TKinterDesigner 使用教程

开发者	Honghaier
版本号	V1.3
更新日期	2020-04-17

GitHub: https://github.com/honghaier-game/TKinterDesigner.git

TKinterDesigner 是什么？

TKinterDesigner 是一款基于 Python 的开发工具，用于进行基于小型界面的 Python 应用项目开发。

TKinterDesigner 都有什么功能？

TKinterDesigner V1.3 目前包括以下九大功能:

1. 项目管理:对于项目进行创建，打开。
2. 文件管理:对于项目进行窗体的创建，文件的创建和资源的导入。
3. 界面设计:对于 TKinter 界面进行设计。
4. 控件设置:对于控件进行基本的属性编辑。
5. 变量绑定:对于 TKinter 的控件进行变量的绑定。
6. 事件响应:对于 TKinter 的控件建立事件与函数之间的映射。
7. 逻辑编写:对于事件函数进行逻辑处理。
8. 编译运行:调用 Python 命令进行工程的编译运行。
9. 打包 EXE:调用 Python 命令对工程进行打包 EXE。
10. 自定义模块导入:将自定义的模块导入并调用。

TKinterDesigner 功能讲解：

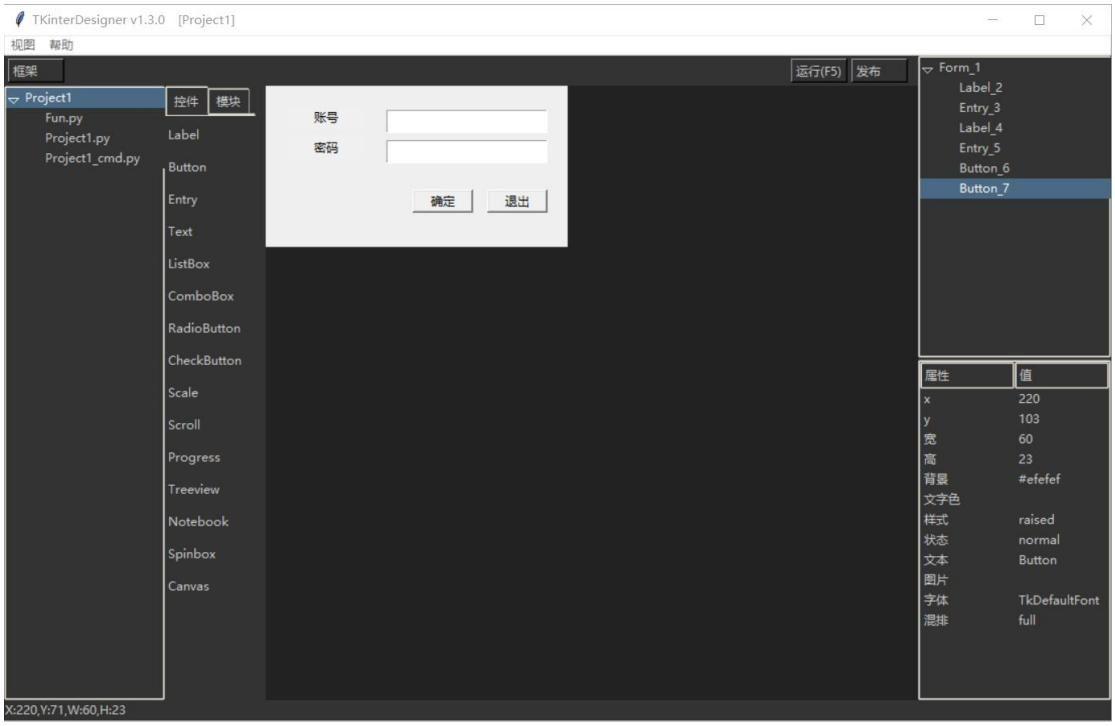
1.项目管理

双击启动 TKinterDesigner.exe，首先进入的是项目管理界面，你可以根据在右上角需要选择所用语言，这里我们选择中文。

在这个界面中，我提供了三个选项的选项卡：

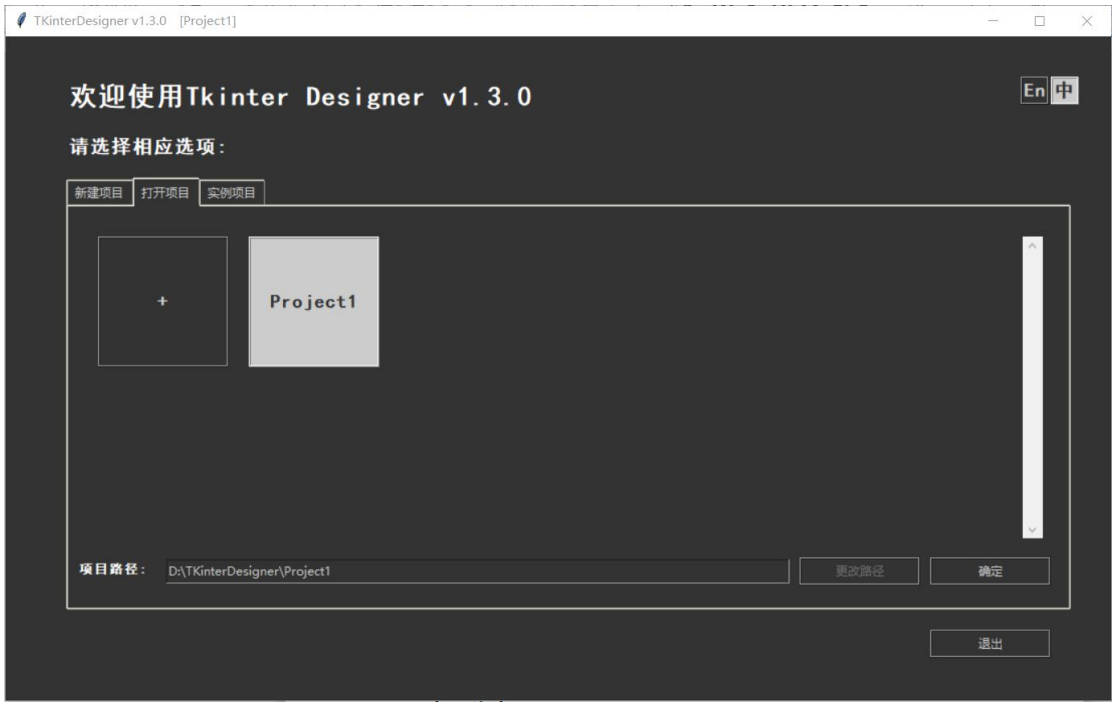
（1）**新建项目：**提供了**空白界面项目**,**对话框界面项目**和**单文档界面项目**三种模版供选择。你只需要选择相应的项目，并点击“确定”，即可完成一个项目的建立，如果需要更改项目路径，可以点击“更改路径”进行修改。

这里我们选中**对话框界面项目**，然后点击“确定”。



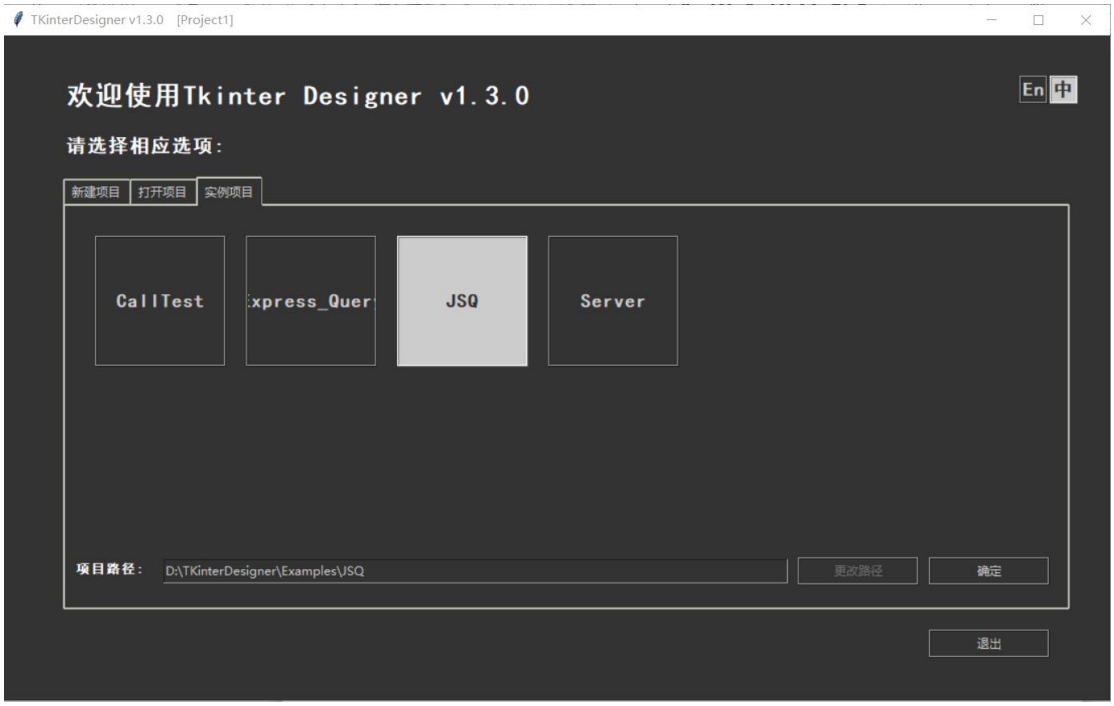
项目创建成功后，我们将立刻进入到主设计界面进行项目开发，我们可以通过右上角的关闭按钮来返回到项目管理控制台。

（2）**打开项目：**所有我们创建的项目，会在这个面板列表中显示，我们只需要选择需要的项目并点击“确定”即可进入项目，第一个显示为加号的按钮，用于打开一个不在列表中的项目。

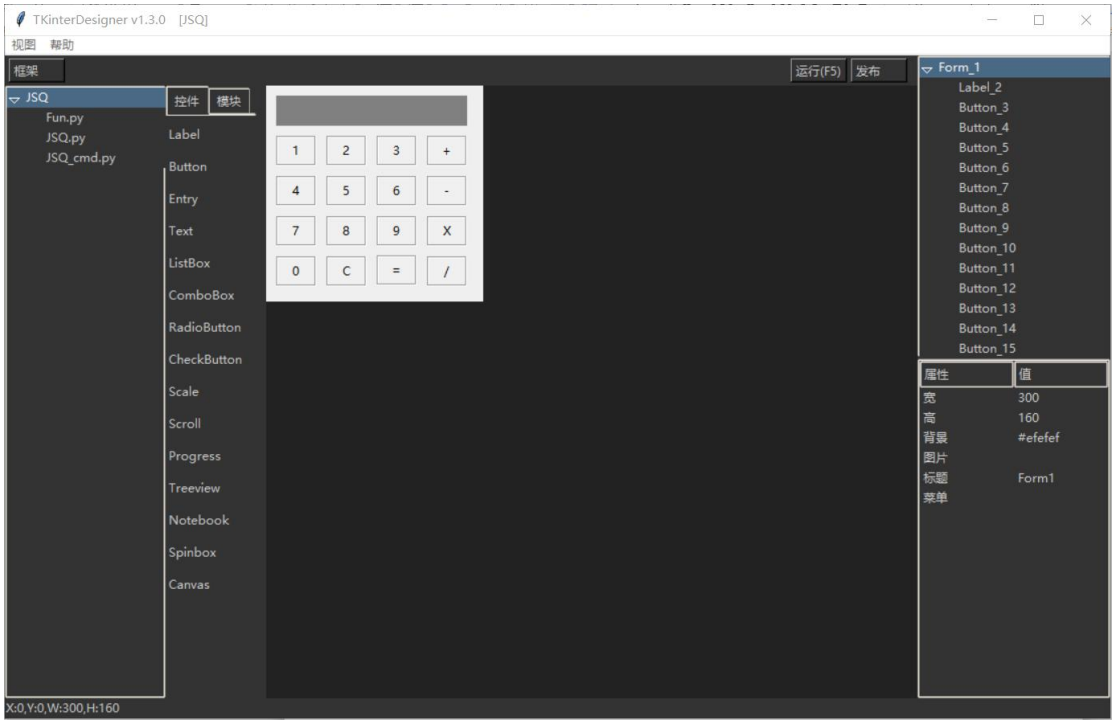


（3）**实例项目：**我提供了一些小的案例作为参考，开发者可以打开进行学习，

以便对于一些类似的小工程的框架和实现有所了解。



比如我们选择“JSQ”，然后点击“确定”，你将看到一个计算器的界面项目:



在这个主设计界面中，我们可以看到它的布局为：

（1）**顶部的主菜单**：1.视图：包括设计时用到的网格和吸附功能，你也可以通过 Ctrl+G,Ctrl+D 来快速调用。2.帮助：一些无用的信息，可能你需要找到我的话，可以看一下。

（2）**主菜单下面的快捷按钮**：“框架”按钮可以显示或隐藏框架结构树，“运行”按

钮可以快速的运行项目进行测试，“发布”按钮则可用于打包发布项目为一个 EXE 程序。

（3）**最左边的框架结构树**：包括项目所有的文件列表，记住：你也可以在上面通过鼠标右键弹出的菜单，增加一个窗体界面，或者增加一个 Python 文件，或者导入一个资源文件。如果你在设计界面时，这个框架结构树会影响你的观察窗口空间，你可以点击“框架”按钮，显示或隐藏它。

（4）**左边的控件和模块列表选择区**：对于界面设计时所需要的常用控件，我在这里有所罗列，虽然还不全，但随着更新，我相信会慢慢丰富起来，这里模块选择区，用于导入一个自定义的模块，在实例项目中有一些自定义的模块类并在项目中使用的案例，比如 Express 或者 ChatServer，你可以稍微看一下，它只需要有一定的设计约束即可。我将在第七部分功能进行详细说明。

