

#

TKinterDesigner API Instruction

Author	Honghaier
Version	V1.6.3
Last Update Date	2021-12-05
Twitter:	honghaier_2020@Honghaier_game
Email:	285421210@qq.com
Twitter:	honghaier_2020@Honghaier_game

WebSite:www.tkinterdesigner.com

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

What's Fun.py?

Fun.py is a function library of TkinterDesigner, which provides common functions to access UI controls and their properties. At the same time, it also contains some functions encapsulated by common functions.

Most of the parameters of these functions start with uiname and elementname, which respectively represent the class name and control name of the interface. In the logic file corresponding to the interface, such as "project1_cmd. Py", if we add an event function for the control, tkinterdesigner will also generate corresponding parameters, such as:

```
def Button_ 6_ onCommand(uiName,widgetName):
    UserName=Fun.GetText(uiName,"Entry_3")
    Password=Fun.GetText(uiName,"Entry_5")
    Fun.MessageBox("UserName:"+UserName+" Password:"+Password)
```

In this function, we also pass similar parameters uiname and widgetname, corresponding to the class name and control name of the interface, which can be directly passed to the function in fun.py.

In fact, the uiname can be directly set to your corresponding interface class name, such as "project1", and the elementname can also be directly set to the control name you need, such as "entry_5"

Note: fun.py is created dynamically according to the needs of the project. It is not recommended to modify fun.py.

What are the functions of Fun.py ?

Tkinterdesigner v1.4.6 currently includes below functions:

1. **Register: Register a control in the control list.**
2. **GetElement: Access the control entity through the control name.**

3. **AddTKVariable:** Add a tkinter variable to control.
4. **SetTKVariable:** Set the tkinter variable value of the control.
5. **GetTKVariable:** Get the tkinter variable value of the control.
6. **AddUserData:** Add a user data to control.
7. **SetUserData:** Set the user data value of the control.
8. **GetUserData:** Get the user data value of the control.
9. **SetTKAttrib:** Set the tkinter attribute value of the control.
10. **GetTKAttrib:** Get the tkinter attribute value of the control.
11. **SetText:** Set the text of the control.
12. **GetText:** Get the text of the control.
13. **SetImage:** Set the image of the control.
14. **GetImage:**Get the image of the control.
15. **SetSelectIndex:**Set selected items for listbox and combobox.
16. **GetSelectIndex:**Get the selected items of listbox and combobox.
17. **InitElementData:**Initialize all control data.
18. **InitElementStyle:**Initialize all control style.
19. **UpdateUIInputDataArray:**Get all the entry data of an interface
20. **CenterDlg:**Center a pop-up interface dialog box
21. **SetRoundedRectangle:**Set the fillet properties of the control
22. **ShowRoundedRectangle:**Set the fillet property of the control immediately.
23. **SetTransparencyFunction:**Set the transparent value of the form.
24. **ExpandAllTreeItem:**Expand or close tree items.
25. **MessageBox:**Pop up a message dialog box
26. **InputDialog:**Pop up a input dialog box
27. **AskBox:**Pop up a question dialog box,You need choose yes or no.
28. **WalkAllResFiles:**Returns all the files in the directory
29. **EventFunction_Adaptor:**Redefine the event response function
30. **SetControlPlace:**Sets the absolute or relative position of the control
31. **SetRootRoundRectangle:**Set the fillet parameter of TK root, support cross platform.
32. **ReadFromFile:**Read content from file.
33. **WriteToFile:**Writing content to a file.
34. **ReadStyleFile:**Read style file

Function explanation:

1. **Register:** To register a control is to bind the name to the control entity. If you want to access the control entity by name, it must be registered. Here, the param 'uiName' is used to distinguish the controls belonging to which interface instance. Because there may be multiple interfaces in the project, the UI name is used to distinguish them.

