Tkinter basics: Python's default GUI framework

To make any app with tkinter, 3 components required required:

1. Widget: any button, text etc.
2. Layout- how widgets are arranged in window.
3. Style- color of button, font of text, size of text, background color etc.

Anything that is GUI is a widget in tkinter. There are 2 types of widgets:

1. Tk widgets 🡪 old styling
2. Ttk widgets🡪 a little modern with a little more functionality.

A picture containing triangle, line, text, font

Description automatically generated

1. There are all the widgets inside ttk while tk gives us the basic logic.
2. To make a window we call this function and store it in a variable/object:



Though nothing will happen if we run the program now.

1. To view the screen we just created, we need to call a mainloop:

A picture containing text, font, screenshot, number

Description automatically generated

This will help us view a blank window like this:

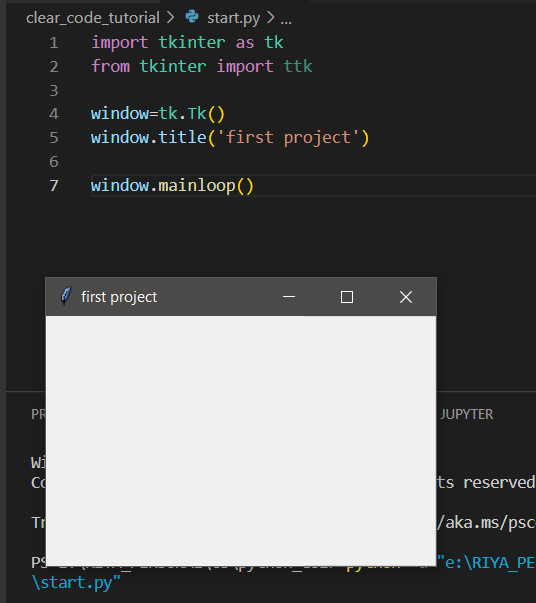
A screenshot of a computer

Description automatically generated with medium confidence

1. We can make changes to this window.
2. To make a title we do:



This will give a title to the window we just created.



1. To set the size of the window we do:



This function takes a width and a height as a string separated by “X”. actual usage:



1. Adding a widget: label.

Label in tkinter means text. It can be created using a function. Arguments are required for this function, we need to set a master, which is the parent. The parent here is the container inside which this label is gonna be. Here we want to put our label inside out window so, the parent here is the container “window”.

The next argument will be the actual text that the label is going to display.

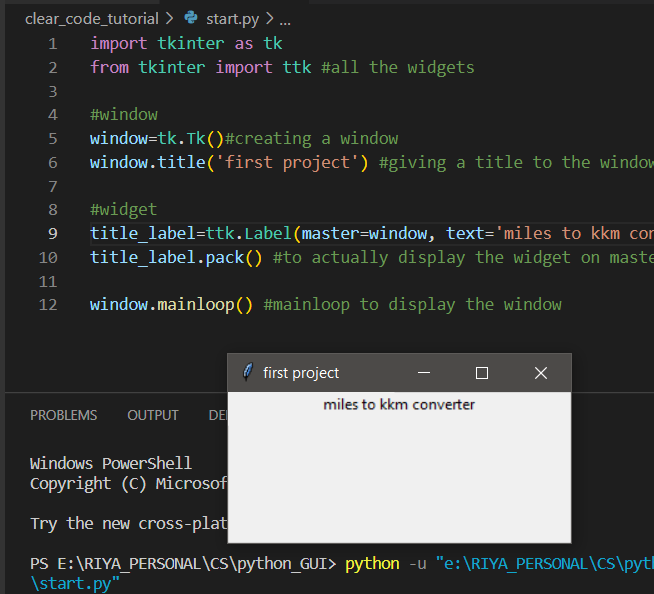


All these steps defined the label widget but it still won’t be included if we run the program.

