**qml实现简单记事本程序**

最近自学qml，尝试用qml写个小程序，熟悉qml简单用法

**#ifndef FILERWRITER**

**#define FILERWRITER**

**#include <QQuickTextDocument>**

**#include <QtGui/QTextCharFormat>**

**#include <QtCore/QTextCodec>**

**#include <qqmlfile.h>**

**#include<QFile>**

**#include<QObject>**

**//该类主要实现文件的读写功能**

**class FileRWritter:public QObject**

**{**

**Q\_OBJECT**

**Q\_PROPERTY(QString m\_fileName READ getFileName WRITE setFileName)**

**public:**

**FileRWritter(QObject \*parent= NULL); //QString filename,**

**// void saveFile(QString filedata);**

**// QString readFile();**

**Q\_INVOKABLE void saveFile(QString filedata);**

**Q\_INVOKABLE QString readFile();**

**QString getFileName();**

**void setFileName(QString filename);**

**private:**

**QString m\_fileName;**

**};**

**#endif // FILERWRITER**

**#include "filerwriter.h"**

**#include <QByteArray>**

**#include<QString>**

**#include<QMessageBox>**

**FileRWritter::FileRWritter(QObject \*parent):QObject(parent)**

**{**

**// if(filename.isEmpty())**

**// {**

**// return;**

**// }**

**// m\_fileName = filename; QString filename ,**

**}**

**void FileRWritter::saveFile(QString filedata)**

**{**

**if(filedata.isEmpty())**

**{**

**return;**

**}**

**// int index = m\_fileName.lastIndexOf(":");**

**QString filePath =m\_fileName.right(m\_fileName.length() - 8);**

**QFile file(filePath);**

**if(file.exists())**

**{**

**file.remove();**

**}**

**if(!file.open(QIODevice::ReadWrite))**

**{**

**QMessageBox::information(NULL,"提示","文件打开失败"+filePath);**

**return;**

**}**

**int length = file.write(filedata.toStdString().data());**

**if(length != filedata.length())**

**{**

**QMessageBox::information(NULL,"提示","保存失败");**

**}**

**file.close();**

**}**

**QString FileRWritter::readFile()**

**{**

**QString filePath =m\_fileName.right(m\_fileName.length() - 8);**

**QFile file(filePath);**

**if(!file.open(QIODevice::ReadWrite))**

**{**

**QMessageBox::information(NULL,"提示","文件打开失败"+filePath);**

**return "";**

**}**

**QByteArray filedata = file.readAll();**

**file.close();**

**return filedata;**

**}**

**QString FileRWritter::getFileName()**

**{**

**return m\_fileName;**

**}**

**void FileRWritter::setFileName(QString filename)**

**{**

**m\_fileName = filename;**

**}**

**#include <QApplication>**

**#include <QQmlApplicationEngine>**

**#include "filerwriter.h"**

**#include <QtQml>**

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

**{**

**QApplication app(argc, argv);**

**QQmlApplicationEngine engine;**

**//为了能在qml中使用上面写的c++类<span style="font-family: Arial, Helvetica, sans-serif; font-size: 12px;">FileRWritter，必须进行注册</span>**

