

Скачайте Яндекс Браузер для образования

Скачать

< Урок PyQTGraph

QT. Установка дополнительных компонентов. PyQTGraph

- 1 PyQTGraph
- 2 Проигрывание музыки

Аннотация

В уроке мы научимся пользоваться дополнительными компонентами PyQT на примере PyQTGraph — библиотеки для построения графиков. А еще научимся проигрывать музыку в своих приложениях.

1. PyQTGraph

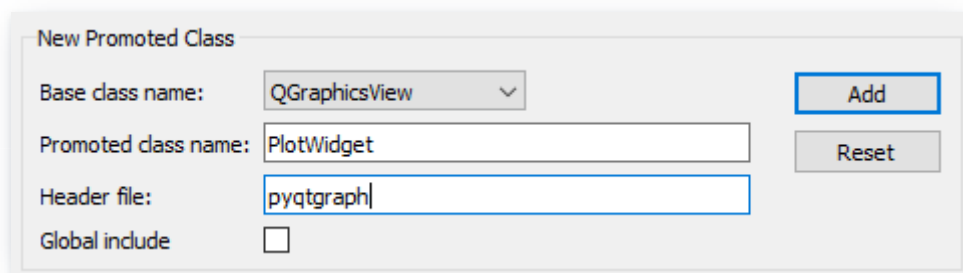
Несмотря на большое количество встроенных виджетов, их не всегда хватает для реализации всех задумок. Хорошая новость заключается в том, что PyQT позволяет создавать собственные виджеты со своими функциями. Некоторые разработчики выкладывают эти библиотеки в Интернет, чтобы их могли использовать другие люди. В этом уроке мы познакомимся с библиотекой **PyQTGraph**, которая включает в себя виджеты для работы с графиками.

Для установки библиотеки выполните стандартную команду в командной строке:

```
pip install pyqtgraph
```

Затем советуем вам настроить QtDesigner для работы с этой библиотекой. Это — необязательное условие, работать с ней можно и «вручную», как мы делали в самом первом уроке. Но это не столь удобно.

1. Открываем новый виджет в QtDesigner и добавляем **Graphics View**.
2. Нажимаем на него правой кнопкой мыши и выбираем вкладку **Преобразовать в...** (Promote to...).
3. Заполняем поля, как показано на картинке:



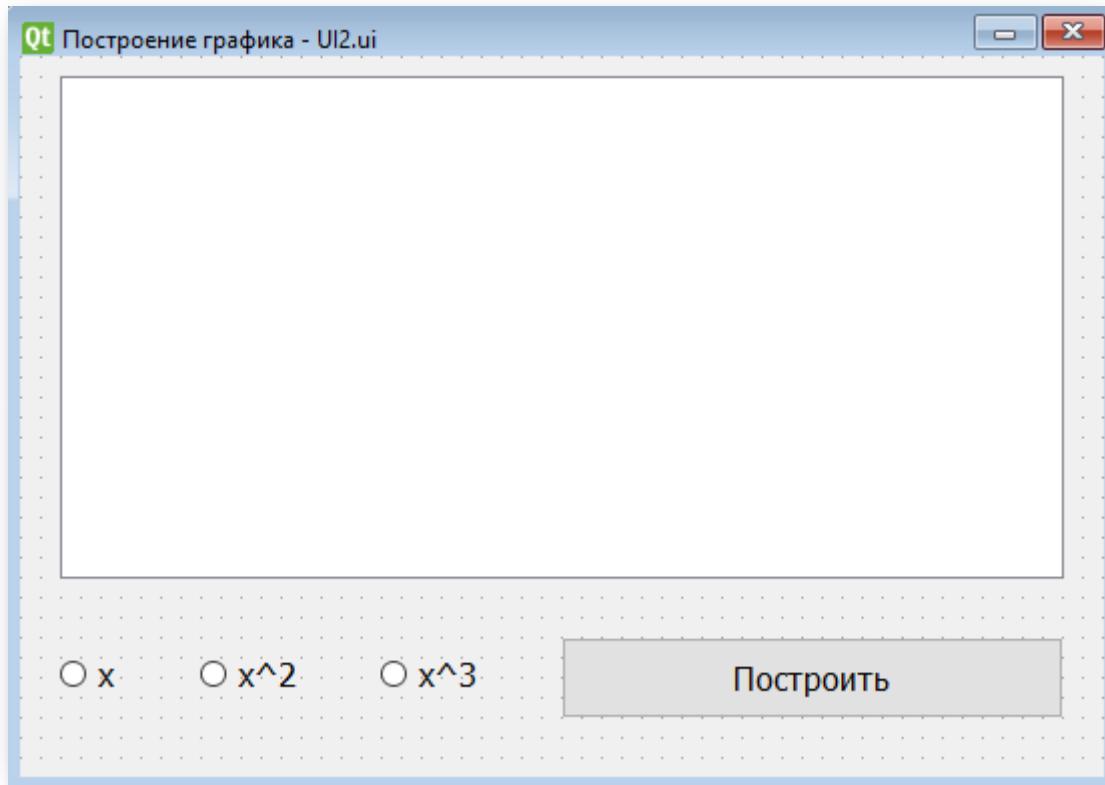
New Promoted Class	
Base class name:	QGraphicsView
Promoted class name:	PlotWidget
Header file:	pyqtgraph
Global include	<input type="checkbox"/>