Parameter 1: interface class name, parameter 2: control name, parameter 3: control.

```
def Register(uiName,elementName,element):  
    if uiName not in G_UIElementArray:  
        G_UIElementArray[uiName]={}  
        G_UIElementArray[uiName][elementName]=element
```

Example:

```
Fun.Register(className, 'Form_1', Form_1)
```

2. **GetElement:** Access the control entity through the control name.

Parameter 1: interface class name, parameter 2: control name

```
def GetElement(uiName,elementName):  
    global G_UIElementArray  
    if uiName in G_UIElementArray:  
        return G_UIElementArray[uiName][elementName]  
    return None
```

Example:

```
Entry_3 = Fun.GetUIEle(className,'Entry_3 ')
```

3. **AddTKVariable:** Add a tkinter variable to control.

parameter 1: interface class name, parameter 2: control name, parameter 3: default value.

```
def AddTKVariable(uiName,elementName,defaultvalue = None):  
    if uiName not in G_UIElementVariableArray:  
        G_UIElementVariableArray[uiName]={}  
        NameLower = elementName.lower()  
        if NameLower.find('combobox_') >= 0:  
            G_UIElementVariableArray[uiName][elementName]=tkinter.IntVar()  
        elif NameLower.find('group_') >= 0:  
            G_UIElementVariableArray[uiName][elementName]=tkinter.IntVar()  
        elif NameLower.find('checkboxbutton_') >= 0:  
            G_UIElementVariableArray[uiName][elementName]=tkinter.BooleanVar()  
        else:  
            G_UIElementVariableArray[uiName][elementName]=tkinter.StringVar()  
        if defaultvalue:  
            G_UIElementVariableArray[uiName][elementName].set(defaultvalue)  
    return G_UIElementVariableArray[uiName][elementName]
```

Example:

```
CheckBox_6_variable = Fun.AddTKVariable(className,'CheckBox_6')  
CheckBox_6_variable.set(False)
```

4. **SetTKVariable:** Set the tkinter variable value of the control.

Parameter 1: interface class name, parameter 2: control name, parameter 3: value.

```
def SetTKVariable(uiName,elementName,value):  
    if uiName in G_UIElementVariableArray:  
        if elementName in G_UIElementVariableArray[uiName]:  
            G_UIElementVariableArray[uiName][elementName].set(value)
```

Example:

```
Fun.SetTKVariable(className,'CheckBox_6',True)
```

5. **GetTKAttrib:** Get the tkinter attribute value of the control.

Parameter 1: interface class name, parameter 2: control name

```
def GetTKVariable(uiName,elementName):  
    if uiName in G_UIElementVariableArray:  
        if elementName in G_UIElementVariableArray[uiName]:  
            return G_UIElementVariableArray[uiName][elementName].get()
```

Example:

```
CheckButton_6_Variable = Fun.GetTKVariable(className,'CheckButton_6')
```

6. **AddUserData:** Add a user data to control. The parameter dataname is the data name and datatype is the data type, which can include int, float, string, list, dictionary, etc. generally, the control is operated by the right mouse button in the design software and set in the pop-up "bind data variable". The parameter datavalue is the data value, and isMapToText is whether to directly reflect the data to the text variable of the control

```
def AddUserData(uiName,elementName,dataname,datatype,datavalue,isMapToText):
    global G_UIElementUserDataArray
    if uiName not in G_UIElementUserDataArray:
        G_UIElementUserDataArray[uiName]={}
    if elementName not in G_UIElementUserDataArray[uiName]:
        G_UIElementUserDataArray[uiName][elementName]=[]
    G_UIElementUserDataArray[uiName]
    [elementName].append([dataname,datatype,datavalue,isMapToText])
```

Example:

```
Fun.AddUserData(className,'Label_7','AAA','int',0,0)
```

```
Fun.AddUserData(className,'Label_7','DDD','list',[],0)
```