（5）**中央的设计预览区**:界面设计的主视区，你可以在这里将所需要的各个界面控件拖动进来并进行摆放和拉伸，所见即所得的搭建界面。

（6）**右上部分当前界面所有控件列表树**:罗列当前界面中的所有控件，你可以点击相应的树项来选中对应控件，也可以通过鼠标右键弹出菜单中对其进行删除。

（7）**右下部分当前所选中控件的属性列表项**：罗列当前所选中的控件对应的所有属性列表，你可以在这里对相应属性项双击进行修改。

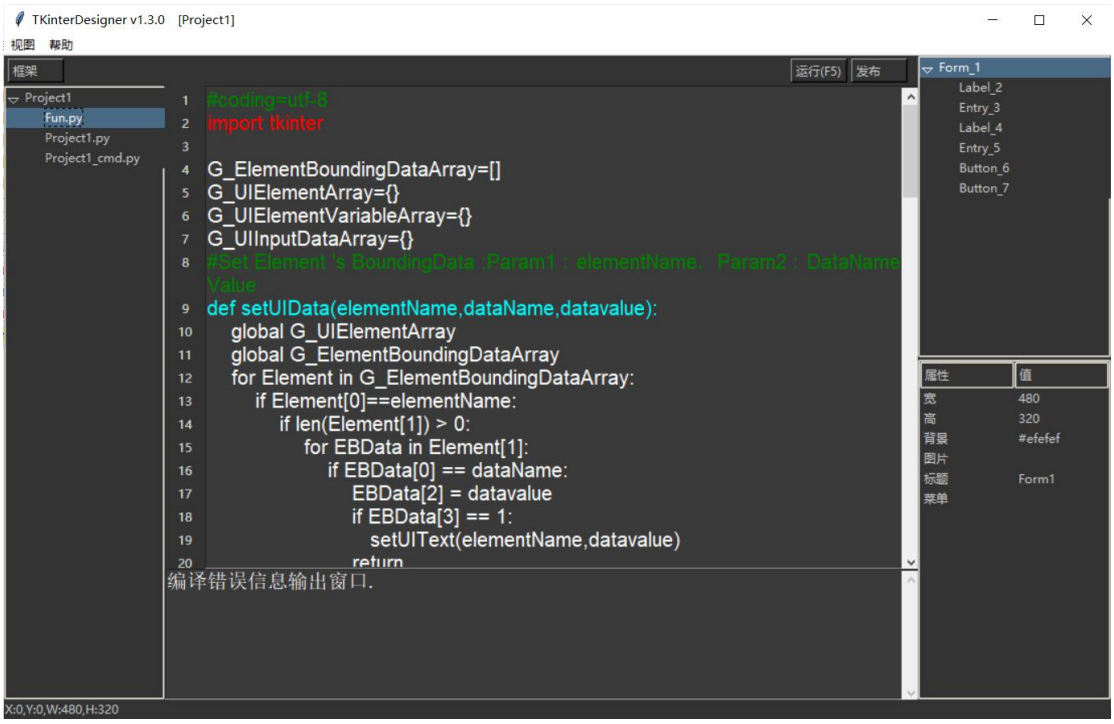
（8）**底部的信息文字**：对当前控件的位置，大小信息进行显示。

2. 文件管理

最左边的框架结构树，对当前项目中的文件进行了罗列，我们以对话框工程为例，可以看到框架结构树下有一个项目名称命名的根结点项，其下有三个文件项：

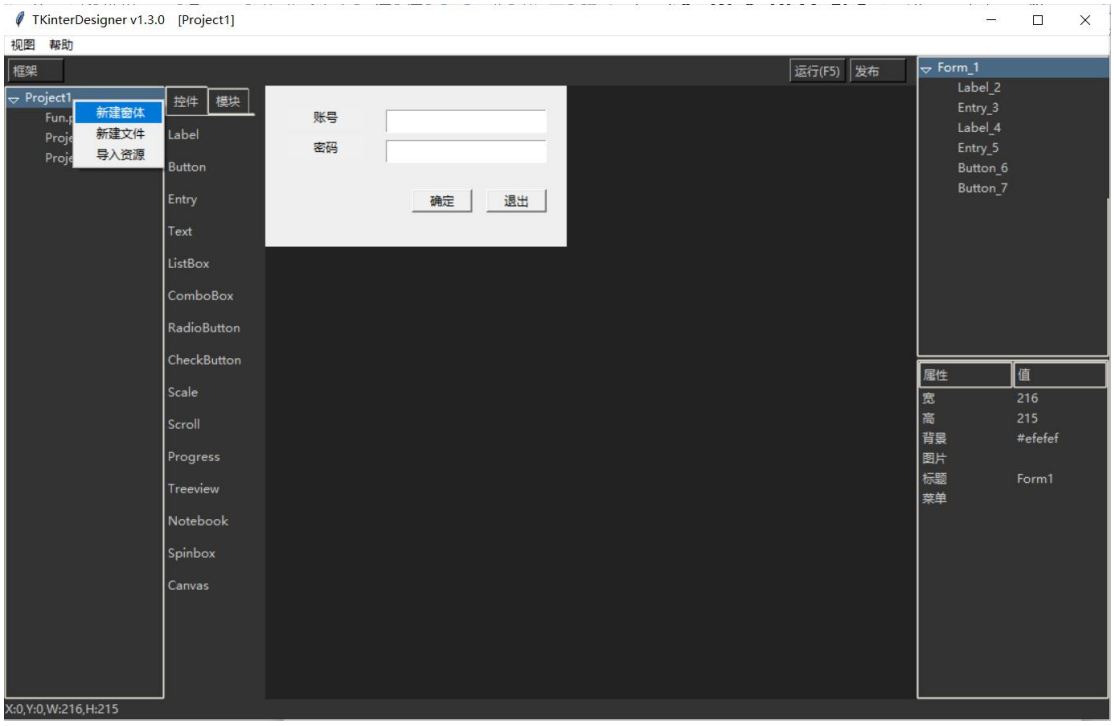
1. Fun.py:这是一个公用的函数库文件，提供对于控件和控件变量的存取访问等函数实现。
2. Project1.py:这是项目主界面的 Python 文件，提供对于界面的基本布局进行代码支撑。
3. Project1_cmd.py:这是项目主界面的逻辑文件，提供对于界面的逻辑进行代码支持，控件的事件函数主要在这里进行编码实现。

当我们点击到一个 Fun.py 或 Project1_cmd.py 时，在主视图区域会变成代码区:



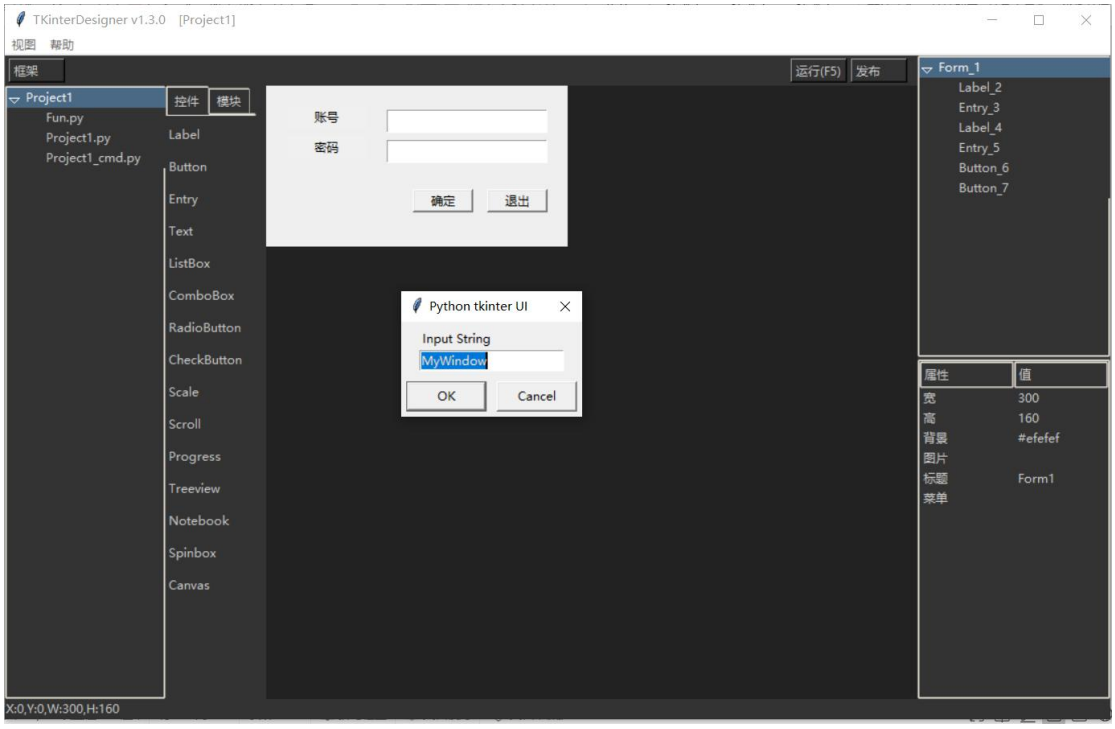
因为软件致立于使用窗体设计器，而不是修改窗体界面的代码，所以在点击 Project1.py 时，主视区显示的是界面，而对于逻辑代码和函数库代码，这是希望开发者多多进行逻辑代码编写，修改以及调错的部分，所以这时显示的是一个代码文本区域和一个信息输出窗口，方便随时进行代码修改以及编译运行时查看输出。

如果我们希望在工程中创建多个窗口，我们可以在框架结构树中新增界面，比如我们打开新创建的对话框界面项目，在左边的框架结构树上用右键点击一下

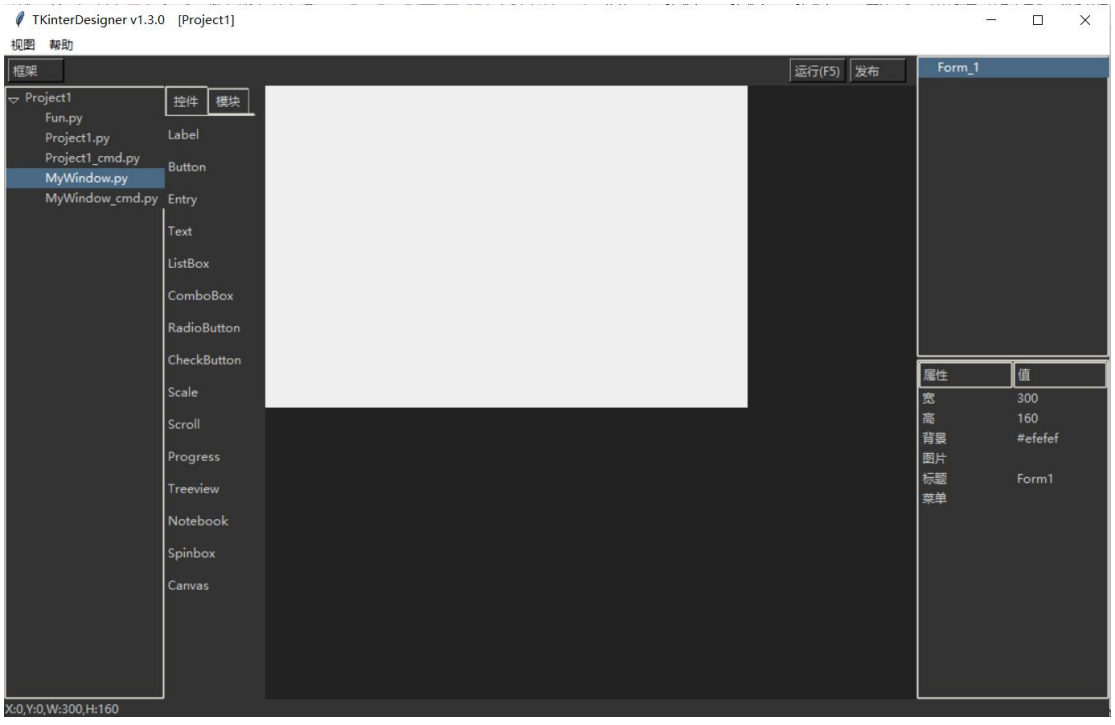


在弹出的菜单中，点击“新建窗体”，这里我们可以看到一个新弹出的对话框，我们可

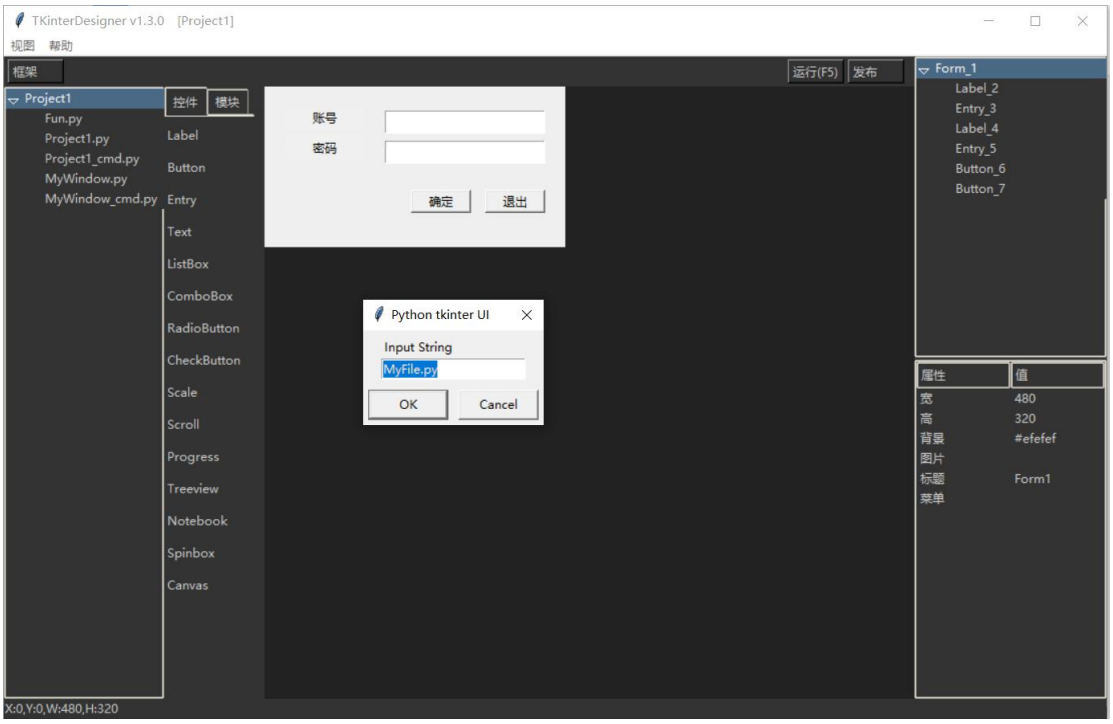
以输入新窗体的名称，然后点击“OK”。



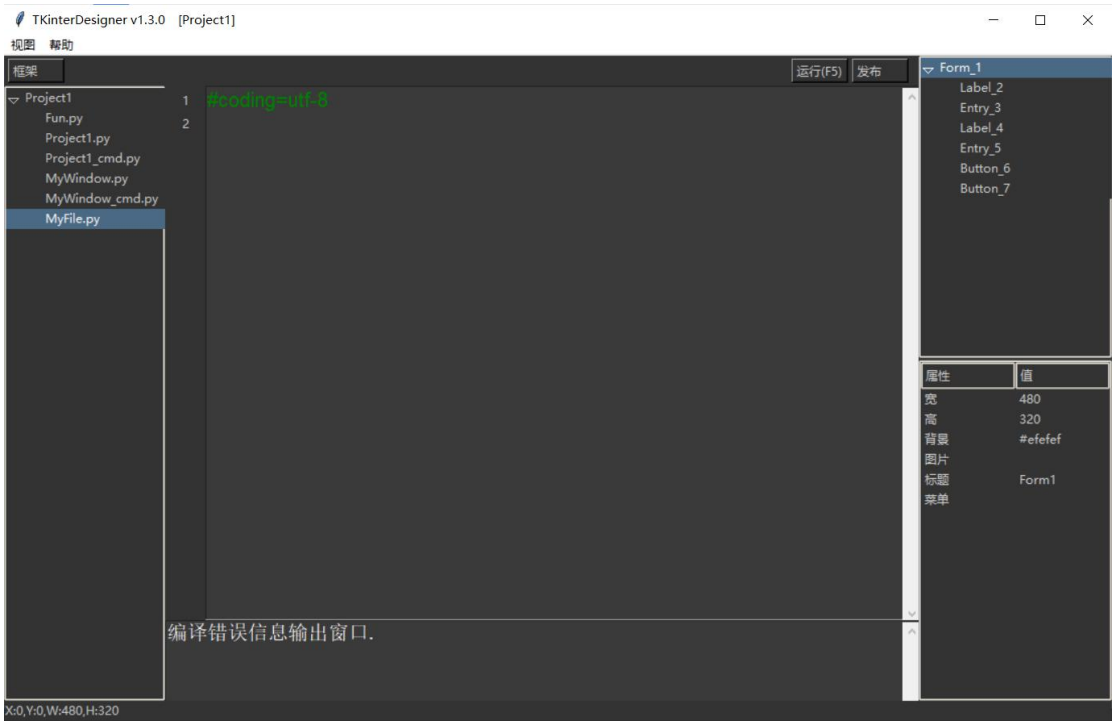
点击“OK”后，我们可以看到一个新增的窗体，包括 MyWindow.py 和 MyWindow_cmd.py 两个文件，分别对应 MyWindow 的窗体布局和逻辑实现。



