

# PYQT6 Course

## 15. Creating QSpinBox in PyQt6

QSpinBox is designed to handle integers and discrete sets of values, QSpinBox allows the user to choose a value by clicking the up/down buttons or pressing up/down on the keyboard to increase/decrease the value currently displayed. The user can also type the value in manually.

there are different methods that you can use in QSpinBox for example we have *value()* method, and this method returns the current selected integer value of spinbox, we have *setMinimum()* and *setMaximum()* methods, also there are some signals for QSpinBox, for example we have *valueChanged()* signal and it is emitted when the value of spinbox is changed also we have *editingFinished()* signal and it is emitted when the focus is lost on the spinbox.

These are some more methods for QSpinBox.

- *value()*: This method returns the current integer value selection from the QSpinBox class.
- *text()*: This method is used to display text in the spinbox.

- `setPrefix()`: This method assigns the prefix text that is prepended to the value returned by the spin box.
- `setSuffix()`: This method assigns the suffix text that is to be appended to the value returned by the spin box.

We can use `QSpinBox` class for creating of the SpinBox, `QSpinBox` class is related to `QtWidgets` module.

```
self.spinbox = QSpinBox()
```

Basically in our example we have used two `QLineEdit`.

```
self.lineedit = QLineEdit()
```

As we have already said that there are different signals that you can use in `QSpinBox` class, for example in here we have used *`valueChanged()`* signal of `QSpinBox`, and it is emitted when the value of spinbox is changed, we have connected the signal with the *`spin_selected()`* method or slot.

```
self.spinbox.valueChanged.connect(self.spin_selected)
```

This is our method or slot, in this method first we are going to check that if our lineedit is not zero, if that is the case we are going to get the value from the lineedit and after that we multiply that with spinbox amount, and at the end we set that to the label that we have.

```
def spin_selected(self):  
    if self.lineEdit.text() != 0:  
        price = int(self.lineEdit.text())  
        totalPrice = self.spinbox.value() * price  
  
    self.total_result.setText(str(totalPrice))  
  
    else:  
        print("Wrong value")
```