7. **SetUserData: Set the user data value of the control.**

Parameter 1: interface class name, parameter 2: control name, parameter 3 dataname is the data name, and parameter 4 datavalue is the data value.

```
def SetUserData(uiName,elementName,dataname,datavalue):
    global G_UIElementArray
    global G_UIElementUserDataArray
    if uiName in G_UIElementUserDataArray:
        if elementName in G_UIElementUserDataArray[uiName]:
            for EBDData in G_UIElementUserDataArray[uiName][elementName]:
                if EBDData[0] == dataname:
                    EBDData[2] = datavalue
                if EBDData[3] == 1:
                    SetText(uiName,elementName,datavalue)
            return
```

Example:

```
Fun.AddUserData(className,'Label_7','AAA',888)
```

8. **GetUserData: Get the user data value of the control.**

Parameter 1: interface class name, parameter 2: control name, parameter 3 dataname is the data name.

```
def GetUserData(uiName,elementName,dataname):
    global G_UIElementUserDataArray
    if uiName in G_UIElementUserDataArray:
        if elementName in G_UIElementUserDataArray[uiName]:
            for EBDData in G_UIElementUserDataArray[uiName][elementName]:
                if EBDData[0] == dataname:
                    if EBDData[1]=='int':
                        return int(EBDData[2])
                    elif EBDData[1]=='float':
```

```

        return float(EBData[2])
    else:
        return EBData[2]
    return None

```

Example:

```
aaa = Fun.GetUserData(className, 'Label_7', 'AAA')
```

9. SetTKAttrib: Set the tkinter attribute value of the control.

Parameter 1: interface class name, parameter 2: control name, parameter 3: attribname is the property name, and parameter 4: attribvalue is the data value.

```

def SetTKAttrib(uiName, elementName, AttribName, attribValue):
    global G_UIElementArray
    if uiName in G_UIElementArray:
        if AttribName in G_UIElementArray[uiName]
[elementName].configure().keys():
            G_UIElementArray[uiName][elementName][AttribName]=attribValue

```

Example:

```

Fun.SetTKAttrib(className, 'Label_7', 'bg', '#000000')
Fun.SetTKAttrib(className, 'Label_7', 'fg', '#ffffff')

```

10. GetTKAttrib: Get the tkinter attribute value of the control.

Parameter 1: interface class name, parameter 2: control name, parameter 3: attribname is property name.

```

def GetTKAttrib(uiName, elementName, AttribName):
    global G_UIElementArray
    if uiName in G_UIElementArray:
        return G_UIElementArray[uiName][elementName].cget(AttribName)
    return None

```

Example:

```

bgColor = Fun.GetTKAttrib(className, 'Label_7', 'bg')
fgColor = Fun.GetTKAttrib(className, 'Label_7', 'fg')

```

11. SetText: Set the text of the control(label, button, entry and text)

Parameter 1: interface class name, parameter 2: control name, parameter 3: text content.

```

def SetText(uiName, elementName, textValue):
    global G_UIElementArray
    global G_UIElementVariableArray
    showtext = str("%s"%textValue)
    if uiName in G_UIElementVariableArray:
        if elementName in G_UIElementVariableArray[uiName]:
            G_UIElementVariableArray[uiName][elementName].set(showtext)
            return
    if uiName in G_UIElementArray:
        if elementName in G_UIElementArray[uiName]:
            if elementName.find('Text_') >= 0:
                G_UIElementArray[uiName][elementName].delete('0.0',tkinter.END)
                G_UIElementArray[uiName]
[elementName].insert(tkinter.END,showtext)

```

```

else:
    G_UIElementArray[uiName][elementName].configure(text=showtext)

```

Example:

```

Fun.SetText(className, 'Label_2', 'Name')
Fun.SetText(className, 'Entry_3', 'Honghaier')

```

12. GetText: Get the text of the control.

Parameter 1: interface class name, parameter 2: control name

```

def GetText(uiName,elementName):
    global G_UIElementArray
    global G_UIElementVariableArray
    if uiName in G_UIElementVariableArray:
        if elementName in G_UIElementVariableArray[uiName]:
            return G_UIElementVariableArray[uiName][elementName].get()
    if uiName in G_UIElementArray:
        if elementName in G_UIElementArray[uiName]:
            if elementName.find('Text_') >= 0:
                return G_UIElementArray[uiName][elementName].get('0.0',
tkinter.END)
            elif elementName.find('Spinbox_') >= 0:
                return str(G_UIElementArray[uiName][elementName].get())
            else:
                return G_UIElementArray[uiName][elementName].cget('text')
    return str("")

