**Demo**

QLabel \*l=new QLabel("Hello,QT!");

l->show();



**Demo2**

QPushButton \*button = new QPushButton("quit");

QObject::connect(button,SIGNAL(clicked()),&a,SLOT(quit()));//信号->槽 机制

button->show();



**Demo3**

QWidget \*window =new QWidget; //创建窗口

window->setWindowTitle("HELLO"); //设置标题

QSpinBox \*spinbox=new QSpinBox; //创建数字微调框

spinbox->setValue(30); spinbox->setRange(0,100);

QSlider \*slider= new QSlider(Qt::Horizontal); //创建滑块

slider->setRange(0,100);slider->setValue(30);

QHBoxLayout \*layout=new QHBoxLayout; //创建水平布局

layout->addWidget(spinbox);

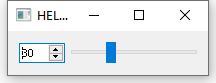
layout->addWidget(slider); //将滑块和数字微调框添加进水平布局中

QObject::connect(slider,SIGNAL(valueChanged(int )),spinbox,SLOT(setValue(int )));//建立滑块和数字微调框之间的联系（信号->槽）

QObject::connect(spinbox,SIGNAL(valueChanged(int)),slider,SLOT(setValue(int )));

window->setLayout(layout); //将水平布局放入建立的窗口

window->show(); //显示窗口



**Demo4**

**From finddialog.h**

#ifndef FINDDIALOG\_H

#define FINDDIALOG\_H

#include<QDialog>//系统对话框类

class QWidget; //声明窗口

class QLabel; //声明标签

class QLineEdit; //声明文本框

class QCheckBox; //声明勾选框

class QPushButton; //声明按钮

class FindDialog: public QDialog //继承系统对话框

{

Q\_OBJECT

public:

FindDialog(QWidget\* parent =0); //构造函数

signals://信号

void findNext(const QString &str,Qt::CaseSensitivity cs); //向后找

void findPrevious(const QString &str,Qt::CaseSensitivity cs); //向前找

private slots://槽

void enableFindButton(const QString &text);//响应FindButton的亮熄

void findClicked(); //响应查找

private: //私有成员

QLabel\* label;

QLineEdit\* lineEdit;

QCheckBox\* caseCheckBox;

QCheckBox\* backwardCheckBox;

QPushButton\* findButton;

QPushButton\* closeButton;

};

#endif // FINDDIALOG\_H

**From finddialog.cpp**

#include "finddialog.h"

FindDialog::FindDialog(QWidget \*parent) :QDialog(parent) //构造函数的具体定义

{ //建立部件

label = new QLabel(tr("找什么(&W):"));

lineEdit = new QLineEdit;

label->setBuddy(lineEdit);

caseCheckBox = new QCheckBox(tr("区分大小写(&C)"));

backwardCheckBox = new QCheckBox(tr("从后查找(&B)"));

findButton = new QPushButton(tr("查找"));

findButton->setDefault(true);

findButton->setEnabled(false);

connect(lineEdit,SIGNAL(textChanged(QString)),this,SLOT(enableFindButton(QString)));

connect(findButton,SIGNAL(clicked()),this,SLOT(findNext(QString,Qt::CaseSensitivity)));

closeButton = new QPushButton(tr("关闭"));

connect(closeButton,SIGNAL(clicked()),this,SLOT(close()));

//建立布局

QHBoxLayout \* topLeftLayout = new QHBoxLayout;

topLeftLayout->addWidget(label);

topLeftLayout->addWidget(lineEdit);

QVBoxLayout \* leftLayout = new QVBoxLayout;

leftLayout->addLayout(topLeftLayout);

leftLayout->addWidget(caseCheckBox);

leftLayout->addWidget(backwardCheckBox);

QVBoxLayout \*rightLayout = new QVBoxLayout;

rightLayout->addWidget(findButton);

rightLayout->addWidget(closeButton);

rightLayout->addStretch();//填充空白

QHBoxLayout \*mainLayout = new QHBoxLayout;

mainLayout->addLayout(leftLayout);

mainLayout->addLayout(rightLayout);

setLayout(mainLayout);

setWindowTitle(tr("Find"));

setFixedHeight(sizeHint().height());//自适应高度

}

void FindDialog::enableFindButton(const QString &text)

{

findButton->setEnabled(!text.isEmpty());

}

void FindDialog::findClicked()