1. To put the label, or the widget onto the window we need another method:

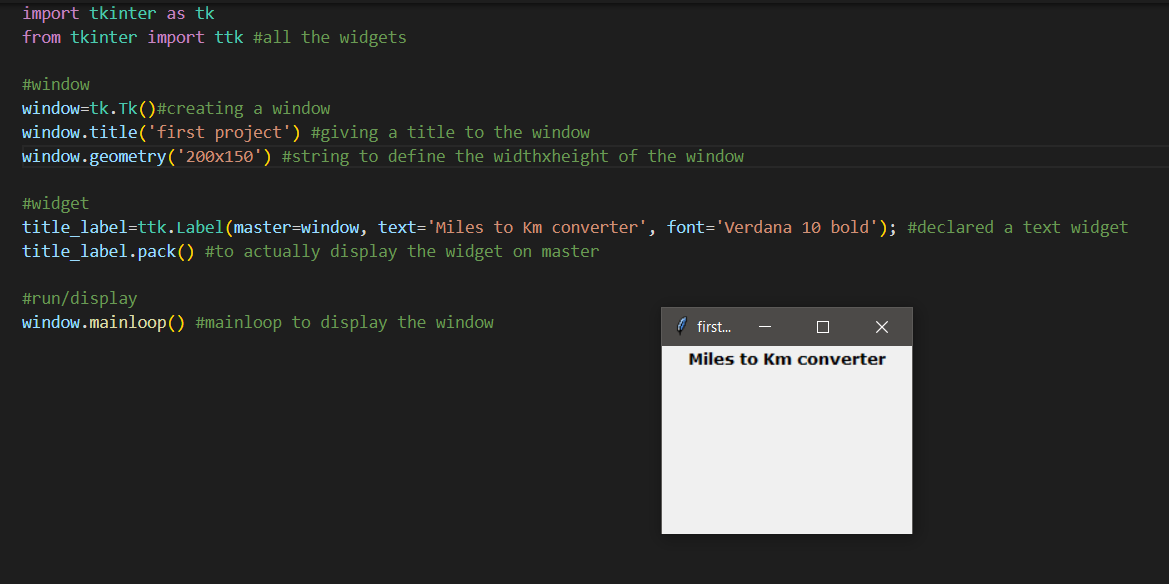
The simplest method to put the label is pack():





1. To insert font and size of font, specify the same in the label argument. We can also add bold and italics.





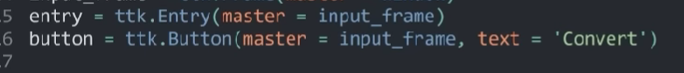
1. Creating an input field with a button. All of this is inside a container.
2. We will create a frame. The frame only needs a parent/master that is out main window here.

A picture containing text, font, screenshot, line

Description automatically generated

The widgets will be put inside this frame.

1. Creating an entry/input box and a button. Both will have the newly created input\_frame as their master.



1. Now we need to put the 2 widgets (entry and button) inside the frame and put the frame inside the window. Till now we have only declared everything, not implemented them.
2. To put everything inside their parent, we will do the same thing that we did to implement the title, we will use pack().

A screenshot of a computer program

Description automatically generated with medium confidence

1. Note the difference:

A screenshot of a computer

Description automatically generated

A picture containing text, screenshot, software, multimedia software

Description automatically generated

1. To make the entry and button side by side, we will add an argument to both the entry and button.

A screenshot of a computer program

Description automatically generated with medium confidence

1. Adding padding. If we add some x padding to the button, then the button and the entry field will have some space between them. If we add some y padding to the frame, then it will create space between the widget container frame and the heading/text label.

A screenshot of a computer program

Description automatically generated with medium confidence

1. To add functionality, we will add a function which will be called when we press the button.



1. When we press the button, we want to get the content of the entry field. This can be done by EASY: use the get method. Input.get() this method will return us whatever is written inside the widget.

But this method is not used or is the efficient method to et the value from a widget.

1. To store the value from a widget, we need to create a different tkinter object that is made for this purpose. So now whatever is in the entry field will update this entry field object variable and whenever the entry field object variable gets updated, the entry field will also get updated.

A screen shot of a computer code

Description automatically generated with low confidence

Declaring an input variable object and connecting it to the entry box. In the convert method we will still access the contents of this object using the get method.

1. Now we will change the output label (the one which will display the output of conversion). In tkinter, the label can also be changed. For this we have to create a tkinter variable object again. This time because it is a label, the variable will be of type string. Then we will connect this variable to the output label.
2. Initial output label:

A picture containing text, screenshot, font

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

1. Making a string object variable and connecting it to output label:

A screenshot of a computer program

Description automatically generated with medium confidence

Note that connecting the string variable to the output label made the label text completely disappear.

This disappeared because the textvariables overrides whatever is written inside the label.

1. Now we can use this output string by updating it. In the command/function that gets called when the button is pressed, we will update the value of the string variable and then this updated label will be displayed on the screen. We can update the value in the object by using the set method.
2. This is how we will get the value and then update the label:

A screenshot of a computer

Description automatically generated with medium confidence

Will display the output like this after updating the output label:

A screenshot of a computer

Description automatically generated with medium confidence

1. The default tkinter styling methods are not very good looking, so we will use some external module, here we will use bootstarp module.

A picture containing text, font, screenshot

Description automatically generated

Ttk bootstrap takes all the ttk widget and adds functionality to them.

1. Now instead of ttk we will import ttkbootstrap

A screenshot of a computer screen

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

And the default style has changed.

1. Another change:

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated with medium confidence

We are using different themes.