QLabel QPushButton

QLineEdit

------------ 文本内容 ----------- QString

text ----- setText(QString& text)

QString test() const;

文本内容 ------------

Int

水平（垂直）滑动条 ---- QSlider 进度条 ------ QProgressBar

内容 ---- int

Value ----setValue(int Value)

Int value() const;

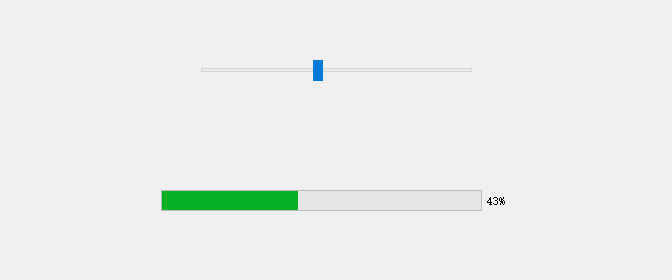
void MainWindow::**on\_horizontalSlider\_sliderMoved**(int position)

{

//将滑动条的值，设置给进度条

ui->progressBar->setValue(position);

}



C++容器

顺序容器： list vector stack queue

关联容器： set map

在QT中，对C++的stl模板容器都进行了一个封装

顺序容器：QList QVector

关联容器：QSet QMap

遍历容器中的数据的方法：

at(i)

[]-------下标运算符

迭代器：

stl迭代器

Java迭代器

for\_each()

项目：调色板

//获取信号的发送者

//QObject \*QObject::sender() const

QObject\* QObject = sender(); //QObject是qt中所有类的基类

1. 头文件：

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QPushButton>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **MainWindow**; }

QT\_END\_NAMESPACE

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

**MainWindow**(QWidget \*parent = nullptr);

~***MainWindow***();

private slots:

void **on\_pushButtonRed\_clicked**();

void **on\_pushButtonGreen\_clicked**();

void **on\_pushButtonBlue\_clicked**();

void **on\_horizontalSliderRed\_valueChanged**(int value);

void **on\_horizontalSliderGreen\_valueChanged**(int value);

void **on\_horizontalSliderBlue\_valueChanged**(int value);

void **on\_pushButtonAddColor\_clicked**();

private:

Ui::MainWindow \*ui;

QPushButton\* buttons[3]; //声明一个调色板容器，用来存放颜色

int index; //数组的索引

};

#endif // MAINWINDOW\_H

1. 源文件

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QString>

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

//给数组初始化

buttons[0] = ui->pushButtonRed;

buttons[1] = ui->pushButtonGreen;

buttons[2] = ui->pushButtonBlue;

index = 0;

}

MainWindow::~***MainWindow***()

{

delete ui;

}

//通过红色按钮来为界面填充红色

void MainWindow::**on\_pushButtonRed\_clicked**()

{

ui->labelPrint->setStyleSheet(ui->pushButtonRed->styleSheet()); //labelprint控件的颜色设置为按钮的颜色

}

//通过绿色按钮来为界面填充绿色

void MainWindow::**on\_pushButtonGreen\_clicked**()

{

ui->labelPrint->setStyleSheet(ui->pushButtonGreen->styleSheet());//labelprint控件的颜色设置为按钮的颜色

}

//通过蓝色按钮来为界面填充蓝色

void MainWindow::**on\_pushButtonBlue\_clicked**()

{

ui->labelPrint->setStyleSheet(ui->pushButtonBlue->styleSheet());

}

//滑动红色控件

void MainWindow::**on\_horizontalSliderRed\_valueChanged**(int value)

{

//1.获取水平滑动条的值，并且设置给对应的LineEdit控件

int green = ui->horizontalSliderGreen->value();

int blue = ui->horizontalSliderBlue->value();

ui->label\_red->setText(QString::number(value));

ui->label\_green->setText(QString::number(green));

ui->label\_blue->setText(QString::number(blue));

//2.将 红 绿 蓝 三个色作为一个样式表的参数，并设置给增加按钮的背景色

QString style = QString("background-color: rgb(%1, %2, %3);")

.arg(value)

.arg(green)

.arg(blue);

ui->pushButtonAddColor->setStyleSheet(style);

}

//滑动绿色控件

void MainWindow::**on\_horizontalSliderGreen\_valueChanged**(int value)

{

//1.获取水平滑动条的值，并且设置给对应的LineEdit控件

int red = ui->horizontalSliderRed->value();

int blue = ui->horizontalSliderBlue->value();

ui->label\_red->setText(QString::number(red));

ui->label\_green->setText(QString::number(value));

ui->label\_blue->setText(QString::number(blue));

//2.将 红 绿 蓝 三个色作为一个样式表的参数，并设置给增加按钮的背景色

QString style = QString("background-color: rgb(%1, %2, %3);")

.arg(red)

.arg(value)

.arg(blue);

ui->pushButtonAddColor->setStyleSheet(style);

}

//滑动蓝色控件

void MainWindow::**on\_horizontalSliderBlue\_valueChanged**(int value)

{

//1.获取水平滑动条的值，并且设置给对应的LineEdit控件

int red = ui->horizontalSliderRed->value();

int green = ui->horizontalSliderGreen->value();

ui->label\_red->setText(QString::number(red));

ui->label\_green->setText(QString::number(green));

ui->label\_blue->setText(QString::number(value));

//2.将 红 绿 蓝 三个色作为一个样式表的参数，并设置给增加按钮的背景色

QString style = QString("background-color: rgb(%1, %2, %3);")

.arg(red)

.arg(green)

.arg(value);

ui->pushButtonAddColor->setStyleSheet(style);

}

//确认按钮

void MainWindow::**on\_pushButtonAddColor\_clicked**()

{

//ui->labelPrint->setStyleSheet( ui->pushButtonAddColor->styleSheet());

buttons[index++]->setStyleSheet(ui->pushButtonAddColor->styleSheet());

if(index == 3)

{

index = 0;

}

}