如果我们想增加一个自己的逻辑代码，我们可以在框架结构树右键弹出菜单项点击“新建文件”，输入新建文件的名称，我们可以新创建一个 Python 文件出来。



点击“OK”后，可以看到新的文件代码，这时你可以开始编写代码了。



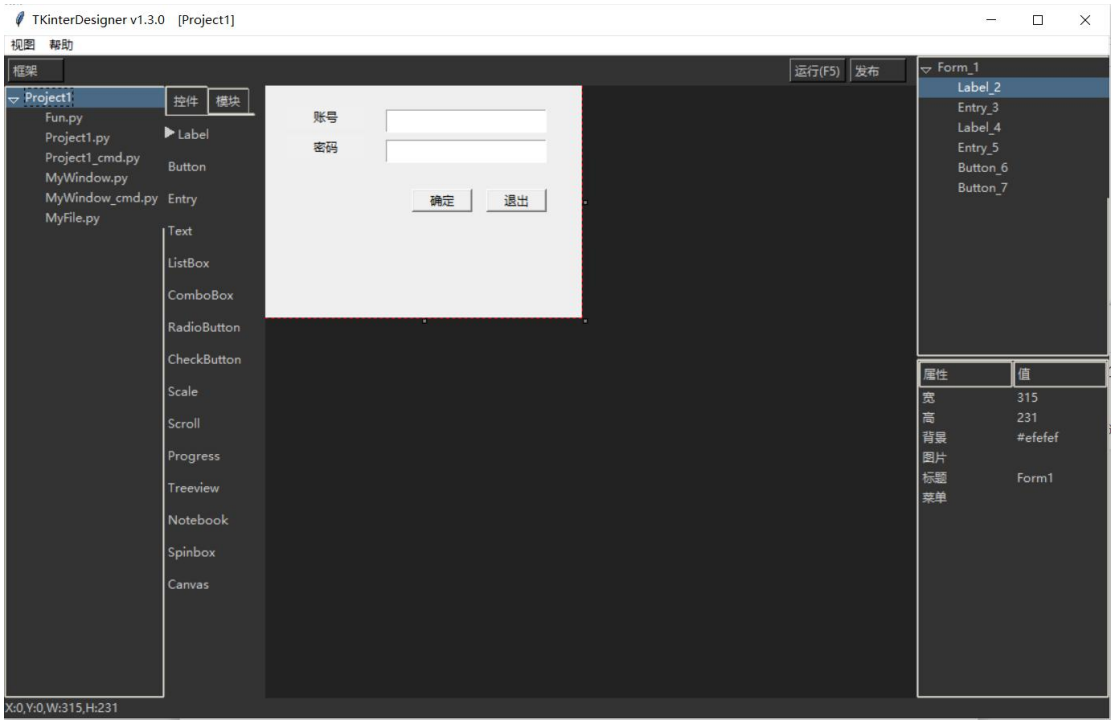
有时，你可能需要一些图片，声音，或者其它什么格式的文件资源放入到工程中，这里也可以通过在框架结构树右键弹出菜单项点击“导入资源”来选取并导入它。

最后，如果你想删除其中一个文件，你需要在相应的文件项上右键弹出菜单项点击“删除文件”，经过确认后，即可删除文件。

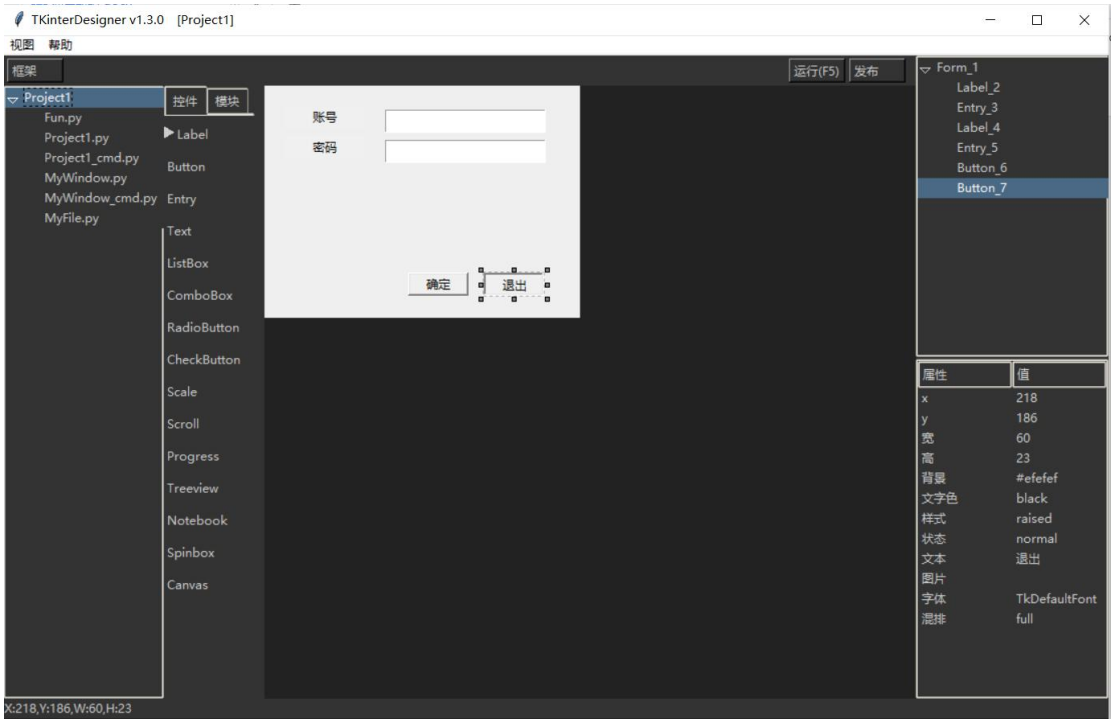
3.界面设计

首先，我们点击 Project1 或 Project1.py 进入到界面设计区域，然后我们就可以开始进行界面设计了，比如我们想对这个基本的账号，密码输入界面增加性别选项，职业分类，以及是否已婚进行输入，我们则需要新增加一些控件，包括两个 RadioButton,一个 ComboBox,一个 CheckButton 和所需的 Label 文字。这都是很常用的控件，比较有代表性。

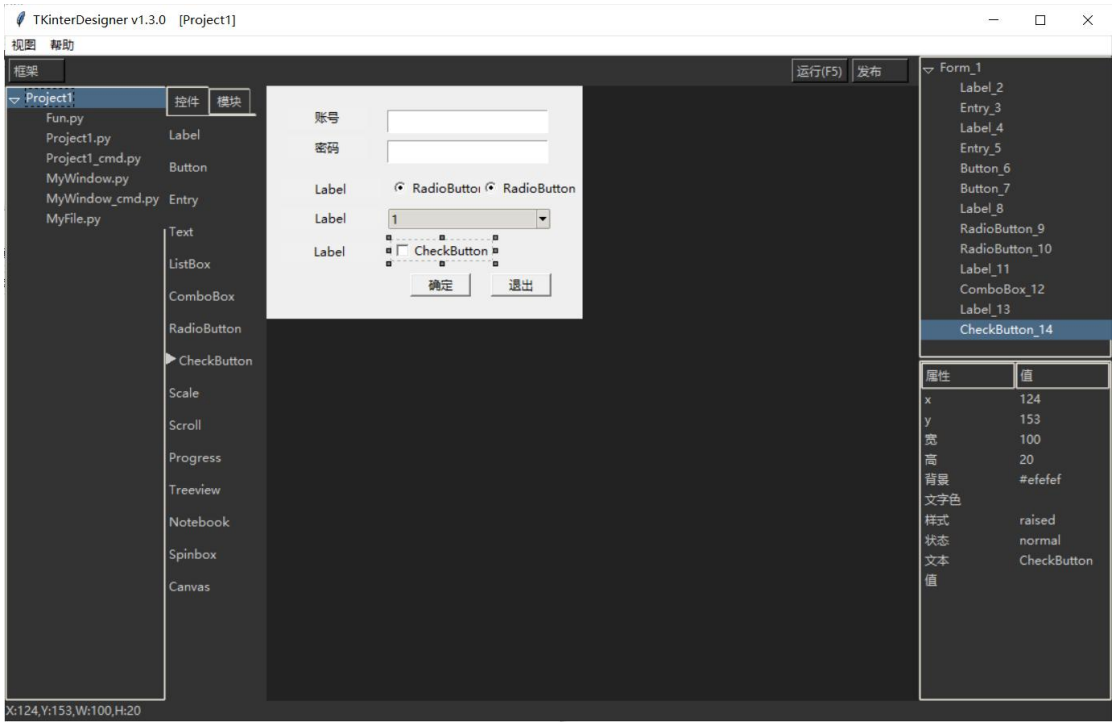
我们首先要对主窗体进行扩展，因为它的大小不太够，这时可以点击右上角的控件树项“Form_1”,或者直接点击设计区中的窗体界面，我们可以看到在窗体的四周，出现了虚线，并在顶点和边线中点位置出现了一个灰白色拖动块。我们可以用鼠标点中右下角的拖动块并拖动它到合适的大小。



我们完成好后，可以再将“确定”和“退出”两个按钮直接鼠标拖动到合适的位置。

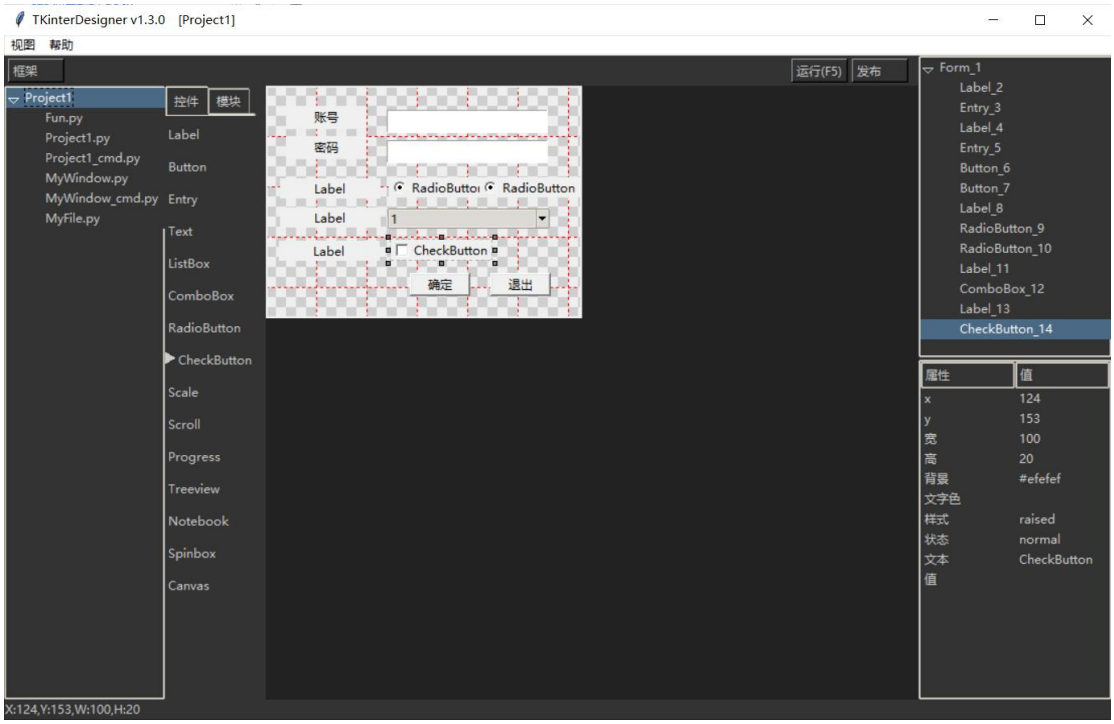


现在我们可以将需要的控件，一一从左边控件拖放区列表中选中并拖放到窗体中。



这里有个小技巧，就是如果你需要重复创建相同的控件，你可以直接在一个控件上选中，并在按下 ALT 键的状态下用鼠标拖动，将可以直接复制出一个控件供你拖动操作。

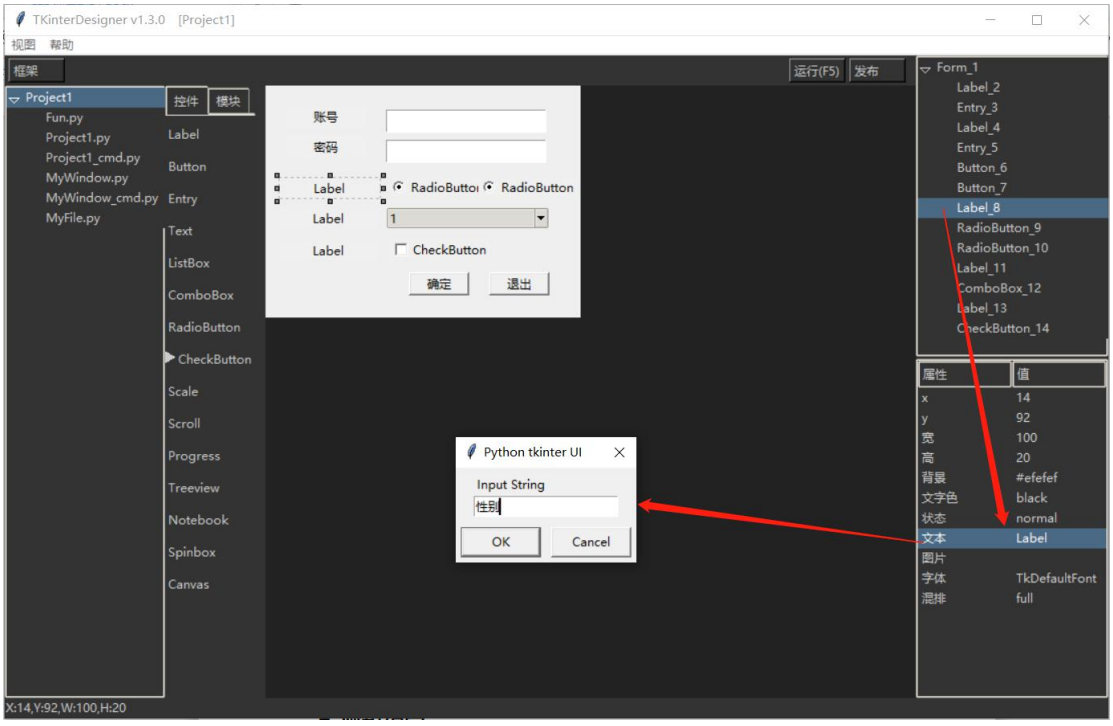
如果你觉得位置不是很好对齐，你可以在主菜单的“视图”项下面选择“网格”和“吸咐”，也可以通过 Ctrl+G,Ctrl+D 来快速调用显示或取消，网格是每 10 像素单位的，方便你进行吸咐后拖动对齐。



