

# QT6 designer

Устанавливается при установке pyqt6-tools

У меня вызвался обычной командой

```
(myenv) D:\projects\calculator_long>pyside6-designer.exe
```

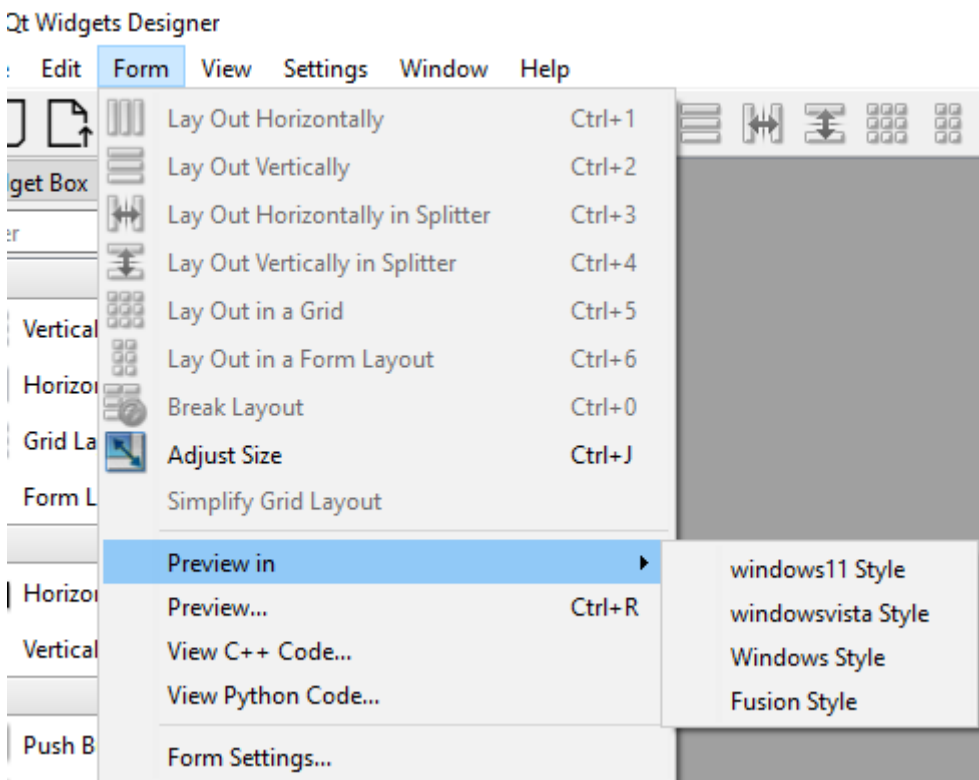
## Типы создаваемых окон:

- 3 типа диалоговых окон,
- Основное окно
- 10 виджетов.

Различаются родительским классом и наличием дополнительных виджетов (кнопки, ...)

## Предпросмотр результата

Блок меню Form - Preview... очень занимательный.