```

Example:

```

Name = Fun.GetText(className, 'Entry_3')

```

13. SetImage: Set the image of the control(Label,Button).

Parameter 1: interface class name, parameter 2: control name, parameter 3: picture name.

```

def SetImage(uiName,elementName,imagePath):
    global G_UIElementVariableArray
    if elementName.find('Label_') == 0 or elementName.find('Button_') == 0 :
        Control = GetElement(uiName,elementName)
        if Control != None:
            if uiName in G_UIElementUserDataArray:
                if elementName in G_UIElementUserDataArray[uiName]:
                    for EBData in G_UIElementUserDataArray[uiName]
[elementName]:
                        if EBData[0] == 'image':
                            EBData[1] = imagePath
                            from PIL import Image,ImageTk
                            image=Image.open(imagePath).convert('RGBA')
                            image_Resize = image.resize((Control.winfo_width(),
Control.winfo_height()),Image.ANTIALIAS)
                            EBData[2] = ImageTk.PhotoImage(image_Resize)
                            Control.configure(image = EBData[2])
                        return
    from PIL import Image,ImageTk

```

```

        image=Image.open(imagePath).convert('RGBA')
        image_Resize = image.resize((Control.wininfo_width(),
Control.wininfo_height()),Image.ANTIALIAS)
        EBData2 = ImageTk.PhotoImage(image_Resize)
        AddUserData(uiName,elementName,'image',imagePath,EBData2,0)
        Control.configure(image = EBData2)

```

Example:

```
Fun.SetImage(className,'Label_2','C:\\bg.jpg')
```

14. GetImage: Get the image file path of the control(Label,Button).

Parameter 1: interface class name, parameter 2: control name.

```

def GetImage(uiName,elementName):
    global G_UIElementVariableArray
    if elementName.find('Label_') == 0 or elementName.find('Button_') == 0 :
        Control = GetElement(uiName,elementName)
        if Control != None:
            if uiName in G_UIElementUserDataArray:
                if elementName in G_UIElementUserDataArray[uiName]:
                    for EBData in G_UIElementUserDataArray[uiName]
[elementName]:
                        if EBData[0] == 'image':
                            return EBData[1]
    return str("")

```

Example:

```
bgImage = Fun.GetImage(className,'Label_2')
```

15. SetSelectIndex: Set the selected index values for listbox and ComboBox.

Parameter 1: interface class name, parameter 2: control name, parameter 3: index value.

```

def SetSelectIndex(uiName,elementName,index):
    if uiName in G_UIElementAlias.keys() and elementName in
G_UIElementAlias[uiName].keys():
        elementName = G_UIElementAlias[uiName][elementName]
        Control = GetElement(uiName,elementName)
        if Control != None:
            if elementName.find('ComboBox_') == 0 :
                Control.current(index)
            elif elementName.find('ListBox_') == 0 :
                Control.select_set(index)

```

Example:

```
Fun.SetSelectIndex(className,'ComboBox_4',2)
```

16. GetSelectIndex: Get the selected index values for listbox and ComboBox.

Parameter 1: interface class name, parameter 2: control name.

```

def GetSelectIndex(uiName,elementName):
    if uiName in G_UIElementAlias.keys() and elementName in
G_UIElementAlias[uiName].keys():
        elementName = G_UIElementAlias[uiName][elementName]
        Control = GetElement(uiName,elementName)

```

```

if Control != None:
    if elementName.find('ComboBox') == 0 :
        return Control.current()
    elif elementName.find('ListBox') == 0 :
        currIndex = Control.curselection()
        if len(currIndex) > 0 and currIndex[0] >= 0:
            return currIndex[0]
return -1

