

DRCheckBox 方块型选择框用法

DRCheckBox 方块型选择框控件是一组纵向或水平排列在一起的选项，可用于接收用户的鼠标点击选择，然后通过回调函数形式编程定义其具体功能和操作，输出值可以是选项字符串，也可以是选项号+设定值。



DRCheckBox 方块型选择框控件的图形显示效果图

在 Python 程序 Tkinter GUI 界面中创建一个 DRCheckBox 方块型选择框控件的命令和语法格式如下：

```
hl=DRCheckBox(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='#003355'
- cb1: 次背景色，默认值='#007799'
- cf: 前景色，默认值='#fff00'
- txt: 标题，默认值='A,B,C,D'，引号中单词数=选项数
- fsize: 字符点数，默认值=12
- val: 当前选项值，默认值=0
- d: 控件显示形式，默认值=0（0=垂直，1=水平）

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。

DRCheckBox 方块型选择框控件有 2 个 get 属性函数，功能如下：

txt=hl.getValueString(): 获取当前选项菜单条文字

v=hl.getValueSingle(): 获取当前选项菜单条号

下面是一个全参数的 DRCheckBox 方块型选择框控件创建实例：

```
hl=dr.DRCheckBox(win,74,80,120,120,'#003355','#007799','#fff00','A,B,C,D',12,0,0)
```

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。例如，下面创建一个水平显示的 DRCheckBox 方块型选择框控件：

```
hl=dr.DRCheckBox(win,55,424,260,30,d=1)
```

创建后可以通过返回的控件句柄对其属性函数进行操作。

DRCheckBox 方块型选择框控件可以通过回调函数绑定的方式将用户的输入传递给绑定的回调函数，数字型格式为：

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

被绑定的回调函数或另一控件的 set 函数是一个单数值函数，收到的是当前选项号+v1a。回调函数形式为：

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

```
.....
```

文本型回调函数绑定格式为：

```
hl.addCallBackString(fun)
```

被绑定的回调函数或另一控件的 set 函数是一个字符串函数，收到的是当前选项的菜单条文字。回调函数形式为：

```
def setValueString(t):
```

```
.....
```

通过回调函数绑定方式，就可以将用户的输入传递到自定义的回调函数中或传递给直连的另一控件。

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。方块型选择框控件可以作为字符串或单数值输出控件，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。选项号或条目可以通过单值端口(addCallBackSingle)或字符串端口(addCallBackString)输出。



下面是 DRCheckBox 方块型选择框控件的创建和应用示例小程序，功能是在 GUI 界面上插入一个方块型选择框控件，一个标签控件、一个用户自定义函数，然后通过直连绑定方式，将方块型选择框控件的选项菜单条文字传递给标签控件，将选项值传递给自定义函数。DRCheckBox 方块型选择框控件应用示例小程序代码如下：

```
##引用的包
import tkinter as tk
import drvi.drviControlls 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")
```

```
#==插入方块型选择框控件控件
mRadio=dr.DRCheckBox(win, 73, 244, 120, 120)
#==插入标签控件
mLab=dr.DRLabel(win, 221, 42, 200, 30, fsize=16)
#==绑定用户回调函数
mRadio.addCallBackSingle(muserSingle3)
mRadio.addCallBackString(mLab.setValueString)
#==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，DRCheckBox 继承自 QCheckBox，因此除了用 DRCheckBox 的参数化外观定义外，还可以直接用 QCheckBox 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)
```