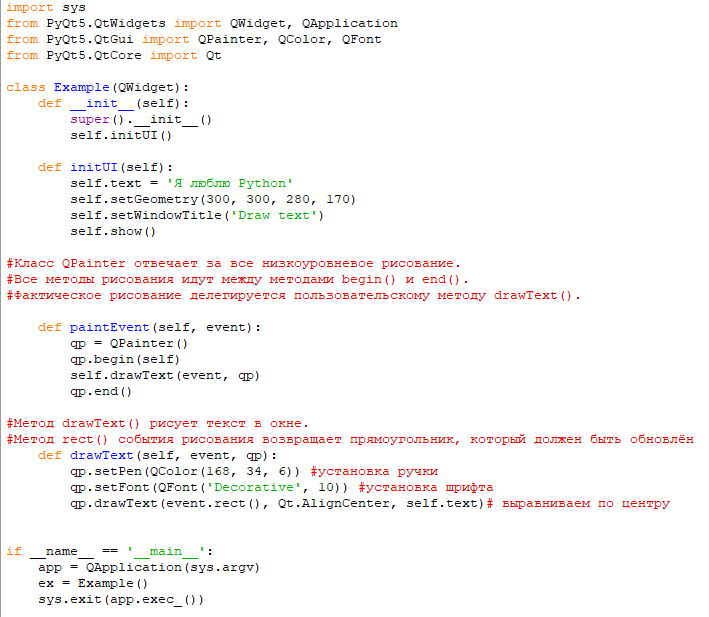
**Рисование в PyQt5**

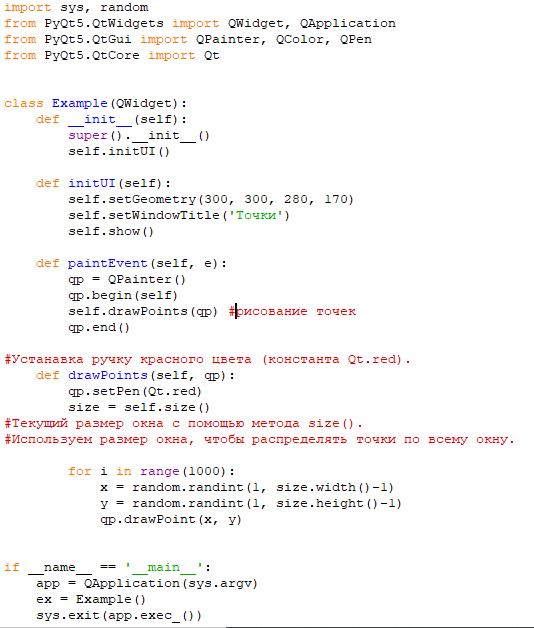
Система рисования PyQt5 способна обрабатывать векторную графику, изображения и шрифты. Рисование необходимо в приложениях, когда мы хотим изменить или улучшить существующий виджет, или мы создаём пользовательский виджет с нуля. Чтобы сделать рисунок, мы используем API рисования, предоставленное инструментарием PyQt5.

Рисование делается в рамках метода paintEvent(). Код рисования размещается между методами begin() и end() объекта QPainter. Он выполняет низкоуровневое рисование на виджетах и других элементах.

