QT6 Buttons

QPushButtons

Командная кнопка является наиболее часто используемым виджетом в любом графическом интерфейсе пользователя. Нажатие (click) кнопки является командой компьютеру выполнить какое-либо действие. Типичными кнопками являются "OK", "Применить", "Отмена", "Закрыть"., Да, Нет и Справка.
Командная кнопка имеет прямоугольную форму и обычно отображает текстовую метку, описывающую ее действие. Можно указать комбинацию клавиш, указав перед нужным символом амперсанд в тексте.
чтобы отобразить кнопку в приложении, вам необходимо создать экземпляр класса QPushButton.

```
from PyQt6.QtWidgets import QApplication, QWidget, QPushButton
import sys

class Window(QWidget):
    def __init__(self):
        super().__init__()
        self.setGeometry(200,200, 700, 400)
        self.setWindowTitle("Python GUI Development")
        self.create_button()

    def create_button(self):
        btn = QPushButton("Click", self)

app = QApplication(sys.argv)
window = Window()
window.show()
sys.exit(app.exec())
```

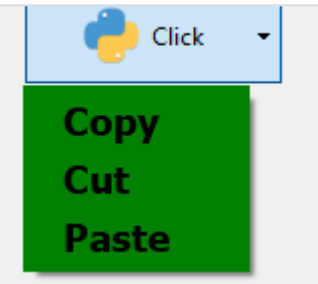
Методы класса:

Метод	Описание
setText()	Изменение текста
setIcon()	Добавление иконки на кнопку btn.setIcon(QIcon("images/python.png"))

Метод	Описание

	Изменение размера иконки
	`from PyQt6.QtCore import QSize`
	`from PyQt6.QtGui import QIcon`
	
	`def create_button(self):`
	`btn = QPushButton("Click", self)`
	`btn.setGeometry(100, 100, 130, 50)`
	
	`btn.setIcon(QIcon('pyqt6lessons\images\python.png'))`
	`btn.setIconSize(QSize(36,36))`

setGeometry()	Настройка положения кнопки,

setMenu()	Всплывающее меню над кнопкой. Сначала создать объект QMenu, класс QMenu связан с модулем QtWidgets, класс QMenu предоставляет виджет меню для использования в строках меню, контекстных меню и других всплывающих меню.
	Виджет меню - это меню выбора. Это может быть как выпадающее меню в строке меню, так и отдельное контекстное меню. Выпадающие меню отображаются в строке меню, когда пользователь щелкает на соответствующем элементе или нажимает указанную комбинацию клавиш.
	`menu = QMenu()`
	`menu.setFont(QFont("Times", 14, QFont.Weight.ExtraBold))`
	`menu.setStyleSheet('background-color:green')`
	`menu.addAction("Copy")`
	`menu.addAction("Cut")`
	`menu.addAction("Paste")`
	`btn.setMenu(menu)`
	 

Метод	Описание
setFont()	Настройка шрифта btn.setFont(QFont("Times", 14, QFont.Weight.ExtraBold))
setCheckable()	Вид кнопки при нажатии меняется. Выделяется и снимается выделение.

QRadioButtons

Кнопка, которую можно включить (установить флажок) или выключить (снять флажок). Переключатели обычно предоставляют
пользователю возможность выбора "из многих". В группе переключателей одновременно может быть установлен только один переключатель, если пользователь выбирает другую кнопку, ранее выбранная кнопка отключается. Существуют различные методы, которые вы можете использовать, например, у нас есть IsChecked(), и он возвращает логическое значение true, если кнопка находится в выбранном состоянии, или у нас есть метод setIcon(), с помощью которого мы можем добавить значок для радиокнопки, а также setText(), который задает текст выбранной кнопки. Также существуют различные сигналы, которые вы можете использовать, например, у нас есть переключаемый сигнал, который используется всякий раз, когда переключатель меняет свое состояние с установленного на снятое и наоборот.

RadioButtons объединенные в одном hbox рассматриваются как зависимые.

```
from PyQt6.QtWidgets import QApplication, QWidget, QHBoxLayout, QRadioButton, QLabel
from PyQt6.QtWidgets import QVBoxLayout
import sys

class Window(QWidget):
    def __init__(self):
        super().__init__()
        self.setGeometry(200,200, 700, 400)
        self.setWindowTitle("Python GUI Development")
        self.create_button()

    def create_button(self):
        hbox = QHBoxLayout()

        self.label = QLabel("", self)
        vbox = QVBoxLayout()
        vbox.addWidget(self.label)
        vbox.addLayout(hbox)
```

```
        rad1 = QRadioButton("Python")
        rad1.toggled.connect(self.radio_selected)
        hbox.addWidget(rad1)
        rad2 = QRadioButton("Java")
        rad2.toggled.connect(self.radio_selected)
        hbox.addWidget(rad2)
        rad3 = QRadioButton("JavaScript")
        rad3.toggled.connect(self.radio_selected)
        hbox.addWidget(rad3)

        self.setLayout(vbox)

    def radio_selected(self):
        radio_btn = self.sender()
        if radio_btn.isChecked():
            self.label.setText(f'Selected: {radio_btn.text()}')

app = QApplication(sys.argv)
window = Window()
window.show()
sys.exit(app.exec())
```

Методы класса:

Метод	Описание
isChecked()	Возвращает True если кнопка выбрана
setChecked()	Переводит кнопку в выбранное состояние
setIcon()	Устанавливает иконку кнопки
setText()	Текст

Revision #7
Created 28 January 2026 07:30:53 by Admin
Updated 31 January 2026 16:06:43 by Admin