PyQt: GUI на Python при помощи Qt

Максим Федотов

tequila.lime@gmail.com

07.03.2018, Изучаем Python,
Хакспейс Вега

Введение

- **Qt** («Кьют») — кросплатформенный фреймворк для разработки приложений
(в первую очередь на C++).

- **PyQt** — расширение для Python, обеспечивающее привязку к фреймворку Qt.

Введение

- И то, и другое — строго говоря, не только графический интерфейс, но и доступ к БД, поддержка локализации интерфейса, интеграция с HTML-движком WebKit, поддержка воспроизведения видео и аудио и кое-что ещё.

Введение

- Первый релиз **PyQt** вышел ещё в 1998 году (ср. Python — в 1991)
- Последняя версия на данный момент — PyQt**5** (на базе Qt 5). Только в ней добавлена (в дополнение к поддерживавшимся уже в PyQt4 Windows, Linux и Mac OS X) также поддержка Android и iOS.
- Тем не менее, PyQt**4** остаётся пока, похоже, более популярной, и под неё гораздо больше тьюториалов и справок, есть книжки и т.д.

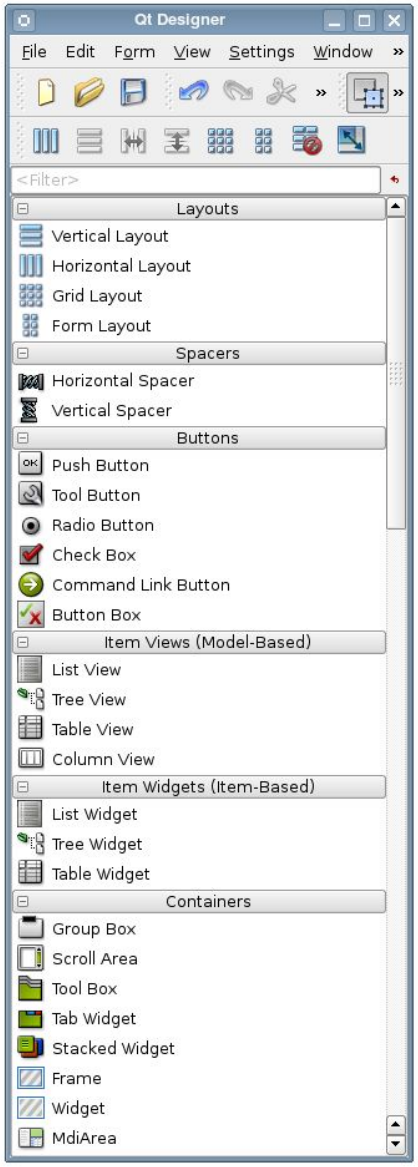
Qt Designer

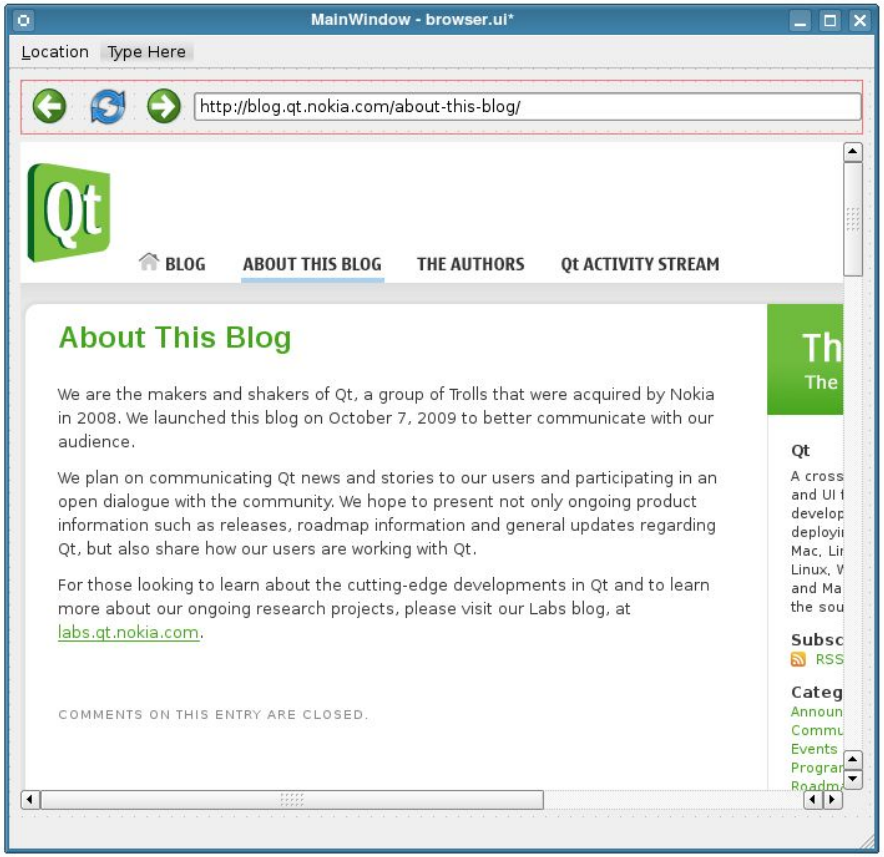
- В PyQt есть интеграция с Qt-шной программой **Qt Designer** (Qt Creator)