```

Example:

```
index = Fun.GetSelectIndex(className, 'ComboBox_4')
```

17. InitElementData:Initialize all user binding data without manual call.

Parameter 1: interface class name,

```

def InitElementData(uiName):
    global G_UIElementUserDataArray
    if uiName in G_UIElementUserDataArray:
        for elementName in G_UIElementUserDataArray[uiName].keys():
            for EBDData in G_UIElementUserDataArray[uiName][elementName]:
                if EBDData[3] == 1:
                    SetText(uiName,elementName,EBDData[2])
                    SetText(uiName,elementName,EBDData[2])

```

18. InitElementStyle:Initialize all control style without manual call.

Parameter 1: interface class name; parameter 2: style name.

```

def InitElementStyle(uiName,Style):
    StyleArray = ReadStyleFile(Style+".py")
    global G_UIElementArray
    if uiName in G_UIElementArray:
        for elementName in G_UIElementArray[uiName].keys():
            widget = G_UIElementArray[uiName][elementName]
            try:
                if widget.wininfo_exists() == 1:
                    winClass = widget.wininfo_class()
                    StyleName = ".T"+winClass
                    if StyleName == '.TLabel':
                        Root = GetElement(uiName,'root')
                        Root['background'] = StyleArray[StyleName]
['background']
                        for attribute in StyleArray[StyleName].keys():
                            widget[attribute] = StyleArray[StyleName][attribute]
            except BaseException:
                continue

```

19. GetInputDataArray:Get all the entry data of an interface

Parameter 1: interface class name,


```

def GetInputDataArray(uiName):
    global G_UIElementArray
    global G_UIInputDataArray
    global G_UIElementVariableArray
    G_UIInputDataArray.clear()
    if uiName in G_UIElementArray:
        for elementName in G_UIElementArray[uiName].keys():
            G_UIInputDataArray[elementName] = []
            widget = G_UIElementArray[uiName][elementName]
            if elementName.find('Text_') >= 0:
                content = widget.get('0.0', tkinter.END)
                G_UIInputDataArray[elementName].append(content)
            elif elementName.find('Entry_') >= 0:
                content = G_UIElementVariableArray[uiName][elementName].get()
                G_UIInputDataArray[elementName].append(content)
    if uiName in G_UIElementVariableArray:
        for elementName in G_UIElementVariableArray[uiName].keys():
            if elementName.find('Group_') >= 0:
                ElementIntValue = G_UIElementVariableArray[uiName]
                [elementName].get()
                G_UIInputDataArray[elementName] = []
                G_UIInputDataArray[elementName].append(ElementIntValue)
    return G_UIInputDataArray

```

Example:

```

import RegDlg

RegDlg.RegDlg(topLevel)

tkinter.Tk.wait_window(topLevel)

InputDataArray = RegDlg.Fun.G_UIInputDataArray

print(InputDataArray)

print('Name: '+InputDataArray['Entry_5'][0])

```

20. **CenterDlg:Center a pop-up interface dialog box,Center a pop-up interface dialog box. If the parameter does not specify the width and height, use the width and height of the dialog box itself. If it is used before registering root, that is, the main window root is in the center, the center of the windows screen is in the center.**

Parameter 1: interface class name, parameter 2: dialog form, parameter 3: form width, parameter 4: form height.

```

def CenterDlg(uiName,popupDlg,dw=0,dh=0):
    if dw == 0:
        dw = popupDlg.wininfo_width()
    if dh == 0:
        dh = popupDlg.wininfo_height()
    root = GetElement(uiName,'root')
    if root != None:
        sw = root.wininfo_width()
        sh = root.wininfo_height()
        sx = root.wininfo_x()
        sy = root.wininfo_y()

```

```

        popupDlg.geometry('%dx%d+%d+%d'%(dw,dh,sx+(sw-dw)/2,sy+(sh-dh)/2))
    else:
        import ctypes
        user32 = ctypes.windll.user32
        sw = user32.GetSystemMetrics(0)
        sh = user32.GetSystemMetrics(1)
        sx = 0
        sy = 0
        popupDlg.geometry('%dx%d+%d+%d'%(dw,dh,sx+(sw-dw)/2,sy+(sh-dh)/2))

```

Example:

```
Fun.CenterDlg(uiName,popDlg)
```

21. **SetRoundedRectangle:**The rounded property of the control is called in the interface Layout Pyfile, but because the interface has not been created, it is necessary to call ShowRoundedRectangle two times later. Note: this function does not cross platform.

