Человеко-машинное взаимодействие

Лекция 7
Мерзлякова Екатерина Юрьевна
к.т.н. доцент ПМиК

Контекстно-независимое представление растровых изображений

QImage scanLine()	GIF BMP JPG XMP XBM PNG

Qimage img(320, 240, QImage::Format_RGB32) //32- глубина цвета.

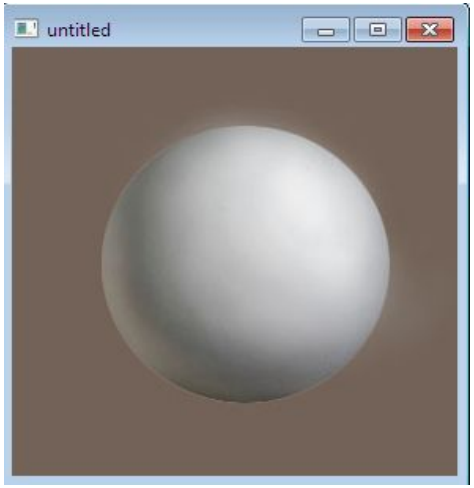
QImage img (“D:\\1.jpg”);

QImage img;
img.load(“D:\\1.jpg”);

img.save(“1.jpg”,”JPG”);

Контекстно-независимое представление растровых изображений

```
#include "mainwindow.h"
#include <QApplication>
#include <QPainter>
#include <QDebug>


class MyWindow: public QMainWindow{
protected:
    virtual void paintEvent(QPaintEvent * e){
        QMainWindow::paintEvent(e);
        QPainter painter(this);
        QImage img("E:\\1.jpg");
        painter.drawImage(0,0,img);
    }
};


int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MyWindow w;
    QImage img("E:\\1.jpg");
    w.setGeometry(50,50,img.width(),img.height());
    w.show();
    return a.exec();
}
```



QPainter::drawImage( )

Контекстно-независимое представление растровых изображений

```
fill();
```

```
QImage img("E:\\1.jpg");
QRgb value;
value = qRgb(122, 163, 39);
img.fill(value);
painter.drawImage(0,0,img);
```

Контекстно-независимое представление растровых изображений

pixel(x,y)

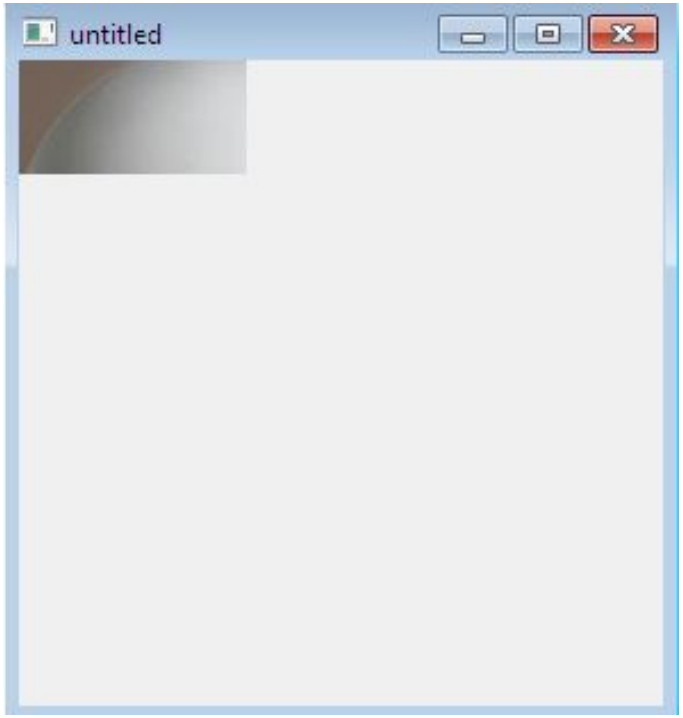
```
QRgb rgb = img.pixel(250,100);
```

setPixel(x,y,rgb);

```
QRgb rgb = qRgb(200,100,0);
img.setPixel(20,50,rgb);
```

Контекстно-независимое представление растровых изображений

```
painter.drawImage(0,0,img,60,60,100,50);
```



