

GUI IN PYTHON

Introduction to PYTHON GUI

Python provides many options for creating a graphical user interface. In this material we will be making use of ***tkinter*** package to build our GUI. It doesn't require any installation since it's embedded in python as a standard package.

Create a Simple GUI Window

Import tkinter

To create a GUI Window, tkinter provides Tk() class. The syntax of Tk() class is:

```
Tk(screenName=None, baseName=None, className='Tk', useTk=1)
```

```
from tkinter import *
```

```
main = Tk()
```

```
main.mainloop()
```

We can add title() and geometry features of Tk() respectively.

```
main.title("This is Python GUI")
```

```
main.geometry("500x400")
```

Create a LABEL WIDGET

To add a label to our previous example, we will create a label using the label class like this:

```
lb = Label(main, text="Hello")
```

Then we will set its position on the form using the grid function and give it the location like this:

```
lb.grid(column=0, row=0)
```

We can format the font of the label as shown below

```
lb = Label(main, text="Hello", font=("Arial Bold", 20))
```

Adding a BUTTON WIDGET

We will create and add the button the main handle like we did for the label e.g.

```
bt = Button(main, text="Submit")  
bt.grid(column=1,row=0)
```

Handle button click **EVENT**

```
def clickButton():
```

```
    lb.configure(text="I was clicked")
```

After creating the method, we will have to attach it to the button as shown below.

```
bt = Button(wd,text="Register Here",command=clickButton)
```

tkinter TEXTBOX

Get input using Entry class

Now let's try getting the user input using Tkinter Entry class (Tkinter textbox).

We can create a textbox using Tkinter Entry class like this:

```
txt = Entry(main,width=20)
```

Then add it to the window using grid function as usual.

To get inputs from the textbox we make use of the `get()` method as shown below:

```
txt.get()
```

To set focus on the Entry widget use the **focus()** method

Add a **COMBOBOX** widget

To add a combobox widget, you can use the Combobox class from ttk library like this:

```
from tkinter.ttk import *  
  
combo = Combobox(main)  
combo['values'] = (1, 2, 3, 4, 5, "Text")  
combo.current(1) #set the selected item  
combo.grid(column=0, row=0)
```

We can also get the combo input with the get() method.

Tkinter checkbox

Add a Checkbutton
widget

To create a checkbutton widget, you can use Checkbutton class like this:

```
chk = Checkbutton(main, text='Choose')
```

Also, you can set the checked state by passing the check value to the Checkbutton like this:

```
chk_state = BooleanVar()
```

```
chk_state.set(True) #set check state
```

```
chk = Checkbutton(main, text='Choose', var=chk_state)
```

```
chk.grid(column=0, row=0)
```

Tkinter checkbox

Set check state of a
Checkbutton

Here we create a variable of type `BooleanVar` which is not a standard Python variable, it's a Tkinter variable, and then we pass it to the `Checkbutton` class to set the check state as the highlighted line in the above example.

You can set the Boolean value to `false` to make it unchecked.

Also, you can use `IntVar` instead of `BooleanVar` and set the value to 0 or 1.

```
chk_state = IntVar()
```

```
chk_state.set(0) #unchecked
```

```
chk_state.set(1) #check
```

Add radio BUTTONS widgets

To add radio buttons, simply you can use RadioButton class like this:

```
rad = Radiobutton(main, text='First', value=1)  
rad.grid(column=0, row=0)
```

Get radio button value

(selected radio button)

To get the currently selected radio button or the radio button value, you can pass the variable parameter to the radio buttons and later you can get its value.

So in the previous code add the following:

For an integer value use:

```
getVal = IntVal()
```

for a string value use:

```
getVal = StringVal();
```

```
rad = Radiobutton(main,text='First', value=1,variable=getVal)
```

Then use `getVal.get()` to get input value from the radio button.

Every time you select a radio button, the value of the variable will be changed to the value of the selected radio button.

Tkinter textarea

Add a ScrolledText
widget

To add a ScrolledText widget, you can use the ScrolledText class like this:

```
from tkinter import scrolledtext
```

```
txt = scrolledtext.ScrolledText(main,width=40,height=10)
```

```
txt.grid(column=0,row=0)
```

Set scrolledtext content

To set scrolledtext content, you can use the insert method like this:

```
txt.insert(INSERT,'You text goes here')
```

Delete/Clear scrolledtext content

To clear the contents of a scrolledtext widget, you can use delete method like this:

```
txt.delete(1.0,END)
```

Create a MessageBox

To show a message box using Tkinter, you can use messagebox library like this:

```
from tkinter import messagebox  
messagebox.showinfo('Message title','Message content')
```

Add a FILEDIALOG

file & directory
chooser

To create a file dialog (file chooser), you can use the `filedialog` class like this:

```
from tkinter import filedialog
```

```
file = filedialog.askopenfilename()
```

Add spacing for widgets

(padding)

You can add padding for your controls to make it looks well organized using padx and pady properties.

Just pass padx and pady to any widget and give them a value.

```
lbl = Label(tab1, text= 'label1', padx=5, pady=5)
```