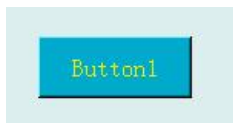


DRButton 按钮用法

按钮控件是最常用的 GUI 控件之一，可用于接收用户的鼠标点击输入，然后通过直连方式驱动另一控件工作，或通过回调函数形式定义其具体功能和操作。



DRButton 按钮控件显示效果图

在 Python 程序 Tkinter GUI 界面中创建一个 DRButton 按钮控件的命令和语法格式如下：

```
hl=DRButton(win,x,y,w,h,option=value, ...)
```

其中输入参数 win 为 Tkinter 库创建的 GUI 窗口句柄，[x,y,w,h]四个参数为该控件插入 GUI 窗口 win 时的左上角坐标[x,y]和该控件的宽、高[w,h]。另外是一组可选的输入参数，包括：

cb: 背景色，默认值='#00aacc'

cf: 前景色，默认值='#eeee00'

txt: 标题，默认值='Button'

fsiz: 字符点数，默认值=12

val: 按键值，默认值=101

返回值 hl 则是该控件创建后的句柄变量名，可以通过 hl 对创建后的控件进行进一步的设定。

下面是一个全参数的 DRButton 按钮控件创建实例：

```
hl=dr.DRButton(win,262,103,100,40,'#00aacc','#eeee00','Button1',12,101)
```

通常，可以选择设定部分选择参数的形式进行设定，除基本参数 win、[x,y,w,h]和用变量名设定的可选参数外，其余可选参数采用默认值。创建实例如下：

```
hl=dr.DRButton(win,262,103,100,40,txt='Run')
```

创建后的 DRButton 按钮控件可以通过直连方式将用户鼠标点击操作传递给另一控件，或通过回调函数绑定的方式将用户的鼠标点击操作传递给被绑定的回调函数。

DRButton 按钮控件的直连或回调函数绑定格式为：

```
hl.addCallBackSingle(fun)
```

其中 fun 是被绑定的另一控件的 set 函数名，或用户自定义的回调函数名。DRButton 按钮控件绑定的 fun 函数是一个单数值函数，其收到的则是该按钮控件传递过来的按键值 val。下面是 fun 函数的形式：

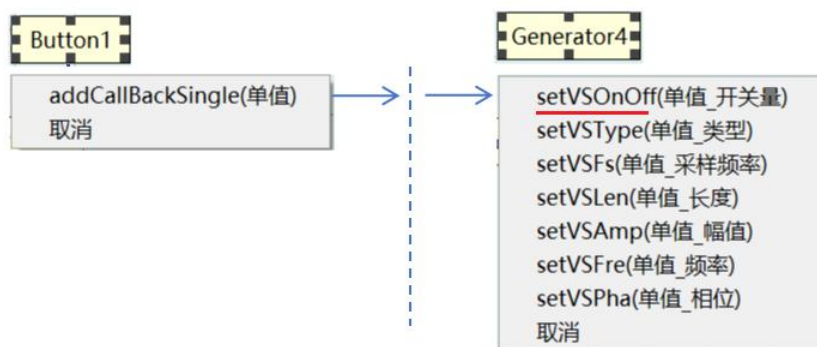
```
def setValueSingle(v):
```

.....

通过回调函数绑定方式，就可以将用户的鼠标点击操作传递到回调函数中，或者传递到另一个控件的 set 方法中，驱动它们运行。

当采用 DRPython 进行图形化编程时，控件的创建代码和直连绑定代码的生成均可以由机器实现；控件创建只需要将所需控件拖拽到 GUI 布局区，而直连绑定则需要在 GUI 布局区的控件间进行连线操

作。按钮控件为单数值输出控件，在连线状态下在控件上点击鼠标，则弹出端选择；选择完成后，点击另一控件，则弹出输入端选择；若输出端和输入端类型一致，则完成连线。输出的单值(addCallBackSingle)为按钮的设定值。例如下图是将按钮控件的输出函数与信号发生器控件的开关设定函数相连。



下面是 DRButton 按钮控件的创建和应用示例小程序，功能是在 GUI 界面上插入一个按钮控件、一个文本框控件和一个用户自定义函数，然后将它们分别绑定到按钮控件的输出上。点击按钮，则在用户点击操作引发的数据流驱动下，文本框控件的输入值显示该按钮控件的按键值 val，而用户自定义的回调函数则 Print 出接收到的按键值 val。DRButton 按钮控件应用示例小程序代码如下：

```

#==引用的包
import tkinter as tk
import drvi.drviControllls as dr
#用户自定义的回调函数
def muserSingle3(v):
    print(v)

#==创建 Tkinter GUI 窗口
win= tk.Tk()
win.geometry('816x497')
win.config(bg="#ddeeef")
#==插入按钮控件
mButton1=dr.DRButton(win, 50, 42, 100, 40, txt=' Run' )
#==插入文本框控件
mEntry2=dr.DREntry(win, 198, 44, 100, 30)
#==直连方式绑定文本框控件
mButton1.addCallBackSingle(mEntry2.setValueSingle)
#==绑定用户回调函数
mButton1.addCallBackSingle(muserSingle3)
#==主程序循环
win.mainloop()

```

pyqt 版本中的用法说明：

DRVI 有继承自 Tkinter 和继承自 pyqt 两个不同库，除引用的 GUI 库(Tkinter/pyqt)和框架程序略有不同外，DRVI 部分控件的创建、布局和连线是完全相同的。在 DRVI 的 pyqt 版本中，DRButton 按钮控件继承自 QPushButton，因此除了用 DRButton 的参数化外观定义外，还可以直接用 QPushButton 的样式表来对其进行外观描述，并继承其样式表加载方法：

```
setStyleSheet(styleString)
```

下面是用样式表来定义 DRButton 外观样例：

```
mButton1=dr.DRButton(win,78,38,100,40)
mButton1.setStyleSheet("background-color: red; color: white; font: bold 12px; border: 2px solid blue; border-radius: 10px;")
```

效果如下：



样式表的具体用法可参阅 [pyqt](#) 或 [QT](#) 中 [QPushButton](#) 样式表的设定部分。

下面 [pyqt](#) 版本中是 [DRButton](#) 按钮控件的用法示例：

```
# 1. 引用的库
import sys
from PyQt5.QtWidgets import *
from PyQt5.QtGui import *
from PyQt5.QtCore import *
import drvi.drviQTControlls as dr

#2.回调函数
def mButClick(v):
    print('Value=',v)

#3.创建 GUI 窗体对象
app=QApplication(sys.argv)
app.setStyle("Fusion")
win=QWidget()
win.setWindowTitle('PyQt 程序')
win.resize(800,500)
win.setStyleSheet("background-color : #ddeeee; ")
#4.在 GUI 窗口插入控件和布局
mButton1=dr.DRButton(win,78,38,100,40)
mButton1.setStyleSheet("background-color: red; color: white; font: bold 12px; border: 2px solid blue; border-radius: 10px;")
#5.控件回调函数绑定=====
mButton1.addCallBackSingle(mButClick)
#6.显示 GUI 主窗口界面，等待用户操作
win.show()
sys.exit(app.exec_())
```