**qmlRegisterType<FileRWritter>("com.mytexteditor.filerwritter", 1, 0, "FileRWritter");**

**engine.load(QUrl(QStringLiteral("qrc:/main.qml")));**

**return app.exec();**

**}**

**import QtQuick 2.4**

**import QtQuick.Controls 1.3**

**import QtQuick.Window 2.2**

**import QtQuick.Dialogs 1.2**

**import QtQuick.Controls 1.2;**

**import QtQuick.Layouts 1.1**

**import QtQuick.Controls.Styles 1.3**

**import com.mytexteditor.filerwritter 1.0**

**ApplicationWindow {**

**title: qsTr("TextEditor")**

**width: 640**

**height: 480**

**visible: true**

**id:mainwindowid;**

**property string filepath: "";**

**toolBar:ToolBar**

**{**

**RowLayout {**

**ToolButton {**

**text:"复制";**

**// iconSource: "image/1.png"**

**onClicked: textareaid.copy();**

**}**

**ToolButton {**

**text:"粘贴"**

**onClicked: textareaid.paste();**

**}**

**ToolButton {**

**text:"剪切";**

**onClicked: textareaid.cut();**

**}**

**ToolButton {**

**text:"撤销";**

**onClicked: textareaid.undo();**

**}**

**ToolButton {**

**text:"字体颜色"**

**onClicked: colorDialog.open();**

**}**

**}**

**}**

**menuBar: MenuBar {**

**Menu {**

**title: qsTr("文件(&F)")**

**MenuItem {**

**text: qsTr("新建")**

**shortcut: "Ctrl+N"**

**onTriggered: newFile();**

**}**

**MenuItem {**

**text: qsTr("打开")**

**shortcut: "Ctrl+O"**

**onTriggered: openfileDialog.open();**

**}**

**MenuItem {**

**text: qsTr("保存")**

**shortcut: "Ctrl+S"**

**onTriggered: saveFile();**

**}**

**// MenuItem {**

**// text: qsTr("另存为")**

**//shortcut: "Ctrl+A"**

**// }**

**MenuSeparator{}**

**MenuItem {**

**text: qsTr("退出")**

**shortcut: "Ctrl+Q"**

**onTriggered: Qt.quit();**

**}**

**}**

**Menu {**

**title: qsTr("编辑(&E)")**

**MenuItem {**

**text: qsTr("撤销")**

**shortcut: "Ctrl+Z"**

**onTriggered: textareaid.undo();**

**}**

**MenuSeparator{}**

**MenuItem {**

**text: qsTr("复制")**

**shortcut: "Ctrl+C"**

**onTriggered: textareaid.copy();**

**}**

**MenuItem {**

**text: qsTr("剪切")**

**onTriggered: textareaid.cut();**

**shortcut: "Ctrl+X"**

**}**

**MenuItem {**

**text: qsTr("粘贴")**

**shortcut: "Ctrl+V"**

**onTriggered: textareaid.paste();**

**}**

**MenuItem {**

**text: qsTr("删除")**

**shortcut: "Del"**

**onTriggered: textareaid.remove(textareaid.selectionStart,textareaid.selectionEnd);**

**}**

**MenuSeparator{}**

**MenuItem {**

**text: qsTr("查找")**

**shortcut: "Ctrl+F"**

**onTriggered:findtextdlg.visible = true;**

**}**

**MenuItem {**

**text: qsTr("查找下一个")**

**// shortcut: "Ctrl+X"**

**}**

**MenuItem {**

**text: qsTr("替换")**

**shortcut: "Ctrl+H"**

**}**

**MenuItem {**

**text: qsTr("转到")**

**shortcut: "Ctrl+G"**

**onTriggered: ;**

**}**

**MenuSeparator{}**

**MenuItem {**

**text: qsTr("全选")**

**shortcut: "Ctrl+A";**

**onTriggered: textareaid.selectAll();**

**}**

**MenuItem {**

**text: qsTr("日期/时间")**

**onTriggered: textareaid.insert(textareaid.cursorPosition,Qt.formatDateTime(new Date(), "dddd yyyy-MM-dd MMM hh-mm-ss"))**

