

TKinterDesigner tutorial

Author	Honghaier
Ver	1.0
Date	2020-03-09

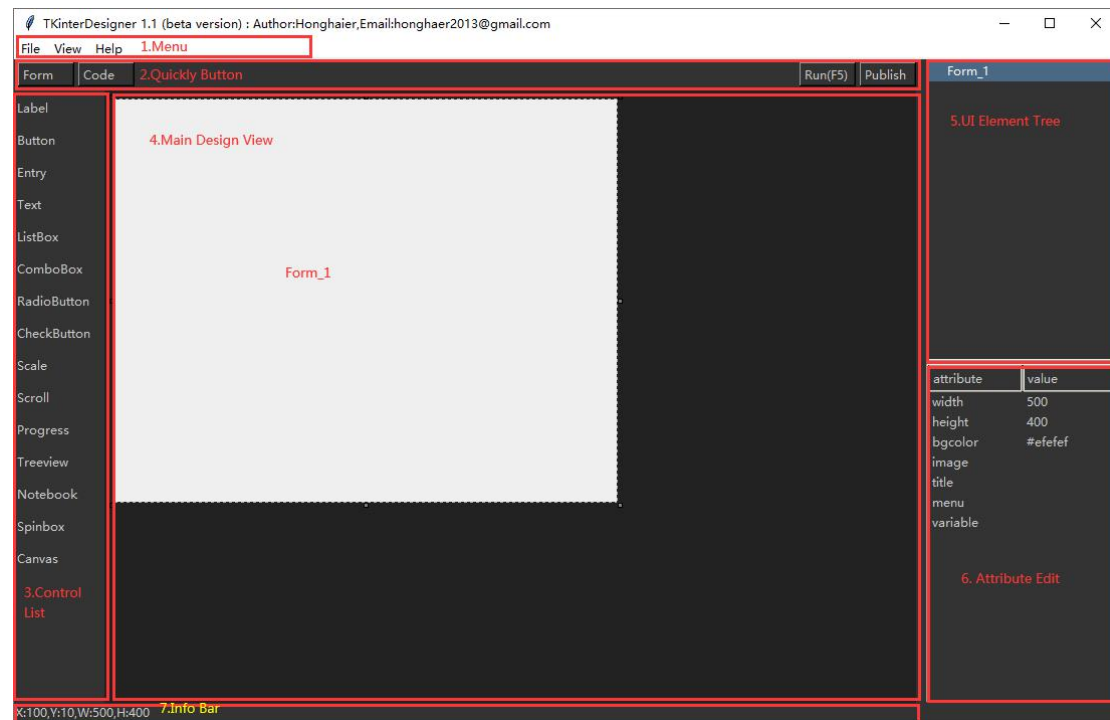
Hello everyone, I am Honghaier. A programmer come from China, I hope to help everyone improve the development efficiency in Python development. So I made a Tool software, It's named "TKinterDesigner", here are some simple instructions. If you have any questions or suggestions, you can send me an email: honghaier2013@gmail.com. I will continue to improve and release updated versions and tutorials. This software is completely free. If you can use it, you can recommend him to your friends and colleagues.

What is "TKinterDesigner"?

TKinterDesigner is a small Python-based TKinter interface editor for software interface design and development when prototyping small and fast Python functions.

Instructions

一. Interface description



The editor consists of seven areas:

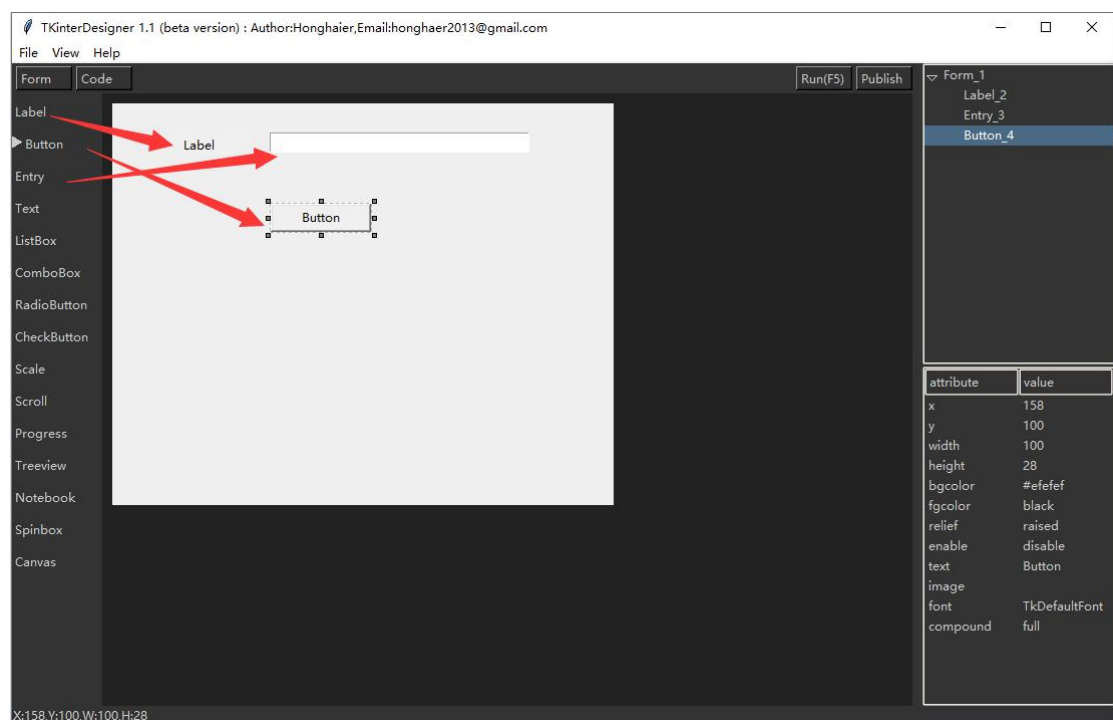
- 1. Main menu:** perform basic file, view, help and other menu item operations.
- 2. Shortcut button:** Quickly switch the form and code view, and view the running status of the interface and publish the EXE.
- 3. List of controls:** commonly used interface controls, used to drag to the interface design area.
- 4. Interface design area:** The main view area for the interface.
- 5. Interface control tree:** a list of all controls in the current interface for quick access and viewing.
- 6. Interface property area:** Display and modify the properties of the corresponding control.
- 7. Bottom information area:** display the position and size of the corresponding control.