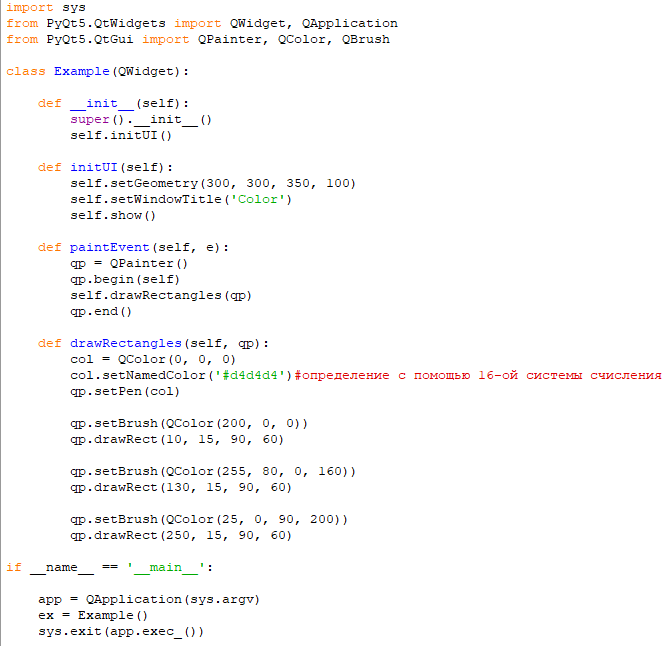
***Рисование текста***



***Рисование точек***



***Цвет***



***QPen***

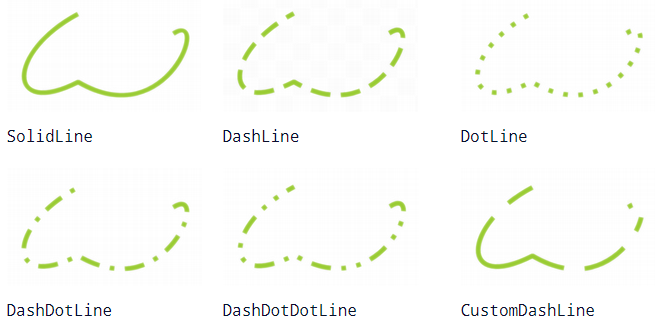
Ручка имеет [style()](https://doc.qt.io/qtforpython/PySide2/QtGui/QPen.html#PySide2.QtGui.PySide2.QtGui.QPen.style), [width()](https://doc.qt.io/qtforpython/PySide2/QtGui/QPen.html#PySide2.QtGui.PySide2.QtGui.QPen.width), [brush()](https://doc.qt.io/qtforpython/PySide2/QtGui/QPen.html#PySide2.QtGui.PySide2.QtGui.QPen.brush), [capStyle()](https://doc.qt.io/qtforpython/PySide2/QtGui/QPen.html" \l "PySide2.QtGui.PySide2.QtGui.QPen.capStyle" \o "PySide2.QtGui.PySide2.QtGui.QPen.capStyle) и [joinStyle()](https://doc.qt.io/qtforpython/PySide2/QtGui/QPen.html#PySide2.QtGui.PySide2.QtGui.QPen.joinStyle).

Стиль пера определяет тип линии.  Ширина пера может быть указана с точностью как целое число ([width()](https://doc.qt.io/qtforpython/PySide2/QtGui/QPen.html" \l "PySide2.QtGui.PySide2.QtGui.QPen.width" \o "PySide2.QtGui.PySide2.QtGui.QPen.width)) и с плавающей точкой ([widthF()](https://doc.qt.io/qtforpython/PySide2/QtGui/QPen.html" \l "PySide2.QtGui.PySide2.QtGui.QPen.widthF" \o "PySide2.QtGui.PySide2.QtGui.QPen.widthF)). Нулевая линия обозначает косметическую ручку (ширина пера всегда рисуется шириной в один пиксель.