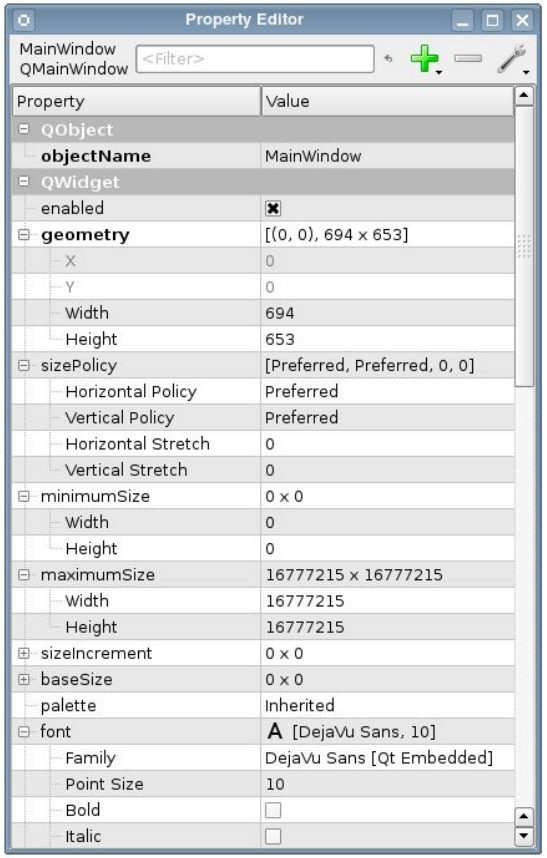
  (дизайнер графического интерфейса пользователя)

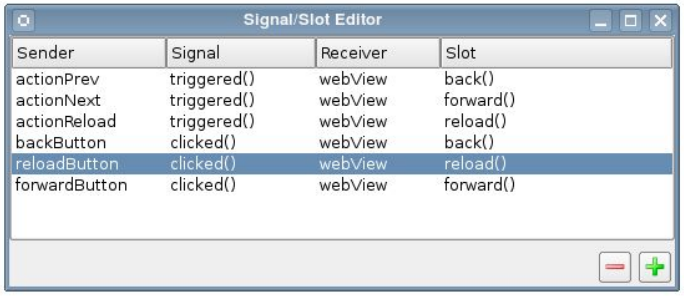
— при помощи приложения ***pyuic*** можно преобразовывать файлы Qt Designer в код на Python.

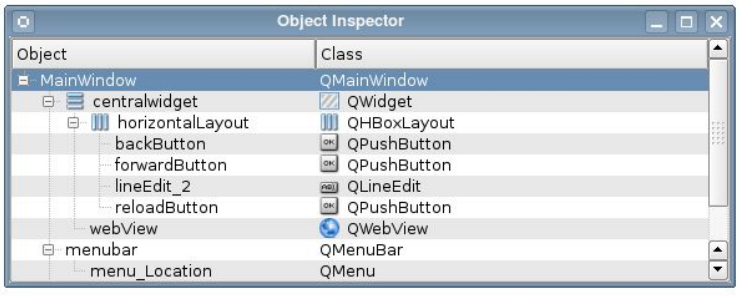
Qt Designer


Qt Designer
File Edit Form View Settings Window
<Filter>
Layouts
Vertical Layout
Horizontal Layout
Grid Layout
Form Layout
Spacers
Horizontal Spacer
Vertical Spacer
Buttons
Push Button
Tool Button
Radio Button
Check Box
Command Link Button
Button Box
Item Views (Model-Based)
List View
Tree View
Table View
Column View
Item Widgets (Item-Based)
List Widget
Tree Widget
Table Widget
Containers
Group Box
Scroll Area
Tool Box
Tab Widget
Stacked Widget
Frame
Widget
MdiArea


MainWindow - browser.ui*
Location Type Here
http://blog.qt.nokia.com/about-this-blog/
BLOG
ABOUT THIS BLOG
THE AUTHORS
Qt ACTIVITY STREAM
About This Blog
We are the makers and shakers of Qt, a group of Trolls that were acquired by Nokia in 2008. We launched this blog on October 7, 2009 to better communicate with our audience.
We plan on communicating Qt news and stories to our users and participating in an open dialogue with the community. We hope to present not only ongoing product information such as releases, roadmap information and general updates regarding Qt, but also share how our users are working with Qt.
For those looking to learn about the cutting-edge developments in Qt and to learn more about our ongoing research projects, please visit our Labs blog, at labs.qt.nokia.com.
COMMENTS ON THIS ENTRY ARE CLOSED.