二. Features description

1. Interface design

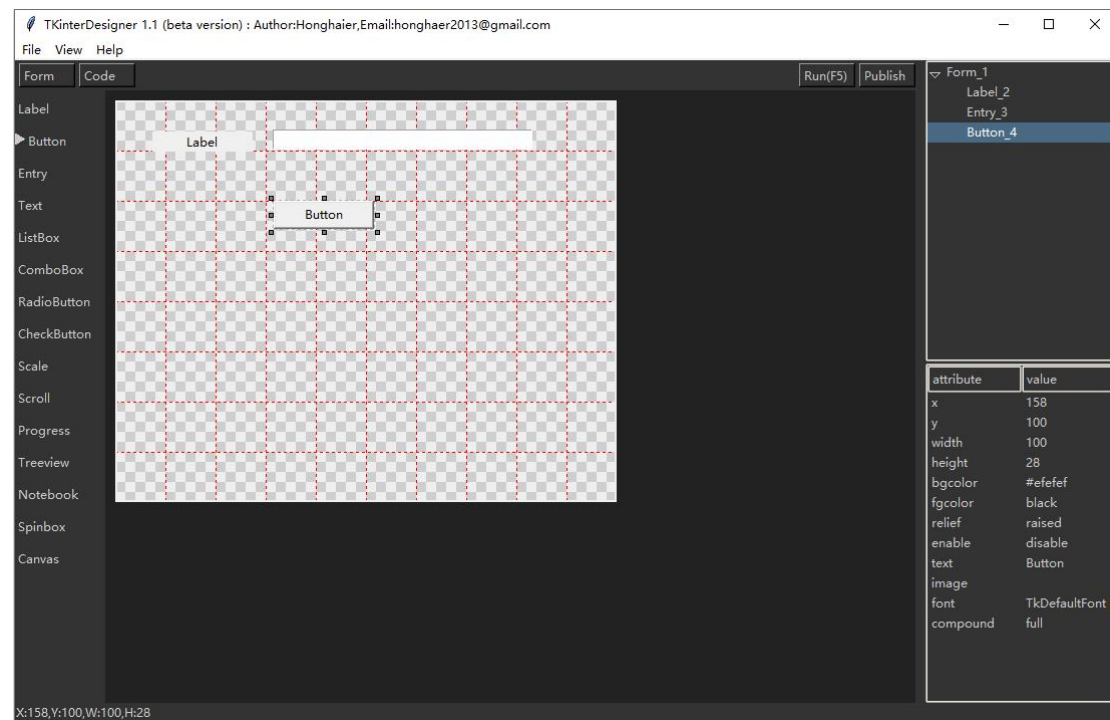
(1) Interface control creation

Use the left mouse button in the control list area to select the corresponding control and drag it to Form in the interface design area to complete the basic control creation. Then you can use the mouse to drag the control to the corresponding position.