**// shortcut: "Ctrl+X"**

**}**

**}**

**Menu {**

**title: qsTr("格式(&O)")**

**MenuItem {**

**id:autoReturnMenuid;**

**text: qsTr("自动换行")**

**//shortcut: "Ctrl+Z"**

**checkable: true;**

**onTriggered: setStatusBarState();**

**}**

**MenuItem {**

**text: qsTr("字体")**

**//shortcut: "Ctrl+C"**

**onTriggered: fontDialog.open();**

**}**

**}**

**Menu {**

**title: qsTr("查看(&V)")**

**MenuItem {**

**id:showStatusBarMenuid;**

**text: qsTr("状态栏")**

**//shortcut: "Ctrl+Z"**

**checkable: true;**

**onTriggered: setStatusBarState();**

**}**

**}**

**Menu {**

**title: qsTr("关于(&V)")**

**MenuItem {**

**text: qsTr("关于记事本")**

**onTriggered: aboutapp();**

**}**

**}**

**}**

**statusBar: StatusBar {**

**id:statusbarid;**

**visible: false;**

**RowLayout {**

**// anchors.fill: parent**

**Label {**

**text: ""**

**id:statuslabel;**

**}**

**Rectangle**

**{**

**width: 100;**

**height: parent.height;**

**color: "red";**

**}**

**Label {**

**text: "555"**

**id:positionlabel;**

**}**

**}**

**}**

**function openFile()**

**{**

**}**

**function aboutapp()**

**{**

**aboutappdialog.visible = true;**

**}**

**function setTextColor()**

**{**

**textareaid.textColor = colorDialog.color;**

**}**

**function setStatusBarState()**

**{**

**positionlabel.text = textareaid.cursorPosition;**

**if(autoReturnMenuid.checked)**

**{**

**statusbarid.visible = false;**

**showStatusBarMenuid.enabled=false;**

**}**

**else**

**{**

**showStatusBarMenuid.enabled=true;**

**if(showStatusBarMenuid.checked)**

**{**

**statusbarid.visible = true;**

**}**

**else**

**{**

**statusbarid.visible = false;**

**}**

**}**

**}**

**function setTextFont()**

**{**

**textareaid.font = fontDialog.font;**

**}**

**function clearTextArea()**

**{**

**textareaid.selectAll();**

**textareaid.remove(0,textareaid.cursorPosition);**

**}**

**function newFile()**

**{**

**// if(textareaid.saved)**

**// {**

**// clearTextArea();**

**// }**

**// else**

**if(textareaid.saved==false && textareaid.text.length > 0)**

**{**

**messageDialog.open();**

**}**

**clearTextArea();**

**mainwindowid.filepath="";**

**}**

**function saveFile()**

**{**

**if(mainwindowid.filepath.length==0)**

**{**

**fileDialog.open();**

**}**

**else**

**{**

**filerwritterid.m\_fileName = mainwindowid.filepath;**

**filerwritterid.saveFile(textareaid.text);**

**}**

**textareaid.saved = true;**

**}**

**function changeCaseSensitiveState()**

**{**

**if(casecheckid.checked)**

**{**

**findtextdlg.casesensitve = true;**

**}**

**else**

**{**

**findtextdlg.casesensitve = false;**

**}**

**}**

**function changeFindTextWard()**

**{**

**if(upbuttonid.checked)**

**{**

**findtextdlg.upward = true;**

**}**

**else if(donwbuttonid.checked)**

**{**

**findtextdlg.upward = false;**

**}**

**}**

**function findNextText()**

**{**

**if( findtextdlg.index === -1 || textareaid.text.length == 0)**

**{**

**return;**

**}**

**var start\_index = 0;**

**if(findtextdlg.index > 0)**

**{**

**start\_index = findtextdlg.index+textfieldid.text.length;**

**if(findtextdlg.upward)**

**{**

**start\_index = findtextdlg.index;**

**}**

**}**

**else if(findtextdlg.upward )**