Property Editor
MainWindow
QMainWindow
<Filter>
Property
Value
QObject
objectName
MainWindow
QWidget
enabled
geometry
[(0, 0), 694 x 653]
X
Y
Width
694
Height
653
sizePolicy
[Preferred, Preferred, 0, 0]
Horizontal Policy
Preferred
Vertical Policy
Preferred
Horizontal Stretch
0
Vertical Stretch
0
minimumSize
0 x 0
Width
0
Height
0
maximumSize
16777215 x 16777215
Width
16777215
Height
16777215
sizeIncrement
0 x 0
baseSize
0 x 0
palette
Inherited
font
A [DejaVu Sans, 10]
Family
DejaVu Sans [Qt Embedded]
Point Size
10
Bold
Italic


Signal/Slot Editor
Sender
Signal
Receiver
Slot
actionPrev triggered() webView back()
actionNext triggered() webView forward()
actionReload triggered() webView reload()
backButton clicked() webView back()
reloadButton clicked() webView reload()
forwardButton clicked() webView forward()


Object Inspector
Object
Class
MainWindow QMainWindow
centralwidget QWidget
horizontalLayout QHBoxLayout
backButton QPushButton
forwardButton QPushButton
lineEdit_2 QLineEdit
reloadButton QPushButton
webView QWebView
menubar QMenuBar
menu_Location QMenu

Установка PyQt5

- Устанавливается через **PIP**:

  pip install PyQt5

  (или *pip3 install PyQt5*, если у Вас Питон обеих версий)

Приложения с оконным интерфейсом

- NB! для приложений с оконным графическим интерфейсом используется расширение файлов не *.py*, а ***.pyw***

— при их запуске не открывается отдельным окном консоль Питона

“Hello, World!” на PyQt5

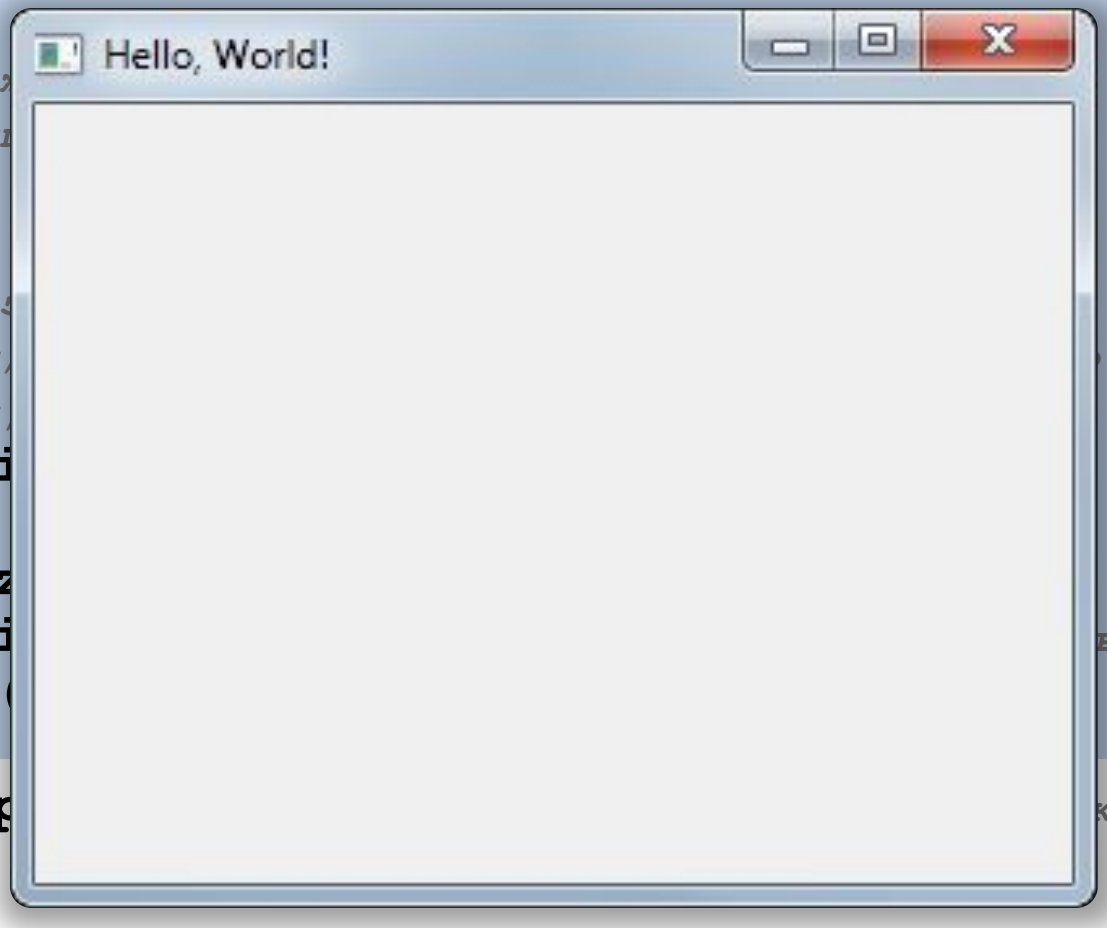
```
import sys
from PyQt5.QtWidgets import QApplication, QWidget

# Каждое прило
# sys.argv - с
application

# QWidget - ба
# пользователя
# без родителя
widget = QWi

widget.resiz
widget.setWi                                    вить заголовок
widget.show(

sys.exit(app                                    кла приложения
```