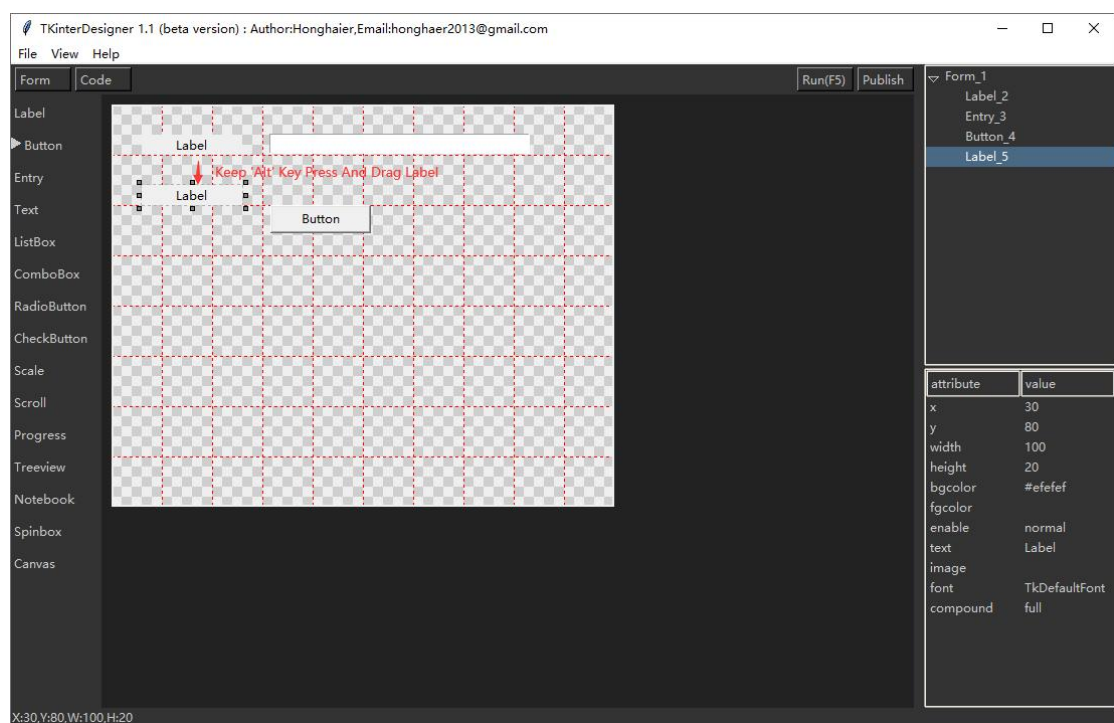
(2) Interface control placement

If you need to place a simple interface, first you can find "Grid" under the "View" item in the main menu. You can also use Ctrl+G to quickly open or close the grid. This will allow you to make basic size observations. Then, you can find "Adsorption" under the "View" item in the main menu, or use Ctrl+D to quickly turn adsorption on or off. After turning on the "grid" and "adsorption", you can drag and drop controls directly to quickly place and align them.

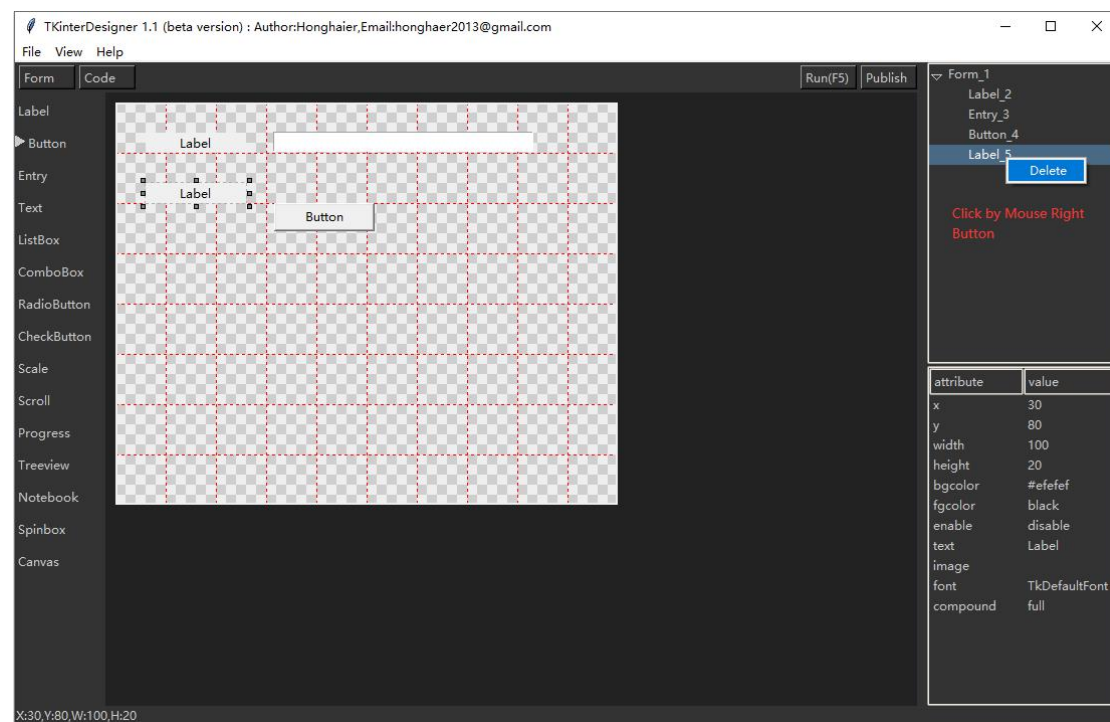


(3) Control copy and delete

Copy control: After selecting a control, hold down the Alt key and drag the control with the mouse to copy a new control.



Deleting a control: You can delete it by using the shortcut key delete or right-clicking a control in the control tree in the upper right corner, and in the pop-up menu. .

