

Python 101

PyQt

Clase de hoy

- Qt - Introducción
- Empezando: QApplication, QWidget, QMessageBox
- Teoría de Señales y Retrollamadas
- Miscelánea de Controles
- Widgets personalizados

Datos útiles

Documentación:

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

Ejemplos y tutoriales:

<http://zetcode.com/gui/qt5/>

<https://wiki.python.org/moin/PyQt/Tutorials>

Alternativas

Ver: <https://wiki.python.org/moin/GuiProgramming>

Las más populares (algunas con soporte multiplataforma, otras no)

- GTK: <http://www.pygtk.org/>
- PySide: <https://wiki.qt.io/PySide>

Info

PyQt5 es un set de instrucciones Python para aplicaciones para el framework Qt5 de Digia. Está disponible para 2.x y 3.x.

Sitio oficial: www.riverbankcomputing.co.uk/news.

Es desarrollado por Riverbank Computing.

Está implementado sobre módulos Python.

620 clases.

6000 funciones y métodos.

Multiplataforma.

Doble licencia. GPL o comercial.

https://es.wikipedia.org/wiki/GNU_General_Public_License



PyQt4 and PyQt5 differences

PyQt5 no es compatible con PyQt4; hay muchas diferencias significantes sobre PyQt5. Sin embargo, no es muy diferente de ajustar el viejo código con la nueva biblioteca de QT. Las diferencias están, entre otras, en lo siguiente:

Los modulos Python fueron reorganizados. Algunos módulos fueron eliminados (QtScript), otros fueron divididos en submodulos (QtGui, QtWebKit).

Nuevos modulos se agregaron QtBluetooth, QtPositioning, or Enginio.

PyQt5 soporta únicamente señales “nuevo estilo” e intercambio por slots. Las llamadas a SIGNAL y SLOT no son más soportadas.

PyQt5 no soporta ninguna parte de QtAPI que fue marcada como deprecada o obsoleta en Qt v5.0

Modulos QT

- QtCore
- QtGui
- QtWidgets
- QtMultimedia
- QtBluetooth
- QtNetwork
- QtPositioning
- Enginio
- QtWebSockets
- QtWebKit
- QtWebKitWidgets
- QtXml
- QtSvg
- QtSql
- QtTest

Events and signals in PyQt5

Todas las aplicaciones gráficas son basadas en eventos. Los Eventos son generados mayormente por el usuario de la aplicación, aunque también pueden ser generados por otros estímulos como ser, internet, un timer, un administrador. Cuando llamamos a `exec_()`, la aplicación entra en lo que se denomina “main loop”. El main loop toma eventos y los envía a los objetos.

- event source
- event object
- event target

Event source es el que cambia de estado, genera eventos. El evento event encapsula la información. El *event target* es el objeto a ser modificado. El event delega el funcionamiento al target.

PyQt5 tiene un único mecanismo de señal y slot para manejar eventos. Estos son utilizados para comunicarse entre objetos. Una señal es emitida cuando un evento particular sucede. Un slot puede ser cualquier python callable (`__callable__`). Un slot es llamado cuando su señal de “connectado” es emitida.