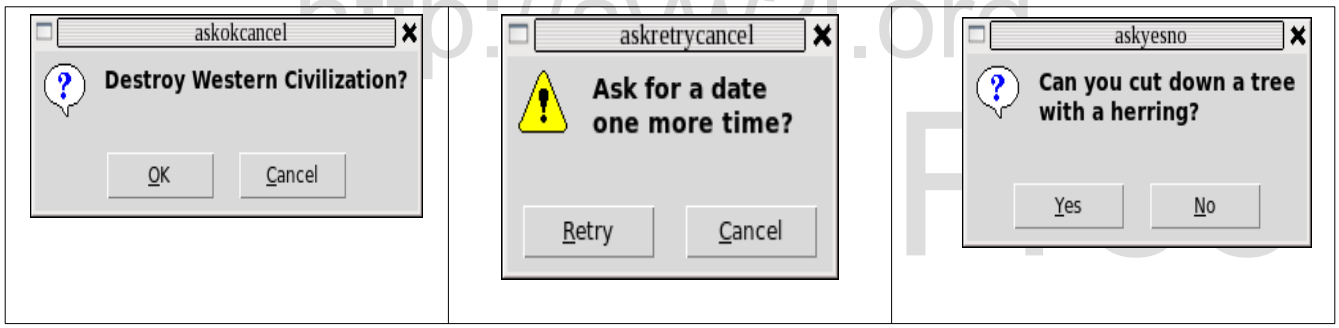
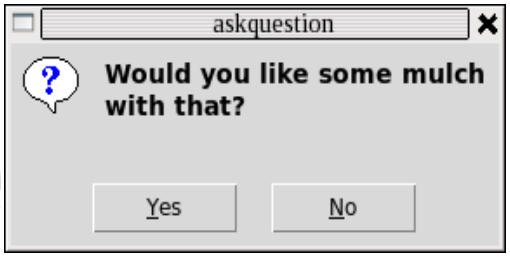


Table 12: True False Message Boxes - Images from https://infohost.nmt.edu/tcc/help/pubs/tkinter/web/tkMessageBox.html

tkinter.messagebox.askquestion(title, message)

returns 'yes' or 'no'



Summary

Goes here

Important Terms

here

Exercises

Here

Word Search

References

https://en.wikipedia.org/wiki/Tk_(software)

https://infohost.nmt.edu/tcc/help/pubs/tkinter/web/tkMessageBox.html