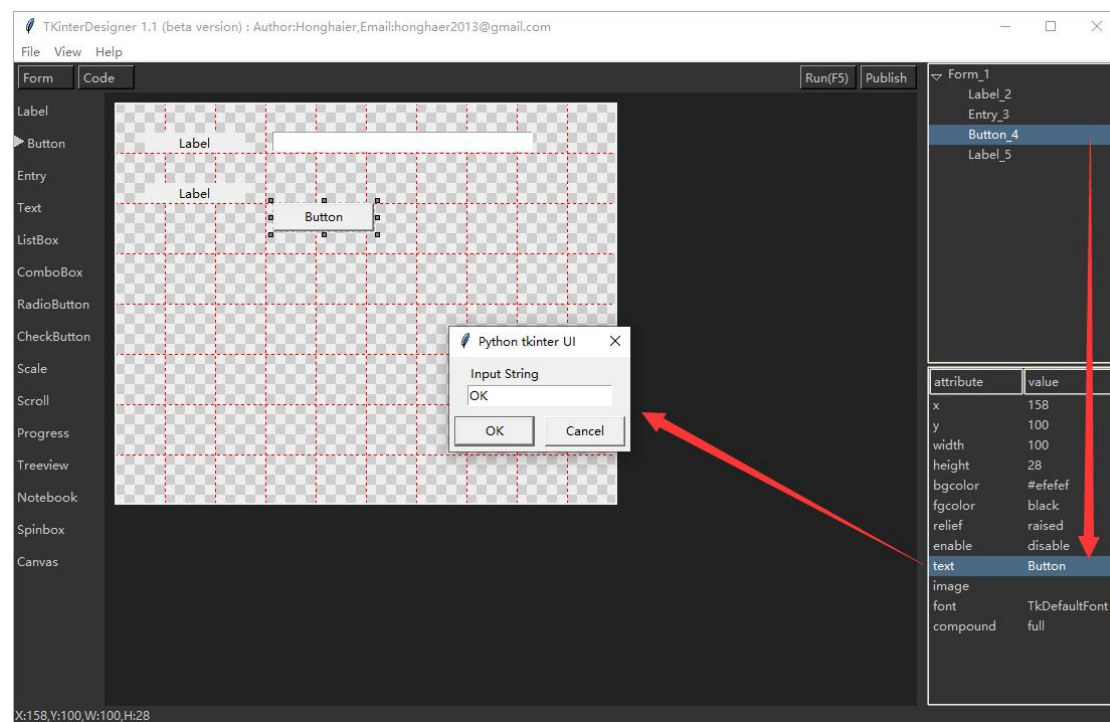


(4) Control property editing

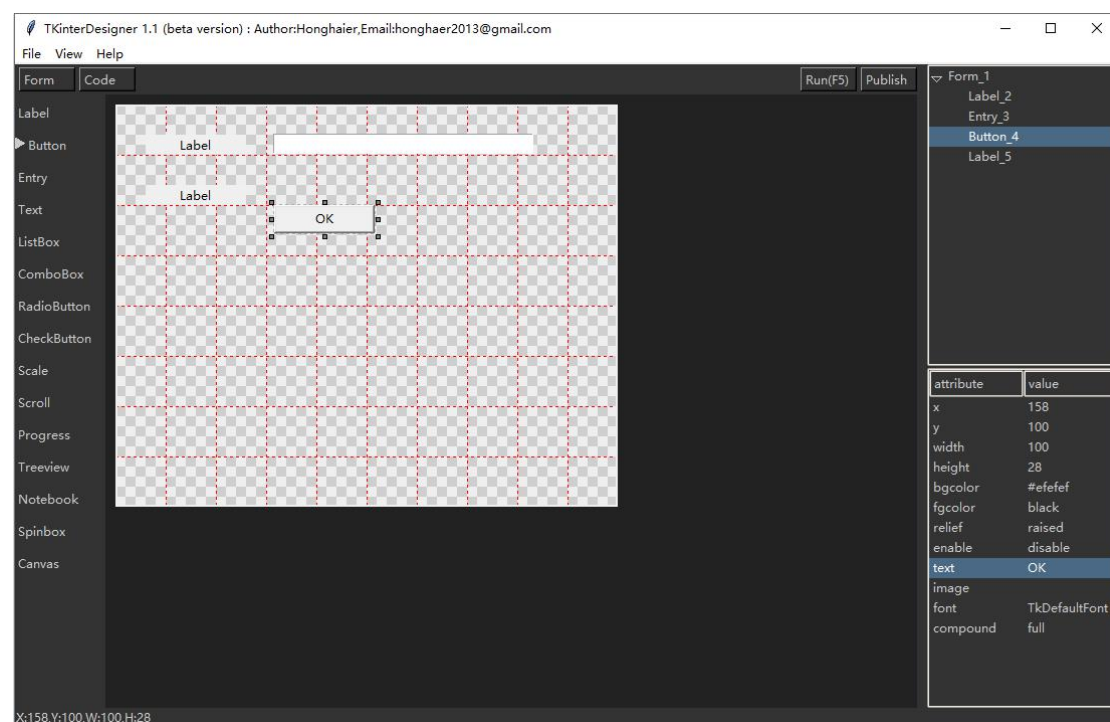
After selecting the corresponding control, in the control list tree in the upper right corner, the corresponding tree item of the corresponding control will be highlighted. At this time, we can modify the corresponding item in the attribute editing area in the lower right corner.

I take the control button as an example, and take a screenshot to show how to modify the text. When we double-click the property "Text" item, a dialog box will pop up, prompting to enter new text.