**{**

**start\_index = textareaid.text.length;**

**}**

**console.log("startindex :"+start\_index)**

**var contentstr;**

**var findText ;**

**if(findtextdlg.casesensitve == true)**

**{**

**contentstr = textareaid.text.substring(start\_index);**

**if(findtextdlg.upward)**

**{**

**contentstr = textareaid.text.substring(0,start\_index).trim();**

**}**

**findText = textfieldid.text;**

**}**

**else**

**{**

**contentstr = textareaid.text.substring(start\_index).toLowerCase();**

**if(findtextdlg.upward)**

**{**

**contentstr = textareaid.text.substring(0,start\_index).toLowerCase().trim();**

**}**

**findText = textfieldid.text.toLowerCase();**

**}**

**console.log("contentstr :"+contentstr)**

**if(contentstr.length == 0)**

**{**

**return;**

**}**

**var index = contentstr.indexOf(findText);**

**if(findtextdlg.upward)**

**{**

**index = contentstr.lastIndexOf(findText);**

**}**

**console.log("index :"+index)**

**if(index < 0 )**

**{**

**completeFindDialog.open();**

**findtextdlg.index = -1;**

**return;**

**}**

**var old\_index = findtextdlg.index;**

**var select\_start = old\_index+index+textfieldid.text.length;**

**var select\_end = textfieldid.text.length+old\_index+textfieldid.text.length+index;**

**if(old\_index == 0)**

**{**

**select\_start = old\_index+index;**

**select\_end = old\_index+textfieldid.text.length+index;**

**}**

**else if(findtextdlg.upward)**

**{**

**select\_start = index;**

**select\_end = textfieldid.text.length+index;**

**}**

**console.log("select\_start :"+select\_start)**

**console.log("select\_end :"+select\_end)**

**textareaid.select(select\_start,select\_end);**

**findtextdlg.index = old\_index+index+textfieldid.text.length;**

**if(findtextdlg.upward)**

**{**

**findtextdlg.index = index;**

**}**

**console.log("findtextdlg.index :"+findtextdlg.index)**

**if((findtextdlg.upward && index === 0))**

**{**

**completeFindDialog.open();**

**findtextdlg.index = -1;**

**return;**

**}**

**}**

**function changeButtonState()**

**{**

**if(textfieldid.text.length != 0)**

**{**

**findnextid.enabled=true;**

**}**

**else**

**{**

**findnextid.enabled=false;**

**}**

**}**

**function reject()**

**{**

**findtextdlg.index= 0;**

**findtextdlg.casesensitve =false;**

**findtextdlg.upward= false;**

**findtextdlg.close();**

**casecheckid.checked = false;**

**upbuttonid.checked = false;**

**donwbuttonid.checked = true;**

**textfieldid.text = "";**

**console.log("close dialog")**

**}**

**ColorDialog {**

**id: colorDialog**

**title: "Please choose a color"**

**onAccepted: {**

**setTextColor();**

**}**

**onRejected: {**

**// console.log("Canceled")**

**// Qt.quit()**

**}**

**//Component.onCompleted: visible = true**

**}**

**FontDialog {**

**id: fontDialog**

**title: "Please choose a font"**

**font: Qt.font({ family: "Arial", pointSize: 24, weight: Font.Normal })**

**onAccepted: {**

**setTextFont();**

**}**

**onRejected: {**

**console.log("Canceled")**

**// Qt.quit()**

**}**

**}**

**FileDialog {**

**id: openfileDialog**

**title: "选择文件"**

**selectExisting:true;**

**nameFilters: [ " files (\*.txt )", "All files (\*)" ]**

**onAccepted: {**

**var filename = openfileDialog.fileUrl;**

**filerwritterid.m\_fileName = filename;**

**var filedata = filerwritterid.readFile();**

**textareaid.text = filedata;**

**mainwindowid.filepath = filename;**

**}**

**}**

**FileDialog {**

**id: fileDialog**

**title: "选择保存路径"**

**selectExisting:false;**

**nameFilters: [ " files (\*.txt )", "All files (\*)" ]**

**onAccepted: {**

**var filename = fileDialog.fileUrl;**

**filerwritterid.m\_fileName = filename;**

**filerwritterid.saveFile(textareaid.text);**

**mainwindowid.filepath = filename;**

**}**

**}**

**MessageDialog {**

**id: messageDialog**

**title: "提示"**

**text: "是否需要保存？"**

**standardButtons:StandardButton.Yes|StandardButton.No**

**onYes: {**

**saveFile();**

**}**

**onNo:**

**{**

**textareaid.saved = true;**

**newFile();**

**}**

**visible: false;**

**}**

**MessageDialog{**

**id:aboutappdialog**

**title: "关于程序"**

**text:"该程序主要实现简单的记事本功能，写该程序的目的是为了解qml的用法"**

**}**

**MessageDialog {**

**id: completeFindDialog**

**title: "提示"**

**text: "查找完毕"**

**standardButtons:StandardButton.Yes**

**onYes: {**

**close()**

**}**

**onNo:**

**{**

**close()**

**}**

**visible: false;**

**}**

**Dialog**

**{**

**id:findtextdlg**

**visible: false;**

**property int index: 0;**

**property bool casesensitve :false;**

**property bool upward: false;**

**contentItem:**

**RowLayout**

**{**

**Layout.maximumWidth: 300;**

**Layout.maximumHeight: 100;**

**// anchors.fill: parent;**

**ColumnLayout**

**{**

**RowLayout**

**{**

**Label {**

**text: "查找内容"**

**}**

**TextField**

**{**

**id:textfieldid;**

**placeholderText: qsTr("输入要查找的内容");**

**onTextChanged: changeButtonState();**

**}**

**}**

**RowLayout**

**{**

**CheckBox**

**{**

**id:casecheckid;**

**text:"区分大小写"**

**onCheckedChanged: changeCaseSensitiveState();**

**}**

**GroupBox {**

**title: "方向"**

**RowLayout {**

**ExclusiveGroup { id: tabPositionGroup }**

**RadioButton {**

**id:upbuttonid;**

**text: "向上"**

**checked: false**

**exclusiveGroup: tabPositionGroup**

**onCheckedChanged: changeFindTextWard();**

**}**

**RadioButton {**

**id:donwbuttonid;**

**text: "向下"**

**exclusiveGroup: tabPositionGroup**

**onCheckedChanged: changeFindTextWard();**

**checked: true;**

**}**

**}**

**}**

**}**

**}**

**ColumnLayout**

**{**

**Button**

**{**

**id:findnextid;**

**text:"查找下一个"**

**onClicked: findNextText();**

**enabled: false;**

**}**

**Button**

**{**

**id:cancelid;**

**text:"取消"**

**onClicked: reject();**

**}**

**}**

**}**

**onVisibleChanged: closedlg();**

**//standardButtons: StandardButton.Save | StandardButton.Cancel**

**}**

**function closedlg()**

**{**

**if(findtextdlg.visible == false)**

**{**

**reject();**

**}**

**}**

**TextArea**

**{**

**property bool saved: false;**

**id:textareaid;**

**anchors.fill: parent;**

**onCursorPositionChanged: setStatusBarState();**

**onTextChanged: saved=false;**

**// verticalScrollBarPolicy:Qt.ScrollBarAlwaysOff**

**}**

**FileRWritter**

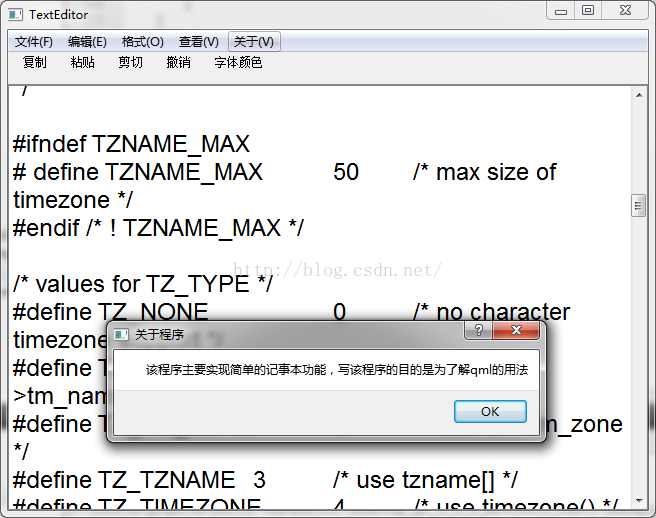
**{**

**id:filerwritterid;**

**m\_fileName: "";**

**}**

**}**



[QML 安卓 开发环境搭建+记事本应用 整理](https://blog.csdn.net/weixin_38416696/article/details/79483407" \t "_blank)