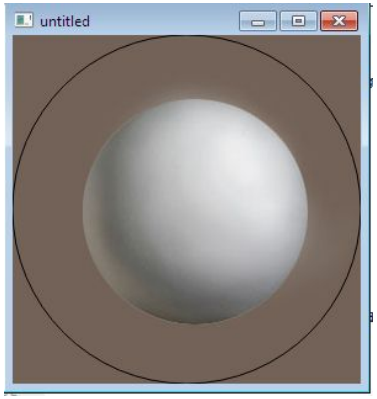
invertPixels()

scaled()

mirrored()

Контекстно-независимое представление растровых изображений

```
class MyWindow: public QMainWindow{
protected:
    virtual void paintEvent(QPaintEvent * e){
        QMainWindow::paintEvent(e);
        QImage img("E:\\1.jpg");
        QPainter painter;
        painter.begin(&img);
        painter.initFrom(this);
        painter.setRenderHint(QPainter::Antialiasing,true);
        painter.drawEllipse(0,0,img.width(),img.height());
        painter.end();

        painter.begin(this);
        painter.drawImage(0,0,img);
        painter.end();

    }
};
```



Контекстно-зависимое представление растровых изображений

QPixmap

```
QPixmap pix(300,300);
```

QPixmap::defaultDepth()

```
QPixmap pix("E:\\1.jpg");
```

load() save()

Контекстно-зависимое представление растровых изображений

**QPainter::drawPixmap(
)**

```
QPainter painter(this);
QPixmap pix("D:\\q\\pict\\untitled\\1.jpg");
painter.drawPixmap(0,0,pix);
```

```
QPainter painter(this);
QPixmap pix("D:\\q\\pict\\untitled\\1.jpg");
QRect r(0,0, pix.width(), pix.height()/3);
painter.drawPixmap(r,pix);
```

Контекстно-зависимое представление растровых изображений

QPixmapCache

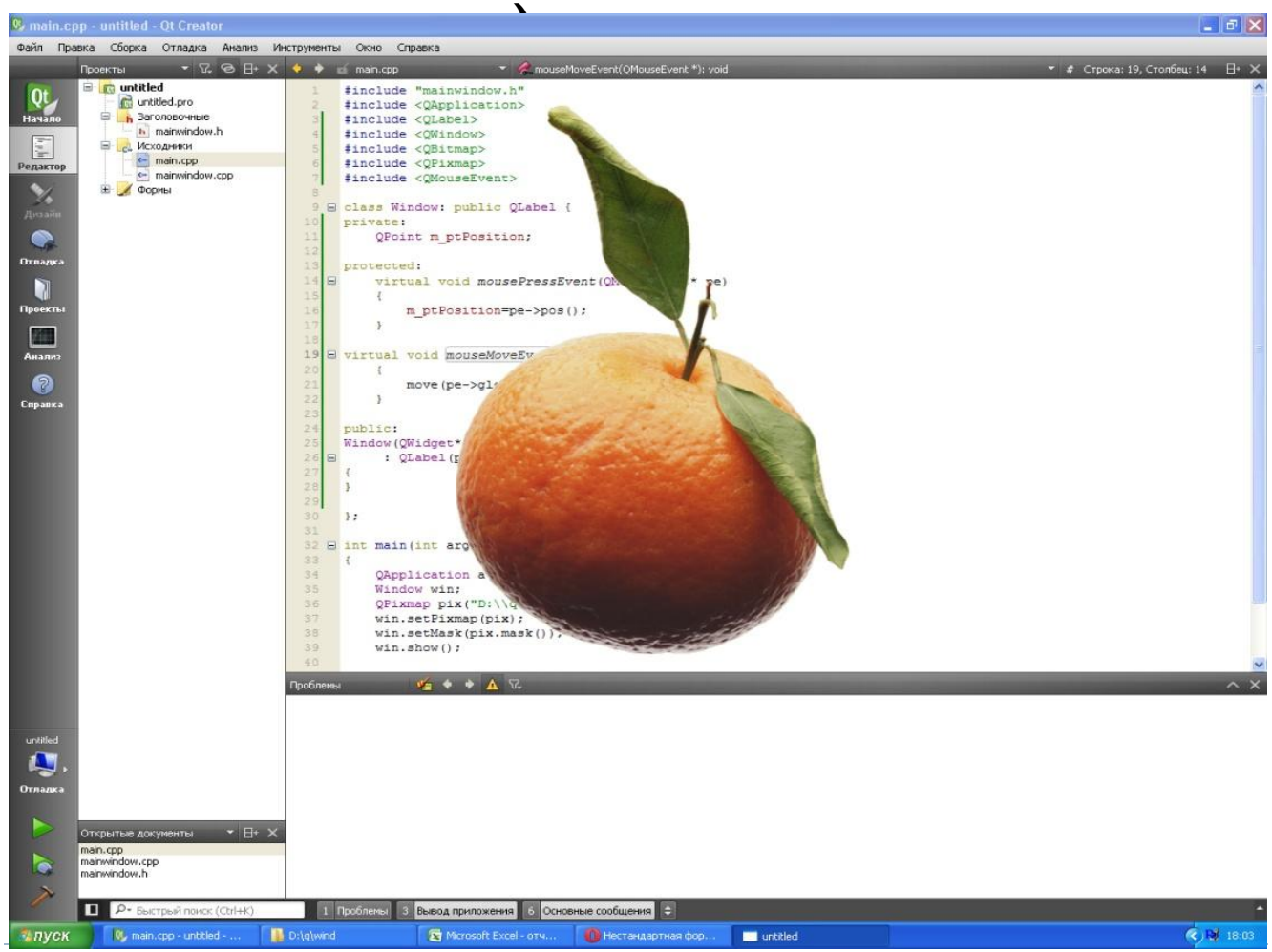
insert()