4. Нажимаем на кнопку «Добавить», а затем — на кнопку «Преобразовать».

Если нам в следующий раз понадобится добавить такой же виджет, то мы сможем выбрать его из выпадающего меню **Преобразовать в....**

Напишем программу, которая будет строить графики одной из трех функций в зависимости от выбранного Radio Button.

Сначала создадим интерфейс в QtDesigner, не забыв при этом выполнить преобразование Graphics View в наш Plot Widget.



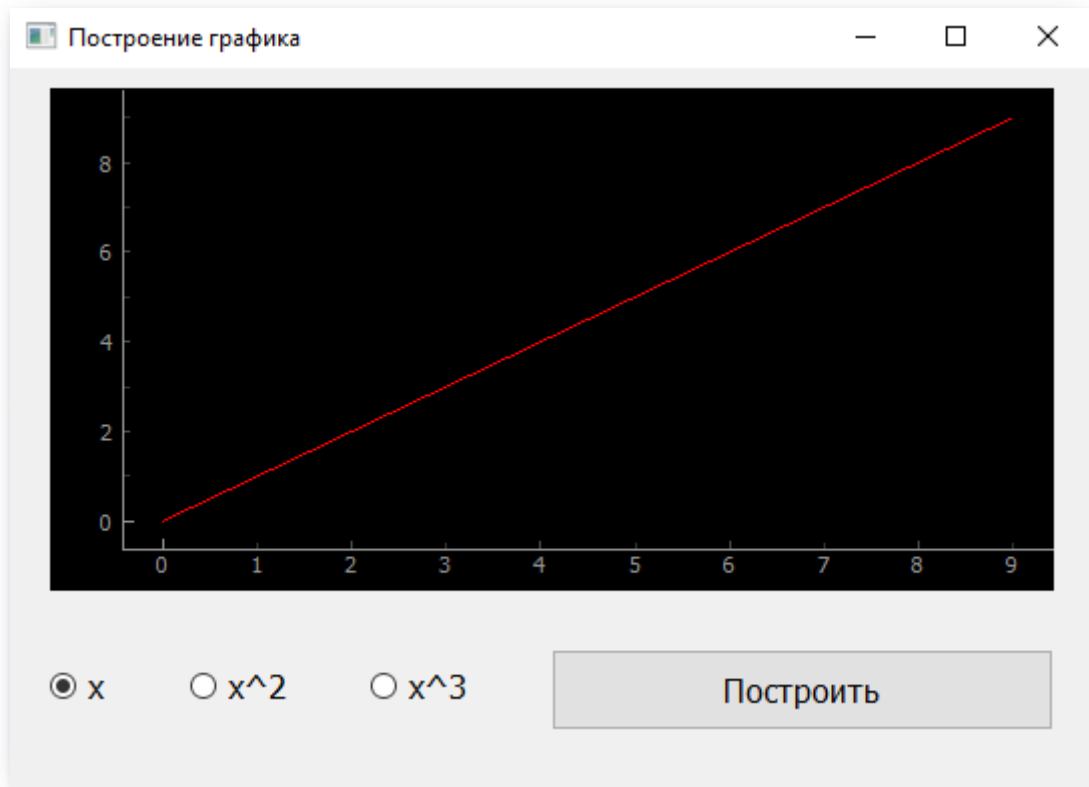
Преобразуем его с помощью **pyuic5** и начнем выполнять построение.