[..](https://blog.csdn.net/weixin_38416696)

https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png 647

[Qt 5.9安卓 开发环境构建 一，安装QtCreator 时，务必选择 和andriod 相关选项。 二、安装java jdk。 下载java jdk 32：java jdk 1.8x86 Java jdk 1.8 官网：http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.h...](https://blog.csdn.net/weixin_38416696/article/details/79483407" \t "_blank)

[qtcalendar:一个简单的QtQML日历应用程序\_qml日历,qt日历-其它...](https://download.csdn.net/download/weixin_42126865/18261892" \t "_blank)

7-9

[CalendarApp:适用于小型企业的日历应用程序 Qt-Qml-Calendar\_qml\_QT\_qml日历控件\_calendar\_ QT/QML的日历控件,纯QML实现。 简单的日历桌面应用程序 由c#开发的简单日历桌面应用程序,刚刚起步的程序员,程序不行请见谅。](https://download.csdn.net/download/weixin_42126865/18261892" \t "_blank)

[玩转Qml(8)-Qml属性\_jared2020的博客](https://blog.csdn.net/D759378563/article/details/111872146" \t "_blank)

6-15

[这种list的实现方式,是在C++中导出了一个特殊类型的属性,即QQmlListProperty。 你也可以自己定义一个这样的属性: Q\_PROPERTY(QQmlListProperty<Fruit> fruit READ fruit) C++里面按它的规则实现几个函数,并注册类型后,就可以在Qml中...](https://blog.csdn.net/D759378563/article/details/111872146" \t "_blank)

[Qml写的备忘录](https://download.csdn.net/download/ray314/4573247" \t "_blank)

09-16

[根据Qt的一个实例改写的 类似备忘录的功能 之前调试没问题了 不过后来不知道怎么可能又有小Bug了 感兴趣的可以看看](https://download.csdn.net/download/ray314/4573247" \t "_blank)

[QML(1)： 什么是QML](https://blog.csdn.net/hitzsf/article/details/108772235" \t "_blank)

[读书笔记](https://blog.csdn.net/hitzsf)

https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png 3884

[文章目录前言一、QML 是什么？二、Qt Quick 又是什么?三、简单代码示例四、察看QML视图效果总结 前言   QML 早已经不限于开发手机应用，也可以支持用户开发传统的桌面程序。Qt5 将 QML 脚本编程提到与传统 C++ 部件编程相同的高度，开始力推 QML 界面编程。  使用 QML 开发界面主要有以下几个优点：     1)  QML 非常灵活，可以做出非常炫酷的效果，例如 QQ、360、迅雷等都不在话下。     2) QML 是标记语言，见名知意，非常容易编写](https://blog.csdn.net/hitzsf/article/details/108772235" \t "_blank)

[ｑｍｌ实现读写一个ｉｎｉ配置文件，实现如何保存上次关闭软件时的大小和位置](https://blog.csdn.net/weixin_39770778/article/details/89788445" \t "_blank)

[weixin\_39770778的博客](https://blog.csdn.net/weixin_39770778)

https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png 1626

[我们知道大部分软件关闭之后再次打开一般是会恢复上一次关闭的之前的大小和位置，所以我们自己用ｑｍｌ开发一个软件项目的时候应该也要按照大部分软件再次打开时会恢复上一次关闭的之前的大小和位置。所以我们自己读写一个ｉｎｉ配置文件来保存关闭该软件时的位置和大小。 写ini配置文件函数代码如下： void Configure::write(QString str, QString titleName) ...](https://blog.csdn.net/weixin_39770778/article/details/89788445" \t "_blank)

[Qt：记事本源代码分享（新手入门，老鸟勿喷）](https://qdebug.blog.csdn.net/article/details/51779273" \t "_blank)

