PyQt5

* **PyQt is** a library that lets **you use** the Qt GUI framework from **Python**
* Qt itself **is** written in C++. By using it from **Python**, **you can** build applications much more quickly while not sacrificing much of the speed of C++.
* **PyQt5** refers to the most recent version 5 of Qt.
* **PyQt5** is implemented as a set of Python **modules**.
* It has over 620 classes and 6000 functions and methods.
* It is a multiplatform toolkit which runs on all major operating systems, including Unix, Windows, and Mac OS

**PyQt5 Modules**

<https://doc.qt.io/qt-5/qtmodules.html>

QObject:

* The **QObject** class is the base class of all **Qt** objects.
* **QObject** is the heart of the **Qt** Object Model.
* The central feature in this model is a very powerful mechanism for seamless object communication called signals and slots.
* You can connect a signal to a slot with connect() and destroy the connection with disconnect().

QtCore

* This module contains the core non-GUI functionality.
* This module is used for working with time, files and directories, various data types, streams, URLs, mime types, threads or processes.
* Classes in QCore
  + QRect
    - The **QRect** class provides a collection of functions that return the various rectangle coordinates, and enable manipulation of these. **QRect** also provides functions to move the rectangle relative to the various coordinates.
  + QSize

QtGui

* deals with the graphical elements.
* contains classes for windowing system integration, event handling, 2D graphics, basic imaging, fonts and text.
* Classes used inside QtGui
  + QIcon
    - Using this class to set/change the icon

QtWidgets

* contains classes that provide a set of UI elements to create classic desktop-style user interfaces.
* Classes used inside QtWidgets
  + QMainWindow
    - Provides a main application window
    - This enables to create a classic application skeleton with a statusbar, toolbars, and a menubar
  + QDialog
    - It is the popup windows that appear when we are trying to do something on the main window
  + QApplication
    - manages the GUI application’s control flow (the order in which the computer executes statements in a script) and main settings.
  + QPushButton
    - Takes 2 parameters, (“string that will be displayed”, self)
  + QGroupBox
    - can group widgets, it provides a frame, title on top and it can display a multiple of widgets inside.
    - It typically has a title and a border.
    - Any **PyQt** widget can be added into the group box.
  + QHBoxLayout
    - Lines up widgets horizontally
  + QVBoxLayout
    - Lines up widgets vertically
  + QGridLayout
    - **QGridLayout** is the most universal layout class.
    - It divides the space into rows and columns.

QtMultimedia

* contains classes to handle multimedia content and APIs to access camera and radio functionality.

QtBluetooth

* contains classes to scan for devices and connect and interact with them.

QtNetwork

* contains the classes for network programming.
* These classes facilitate the coding of TCP/IP and UDP clients and servers by making the network programming easier and more portable.

QtPositioning

* contains classes to determine a position by using a variety of possible sources, including satellite, Wi-Fi, or a text file.

Enginio

* implements the client-side library for accessing the Qt Cloud Services Managed Application Runtime.

QtWebSockets

* contains classes that implement the WebSocket protocol.

QtWebEngine

* provides lasses for integrating QML Web Engine objects with Python

QtWebEngineCore

* contains the core Web Engine classes.

QtWebEngineWidgets

* contains the Chromium based web browser.

QtXml

* contains classes for working with XML files.
* This module provides implementation for both SAX and DOM APIs.

QtSvg

* provides classes for displaying the contents of SVG files.
* Scalable Vector Graphics (SVG) is a language for describing two-dimensional graphics and graphical applications in XML.

QtSql

* provides classes for working with databases.

QtTest

* contains functions that enable unit testing of PyQt5 applications.

Other methods

Import sys

* The **sys** module in **Python** provides various functions and variables that are used to manipulate different parts of the **Python** runtime environment.
* It allows operating on the interpreter as it provides access to the variables and functions that interact strongly with the interpreter.