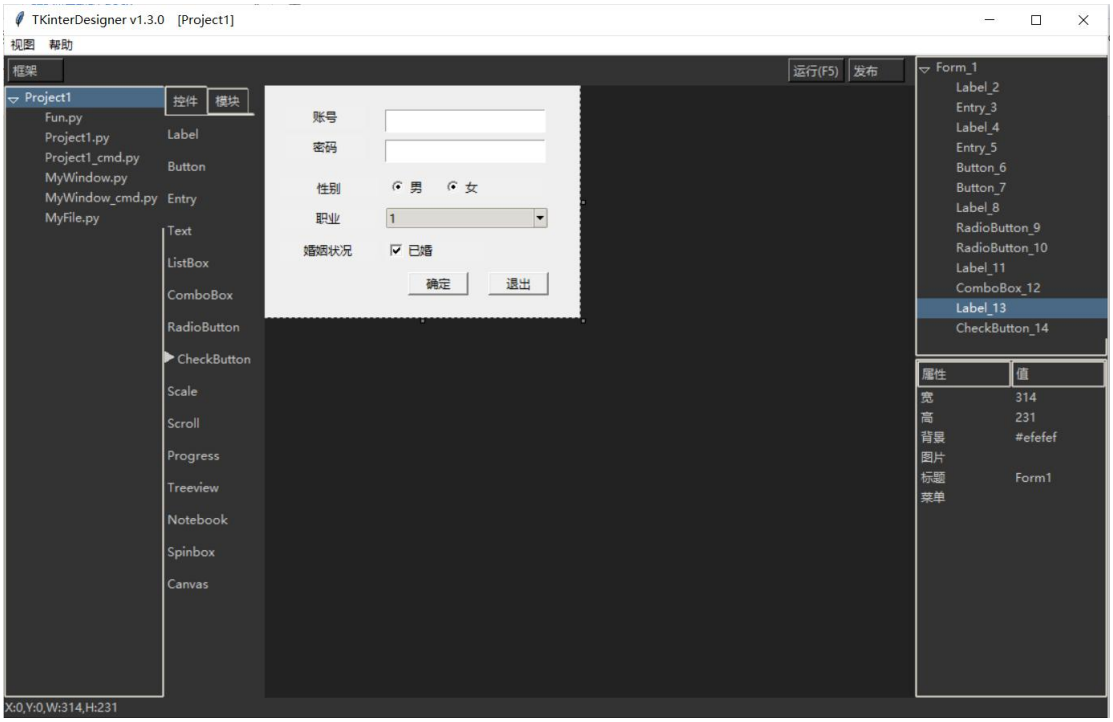
好，现在我们已经完成了所需要控件的创建部分，是不是非常简单？
下面我们来对这些控件进行设置。

4. 控件设置

选中性别单选按钮之前的 Label，然后我们在右边的属性框里找到“文本”属性，双击它，在弹出的对话框里，输入“性别”，点击“OK”即可对相应 Label 的文本进行更改。

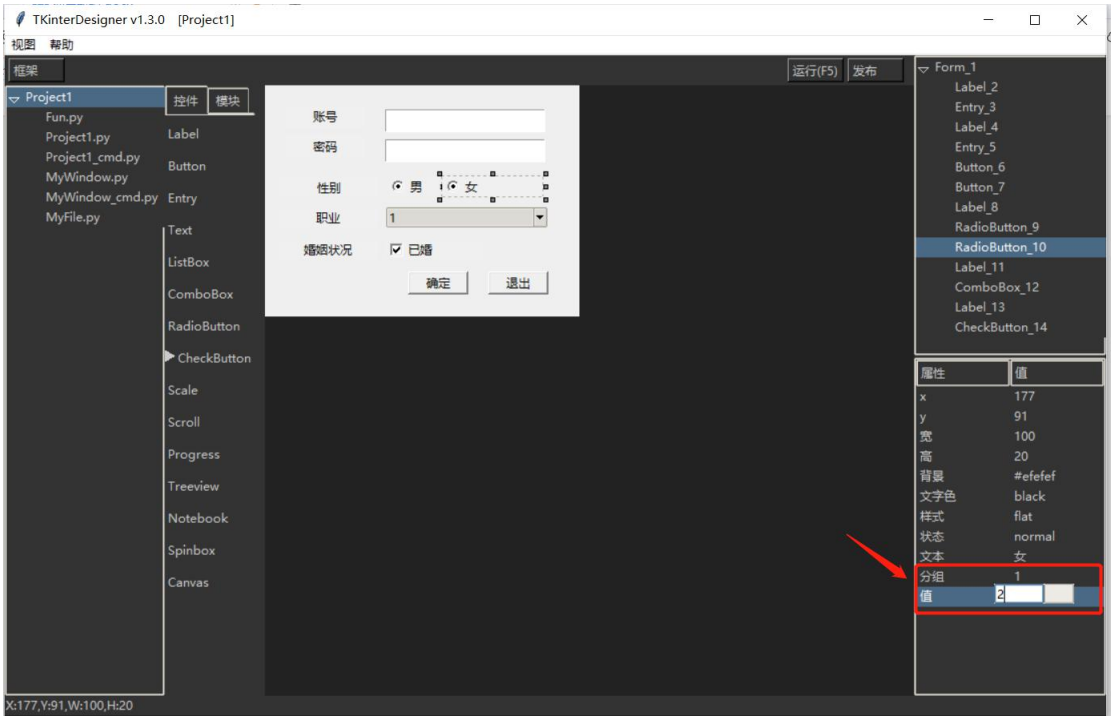


于是，很快我们完成了所有 Label，两个 RadioButton 的文本和 CheckButton 的文本设置。



看起来还不错是不是？ 当然，可以设置的属性很多，比如你可以修改背景色和文字色，也可以加入图片进行混排，或者修改字体等等。

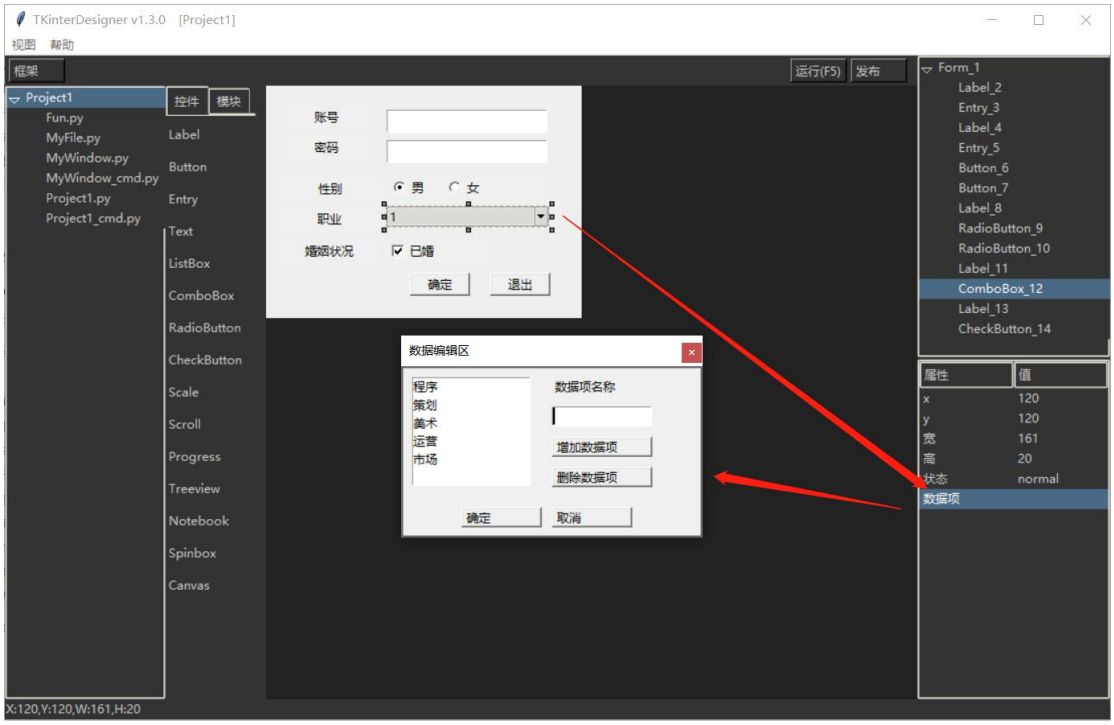
当然，性别不可能同时为男和女，这样的双性人种是不符合我们的取向的，对吧？我们选中文字为“女”的 RadioButton，在分组和值这一栏，双击值项，在输入框里改为 2，然后点击按钮，这时我们可以看到正确的 RadioButton 分组。



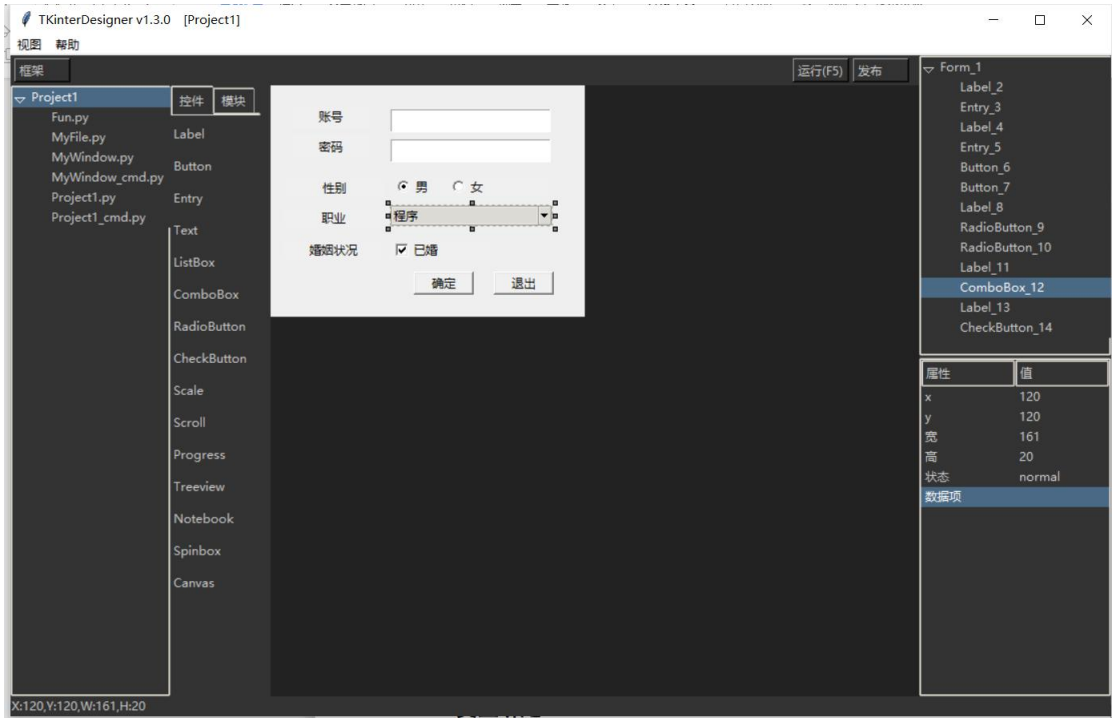
这是怎么回事？首先我们要清楚，RadioButton 在一个界面上可以有许多个，它们可能

某几个针对一个选项，比如性别，某几个又针对另一个选项，比如居住在本市的几个区域，这两部分需要归到两个组中，而且它们在每个组中需要有唯一的值进行区分。所以这两个代表“男”和“女”的单选按钮在使用默认分组号 1 的情况下，只需要修改“女”对应 RadioButton 的值为 2 即可。

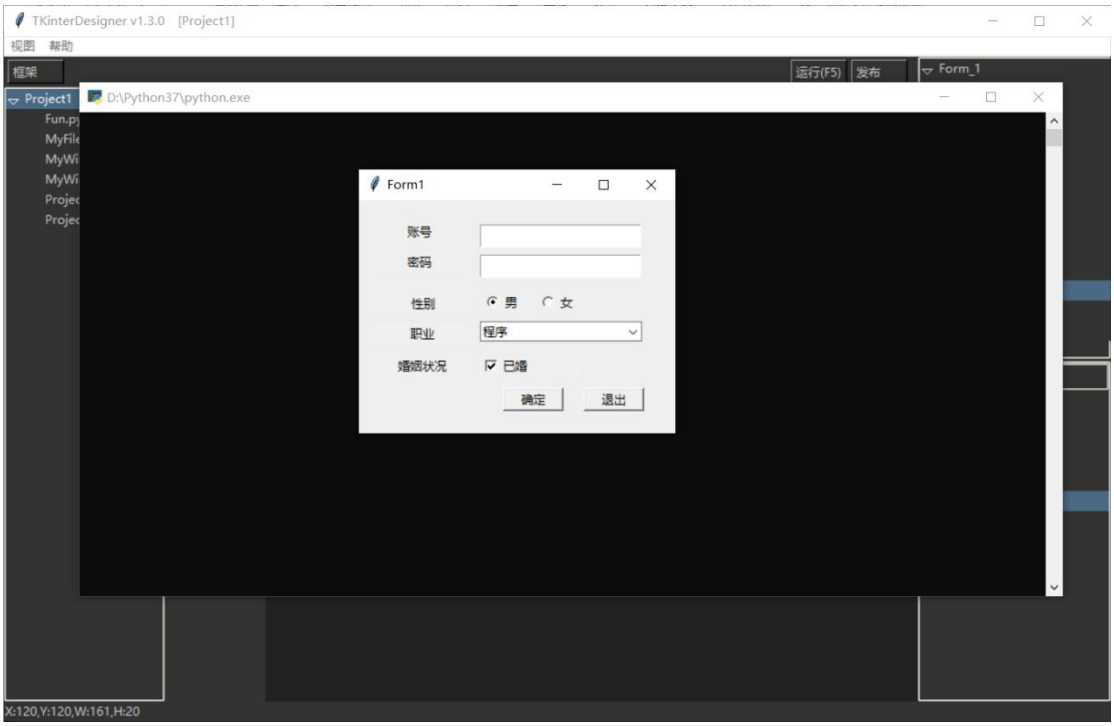
下面我们为输入增加职业选项，我们选中 ComboBox,在其属性框中找到“数据项”，双击后，在弹出的数据项编辑区对话框，即可编辑 ComboBox 对应的数据项。