{

QString text = lineEdit->text();

Qt::CaseSensitivity cs = caseCheckBox->isChecked()? Qt::CaseSensitive : Qt::CaseInsensitive;

if(backwardCheckBox->isChecked())

{

emit findPrevious(text,cs);

}

else

{

emit findNext(text,cs);

}

}

**From main.cpp**

#include<finddialog.h>

#include <QApplication>

int main(int argc, char \*argv[])

{

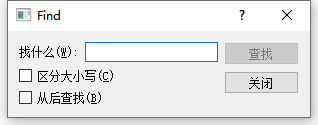
QApplication a(argc, argv);

FindDialog \*diglog = new FindDialog;

diglog->show();

return a.exec();

}



**Demo5**

**From dialog.h**

#ifndef DIALOG\_H

#define DIALOG\_H

#include <QDialog>

QT\_BEGIN\_NAMESPACE

namespace Ui { class Dialog; }

QT\_END\_NAMESPACE

class Dialog : public QDialog

{

Q\_OBJECT

public:

Dialog(QWidget \*parent = nullptr);

~Dialog();

private:

Ui::Dialog \*ui;

};

#endif // DIALOG\_H

**From dialog.cpp**

#include "dialog.h"

#include "ui\_dialog.h"

Dialog::Dialog(QWidget \*parent)

: QDialog(parent)

, ui(new Ui::Dialog)

{

ui->setupUi(this);

}

Dialog::~Dialog()

{

delete ui;

}

**From main.cpp**

#include "dialog.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

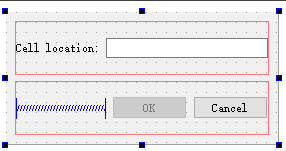
Dialog w;

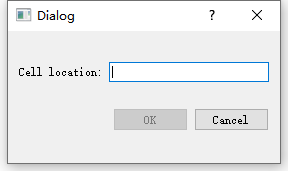
w.show();

return a.exec();

}

**From dialog.ui**





**Demo6**

**From dialog.h**

#ifndef DIALOG\_H

#define DIALOG\_H

#include <QDialog>

QT\_BEGIN\_NAMESPACE

namespace Ui { class Dialog; }

QT\_END\_NAMESPACE

class Dialog : public QDialog

{

Q\_OBJECT

public:

Dialog(QWidget \*parent = nullptr);

~Dialog();

void setColumnRange(QChar frist ,QChar last);

private:

Ui::Dialog \*ui;

};

#endif // DIALOG\_H

**From dialog.cpp**

#include "dialog.h"

#include "ui\_dialog.h"

Dialog::Dialog(QWidget \*parent)

: QDialog(parent)

, ui(new Ui::Dialog)

{

ui->setupUi(this);

ui->groupBox\_2->hide(); //隐藏框2和框3

ui->groupBox\_3->hide();

this->layout()->setSizeConstraint(QLayout::SetFixedSize);//窗口布局自适应

setColumnRange('A','Z');

}

Dialog::~Dialog()

{

delete ui;

}

void Dialog::setColumnRange(QChar frist,QChar last)//添加选项

{

ui->comboBox->clear();

ui->comboBox\_3->clear();

ui->comboBox\_5->clear();

ui->comboBox\_3->addItem("None");

ui->comboBox\_5->addItem("None");

QChar ch=frist;

while(ch<=last)

{

ui->comboBox->addItem(QString(ch));

ui->comboBox\_3->addItem(QString(ch));

ui->comboBox\_5->addItem(QString(ch));

ch=ch.unicode()+1;

}

}

**From main.cpp**

#include "dialog.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

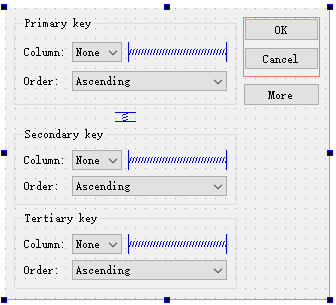
Dialog w;

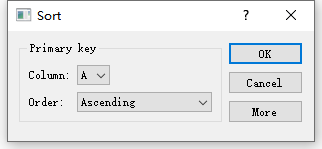
w.show();

return a.exec();

}

**From dialog.ui**





setFixedSize(width,height) //改变窗口大小

setWindowTitle(Qstring) //改变窗口名称