Parameter 1: control, parameter 2: fillet width, parameter 3: fillet height.

```

def SetRoundedRectangle(control,widthEllipse=20,HeightEllipse=20):
    if control != None:
        control.after(10, lambda:
ShowRoundedRectangle(control,widthEllipse,HeightEllipse))

```

Example:

```
SetRoundedRectangle(Button_2,20,20)
```

22. **ShowRoundedRectangle:** Set the fillet property of the control immediately.

参数1: 控件, 参数2:圆角宽度, 参数3:圆角高度。

```

def ShowRoundedRectangle(control,widthEllipse,HeightEllipse):
    import win32gui
    HRGN =
win32gui.CreateRoundRectRgn(0,0,control.wininfo_width(),control.wininfo_height(),wi
dthEllipse,HeightEllipse)
    win32gui.SetWindowRgn(control.wininfo_id(), HRGN,1)

```

Example:

```
Fun.ShowRoundedRectangle(Button_2,20,20)
```

23. **SetTransparencyFunction:** Sets the transparent value of the form. Sets the transparent value of the form.**Sets the transparent value of the form.

Parameter 1: control, parameter 2: transparency.

```

def SetTransparencyFunction(root,alpha):
    if root != None:
        try :
            import ctypes
            from ctypes import windll
            hwnd = windll.user32.GetParent(root.wininfo_id())
            _winlib = ctypes.windll.user32
            style = _winlib.GetWindowLongA( hwnd, 0xffffffffec ) | 0x00080000
            _winlib.SetWindowLongA( hwnd, 0xffffffffec, style )
            _winlib.SetLayeredWindowAttributes( hwnd, 0, alpha, 2 )

```

```
except ImportError:
    pass
```

Example:

```
Fun.SetTransparencyFunction(root,128)
```

24. **ExpandAllTreeItem: Expand or close tree items**

Parameter 1: target tree item, parameter 1: expand or close, parameter 3: recursive call, no need to pass in manually.

```
def ExpandAllTreeItem(targetTree,isOpen,parentItem = None):
    ParentItemArray = [parentItem]
    if parentItem == None:
        ParentItemArray = targetTree.get_children()
    for Item in ParentItemArray:
        targetTree.item(Item,open=isOpen)
        for childItem in targetTree.get_children(Item):
            targetTree.item(childItem,open=isOpen)
            ExpandAllTreeItem(targetTree,isOpen,childItem)
```

Example:

```
Fun.ExpandAllTreeItem(TreeView_4,True)
```

25. **MessageBox:Pop up a message dialog box**

Parameter 1: the dialog box displays text.

```
def MessageBox(text):
    tkinter.messagebox.showwarning('info',text)
```

Example:

```
Fun.MessageBox("Thank you")
```

26. **InputDialog:Pop up a input dialog box**

Parameter 1: dialog box title text, parameter 2: dialog box default box input text.

```
def InputBox(title,text):
    res = tkinter.simpledialog.askstring(title,'Input Box',initialvalue=text)
    return res
```

Example:

```
Fun.InputBox("Please input the name")
```

27. **AskBox:Pop up a question dialog box,You need choose yes or no.**

Parameter 1: dialog title text, parameter 2: dialog display text.

```
def AskBox(title,text):
    res = tkinter.messagebox.askyesno(title,text)
    return res
```

Example:

```
Result = Fun.AskBox("Are you sure to delete?")
```

```
if Result == True:
```

```
    ....
```