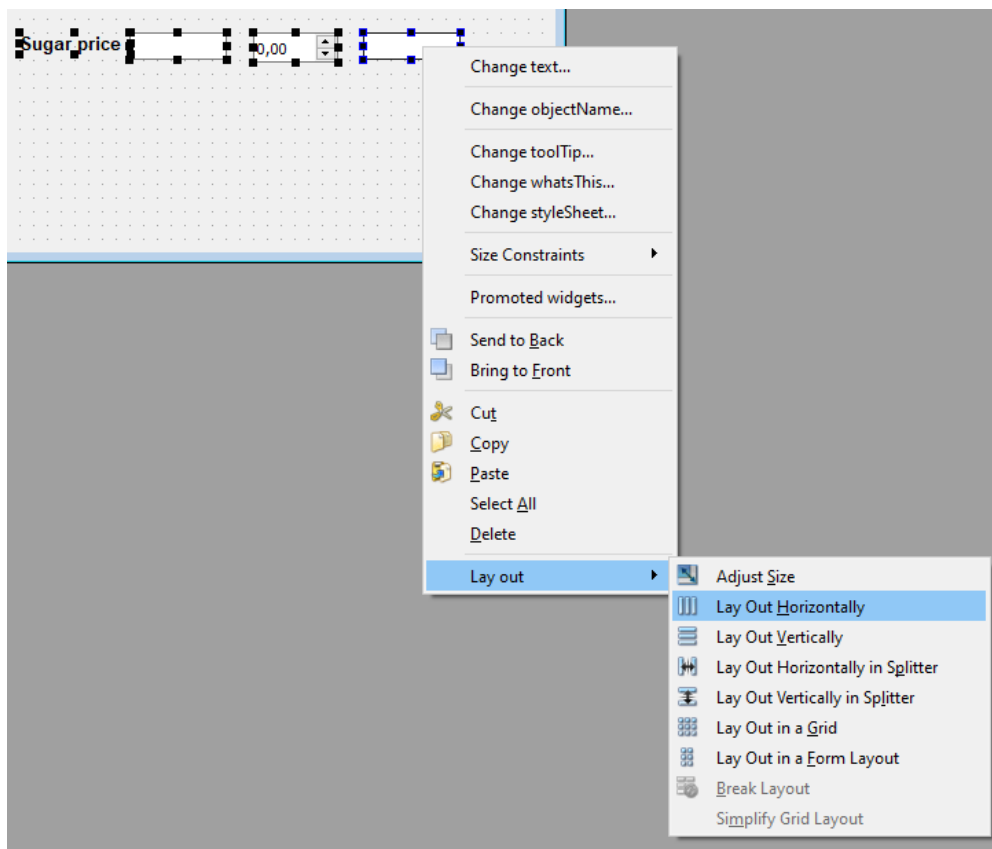


QT использует стили операционных систем, поэтому вид будет отличаться на разных ОС.

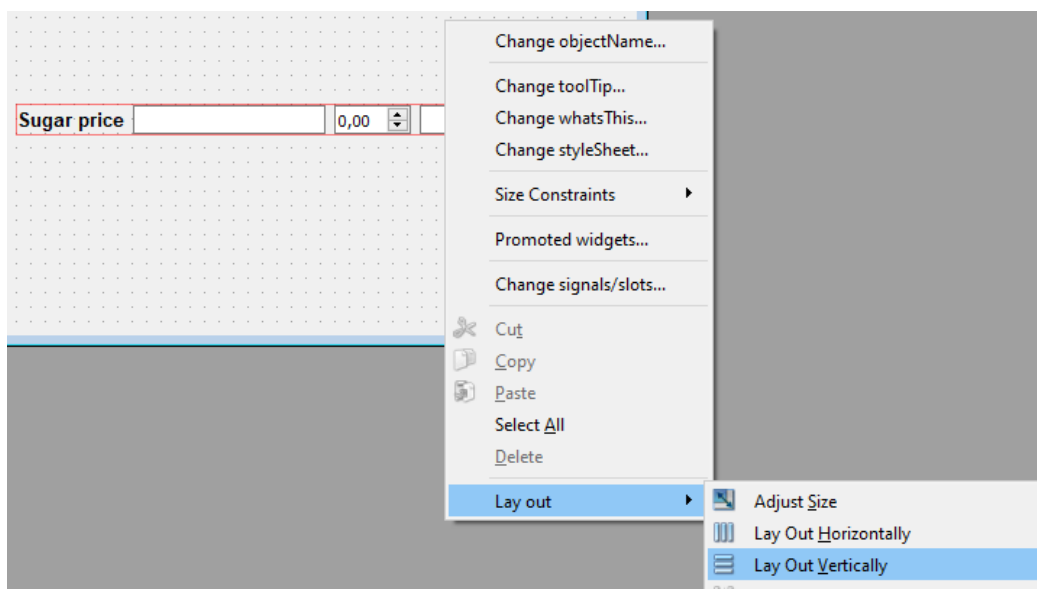
## Слои (Layouts)

Настраивается отдельно тип для всей формы и для группы элементов.