比如我们输入“程序员”，“策划”，“美术”，“运营”，“市场”五个数据，点击“确定”后，我们可以看到 ComboBox 变成了想要的样子。



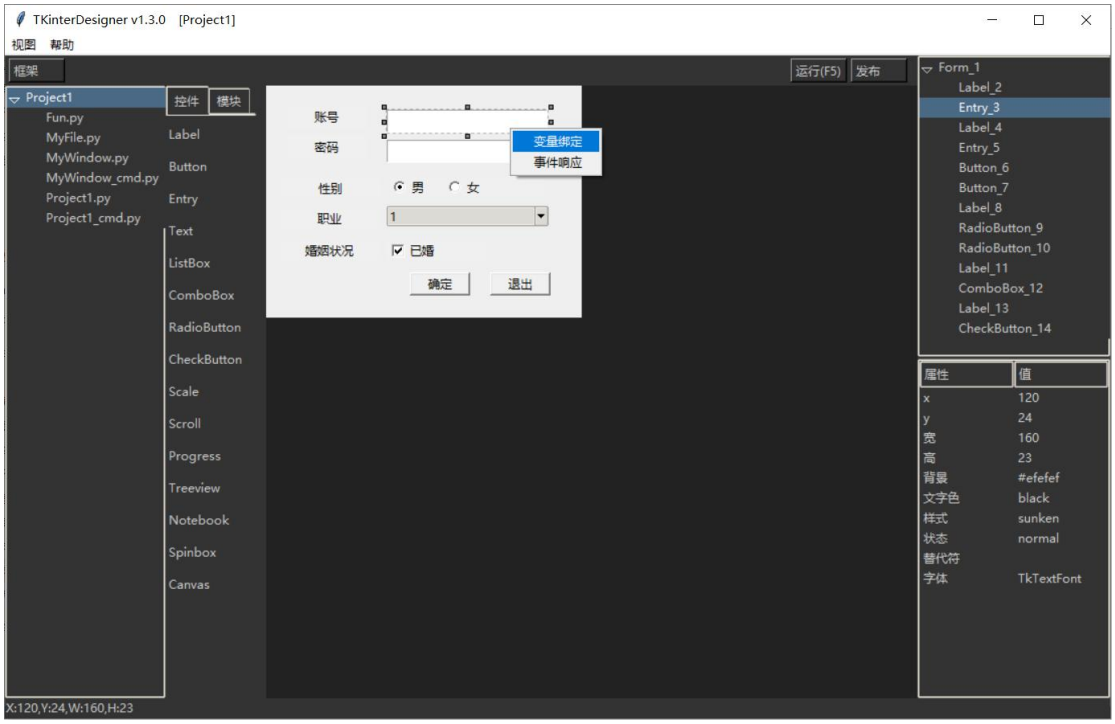
OK，点击一下“运行”或按 F5，这将会自动保存设计和代码，并编译运行结果。



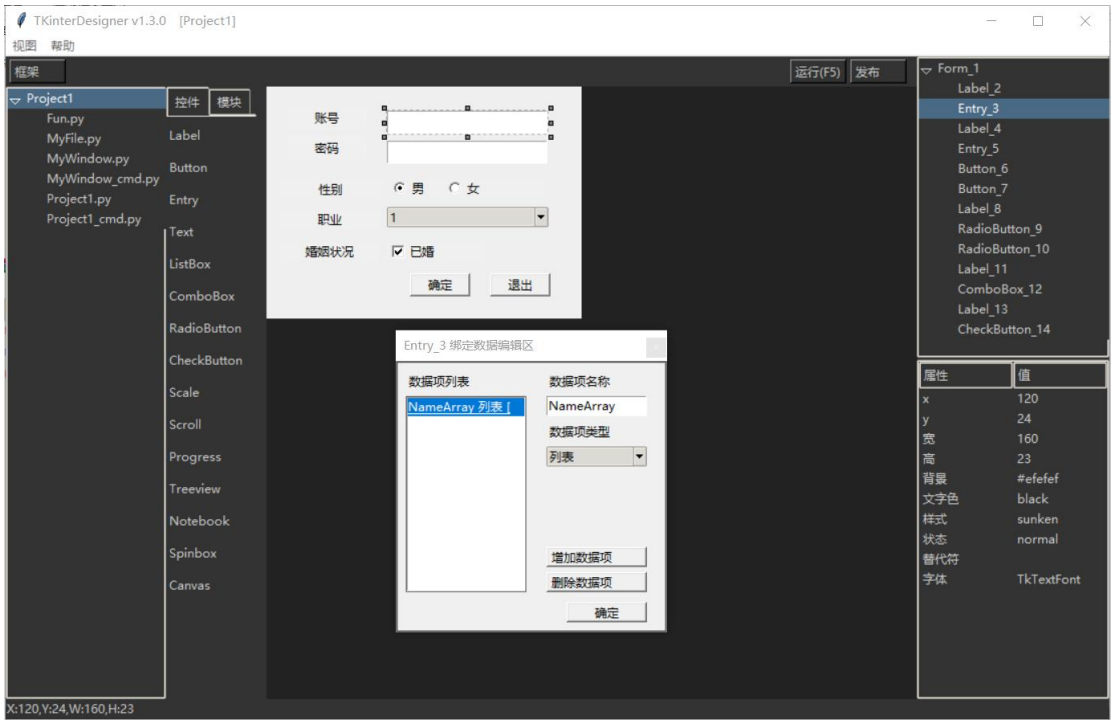
本实例演示了几个常见控件的属性设计和使用，其它的控件您可以自行尝试，在这里不再赘述。

5. 变量绑定

在开发中，我们经常需要存储一些数据，也许只是简单的结果存储，也许是控件的输入值，比如在实例项目 JSQ 的开发中，我们为显示数据的 Label 绑定一个临时变量存储中间值，以方便进行加减乘除的操作。我们在上面这个实例中，假设在点击“确定”时判断账号和上一次输入值相同时，我们弹出对话框提示“账号已用”，我们可以为账号增加一个自定义变量绑定一下。



我们在账号对应的输入框上用鼠标右键点击一下，在弹出的菜单中点击“变量绑定”，在弹出的对话框里，我们输入要绑定的数据项名称“NameArray”，选择为“列表”类型，我们如果是使用数字类型或字符串类型，可以看到有一个“映射到’text’”的选项框，点选这一项将意味着这个变量在调用 Fun.setUIData 函数进行设置时将同时将变量更新到 Label 或 Entry 控件的文本。这个变量对于同一个控件只允许有一个，如果未点选这一项，你将可以为一个控件创建多个变量。在这里我们不需要用就不点选。

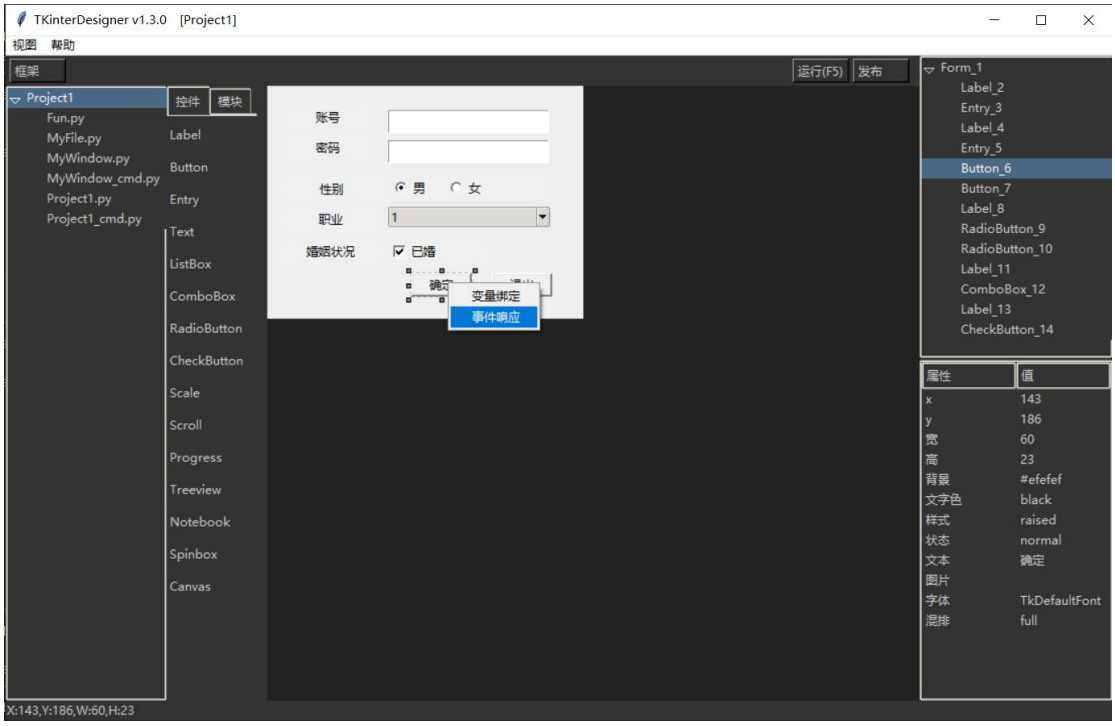


创建完成后，输入框将有一个绑定列表变量，你可以通过 Fun.getUIData(‘Entry_3’,’NameArray’)随时获取它。

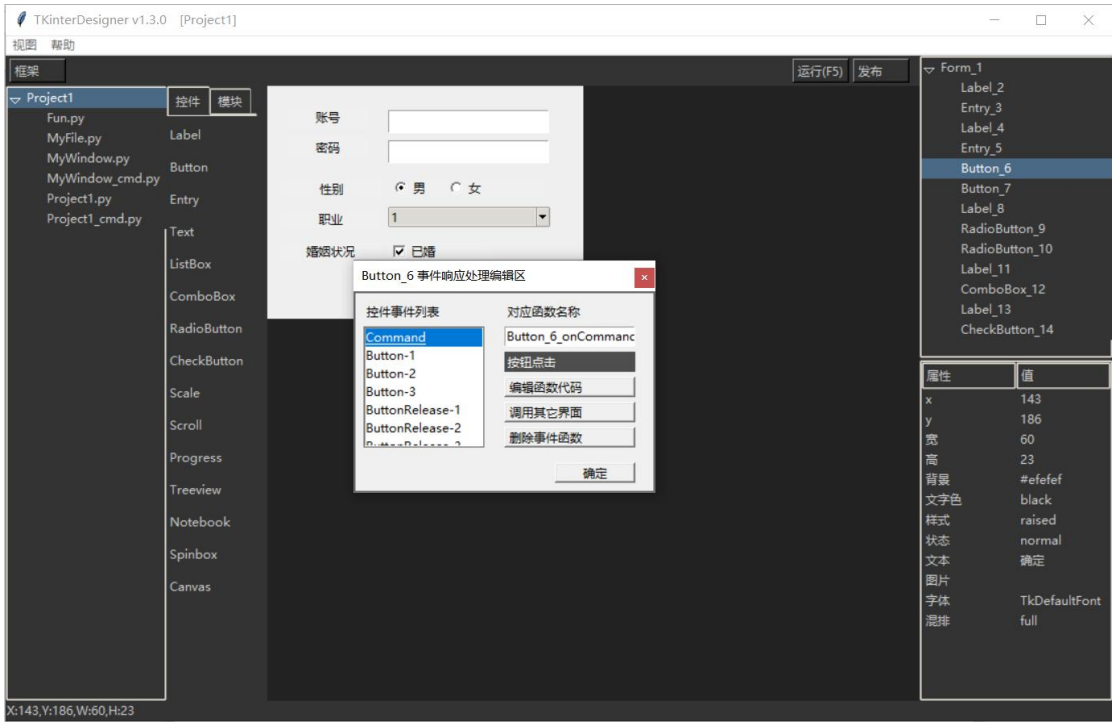
下面，我们来处理点击“确定”按钮时做相关判断，这就需要为“确定”按钮增加一个 Command 事件映射。

6. 事件响应

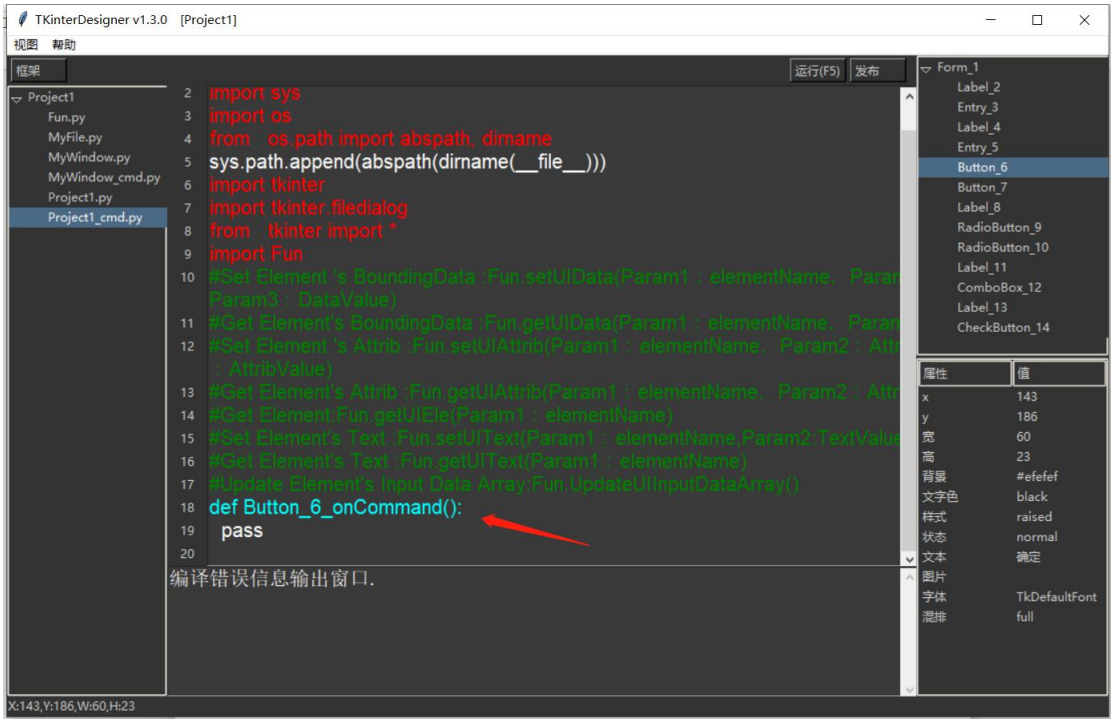
所谓事件响应，就是对于控制可以绑定的事件，做一个函数映射，使其触发对应事件时，调用设置的函数。