[好儿郎~志在四方](https://blog.csdn.net/rl529014)

https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png 5334

[主要功能： 1、获取系统时间和日期。 2、将界面的数据保存为txt文件。 3、播放背景音乐。](https://qdebug.blog.csdn.net/article/details/51779273" \t "_blank)

[用qml简单的文本编辑器，可以跨平台，linux，安卓，Windows下都可以](https://download.csdn.net/download/lvdepeng123/10926068" \t "_blank)

01-18

[我们尝试实现一个编辑器。这是一个简单的文本编辑器，具有新建、剪切、复制和粘贴等操作](https://download.csdn.net/download/lvdepeng123/10926068" \t "_blank)

[用Qt 5.3 纯 qml 写的文件查看器](https://download.csdn.net/download/surfsky/7995831" \t "_blank)

10-01

[用Qt 5.3 纯 qml 写的文件查看器 - 支持文本、图像、音频、视频 - 支持文字大小、色彩 - 应用程序图标 - xmlhttp 读取本地文件](https://download.csdn.net/download/surfsky/7995831" \t "_blank)

[Qml文件对话框](https://zhengtianzuo.blog.csdn.net/article/details/78759892" \t "_blank)

[zhengtianzuo的博客](https://blog.csdn.net/zhengtianzuo06)

https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png 2793

[使用Qt.labs.platform 1.0中的FileDialog 代码非常的简单FileDialog { id: fileDialog1 fileMode: FileDialog.OpenFile nameFilters: ["图像文件 (\*.png \*.jpg \*.gif \*.bmp)", "全部文件 (\*.\*)"] optio](https://zhengtianzuo.blog.csdn.net/article/details/78759892" \t "_blank)

[Qt-QML-自定义个自己的文本Text](https://dreamlife.blog.csdn.net/article/details/79734832" \t "_blank)

[DreamLife](https://blog.csdn.net/z609932088)

https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png 1442

[好久都没有正经的更新自己的文章了，这段时间也辞职了，听了小爱的，准备买个碗，自己当老板，下面请欣赏效果图这个界面布局就是自己是在想不到啥了，按照常规汽车导航的布局布局了一下，主要看内容哈，看看这个文本文件问个文本文件的状态了有一下三种1. 正常 normal2. 激活 active3. 不可用 enable位置可以有一下方式组合1. 上2. 下3. 左4. 右5. 中心等等，自己组合了就不写了...](https://dreamlife.blog.csdn.net/article/details/79734832" \t "_blank)

[QML 从入门到放弃 第二卷](https://blog.csdn.net/weixin_34080571/article/details/94278164" \t "_blank)

[weixin\_34080571的博客](https://blog.csdn.net/weixin_34080571)

https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png 284

[第二卷如何更快速的放弃，注重的是C++和QML的交互 <1>记事本。。 (1) 先测试下不在QML创建C++对象，仅仅在main.cpp添加一个属性函数供调用. 注意只使用槽函数来做到。 TextStreamLoader.h #ifndef TEXTSTREAMLOADER\_H #define TEXTSTREAMLOADER\_H #inclu...](https://blog.csdn.net/weixin_34080571/article/details/94278164" \t "_blank)

[Qt Quick开发中利用qml的单例模式（1）](https://blog.csdn.net/BlueBuger/article/details/52744748" \t "_blank)

[BlueBuger的博客](https://blog.csdn.net/BlueBuger)

https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png 1972

[从C/C++的模式转到qml开发中，跨文件访问变量和函数是很头疼的事情。通过id访问，虽然可以实现，但是代码量大了就不太好控制了，id可以覆盖、跨文件访问就不太好统一规范起来。这时候qml的单例模式就变得很有用了，qml的单例模式，在整个应用中只会被实例化1次，没有副本，所以在一定程度上可以代替全局变量。有时候跨文件的调用、访问可以间接通过单例模块进行。 如下： 创建单例模块 pragma S](https://blog.csdn.net/BlueBuger/article/details/52744748" \t "_blank)