Настройка для группы элементов: выделяются виджеты, ПКМ - Lay Out - Нужный тип.



Для всей формы: ПКМ на пустом месте формы - Lay Out - Нужный тип.



## Использование .ui файла

### Использование в скрипте

Два варианта: преобразование в ru файл и загрузка в ru скрипт ui-файла во время выполнения.

Преобразование в python файл:

```
(myenv) D:\projects\calculator_long>pyuic6 -x testui.ui -o testuicreated.py
```

После этого запуск \*.py файла откроет пользовательский интерфейс. При изменении \*.ui файла необходимо обновить файл исходного кода и связанных процедур.

Загрузка ui файла во время выполнения

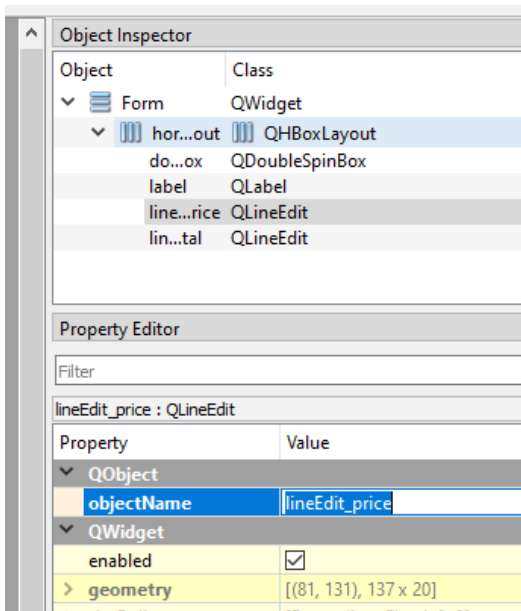
```
from PyQt6.QtWidgets import QApplication, QWidget
import sys
from PyQt6 import uic

class UI(QWidget):
    def __init__(self):
        super().__init__()
        uic.loadUi("WindowUI.ui", self)

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

### **Доступ к виджетам внутри .ui файла**

Для дальнейшего доступа к виджетам из ru скрипта необходимо знать тип виджета и имя объекта. Например, есть виджет типа QLineEdit, имя объекта lineEdit\_price.



Создадим свойство объекта через FindChild

```
class UI(QWidget):
    def __init__(self):
        super().__init__()
        uic.loadUi("double_spin.ui", self)

        self.linePrice = self.findChild(QLineEdit, "lineEdit_price")
```

Дальше с этим свойством работать также как с созданным объектом. Пример:

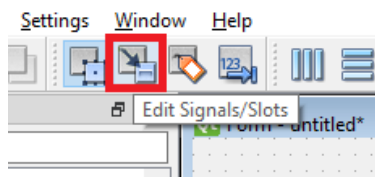
```
class UI(QWidget):
    def __init__(self):
        super().__init__()
        uic.loadUi("double_spin.ui", self)

        self.linePrice = self.findChild(QLineEdit, "lineEdit_price")
        self.doublespin = self.findChild(QDoubleSpinBox, "doubleSpinBox")
        self.doublespin.valueChanged.connect(self.spin_selected)
        self.lineresult = self.findChild(QLineEdit, "lineEdit_total")