find()

Контекстно-зависимое представление растровых изображений

setMask(

)

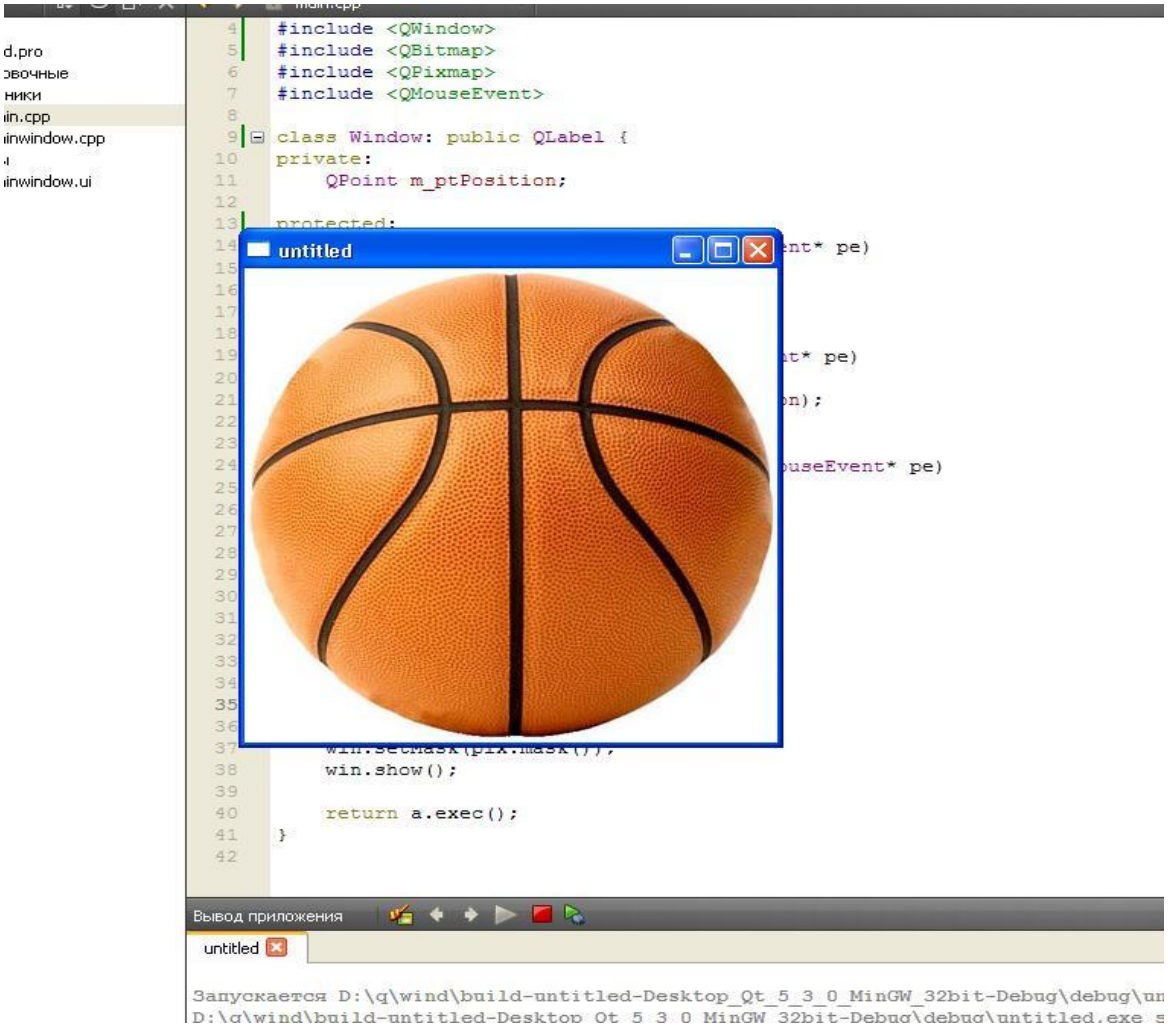


Контекстно-зависимое представление растровых изображений

```
30
31  int main(int argc, char *argv[])
32  {
33      QApplication a(argc, argv);
34      Window win;
35      QPixmap pix("D:\\q\\wind\\3.png");
36      win.setPixmap(pix);
37      win.setMask(pix.mask());
38      win.show();
39
40      return a.exec();
```

```
#include "mainwindow.h"
#include <QApplication>
#include <QLabel>
#include <QWindow>
#include <QBitmap>
#include <QPixmap>
#include <QMouseEvent>

class Window: public QLabel {
private:
    QPoint m_ptPosition;

protected:
    virtual void mousePressEvent(QMouseEvent* pe)
    {
        m_ptPosition=pe->pos();
    }

    virtual void mouseMoveEvent(QMouseEvent* pe)
    {
        move(pe->globalPos() - m_ptPosition);
    }

    virtual void mouseDoubleClickEvent(QMouseEvent* pe)
    {
        exit(0);
    }

};
```