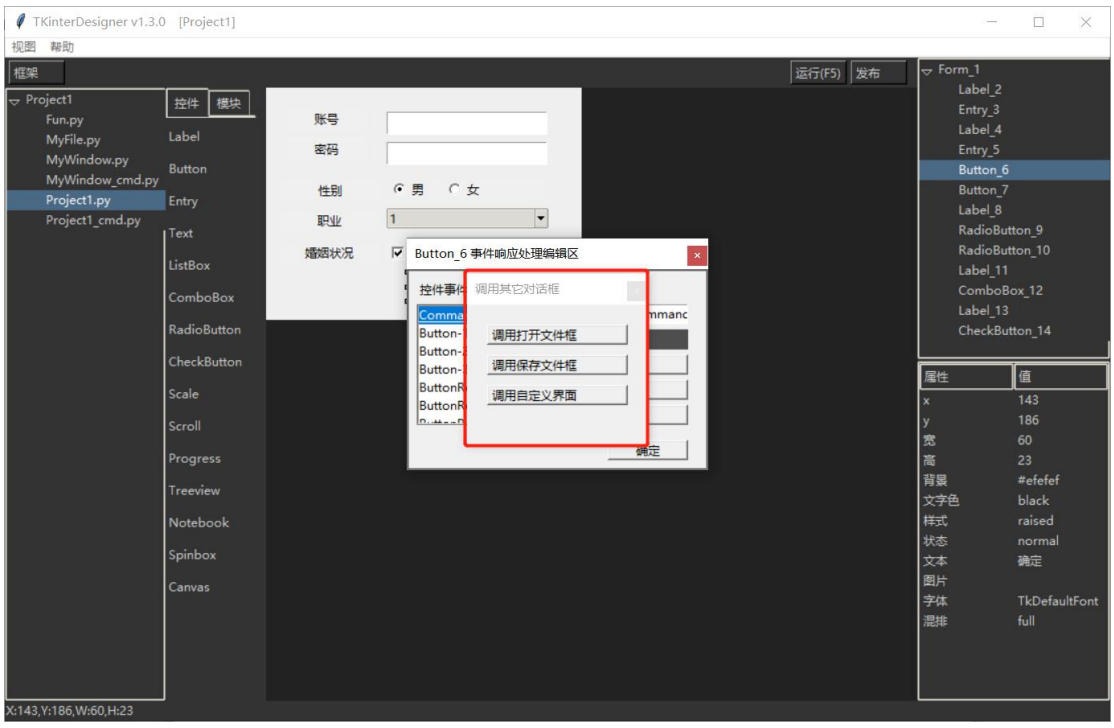
我们在“确定”按钮上用鼠标右键点击一下，在弹出的菜单中选择“事件响应”。



在弹出的事件响应处理编辑区，我们可以看到左边有一个事件列表，罗列了常用的 Python 事件，在右边有一个输入框，显示了默认的函数名称，我们也可以修改它，点击“编辑函数代码”即可以直接进入逻辑文件的代码编辑区，这时，我们可以看到已经增加的事件响应函数，我们可以在这个函数里手动进行代码的编辑。



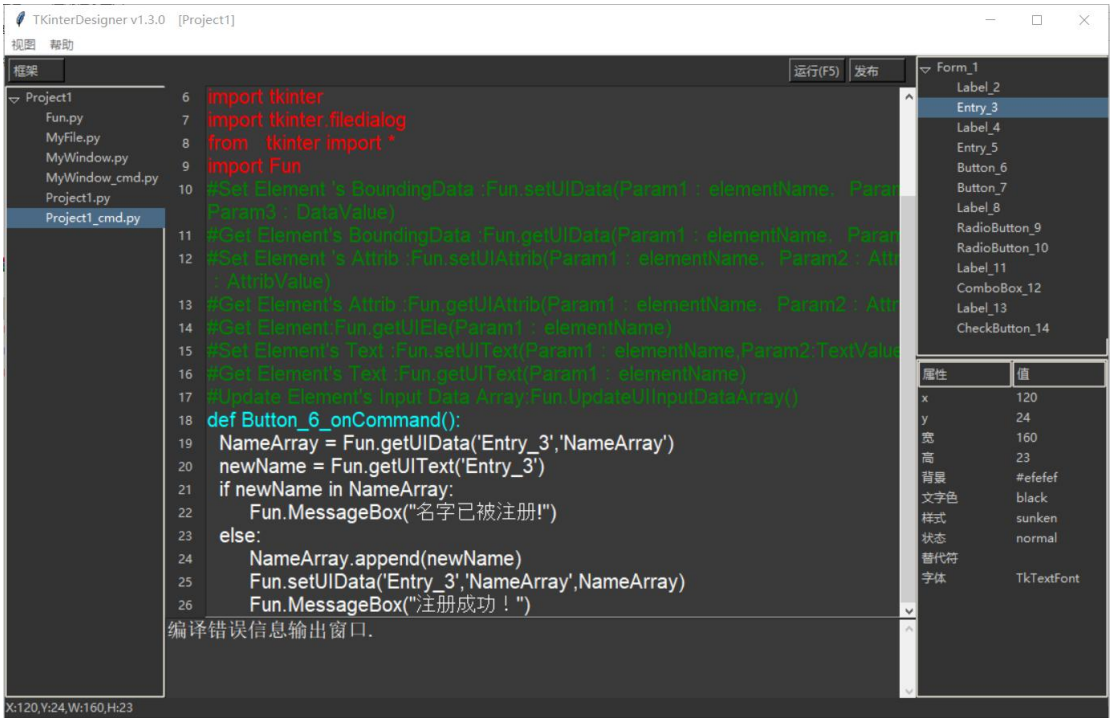
除此外，我们如果想在事件响应时调用其它界面，也可以点击“调用其它界面”按钮，这时会弹出一个选项对话框，由我们根据需要进行调用。



最常用的打开和保存文件框，在这里都可以直接选择，但如果你创建的是多窗口程序，你需要在这里调用另一个窗口，就选择“调用自定义界面”来查找它的 Py 文件进行调用即可。我在实例工程里有一个 CallTest 工程对此做了演示。

7. 逻辑编写

在逻辑文件的代码编辑区的 Button_6_onCommand 函数内，我们可以编写以下代码：

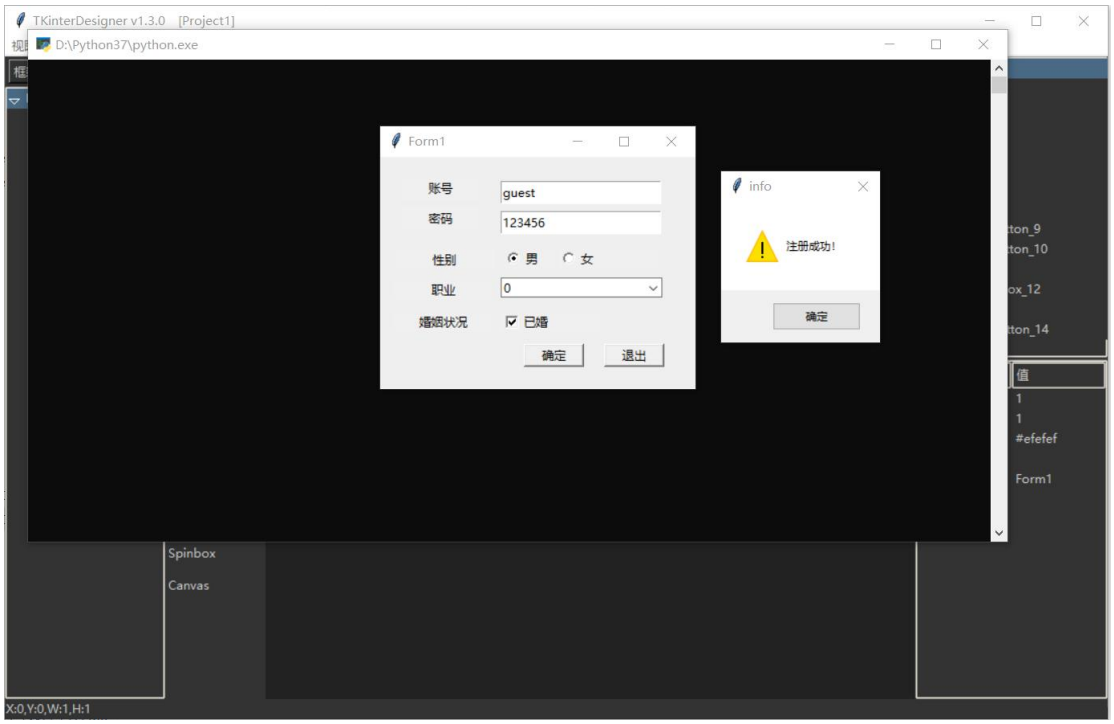


这段代码通过 Fun.getUIData 对于输入框绑定的列表变量进行了获取，并通过 Fun.getUIText 直接获取当前输入值，然后进行对比，如果相同，弹出对话框“名字已被注册”，如果不同，将输入值加入到列表变量中，并弹出对话框“注册成功”。

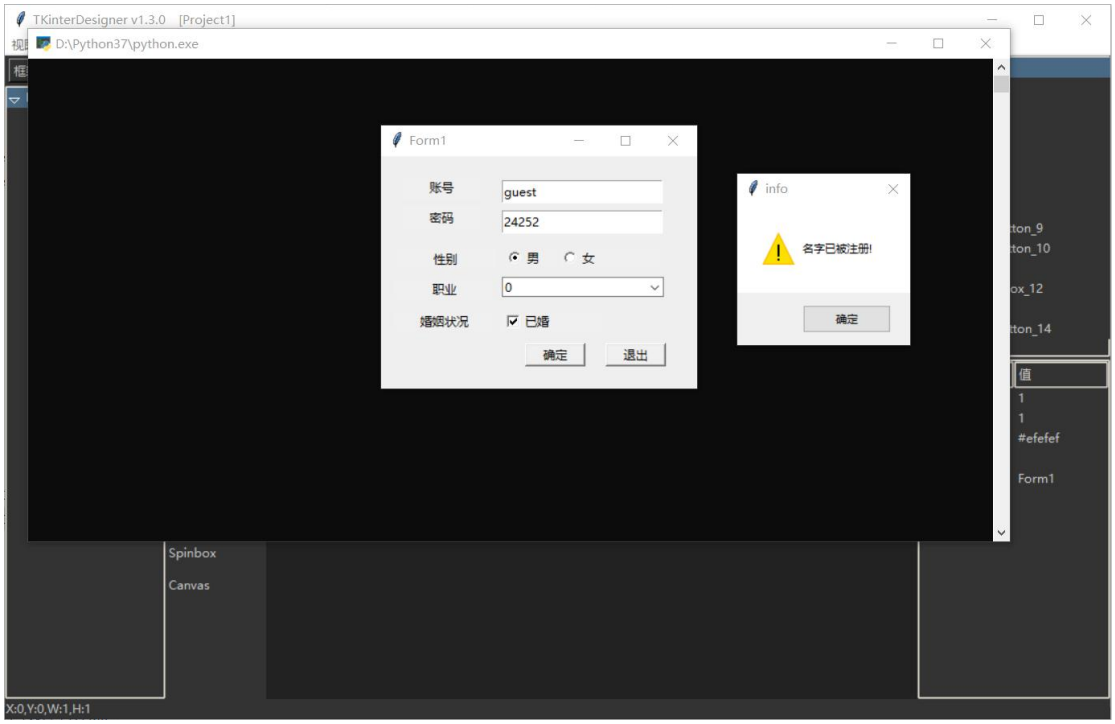
8. 编译运行

开发到这里，你一定想立刻编译运行，那就直接按 F5 或者点击右上部的“运行 F5”按钮，程序代码将会自动保存并开始编译运行。如果代码有错误，将会在代码区的编译错误信息输出窗口显示出来，如果你在函数中加入 print 打印，也会实时显示出来。