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QMainWindow
from ui import Ui_Form

class MyWidget(QMainWindow, Ui_Form):
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.pushButton.clicked.connect(self.run)

    def run(self):
        if self.radioButton.isChecked():
            self.widget.clear()
            self.widget.plot([i for i in range(10)], [i for i in range(10)], pen='r')
        elif self.radioButton_2.isChecked():
            self.widget.clear()
            self.widget.plot([i for i in range(10)], [i ** 2 for i in range(10)], pen='g')
        elif self.radioButton_3.isChecked():
            self.widget.clear()
            self.widget.plot([i for i in range(10)], [i ** 3 for i in range(10)], pen='b')
```

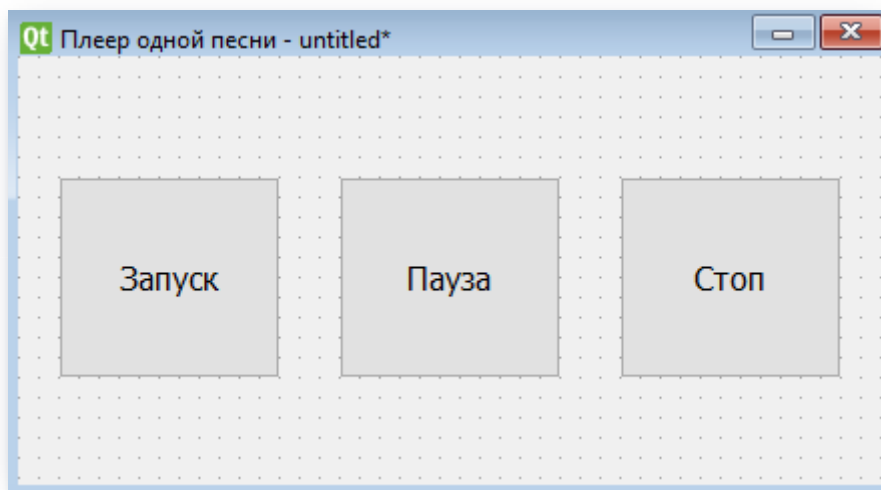
```
app = QApplication(sys.argv)
ex = MyWidget()
ex.show()
sys.exit(app.exec_())
```



Подробнее про модуль `pyqtgraph` можно почитать в его официальной [документации](#).

2. Проигрывание музыки

Часто в программах хочется добавить какую-либо аудио-информацию. Злоупотреблять этим не следует, но работать со звуками надо уметь. Напишем простейшее приложение-плеер, который будет проигрывать одну песню. Создадим интерфейс нашего приложения в QtDesigner.



Для работы с музыкальными файлами нам понадобится импортировать виджет **QtMultimedia**, а чтобы получить полный путь к файлу — **QtCore**.

```
from PyQt5 import QtCore, QtMultimedia
```

Посмотрим, как выглядит наше приложение:

```
from player import Ui_Form
from PyQt5 import QtCore, QtMultimedia
from PyQt5.QtWidgets import QApplication, QMainWindow

class MyWidget(QMainWindow, Ui_Form):
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.load_mp3('/Users/user/Documents/Yandex/FileLesson/test.mp3')
        self.playBtn.clicked.connect(self.player.play)
        self.pauseBtn.clicked.connect(self.player.pause)
        self.stopBtn.clicked.connect(self.player.stop)

    def load_mp3(self, filename):
        media = QtCore.QUrl.fromLocalFile(filename)
        content = QtMultimedia.QMediaContent(media)
        self.player = QtMultimedia.QMediaPlayer()
        self.player.setMedia(content)

app = QApplication(sys.argv)
ex = MyWidget()
ex.show()
sys.exit(app.exec_())
```

В нем всего две функции. Начнем с конца, с функции `load_mp3`. Она отвечает за загрузку музыкального файла и создание объекта **плеера**, в котором эта музыка будет содержаться.

Важно, что сначала мы передаем в эту функцию не название файла, а полный путь, после чего происходит «оборачивание» в формат `QUrl`. Затем создается медиа-объект, который помещается в плеер.

Во время инициализации, после того, как мы подключили наш интерфейс, мы вызываем эту функцию, и в нашем классе появится поле `player`. Затем необходимо лишь настроить кнопки, привязав к каждой из них соответствующую функцию нашего объекта. Функции объекта `player` достаточно очевидны по их названиям.

Справка

Исключительное право на учебную программу и все сопутствующие ей учебные материалы, доступные в рамках сервиса, принадлежат АНО ДПО «Образовательные технологии Яндекса». Воспроизведение, копирование, распространение и иное использование программы и материалов допустимо только с предварительного письменного согласия АНО ДПО «Образовательные технологии Яндекса».

[Пользовательское соглашение.](#)

© 2018 – 2024 ООО «Яндекс»