Контекстно-зависимое представление растровых изображений



Контекстно-зависимое представление растровых изображений

```
public:
Window(QWidget* pwgt = 0)
        :QLabel(pwgt, Qt::FramelessWindowHint |
Qt::Window)
 {}
```

Использование каскадных стилей документа

CSS *.qss

QApplication::setStyleSheet()

QWidget::setStyleSheet()

a.setStyleSheet("*описание стиля*")

```
#include "mainwindow.h"
#include <QApplication>
#include <QFile>
#include <QStyle>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;

 QFile file("D:\\q\\styl\\s1.qss");
 file.open(QFile::ReadOnly);
 QString strCSS = QLatin1String(file.readAll());
 a.setStyleSheet(strCSS);
 w.show();
    return a.exec();
}
```

Использование каскадных стилей документа

селектор {свойство: значение}

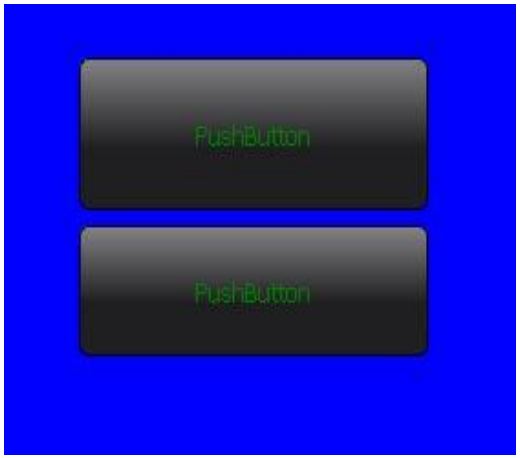
```
QPushButton {color: blue}
```

```
QLabel {
      color: black;
      background-color: red;
}
```

```
QLabel {
            color: rgb(255,0,0);
            background-color: #FFFFFF;
}
```