我们在运行的程序的账号里输入 guest 名称，第一次点击“确定”，会弹出“注册成功”对话框。



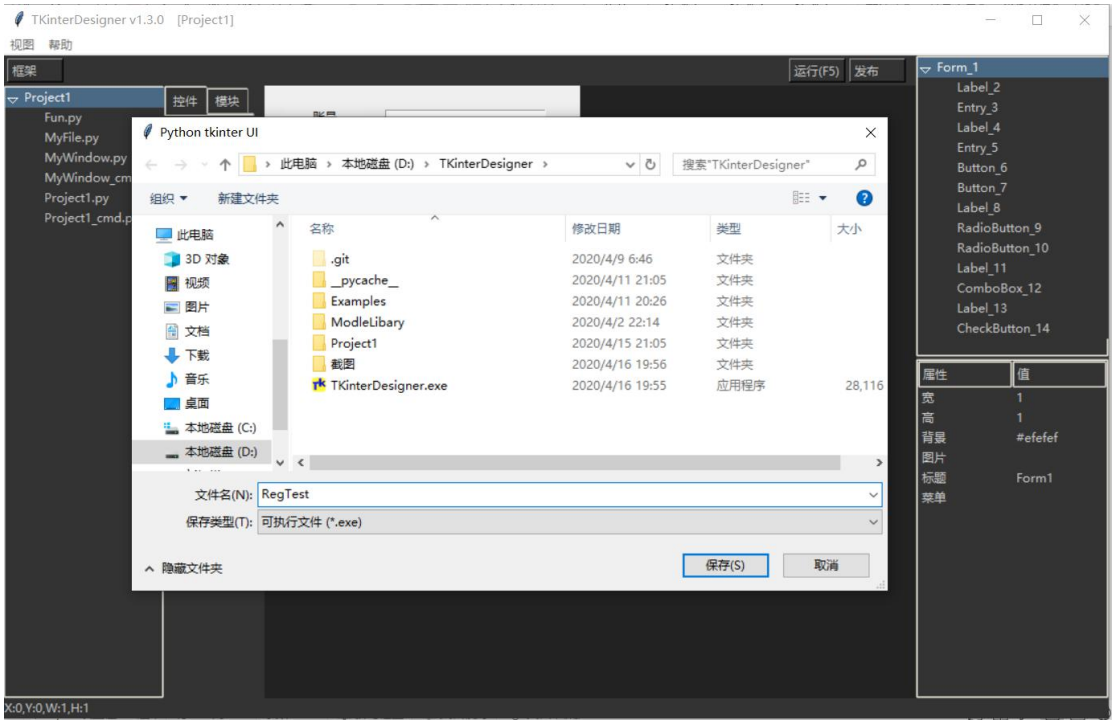
然后我们再次点击确定，则弹出“名字已被注册”对话框。



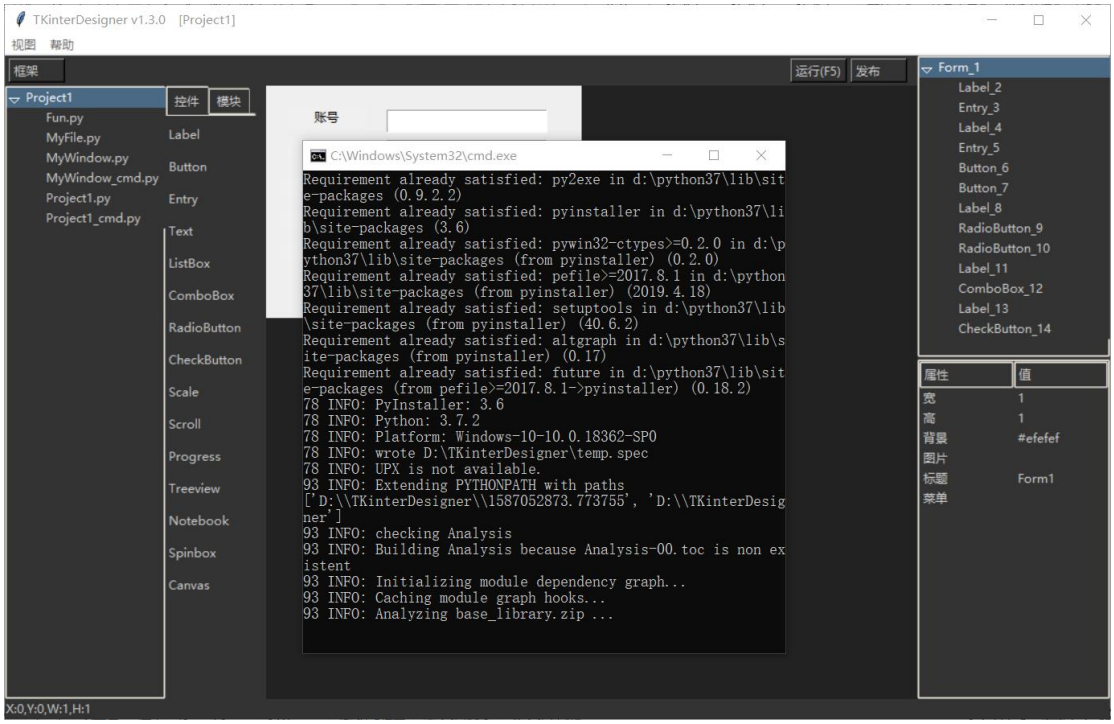
我们换个名字仍然可以注册成功，看起来一切都和我们期望的一样。

9. 打包 EXE

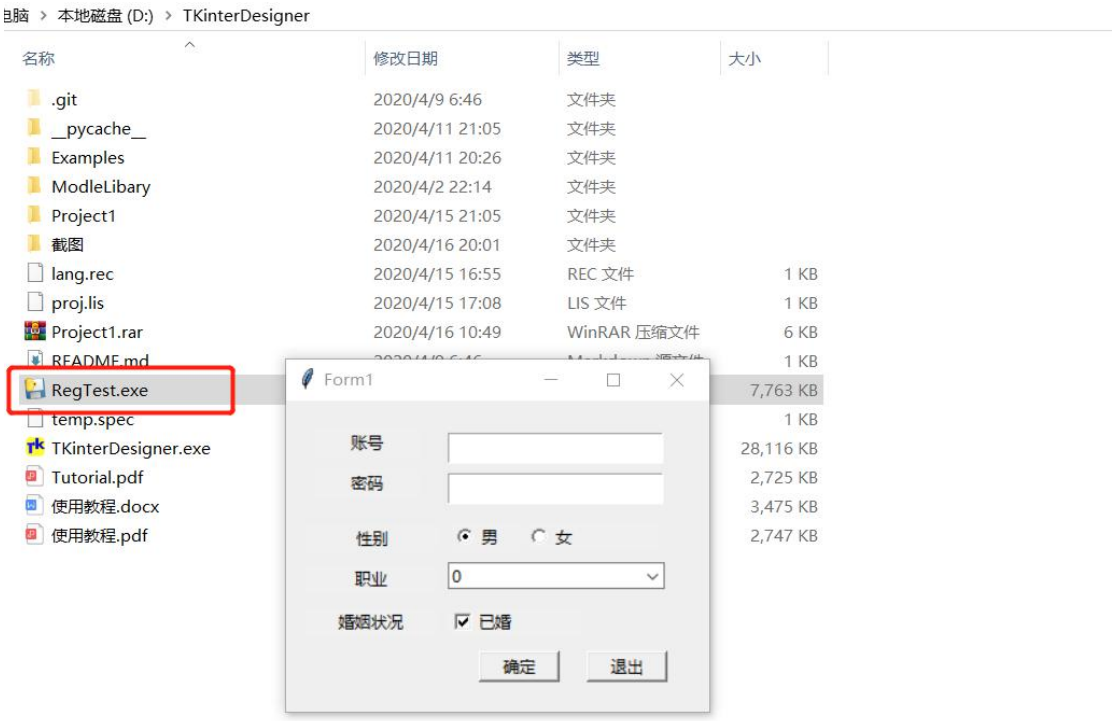
我们在完善好自己的程序后，我们希望将程序打包成 EXE 发布给使用者，我们可以直接点击右上部的“发布”按钮，在选择好输出目录后，输入要打包的 EXE 名称。



点击“确定”按钮，TKinterDesigner 会开始调用打包程序对项目进行打包。



顺利的话，最终可以在输出目录中找到对应的 EXE 程序：



10. 自定义模块导入

自定义模块我放在最后，是因为你并不是必须去用，但它可以很方便的对你的项目进行扩展，我在实例项目 Express_Query 和 Server 中都用到了它。

简单的说，你可以编写一个自定义的模块类提供给多个项目去使用，你可以方便的在设计器中对模块类进行属性设置，包括将界面控件作为一个参数丢给它。

我们以实例项目 Express_Query 的开发过程来进行简单的讲解。

首先，我们创建需要一个空白项目，并在框架文件树项上右键，在弹出菜单中产建一个 Python 文件，然后命名为 Express.py，在这个文件中，我们需要创建一个 Express 类，用于通过关键字来查询快递的结果。这个类的完整代码如下：

```
iimport urllib.request
import json
import msvcrt
import tkinter


class   Express:
    def __init__(self):
        self.Company_Dict = {1:'shentong',2:'youzhengguonei',3:'yuantong',4:'shunfeng',5:'yund
a',6:'zhongtong',7:"tiantian",8:"debang"}
        self.CompanyID = 4
        self.ExpressNumber = '0000001'
        self.ComboBox = None
```

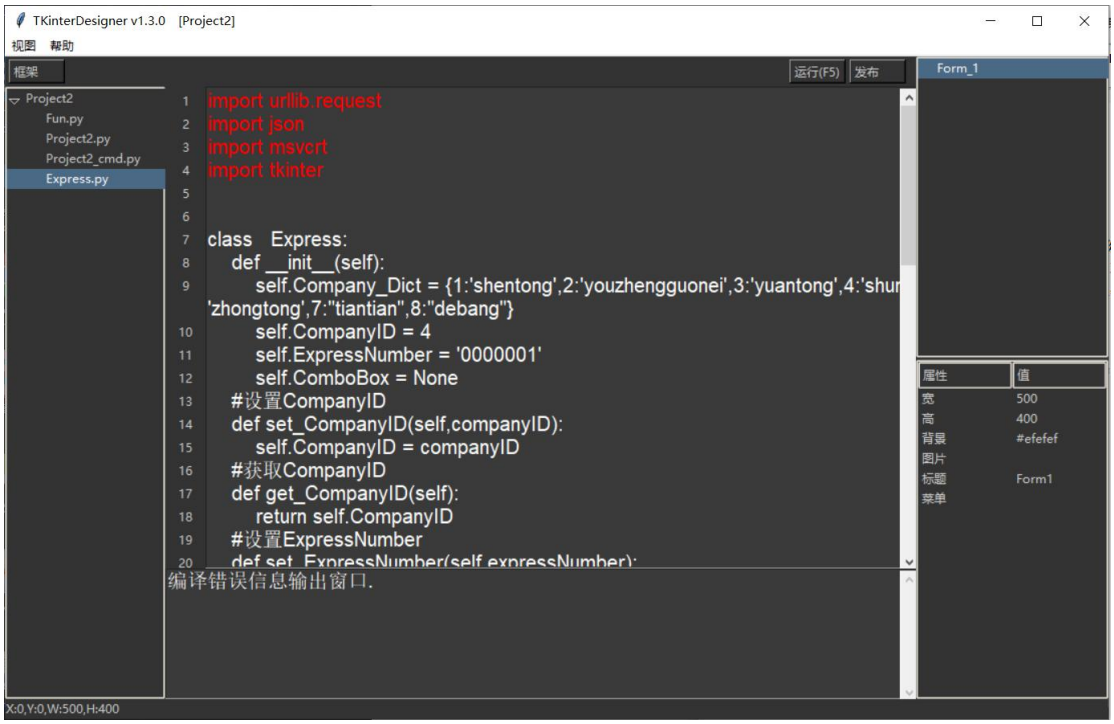
```
    #设置 CompanyID
    def set_CompanyID(self,companyID):
        self.CompanyID = companyID
    #获取 CompanyID
    def get_CompanyID(self):
        return self.CompanyID
    #设置 ExpressNumber
    def set_ExpressNumber(self,expressNumber):
        self.ExpressNumber = expressNumber
    #获取 ExpressNumber
    def get_ExpressNumber(self):
        return self.ExpressNumber
    #设置 ComboBox
    def set_ComboBox(self,comboBox):
        self.ComboBox = comboBox
        self.ComboBox['values'] = ['申通快递','EMS 邮政','圆通快递','顺丰快递','韵达快递','中通快递
','天天快递','德邦快递']
        self.ComboBox.current(4)
    #获取 ComboBox
    def get_ComboBox(self,comboBox):
        return self.ComboBox
    #查询
    def Query(self,ListBox):
        self.CompanyID = self.ComboBox.current() + 1
        ListBox.delete(0,tkinter.END)
        url = "http://www.kuaidi100.com/query?type=%s&postid=%s" % (self.Company_Dict[self.Com
panyID], self.ExpressNumber)
        response = urllib.request.urlopen(url)
        html = response.read().decode('utf-8')
        target = json.loads(html)
        #print(target)
        status = target['status']
        if status == '200':
            data = target['data']
            #print(data)
            data_len = len(data)

            for i in range(data_len):
                time_text = "时间: " + data[i]['time']
                ListBox.insert(tkinter.END,time_text)
                state_text = "状态: " + data[i]['context']
                ListBox.insert(tkinter.END,state_text)
        else:
            ListBox.insert(tkinter.END,"查询出现错误")
```

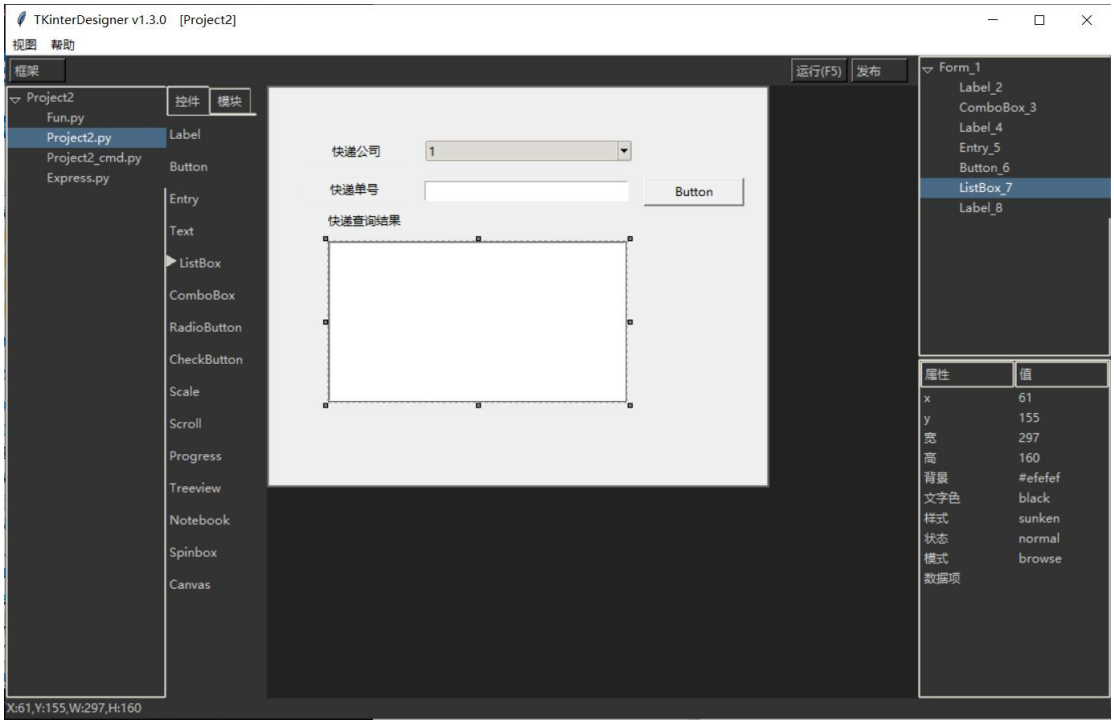
自定义模块类的写法要求只有一个：**如果你希望在设计器中传参数给它，你需要使用 set_和 get_为前缀的变量存取方法。**所以这里共对三个变量进行了相关的函数设计，包括快递公司的 CompanyID，快递单号 ExpressNumber，以及我们希望传入一个控件 ComboBox 来接受公司名称列表。