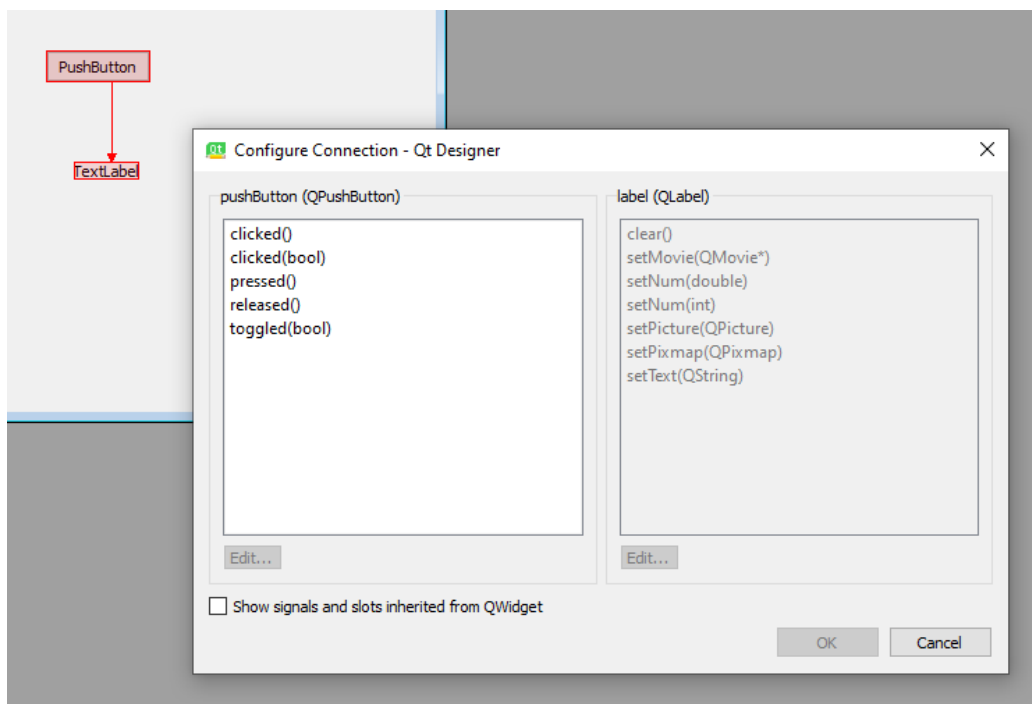
    def spin_selected(self):
        if self.linePrice.text() != 0:
            price = int(self.linePrice.text())
            totalPrice = self.doublespin.value() * price
            self.lineresult.setText(str(totalPrice))
```

## Соединение действий виджетов

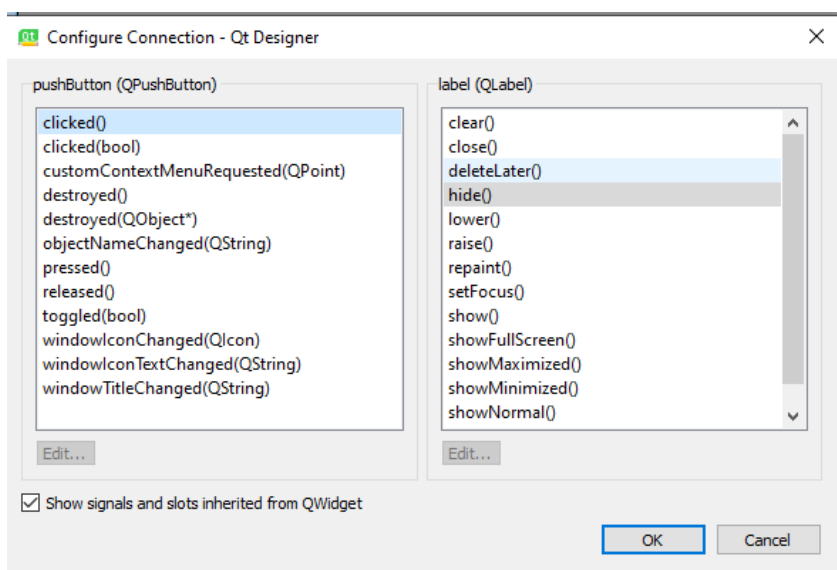
Функционал урезан. Нажать кнопку Edit signals/slots



При наведении мыши виджет становится красным. Если нажать виджет, с которого сигнал будет исходить и, удерживая мышью, переместить указатель на виджет, на который будет влиять сигнал, появится стрелка и элемент выбора сигналов/слотов.



Нужно установить флажок внизу. Выбирается сигнал, слот и все работает даже на превью.



---

Revision #4

Created 28 January 2026 05:51:33 by Admin

Updated 7 February 2026 09:43:54 by Admin