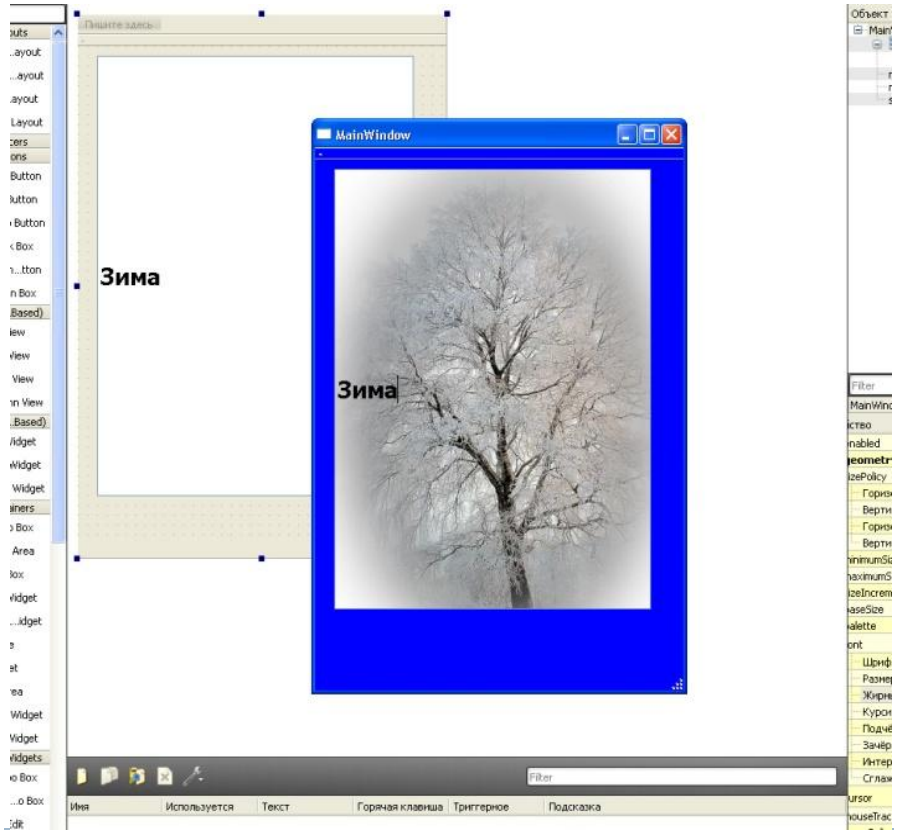
Использование каскадных стилей документа

```
QPushButton {
color: green;
border: 1px solid black;
border-radius: 5px;
background: qlineargradient(x1:0, y1:1, x2:0, y2: 0,
                            stop:1 rgb(133,133,135),
                            stop:0.4 rgb(31,31,33));
}
```



Использование каскадных стилей документа

```
QLineEdit {
background-image: url(D:/q/1.png);
}
QMainWindow {
background-color: blue;
}
```



Использование каскадных стилей документа

```
QLineEdit, QLabel, QPushButton {color: red}

.PushButton {color: red}


QLabel#MyLabel
```

Использование каскадных стилей документа

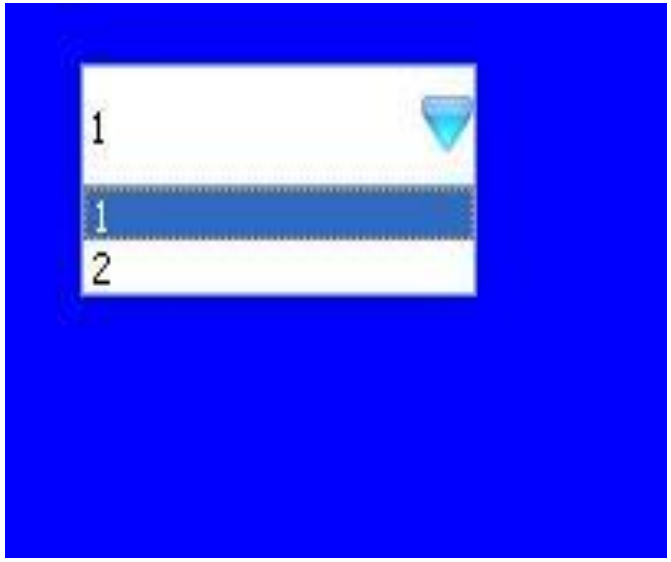
```
Namespase My {
Class My : public QWidget {…..}; }
```

```
/*.qss*/
My {
    color: red;
    background-color: blue
}
```