TKinterDesigner tutorial



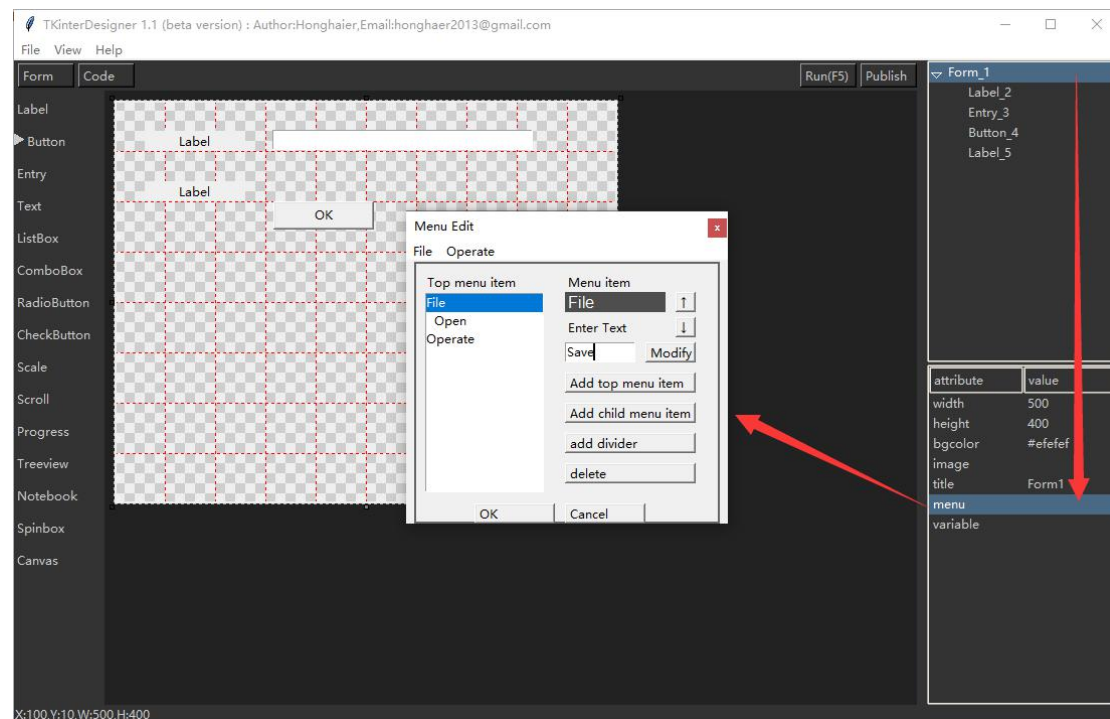
After inputting, click "OK", you can see that the corresponding control is changed to the currently entered text.



Regarding other attributes, we will not introduce them one by one here, and you can try it yourself.

(5) Menu editing

If you need to add a menu to the program, you can select "Form_1" in the control tree list in the upper right corner. At this time, you can see that there are "menu" items in its properties. Double-click the menu edit dialog box to pop up.

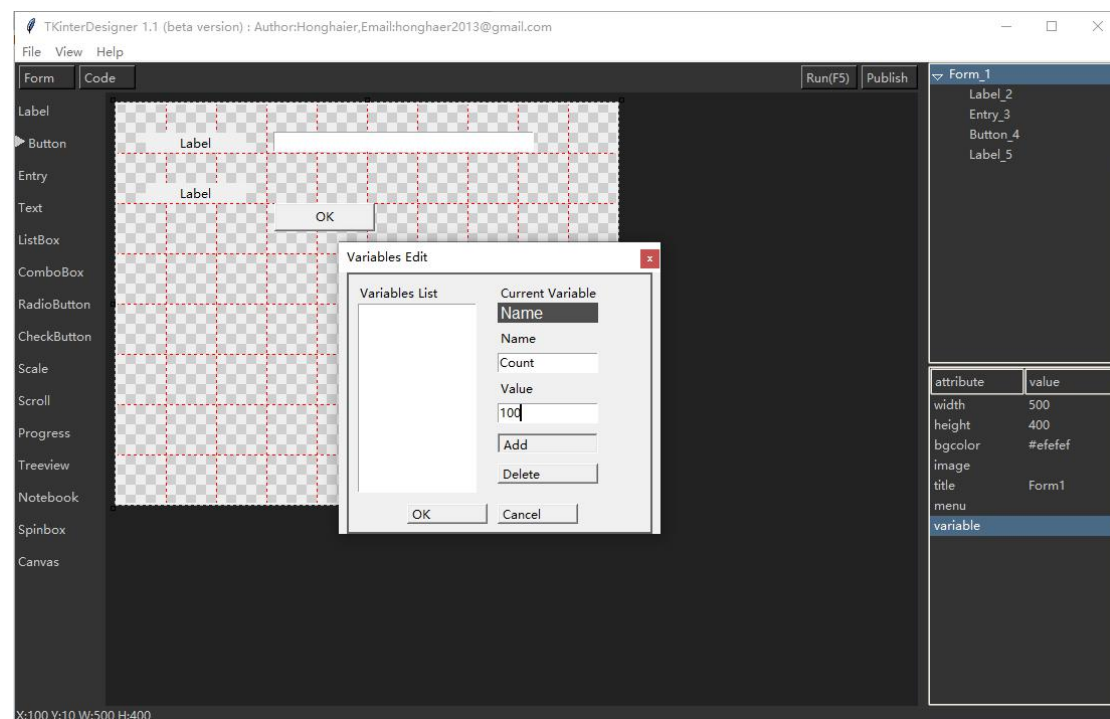


For example, In the menu editing dialog, we first enter "File" in the edit box for entering text, and then click "Add top menu". At this time, we can see that the "File" item appears in the list box of "Top menu items" on the left. , And added the menu and "File" menu items in the dialog box, it can facilitate you to preview the menu, yes, this is just a preview. After we add the top-level menu, click the corresponding menu item in the "Top-level menu item" list on the left, and you can add new sub-items to it by clicking the "Add sub-menu item" button on the right. And use the up and down arrow buttons to modify the order of menu items at the same level.

The processing of this part is slightly complicated, but I try to make the operation buttons simple and clear, I hope you can understand.