28. **WalkAllResFiles>Returns all the files in the directory**

Parameter 1: directory name, parameter 2: enter subdirectory, parameter 3: whether there is extension filtering.

```
def walkAllResFiles(parentPath,alldirs=True,extName=None):
    ResultFilesArray = []
    if os.path.exists(parentPath) == True:
        for fileName in os.listdir(parentPath):
            if '__pycache__' not in fileName:
                if '.git' not in fileName:
                    newPath = parentPath + '\\'+ fileName
                    if os.path.isdir(newPath):
                        if extName == None:
                            ResultFilesArray.append(newPath)
                        if alldirs == True:
                            ResultFilesArray.extend(walkAllResFiles(newPath,alldirs,extName))
                    else:
                        if extName == None:
                            ResultFilesArray.append(newPath)
                        else:
                            file_extension = os.path.splitext(fileName)
                            [1].replace('.', '')
                            file_extension_lower =
file_extension.lower().strip()
                            file_extName_lower = extName.lower().strip()
                            if file_extension_lower == file_extName_lower:
                                ResultFilesArray.append(newPath)
    return ResultFilesArray
```

Example:

```
jsonFileArray = Fun.walkAllResFiles(path,False,'json')
```

29. EventFunction_Adaptor:Redefine the event response function to have the specified parameter.

```
def EventFunction_Adaptor(fun, **params):
    return lambda event, fun=fun, params=params: fun(event, **params)
```

Example:

```
Button_7.configure(command=lambda:Project1_cmd.Button_7_onCommand(className,"Button_7"))
```

30 SetControlPlace:Sets the absolute or relative position of the control

Parameter 1: control, parameter 2: x position, parameter 3: y position, parameter 4: height, parameter 5: width.

```
`def SetControlPlace(control,x,y,w,h):
    control.place(x=0,y=0,width=0,height=0)
    control.place(relx=0,relx=0,relwidth=0,relheight=0)
    if type(x) == type(1.0):
        if type(y) == type(1.0):
            if type(w) == type(1.0):
                if type(h) == type(1.0):
```

```

        control.place(relx=x, rely=y, relwidth=w, relheight=h)
    else:
        control.place(relx=x, rely=y, relwidth=w, height=h)
    else:
        if type(h) == type(1.0):
            control.place(relx=x, rely=y, width=w, relheight=h)
        else:
            control.place(relx=x, rely=y, width=w, height=h)
    else:
        if type(w) == type(1.0):
            if type(h) == type(1.0):
                control.place(relx=x, y=y, relwidth=w, relheight=h)
            else:
                control.place(relx=x, y=y, relwidth=w, height=h)
        else:
            if type(h) == type(1.0):
                control.place(relx=x, y=y, relwidth=w, relheight=h)
            else:
                control.place(relx=x, y=y, relwidth=w, height=h)
    else:
        if type(y) == type(1.0):
            if type(w) == type(1.0):
                if type(h) == type(1.0):
                    control.place(x=x, rely=y, relwidth=w, relheight=h)
                else:
                    control.place(x=x, rely=y, relwidth=w, height=h)
            else:
                if type(h) == type(1.0):
                    control.place(x=x, rely=y, width=w, relheight=h)
                else:
                    control.place(x=x, rely=y, width=w, height=h)
        else:
            if type(w) == type(1.0):
                if type(h) == type(1.0):
                    control.place(x=x, y=y, relwidth=w, relheight=h)
                else:
                    control.place(x=x, y=y, relwidth=w, height=h)
            else:
                if type(h) == type(1.0):
                    control.place(x=x, y=y, width=w, relheight=h)
                else:
                    control.place(x=x, y=y, width=w, height=h)`

```

****Example:****

```
`Fun.SetControlPlace(Button_1, 10, 10, 100, 24)`
```

31. **SetRootRoundRectangle:**Set the fillet parameter of TK root, support cross platform.