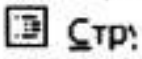
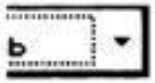
:/* коммент */

Использование каскадных стилей документа

```
QComboBox::drop-down {image: url(D:/q/styl/3.png)}
```



Использование каскадных стилей документа

Подэлемент	Описание	Возможные виды
::down-arrow	Стрелка вниз. Имеется, например, у виджета выпадающего списка и у счетчика	
::down-button	Кнопка вниз. Имеется у виджета счетчика	
::drop-down	Стрелка виджета выпадающего списка	
::indicator	Индикатор кнопки флажка или переключателя, а также группировки кнопок	
::item	Элемент меню, строки состояния	
::menu-indicator	Индикатор меню кнопки нажатия, обычно это стрелка	

Подэлемент	Описание	Возможные виды
::title	Надпись группы	Title
::up-arrow	Стрелка вверх. Имеется у виджета счетчика	
::up-button	Кнопка вверх. Имеется у виджета счетчика	

Использование каскадных стилей документа

```
QPushButton: hover {color: red}
QLineEdit: hover {color: red}
```

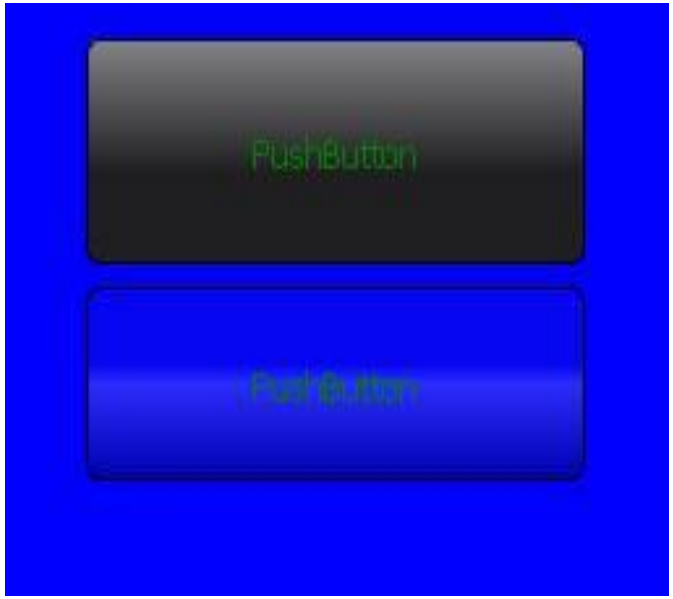
Обозначение	Описание
:checked	Активировано
:closed	Виджет находится в закрытом либо свернутом состоянии
:disabled	Виджет недоступен
:enabled	Виджет доступен
:focus	Виджет находится в фокусе ввода
:hover	Указатель мыши находится над виджетом
:indeterminate	Кнопка находится в промежуточном неопределенном состоянии
:off	Выключено (для виджетов, которые могут быть в фиксированном состоянии нажато/не нажато)
:on	Включено (для виджетов, которые могут быть в фиксированном состоянии нажато/не нажато)
:open	Виджет находится в открытом или развернутом состоянии
:pressed	Виджет был нажат мышью
:unchecked	Деактивировано

Использование каскадных стилей документа

```
QCheckBox:hover:checked {color: red}

QCheckBox:hover, QCheckBox:checked {color: red}

QLineEdit:!hover {color: red}
```

Использование каскадных стилей документа

```
QPushButton:hover {
background: qlineargradient(x1:0, y1:1, x2:0, y2:0,
                            stop:1 rgb(133,133,135),
                            stop:0.4 rgb(31,31,33),
                            stop:0.2 rgb(0,0,150));
}
```

Использование каскадных стилей документа

```
QPushButton:pressed {
background: qlineargradient(x1:0, y1:1, x2:0, y2:0,
                            stop:0 rgba(1,1,5,80),
                            stop:0.6 rgba(18,18,212,80),
                            stop:0.5 rgba(142,142,245,80));
}
```