“Hello, World!” на PyQt4

```
import sys
from PyQt4.QtGui import QApplication, QWidget

# Каждое приложение должно создать объект QApplication
# sys.argv - список аргументов командной строки
application = QApplication(sys.argv)

# QWidget - базовый класс для всех объектов интерфейса
# пользователя; если использовать для виджета конструктор
# без родителя, такой виджет станет окном
widget = QWidget()

widget.resize(320, 240) # изменить размеры виджета
widget.setWindowTitle("Hello, World!") # установить заголовок
widget.show() # отобразить окно на экране

sys.exit(application.exec_()) # запуск основн. цикла приложения
```

Добавляем кнопочку

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget,
                                                QPushButton

application = QApplication(sys.argv)

widget = QWidget()

widget.resize(200, 120)
widget.setWindowTitle("Button test")

btn = QPushButton('Close me', widget) # добавляем кнопку
btn.clicked.connect(QApplication.instance().quit)
# присоединяем к ней метод, который будет выполняться при нажатии
btn.resize(btn.sizeHint())
# устанавливаем размер кнопки;
# sizeHint() подстраивает размер под текст
btn.move(50, 50)# устанавливаем расположение кнопки в окне

widget.show()  # только потом показываем окно!
```

Вызываем свой метод кнопочкой

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton,
                                                    QMessageBox

class MyWidget(QWidget): # создаём на основе стандартного свой виджет с
                                                    блэкджеком и шлюхами

    def __init__(self):
        super().__init__()

        self.initUI()

    def initUI(self): # переопределяем стандартную инициализацию интерфейса
        self.resize(200, 120) # NB! теперь всё через self
        self.setWindowTitle("Button test 2")
```

Вызываем свой метод кнопочкой

```
        btn = QPushButton('Pop up!', self)
        btn.clicked.connect(self.pop_up_hello_world)
    # вешаем на кнопку на кастомный метод (см. ниже)
        btn.resize(btn.sizeHint())
        btn.move(50, 50)

        self.show()

    def pop_up_hello_world(self):
# создаём свой метод (который будет вызываться кнопкой) ВНУТРИ нашего класса
# — тогда можно будет его повесить на кнопку
        QMessageBox.information(self, "Title", "Hello, World!")
        # выплёваем окошко на экран с заданным заголовком и текстом

if __name__ == '__main__':
    app = QApplication(sys.argv)
    my_widget = MyWidget() # создаём экземпляр нашего виджета, и он
                                                    запускается
    sys.exit(app.exec_())
```

Дальнейшее чтение:

- Хороший тьюториал (англоязычный):

http://zetcode.com/gui/pyqt5/

- Reference guide:

http://pyqt.sourceforge.net/Docs/PyQt5/

(но он не слишком полезен, по поводу внутренних классов и методов постоянно отсылает к общему reference guide'у по Qt, а там уже C++ реализация)

PyIstaller

- Приложение, использующее PyQt, можно потом ещё и собрать вместе с интерпретатором и всеми модулями в **исполняемый файл** (не требующий отдельной установки всего этого) для Windows / Linux / Mac OS и др.
  при помощи **PyInstaller**
- См. тьюториал на русском:

https://habrahabr.ru/post/325626/

Спасибо