Parameter 1: Canvas control, parameter 2: x position, parameter 3: y position, parameter 4: height, parameter 5: width, parameter 6: fillet radius.

```

def SetRootRoundRectangle(canvas,x1, y1, x2, y2, radius=25,**kwargs):
    points = [x1+radius, y1,
              x1+radius, y1,
              x2-radius, y1,
              x2-radius, y1,

```

```

        x2, y1,
        x2, y1+radius,
        x2, y1+radius,
        x2, y2-radius,
        x2, y2-radius,
        x2, y2,
        x2-radius, y2,
        x2-radius, y2,
        x1+radius, y2,
        x1+radius, y2,
        x1, y2,
        x1, y2-radius,
        x1, y2-radius,
        x1, y1+radius,
        x1, y1+radius,

    return canvas.create_polygon(points, **kwargs, smooth=True)

```

Example:

```
Fun.SetControlPlace(Button_1,10,10,100,24)
```

32. ReadFromFile:Read content from file.

Parameter 1: file path.

```

def ReadFromFile(filePath):
    content = None
    if filePath != None:
        if os.path.exists(filePath) == True:
            f = open(filePath,mode='r',encoding='utf-8')
            if f != None:
                content = f.read()
                f.close()
    return content

```

Example:

```

content = Fun.ReadFromFile('test.txt')

print(content)

```

33. WriteToFile:Writing content to a file

Parameter 1: file path, parameter 2: written content.

```

def writeToFile(filePath,content):
    if filePath != None:
        f = open(filePath,mode='w',encoding='utf-8')
        if f != None:
            if content != None:
                f.write(content)
            f.close()
        return True
    return False

```

Example:

```
content = "welcome to use tkinter designer."
```

```
Fun.WriteToFile('test.txt',content)
```

34. ReadStyleFile:Read style file

Parameter 1: file path.

```
def ReadStyleFile(filePath):
    StyleArray = {}
    if len(filePath)==0 :
        return StyleArray
    if os.path.exists(filePath) == False:
        return StyleArray
    f = open(filePath,encoding='utf-8')
    line = ""
    while True:
        line = f.readline()
        if not line:
            break
        text = line.strip()
        if not text:
            continue
        if text.find('style = tkinter.ttk.Style()') >= 0:
            continue
        if text.find('style.configure()') >= 0:
            splitarray1 = text.partition('style.configure()')
            stylename = None
            splitarray2 = None
            if splitarray1[2].find(',') >= 0:
                splitarray2 = splitarray1[2].partition(',')
                stylename = splitarray2[0].replace('\"', '')
            else:
                splitarray2 = splitarray1[2].partition(')')
                stylename = splitarray2[0].replace('\"', '')
            sytleValueText = splitarray2[2]
            fontindex_begin = sytleValueText.find('font=')
            fontindex_end = fontindex_begin
            StyleArray[stylename] = {}
            othertext = sytleValueText
            if fontindex_begin >= 0:
                fontindex_end = sytleValueText.find(')')
                fonttext = sytleValueText[fontindex_begin+6:fontindex_end]
                fontsplitarray = fonttext.split(',')
                StyleArray[stylename]['font'] =
tkinter.font.Font(family=fontsplitarray[0].replace('\"', '').strip(),
size=int(fontsplitarray[1].replace('\"', '').strip()),weight=fontsplitarray[2].re
place('\"', '').strip())
                othertext = sytleValueText[0:fontindex_begin] +
sytleValueText[fontindex_end+1:-1]
            else:
                splitarray4 = sytleValueText.partition(')')
                othertext = splitarray4[0]
```

```
        splitarray3 = othertext.split(',')
        for stylecfgtext in splitarray3:
            if stylecfgtext.find('=') > 0:
                splitarray4 = stylecfgtext.partition('=')
                key = splitarray4[0].replace('"','').strip()
                value = splitarray4[2].replace('"','').strip()
                StyleArray[stylename][key] = value
            continue
        if text.find('style.map()') >= 0:
            continue
    f.close()
    return StyleArray
```

Example:

```
StyleArray= ReadStyleFile('Project1_sty.py')
```