[QtQuick实现文本编辑和显示文本行号](https://blog.csdn.net/qyvlik/article/details/40402069" \t "_blank)

[qyvlik的专栏](https://blog.csdn.net/qyvlik)

https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png 4310

[由于某个项目的需要的，需要使用Qt实现文本行号 在参考网上的一些](https://blog.csdn.net/qyvlik/article/details/40402069" \t "_blank)

[QtQuick实现文本编辑和显示文本行号-改进版](https://blog.csdn.net/qyvlik/article/details/40451731" \t "_blank)

[qyvlik的专栏](https://blog.csdn.net/qyvlik)

https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png 2834

[可以查看这个简单的项目](https://blog.csdn.net/qyvlik/article/details/40451731" \t "_blank)

[Qt Quick QML (包括PySide，QtCreator)的各种坑](https://blog.csdn.net/ztzata/article/details/118487452" \t "_blank)

[最新发布](https://blog.csdn.net/ztzata/article/details/118487452" \t "_blank)

[CSDN的博客](https://blog.csdn.net/ztzata)

https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png 619

[实在是无力吐槽。Qt6，PySide6还不成熟也就算了，QtQuick，QtQuick.Controls 2重写了1然后改名去掉了2也就算了，QtCreator这么多年了还有各种不支持的 语句和Item，然后可视化图形界面设计QtQuickDesigner就直接白板了，现在直接用代码写界面反而更快。但是QtCreator的IDE也真烂，QML写起来就和用记事本差不多，各种假警报，错误提示也没什么参考价值。 QtWebEngine在Qt6.1中没有，要等到6.2才有也就算了，Qt5的QtWebEngine也不](https://blog.csdn.net/ztzata/article/details/118487452" \t "_blank)

[qml 实现文本编辑器](https://blog.csdn.net/liqiangzju/article/details/54645433" \t "_blank)

[liqiangzju的博客](https://blog.csdn.net/liqiangzju)

https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png 2876

[实现文本编辑器要文件的 IO 功能，但是 QML 并没有提供，所以需要自己写一个。 #ifndef FILEIO\_H #define FILEIO\_H #include #include #include class FileIO : public QObject { Q\_OBJECT public: Q\_PROPERTY(QString source](https://blog.csdn.net/liqiangzju/article/details/54645433" \t "_blank)

[qml开发笔记(七)：输入元素鼠标输入MouseArea和键盘输入Keys](https://hpzwl.blog.csdn.net/article/details/78552121" \t "_blank)

[热门推荐](https://hpzwl.blog.csdn.net/article/details/78552121" \t "_blank)

https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png 2万+

[qml学习笔记(七)：输入元素鼠标输入MouseArea和键盘输入Keys 前话 本章节笔记开始进入qml用户输入元素的介绍和学习，本篇章主要学习鼠标捕捉MouseArea和键盘捕捉Keys](https://hpzwl.blog.csdn.net/article/details/78552121" \t "_blank)

**热门文章**

* [Qt线程QThread简析 [https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png](https://blog.csdn.net/hai200501019/article/details/9535069) 16868](https://blog.csdn.net/hai200501019/article/details/9535069)
* [QTableWidget排序问题 [https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png](https://blog.csdn.net/hai200501019/article/details/45746477) 12899](https://blog.csdn.net/hai200501019/article/details/45746477)
* [QCompleter的一些用法 [https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png](https://blog.csdn.net/hai200501019/article/details/8455808) 11159](https://blog.csdn.net/hai200501019/article/details/8455808)
* [QVector 和vector的比较 [https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png](https://blog.csdn.net/hai200501019/article/details/11713519) 9883](https://blog.csdn.net/hai200501019/article/details/11713519)
* [QThread 与 QObject的关系 [https://csdnimg.cn/release/blogv2/dist/pc/img/readCountWhite.png](https://blog.csdn.net/hai200501019/article/details/9748173) 9063](https://blog.csdn.net/hai200501019/article/details/9748173)