2. Custom variables

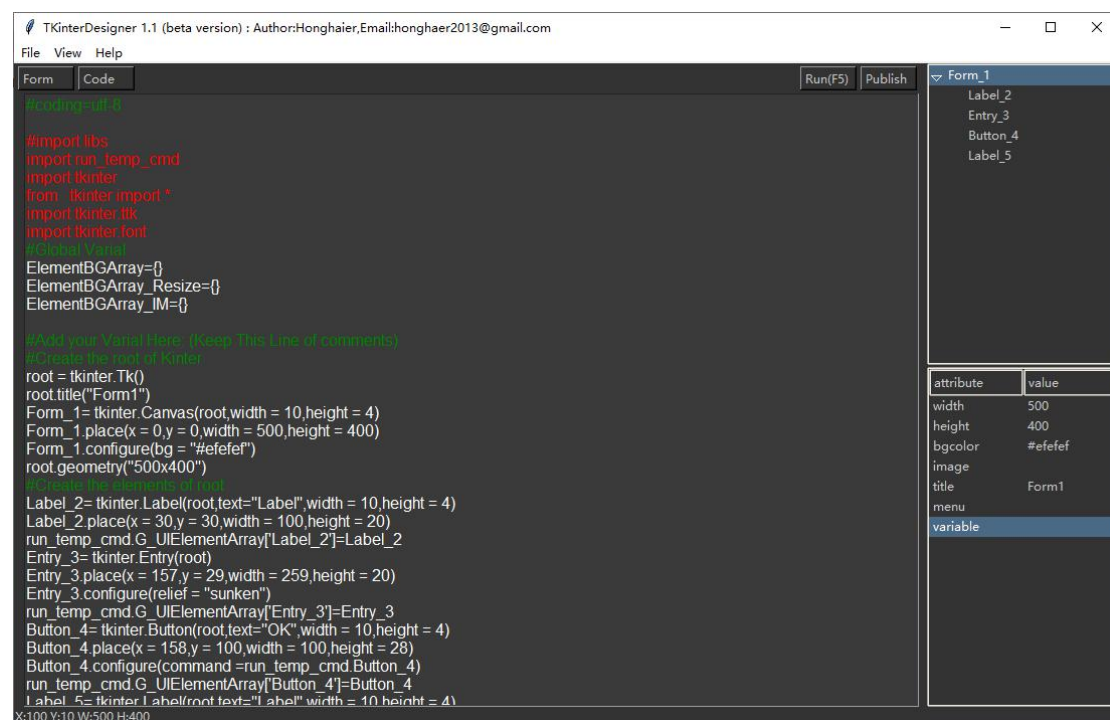
We can add some global custom variables to the program to facilitate our logical access. Select "Form_1" in the control tree list in the upper right corner. At this time, you can see that there are "variables" in its properties. Double-click the variable edit dialog box to pop up.



We enter a Count variable here, the default value is 100, and click "OK". We will record the information of this variable in the saved file.

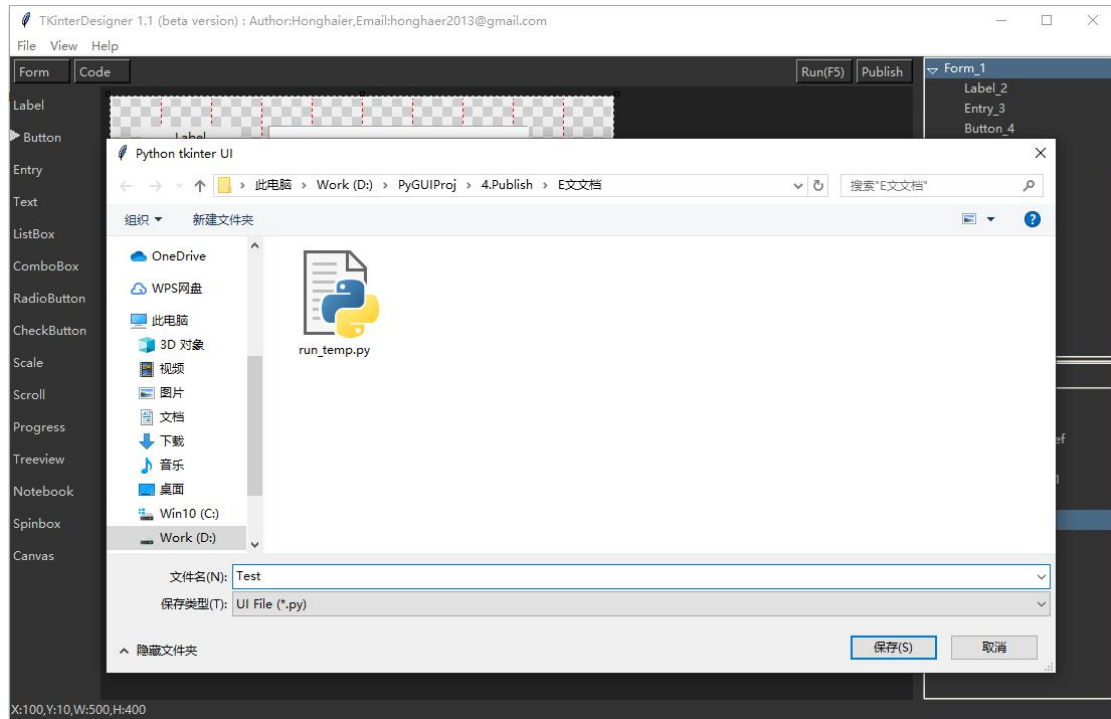
3. Form and code

If you want to view the code, you can click "Code" in the shortcut button area. At this time, we can see the interface code display. Although it cannot be edited directly at the moment, it can make you understand the code.



