Python, Day 9.5: Tkinter continued

Andrew Bydlon

January 19, 2019

Toplevel()

Often times 1 window isn't enough for a programs desired purpose. For this, we can use Toplevel() widget:

Syntax (Toplevel)

MyExtraWindow = Toplevel()

Much like a frame, you can then insert other objects in this toplevel window, such as Frames, Labels, Buttons, and really anything that could be put in master.

Entry

Next up we have the tk-analogue of **input()**: Entry(). As you are hopefully convinced, one of the most important aspects of coding is the ability to reuse code on different user input.

Syntax

self.MyEntry = Entry(master, option, ...)

Somewhat more importantly:

Example (Username acceptance)

```
MyString = ""
self.L1 = Label(self.master, text="User Name")
self.L1.pack(side=LEFT)
self.E1 = Entry(master, bd=5,) # bd determines border size
self.E1.pack(side=RIGHT)
```

Getting the user info

Right now, this allows a user to interact with the program via keyboard. However, none of this data is shown/saved! For this, we can use the .get() function.

Example

```
def ShowEntry(self):
    MyNewString = self.E1.get()
    print(MyNewString)
    return MyNewString
MyNewButton = Button(master, text='Save', command=ShowEntry)
MyNewButton.pack()
```

Message

In addition to the traditional labels, sometimes it is preferable to list many lines of text to the user.The message widget performs this task.

Example (My Message)

self.MyElaborateString = "When I encode a message for someone, I use AES 512. I'm worried that if quantum computers reach 128 cubits all of my information will be abducted"

self.Secret = Message(self.top, text=self.MyElaborateString, relief=RAISED)
self.Secret.config(bg='lightgreen', font=('times', 12, 'italic'))
self.Secret.pack()

Radio Buttons

Very similar to the menu button command, we can add RadioButton widgets to a tkinter driven program. They differ from the standard Menu Buttons in that only one of the choices can be selected.

Syntax

MyRadio = Radiobutton (master, option, ...)

Example

```
self.MyVariable = IntVar()
self.R1 = Radiobutton(self.top, text="Option 1",
variable=self.MyVariable, value=1, command=self.selection)
self.R1.pack(anchor=W)
self.R2 = Radiobutton(self.top, text="Option 2",
variable=self.MyVariable, value=2, command=self.selection)
self.R2.pack(anchor=W)
⋮
```

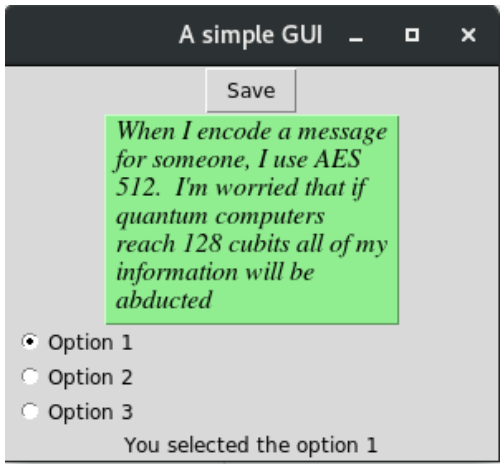
Radio Buttons Continued

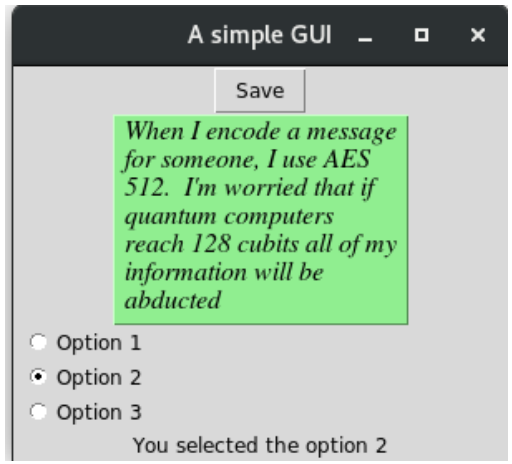
Example

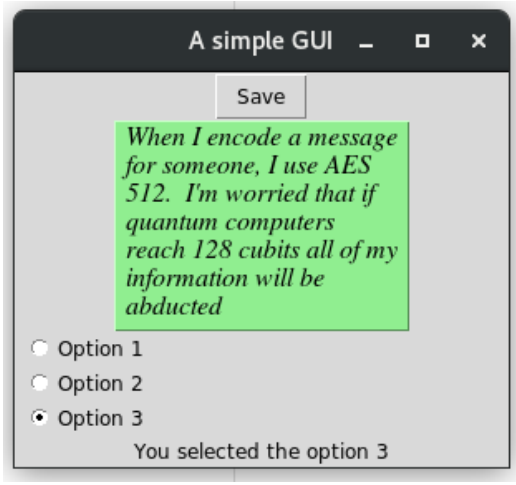
```
self.R3 = Radiobutton(self.top, text="Option 3",
variable=self.MyVariable, value=3, command=self.selection)
self.R3.pack(anchor=W)
self.MyLastLabel = Label(self.top)
self.MyLastLabel.pack()
⋮

def selection(self):
    selection = "You selected the option " + str(self.MyVariable.get())
    self.MyLastLabel.config(text=selection)
```

Some additional notes about the previous code

1. To get different groupings of radio buttons, choose a different **MyVariable**.
2. self.top puts it in the second window that I created earlier. It could also be master.
3. The set of values for each option is the value= command. It is stored within self.MyVariable
4. The way the code is set up is as follows:
    1. Selecting a new button runs the command self.selection
    2. self.selection runs a function which manipulates the self.MyLastLabel Label to contain a specific text string.
    3. This updates The text underneath the buttons, indicating your choice.
5. Of course, you can use this to run more elaborate functions.







Let's try running some code to demonstrate the effects of these different types of widgets.

Practice II

Write a Tkinter selection to create an applet for ordering sandwiches.

This app should have:

1. A menu for different toppings.
2. A set of radio button for type of protein (I'll leave the options to your own imagination)
3. A button to open a new window (.top) with a Message of all possible veggies.
4. (inside this new window) a text entry box for the user to save their information.

For an additional challenge, try to save their choices to a file!