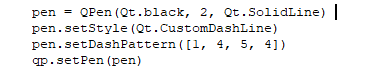
Различные параметры могут быть легко изменены с помощью соответствующего [setStyle()](https://doc.qt.io/qtforpython/PySide2/QtGui/QPen.html#PySide2.QtGui.PySide2.QtGui.QPen.setStyle), [setWidth()](https://doc.qt.io/qtforpython/PySide2/QtGui/QPen.html#PySide2.QtGui.PySide2.QtGui.QPen.setWidth), [setBrush()](https://doc.qt.io/qtforpython/PySide2/QtGui/QPen.html" \l "PySide2.QtGui.PySide2.QtGui.QPen.setBrush" \o "PySide2.QtGui.PySide2.QtGui.QPen.setBrush), [setCapStyle()](https://doc.qt.io/qtforpython/PySide2/QtGui/QPen.html" \l "PySide2.QtGui.PySide2.QtGui.QPen.setCapStyle" \o "PySide2.QtGui.PySide2.QtGui.QPen.setCapStyle) и [setJoinStyle()](https://doc.qt.io/qtforpython/PySide2/QtGui/QPen.html" \l "PySide2.QtGui.PySide2.QtGui.QPen.setJoinStyle" \o "PySide2.QtGui.PySide2.QtGui.QPen.setJoinStyle) функций.



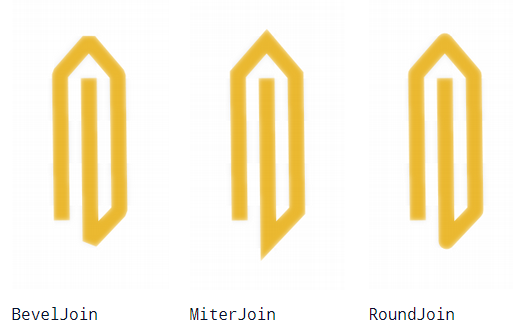
Qt предоставляет несколько встроенных стилей, представленных PenStyle:



Также можно указывать пользовательский шаблон тире, используя [setDashPattern()](https://doc.qt.io/qtforpython/PySide2/QtGui/QPen.html#PySide2.QtGui.PySide2.QtGui.QPen.setDashPattern) функцию, которая преобразует стиль пера CustomDashLine. Аргумент шаблона: нечётные числа определяют сплошную линию, чётные числа – промежутки. Чем больше число, тем больше промежуток или штрих.



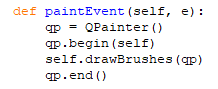
Стиль соединения определяет, как можно нарисовать соединения между двумя связанными линиями [QPainter](https://doc.qt.io/qtforpython/PySide2/QtGui/QPainter.html" \l "PySide2.QtGui.QPainter" \o "PySide2.QtGui.QPainter). Стиль соединения применяется только к широким линиям. PenJoinStyle. Перечисление предоставляет следующие стили:

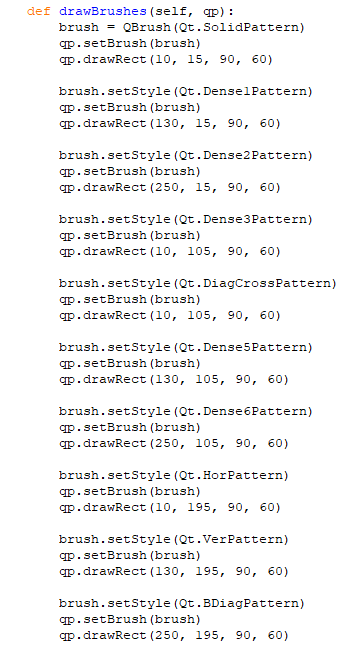


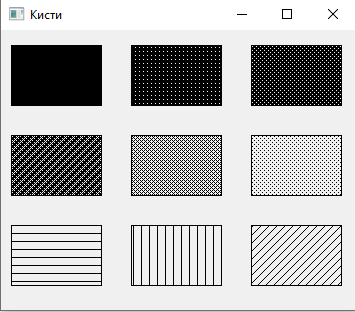
Самостоятельно: нарисовать 5 линий с различными настройками (цветом, шириной, стилем).

***QBrush***

QBrush – это элементарный графический объект. Он используется для рисования фона графических форм, таких как прямоугольники, эллипсы или многоугольники. Кисть может быть трёх разных типов: предопределённая кисть, градиент, образец текстуры.







1. Самостоятельно нарисовать любую блок схему с помощью графики в Python.
2. Нарисовать копир с анимацией работы.