4. Saving and opening files

OK, everything is done, we just need to find "Save" under the "File" menu item in the main menu, or Ctrl + S to save the file.



Here we define the file name as "Test", click "Save", if successful, a "Save Successful" prompt will pop up. We will see the generated Test.py and Test_cmd.py files in the corresponding directories, where Test.py is the interface control and layout file, and Test_cmd.py is the corresponding logical processing file.

5. Logical processing

Open Test_cmd.py with a text editor and we will see the following screenshot:

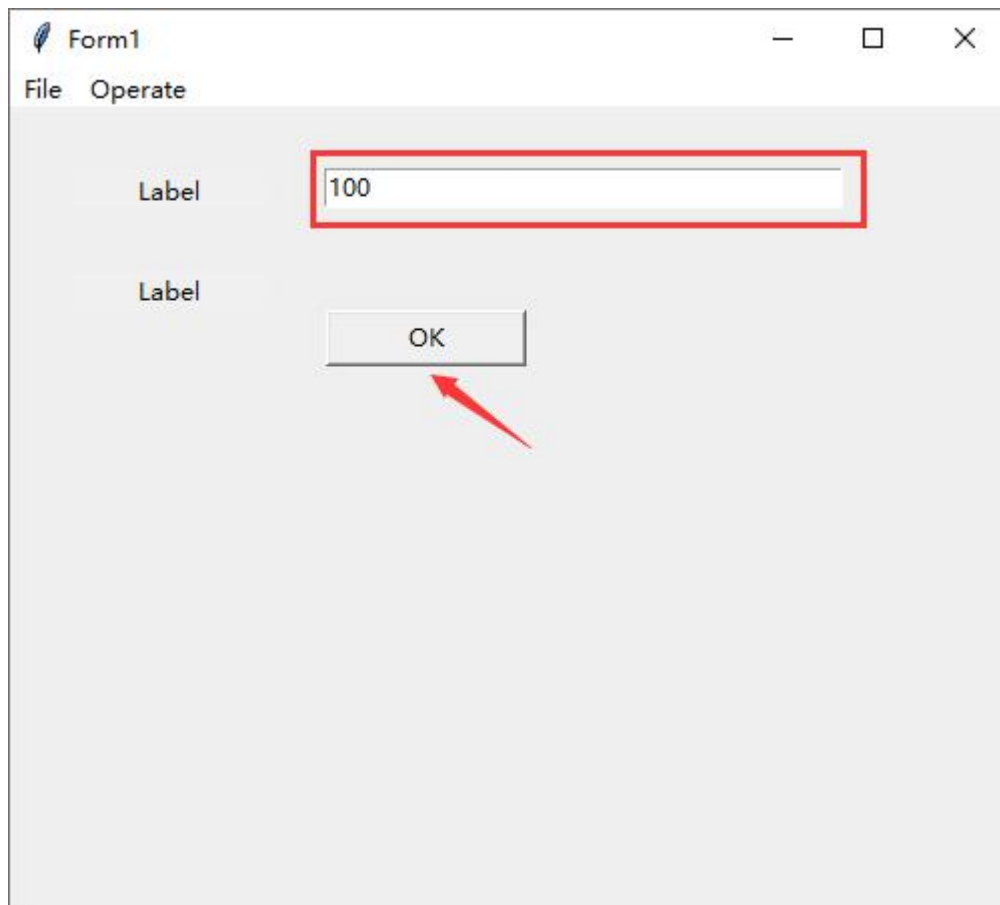
```
D: > PyGUIProj > 4.Publish > E文档 > Test_cmd.py > ...
1  #coding=utf-8
2
3  G_UIElementArray={'Label_2':None,'Entry_3':None,'Button_4':None,'Label_5':None}
4  def Button_4():
5      pass
6  def Menu_Open():
7      pass
8  def Menu_Save():
9      pass
10 def Menu_Operate():
11     pass
12
```

G_VarArray is the global variable dictionary we defined, and the variables we defined in the editor are displayed here. There is also a G_UIElementArray dictionary, which will store all the

control objects at runtime. You can access it directly through the Key. For example, we change the code:

