Qt for Python: past, present, and future!

Dr. Cristián Maureira-Fredes @ TQtC

GHCon2020
What's the deal with **Python**?
What's the connection with Qt?
The story of PySide

2008
Qt4 Development (PySide)

2015
Qt5 Port (PySide2)

2016
Back to the Qt Project

2018
Released (Qt for Python)

2020
Qt6 Release (PySide6)
The Qt5 Challenge

- Porting
- Building from source
- New territory
- Official project
- Spread the word
- Keep the C++ users
import sys
from PySide2.QtWidgets import QApplication, QLabel

# Qt Application
app = QApplication([])

# Qt Widget
label = QLabel("Hello World!")
label.show()

# Executing app
sys.exit(app.exec_())
For C++ Users (1/2)
For C++ Users (2/2)
Webinars & Conferences

• Creating user interfaces with Qt for Python
• Develop your first Qt for Python Application
• Python and C++ interoperability with Shiboken
• doc.qt.io/qtforpython/videos.html
Documentation

- General: doc.qt.io/qtforpython
- Shiboken: doc.qt.io/qtforpython/shiboken2
- Resources: resources.qt.io
Binding generation

Qt Headers → ApiExtractor (libclang) → Shiboken (Generator) → PySide

Typesystem (xml) ← Shiboken (Generator) ← ApiExtractor (libclang) ← Qt Headers

CPython (Wrappers) → Shiboken (Generator)
Setup
Installing PySide

$ pip install pyside2

What does that install?

- PySide2
  - Qt modules
  - pyside2-designer
  - pyside2-rcc
  - pyside2-uic
- Shiboken2 (module)
  - shiboken2
- Shiboken2_Generator (exe)
Installing Shiboken Generator

the simple way

pip install \
--index-url=http://download.qt.io/official_releases/QtForPython/ \
--trusted-host download.qt.io \
shiboken2 pyside2 shiboken2_generator

But one needs to:

- Set CLANG_INSTALL_DIR to the libclang directory
- Add to PATH a Qt bin path with the same version
- Add to LD_LIBRARY_PATH the Qt lib path with the same version
Installing Shiboken Generator the hard way?

- Set `CLANG_INSTALL_DIR` to the libclang directory

```
python setup.py install
# there are many other options!
```

doc.qt.io/qtforpython/gettingstarted.html
Look at some code

- Widgets (.ui, .qrc)
- QML
- Matplotlib and Pandas interaction
- Python Bindings from a C++ project
- Extend Qt/C++ project with Python

[maureira.xyz/qtvts2020/code](maureira.xyz/qtvts2020/code)
Qt for Python in Qt6 times

$ pip install pySide6
Naming

• Can you think of a new name?
• ...believe me, it's difficult.
• we decided to follow Qt numbering.
Adding optional features (1/3)

```python
from __feature__ import snake_case, true_property
```
# Normal code

```python
# ...
table = QTableWidget()
table.setColumnCount(2)

button = QPushButton("Add")
button.setEnabled(False)

layout = QVBoxLayout()
layout.addWidget(table)
layout.addWidget(button)
```

```python
# ...
from __feature__ import snake_case

table = QTableWidget()
table.set_column_count(2)

button = QPushButton("Add")
button.set_enabled(False)

layout = QVBoxLayout()
layout.add_widget(table)
layout.add_widget(button)
```
Adding optional features (3/3)

# Normal code

table = QTableWidget()
button = QPushButton("Add")
button.setEnabled(False)

layout = QVBoxLayout()
layout.addWidget(table)
layout.addWidget(button)

from __feature__ import (snake_case, true_property)

table = QTableWidget()
table.column_count = 2

button = QPushButton("Add")
button.enabled = False

layout = QVBoxLayout()
layout.addWidget(table)
layout.addWidget(button)
QML

- Remove type limit registration
- Different type registration
- QML_ELEMENT equivalent

```cpp
#ifndef PYSIDE_MAX_QML_TYPES
// QML engine creates objects by calling a function
// with one argument. Bypass while creating a factory
// function for each time at compile-time.
# define PYSIDE_MAX_QML_TYPES 50
#endif
```
QML

- Remove type limit registration
- **Different type registration**
- QML_ELEMENT equivalent

```cpp
// what we had
qmlRegisterType()

// what we added
qmlRegisterSingletonType()
qmlRegisterUncreatableType()
```
QML

- Remove type limit registration
- Different type registration
- QML_ELEMENT equivalent

```qml
QML_IMPORT_NAME = "com.library.name"
QML_IMPORT_MAJOR_VERSION = 1
QML_IMPORT_MINOR_VERSION = 0 # Optional

@QmlElement
class ClassForQml(QObject):
    # ...

import com.library.name 1.0

ClassForQml {
    // ...
}
```
Shiboken

- Typesystem extra options
- Reduce the effort for the workflow
- Extend C++ support
Collaborations

- Approach compatible projects
- Increase priority
- Improve the collaboration process
- Community messaging platforms
What's next?

- Currently in active development 📚
- Improvements to the documentation 📚
- A campaign to add more examples started 🤖
- Tooling to improve our lives 🔧
but most importantly...

What do you need?

we are community driven.

🌟

bugreports.qt.io
Let's talk!
pyside.org

IRC
#qt-pyside on Freenode
Matrix
#qt-pyside:matrix.org

Telegram
t.me/qtforpython

Gitter
gitter.im/PySide/pyside2