

PYQT6 Course

10. Working with QRadioButton in PyQt6

A QRadioButton is an option button that can be switched on (checked) or off (unchecked). Radio buttons typically present the user with a "one of many" choice. In a group of radio buttons, only one radio button at a time can be checked, if the user selects another button, the previously selected button is switched off. there are different methods that you can use, for example we have *isChecked()* and it returns a boolean value true if the button is in the selected state or we have *setIcon()* method that we can add icon for the radiobutton, also *setText()* which set the text of the radiobutton. also there are different signals that you can use for example we have toggled signal which is used whenever the radio button changes it is state from checked to unchecked and vis versa.

The QRadioButton class provides the following methods:

- *isChecked()*: This method returns the Boolean value true if the button is in the selected state.
- *setIcon()*: This method displays an icon with the radio button.
- *setText()*: This method assigns the text to the radio button.

- `setChecked()`: To make any radio button appear selected by default, pass the Boolean value `true` to this method.

To create `RadioButton`, you need to use `QRadioButton` class and `QRadioButton` is related to `QtWidgets` module, first you need to create the object of `QRadioButton`.

```
rad1 = QRadioButton("Python")
```

In here we have added an icon to the `RadioButton`, make sure that you have already added some icons to the project.

```
rad1.setIcon(QIcon("images/py.png"))
```

With `setIconSize()` method we can set the icon size, it expects a `QSize` and you need to add the width and height of the icon.

```
rad1.setIconSize(QSize(40, 40))
```

With this method we can change the font of the QRadioButton.

```
rad1.setFont(QFont("Times", 14))
```

As I have already said that we can use toggled() signal in QRadioButton. In here we have connected the toggled signal to the radio_selected() method or slot.

```
rad1.toggled.connect(self.radio_selected)
```

And this is the method that we have already connected to our signal, basically in this method we want to check the state of the QRadioButton using isChecked() method and according to the state we want to change the text of the label.

```
def radio_selected(self):  
    radio_btn = self.sender()  
    if radio_btn.isChecked():  
        self.label.setText("You have selected :  
{ } ".format(radio_btn.text()))
```