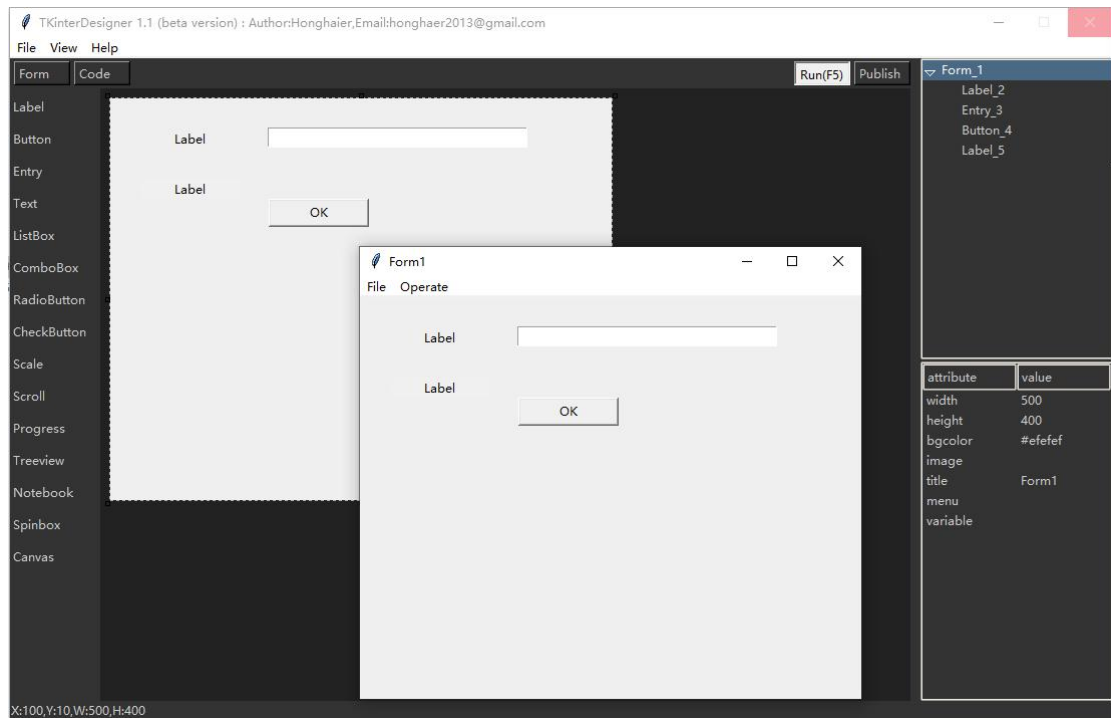
```
Test.py Test_cmd.py ×
D: > PyGUIProj > 4.Publish > E文文档 > Test_cmd.py > {} *
1  #coding=utf-8
2  import tkinter
3  from tkinter import *
4  G_UIElementArray={'Label_2':None,'Entry_3':None,'Button_4':None,'Label_5':None}
5  Entry3_Value = None
6  def Button_4():
7      global G_UIElementArray
8      global Entry3_Value
9      Entry3_Value = tkinter.StringVar()
10     G_UIElementArray['Entry_3'].configure(textvariable = Entry3_Value )
11     Entry3_Value.set('100')
12     pass
13 def Menu_Open():
14     pass
15 def Menu_Save():
16     pass
17 def Menu_Operate():
18     pass
19
```

After saving, you can execute Test.py and click the "OK" button. At this point you will find that the value of Label becomes 100.

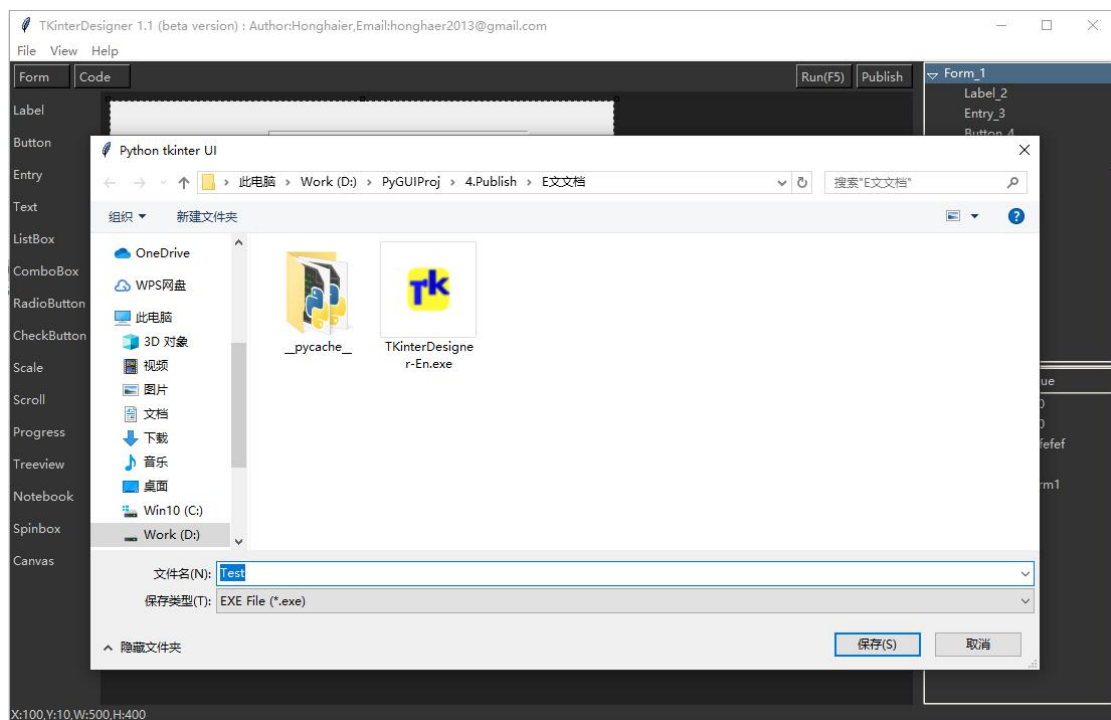


6. Run and Publish

Sometimes, we have designed an interface, and we want to quickly see the interface execution status. At this time, we only need to click the "Run" button in the shortcut button area to quickly see the results.



If you have completed it, you can also click "Publish" to generate the corresponding execution file.



Click “Save” Button, You can see the console print some compile info, Finally, The execute file can be made.

三. About

First of all, because it's just a little motivation to learn Python intermittently for two months in my spare time, it has a lot of problems, but I will continue to improve, and I hope that interested Python enthusiasts will communicate with me, because I'm writing before I haven't finished a Python book, there are too many not getting started.

In addition, I am very optimistic about Python's interface tool requirements for rapid prototyping, and hope that Python is getting better and better.

Finally, I wish you every success in your work and good health ~

Honghaier

2020/03/09