

INSTRUMENTACION VIRTUAL

CHECK BOX

Otro widget muy usado es el Check Box muy similar al Radio Button la diferencia está en que con el Check Box vamos a poder seleccionar varias opciones al mismo tiempo. Un Check Box puede tener uno de estos tres estados, Seleccionado (checked), no seleccionado (unchecked) y tristate (sin cambio). El estado tristate es un estado de no cambio, el usuario tiene entre seleccionar o no seleccionar el Check Box. El Check Box es instancia de la clase QCheckBox y tiene los siguientes métodos.

METODOS

- **isChecked()**. Este método retorna un valor True si el Check Box es seleccionado en otro caso retornara False.
- **setTristate()**. Si no se quiere que el Check Box tenga la opción de seleccionarse o deseleccionarse únicamente tenemos que pasar el valor True a este metodo y el usuario será incapaz de cambiar el estado del Check Box.
- **setIcon()**. Este metodo es usado para desplegar un icono con el Check Box.
- **setText()**. Este metodo permite cambiar el texto al Check Box. Para indicar un atajo de teclado se precede a la letra del texto con un (&).
- **setChecked()**. Para poner el Check Box seleccionado por default pasamos a este metodo el valor True.

SEÑALES

Las señales emitidas por el Check Box son las siguientes.

- **clicked()**. Esta señal es emitida cuando un Check Box es activado o cuando se activa a través de su atajo de teclado (Alt + Letra subrayada).
- **stateChanged()**. Esta señal es emitida cada vez que un Check Box cambia su estado de seleccionado a no seleccionado o viceversa.

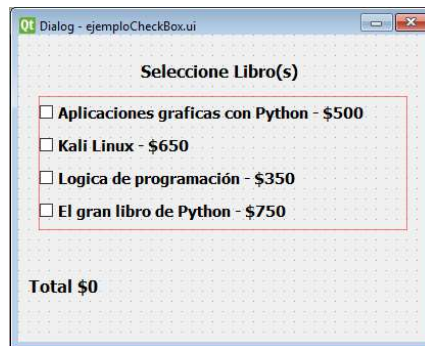
Para entender mejor el funcionamiento del Check Box vamos a hacer un ejemplo simulando un punto de venta de libros. En este ejemplo tendremos únicamente cuatro libros cada libro tendrá un precio diferente, cuando el usuario seleccione el libro nos deberá de aparecer la suma de los libros que se hayan vendido.

Para hacer esto vamos a abrir el Designer y realizaremos los siguientes pasos.

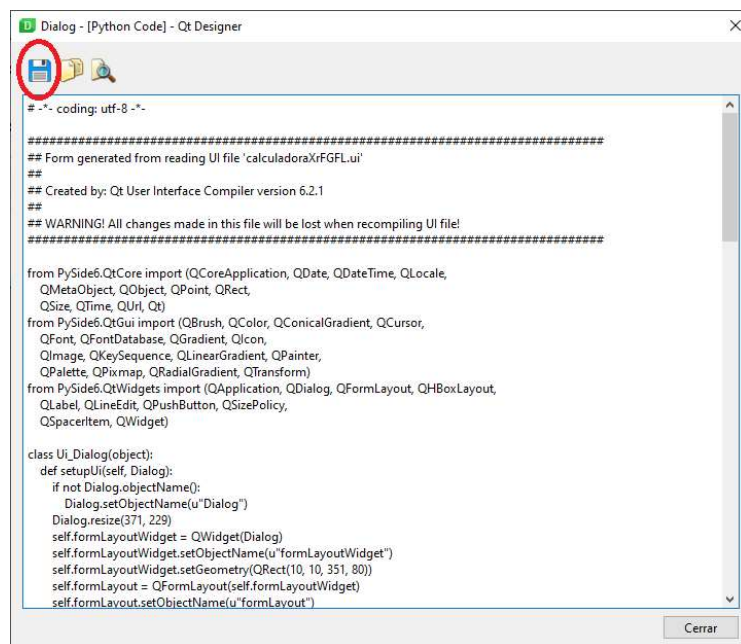
1. Empezaremos creando un nuevo archivo y seleccionamos un MainWindow.
2. Arrastramos dos Label, y cuatro Check Box dentro de la ventana.
3. Cambiamos las propiedades *text* de los Label a uno le asignamos el siguiente texto "Seleccione Libro(s)" y al segundo Label le borramos el texto.

4. Cambiamos las propiedades *text* de cada Check Box y ponemos lo siguiente primer Check Box “Aplicaciones graficas con Python - \$500”, segundo Check Box “Kali Linux - \$650”, tercer Check Box, “Logica de programación - \$350” y el ultimo Check Box “El gran libro de Python - \$750”.
5. En la propiedad *objectName* del Label que no tiene texto ponemos lblResultado y en cada propiedad *objectName* ponemos para el primer Check Box (siguiendo el orden del paso anterior) cbLibro1, el siguiente como cbLibro2, el tercer libro como cbLibro3 y el último libro como cbLibro4.
6. Guardamos la interfaz como checkBox.ui.

La interfaz tendrá el siguiente aspecto usando la pre - visualización del Designer.



7. Después de crear la interfaz convertimos el archivo “ui” en código Python. Primer vamos al menú Formulario/View Python Code...



Damos click en el icono de guardar (en la figura del disco) y guardamos el archivo Python en la ubicación deseada de preferencia en la misma ubicación del proyecto.

Lo siguiente es crear un nuevo proyecto en PyCharm o PyScripter y guardamos el proyecto en la misma carpeta donde se tiene el archivo ui y escribir el siguiente código.

```
import sys
from PySide6.QtWidgets import QApplication, QMainWindow
from checkBox import Ui_MainWindow

class MiFormulario(QMainWindow, Ui_MainWindow):
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.cbLibro1.stateChanged.connect(self.sumarTotal)
        self.cbLibro2.stateChanged.connect(self.sumarTotal)
        self.cbLibro3.stateChanged.connect(self.sumarTotal)
        self.cbLibro4.stateChanged.connect(self.sumarTotal)

    def sumarTotal(self):
        total = 0
        if self.cbLibro1.isChecked():
            total += 500
        if self.cbLibro2.isChecked():
            total += 650
        if self.cbLibro3.isChecked():
            total += 350
        if self.cbLibro4.isChecked():
            total += 750

        self.lblResultado.setText("Total $" + str(total))

if __name__ == "__main__":
    app = QApplication(sys.argv)
    w = MiFormulario()
    w.show()
    sys.exit(app.exec())
```

Como resultado obtendremos la siguiente ventana.



Podremos ver que si seleccionamos o deseccionamos diferentes Check Box la suma será diferente y obtendremos el total de los libros seleccionados.