如果你觉得目前的工程代码编写很不方便，你也可以用 VSCode 或喜欢的代码编辑器来编写代码，或者直接将 Express.py 从实例项目中导入或拷贝过来。

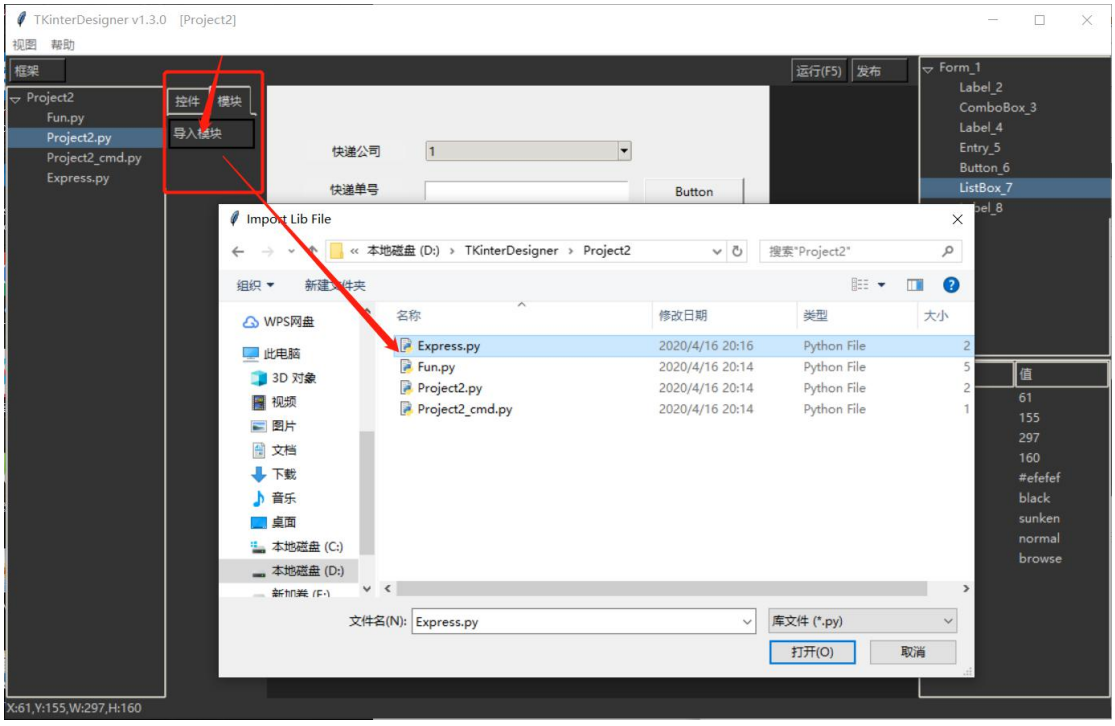
总之，你有这个代码就好。



回到主界面设计区，快速搭个界面：

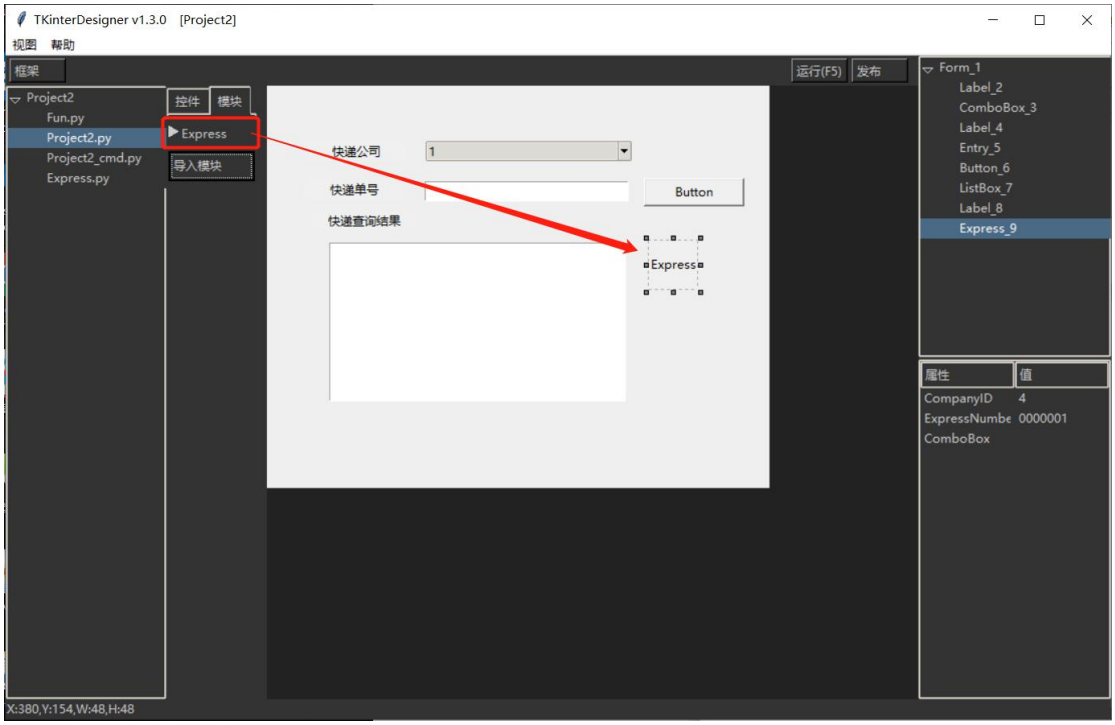


现在我们完成了界面的设计工作，我们在左边的“控件”和“模块”列表选择区域，切换到“模块”选项。

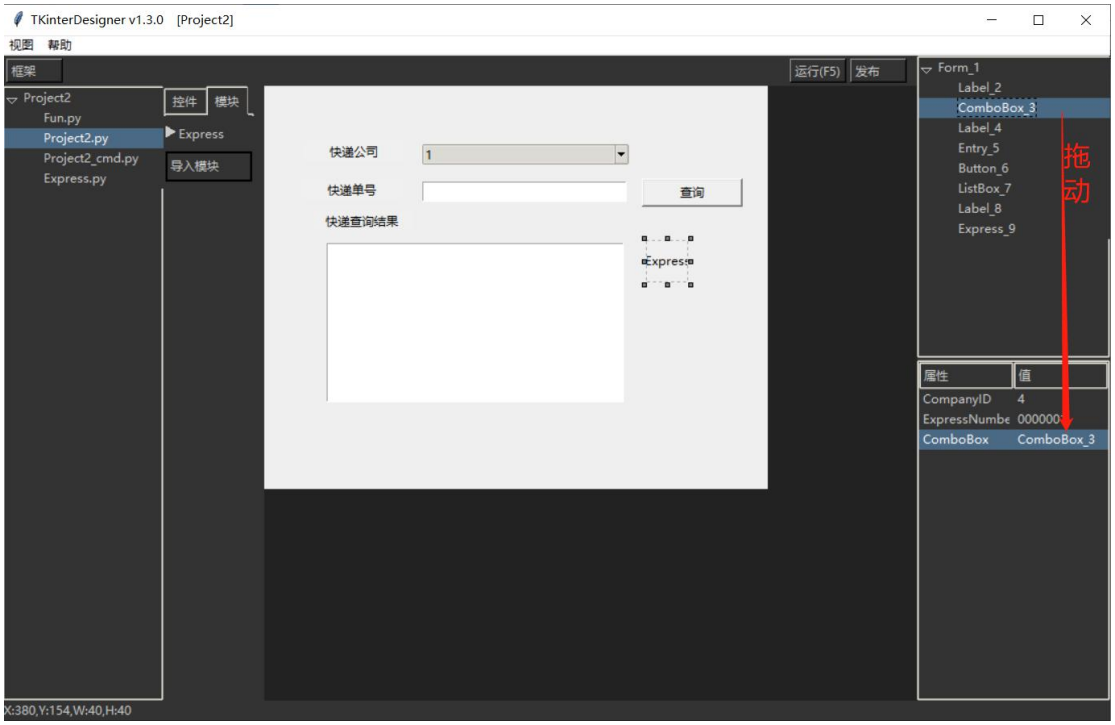


在“模块”选项面板里，我们点击“导入模块”按钮，然后找到 Express.py，点击“打开”。这里要注意：其实 Express.py 并不是必须要在当前项目目录下哦，你可以为多个项目使用到一个模块，只需要在这里导入就行了，并不需要每个使用到同一模块的项目都创建一个模块类文件。

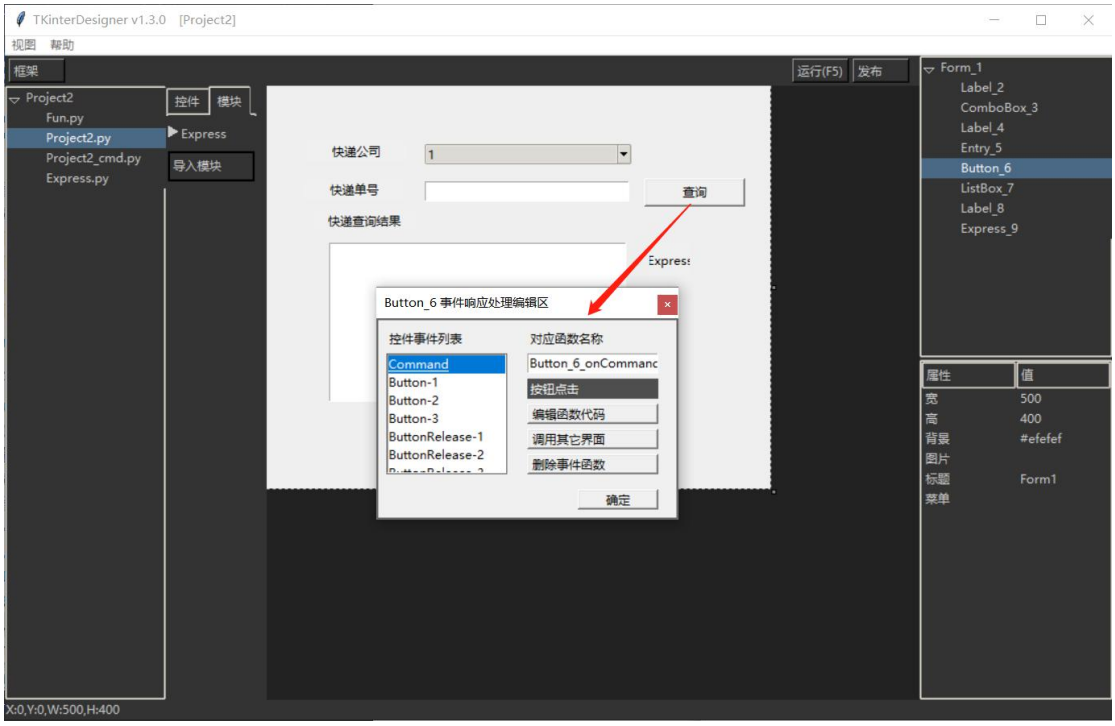
好，现在 Express 模块项出现在模块面板上了，我们拖动它放到界面上。



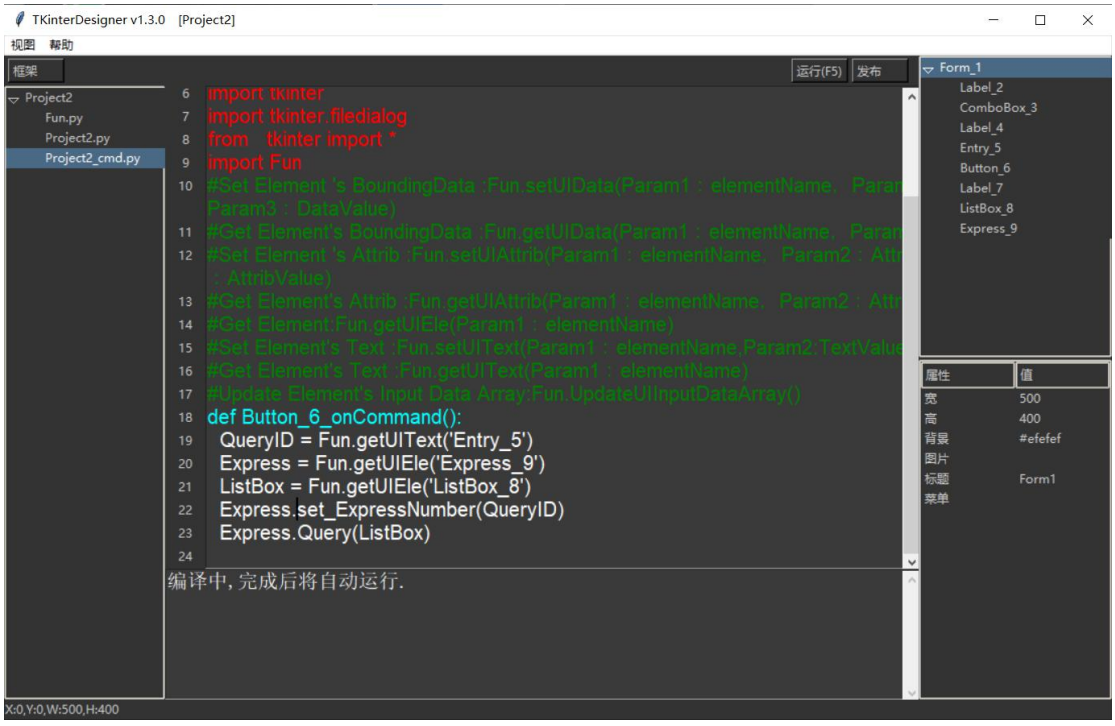
我们可以看到右下部分的属性框中，Express 模块的 3 个变量显示在属性框中，我们可以手动对 CompanyID 和 ExpressNumber 进行设置，但是 ComboBox 怎么设置呢？这里只需要在右上部的控件树下找到对应的 ComboBox_3,并拖动到属性框的 ComboBox 的值项位置即可。



然后我们为“查询”按钮增加 Command 的响应函数。

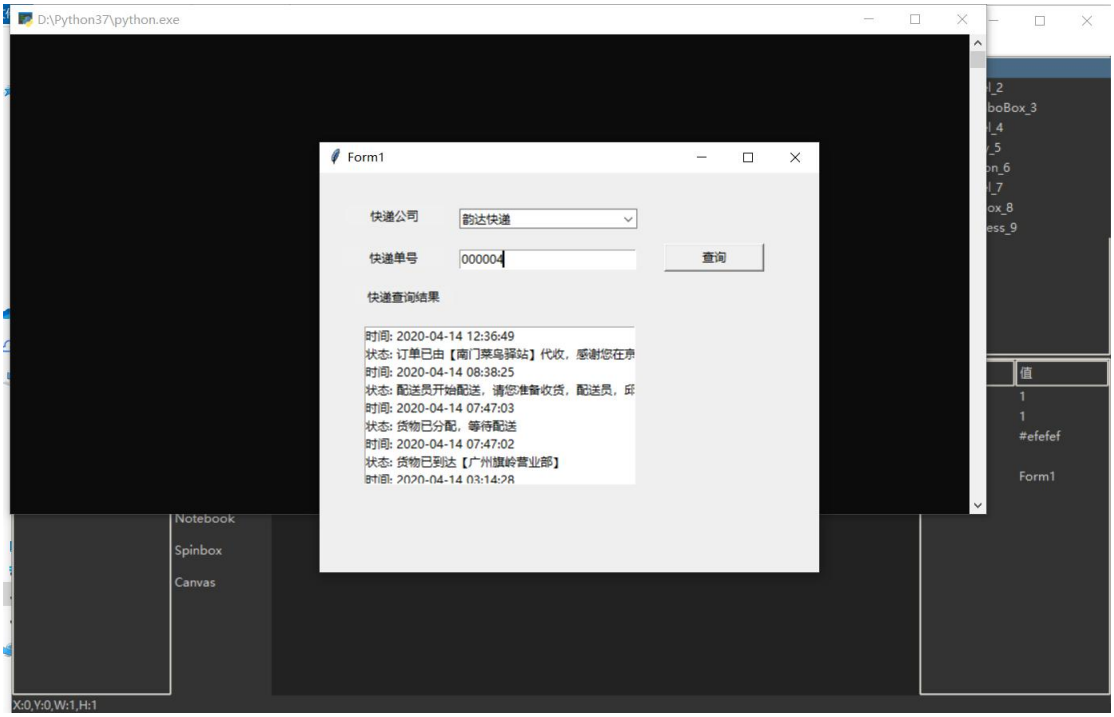


在 Button_6_onCommand 函数内我们可以编写相应代码：



这部分代码实现了获取 Entry_5 输入的快递单号，并通过模块名“Express_9”调用 Fun.getUIEle 称获取 Express 模块，并用同样的方法获取 ListBox_8 对象，然后调用 Express 的 set_ExpressNumber 方法，设置快递单号，最后调用 Query 方法进行查询，参数就是输出显示的 ListBox 对象。

代码就这么多，我们完成后按 F5 运行一下，我们可以看到运行的程序，在快递单号尝试输入数字后，点击“查询”，很快可以看到在列表框中显示出的快递信息。



这就是自定义模块的使用方法了。

作者介绍

网名：火云红孩儿

作品：红孩儿工具箱，CocosVR。

教程：《Cocos 引擎深入分析系列》，《Shader 从入门到精通系列》

成就：原无限世界引擎总监，原 COCOS 引擎总监，CSDN 博客专家，Cocos 最有价值专家。

联系方式：QQ：285421210

关于 TKinterDesigner

因为它只是我业余时间断断续续学了三个月 Python 的一点小动机，它有很多问题，不过我会继续完善的，在此希望有兴趣的 Python 爱好者与我交流，给我更多的建议，因为我没看完一本 Python 书，有太多的需求还不太了解，不过我很看好 Python 在快速化原型开发上的界面工具需求，希望 Python 越来越好。

最后，祝各位工作顺心，身体健康~

火云红孩儿

2020/04/17