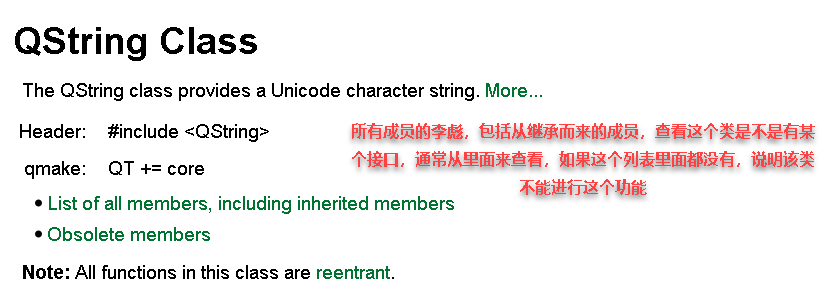
### QString --- 字符串类



### 二、QMssageBox----消息盒子

### 三、QLabel -------文本标签--文本组件

### 四、QPushButton---按钮 ---按钮组件

五、多界面编程

一 QString类

功能：显示一个字符串内容

主要接口函数

构造函数：

QString(const char \*str)

QString(const QString &other)

QString()

赋值运算符重载：

QString &operator=(const QString &other)

功能函数：

1. 基本数据类型（int, float, double, char\*）转换成字符串

Number(long, int):QString

Number(int, int):QString

Number(uint, int):QString

Number(ulong, int):QString

Number(qlonglong, int):QString

Number(qulonglong, int)QString

Number(double, char , int)QString

该函数都是一个静态成员并且重载函数

[static]QString QString::number(int n, int base = 10)

参数说明：

参数 一： int n ---- 要转换的数据

参数 二： int base = 10 -- 设置数据的进制，默认为十进制

2 8 16

返回值：QString

比如：

Ui->testLabel->setText(QString::number(20, 16));

1. 将字符串转换成基本数据类型

toDouble(bool \*) const : double

toFloat(bool \*) const : float

toInt(bool \*, int) const : int

Int QString::toInt(bool \*ok = nullptr, int base = 10) const

该函数为普通成员函数，将字符串转换成基本数据类型

参数说明：

参数一：bool \*ok ----> 输出参数 转换成功 返回true ， 否则 返回false

参数二：int base = 10 --> 设置数据的进制，默认为十进制

返回值：得到基本类型数据

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QtDebug>

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

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

//在构造函数中设置窗口的大小固定

this->setFixedSize(this->width(), this->height());

//QString

QString s1("hello"); // ----- char \* ---- 有参构造

QString s; //创建一个字符串对象 ----- 默认构造

s = s1; //使用一个对象赋值给另一个对象 --- 赋值运算符重载函数

qDebug() << "s1 = " << s1 << " s = " << s ;

//[2] 功能接口函数 --number---将基本数据类型转换成字符串类型，比如：int double

//在修改完ui文件之后，为了保证在代码中可以找到该空间，必须要重新构建工程或者编译工程

ui->testLabel->setText(QString::number(20,10)); //20 ---> "20" ,快捷编译工程的方式：ctrl + r

QString data = ui->testLabel->text();

qDebug() << "data = " << data.toInt(); //一般不传参数，就使用默认参数

}

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

{

delete ui;

}

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

{

//重新给指定控件设置背景色

//ui->testLabel->styleSheet(); //得到一个控件的样式表

QString red = ui->redEdit->text();

QString style("background-color: rgb(%1, %2, 127);");

QString s = style.arg(red.toInt());

QString ss = s.arg(ui->greenEdit->text().toInt());

qDebug() << "style = " << style;

ui->testLabel->setStyleSheet(ss); //给控件在代码中设置样式表

}



### 项目练习：人品测试

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QtDebug>

#include <ctime>

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

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

this->setFixedSize(this->width(), this->height());

}

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

{

delete ui;

}

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

{

//获取用户输入

QString name = ui->inputNameEdit->text();

ui->NameLabel->setText(name); //将输入显示到NameLabel

qsrand(time(nullptr)); //随机数种子

int i = qrand()%101; //生成随机数 0-100

ui->numLabel->setText(QString::number(i));

if(i > 70 && i <= 100)

{

ui->pjLabel->setText("你的人品很不错哦！一看就是干大事的人");

}

else if(i < 40 && i >= 0)

{

ui->pjLabel->setText("你的人品太low了，一定是干过坏事");

}

else{

ui->pjLabel->setText("你的人品有待改善哦，多读书");

}

}

运行：



### 二、QMssageBox----消息盒子

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QtDebug>

#include <QMessageBox>

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

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

//定义一个整形变量，初始值为3，记录用户输入错误的次数

accoutTimes = 3;

//设置界面固定大小

this->setFixedSize(this->width(), this->height());

}

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

{

delete ui;

}

void MainWindow::**on\_root\_pushButton\_clicked**()

{

//1.获取用户输入的管理员账号密码

QString input = ui->userEdit->text();

QString passwd = ui->passwdEdit->text();

//2.跟系统内置的账号密码进行比对

if(input == "root" && passwd == "12345")

{

//账号密码正确，则进入管理员界面

}

else if(input == "root" && passwd != "12345")

{

//账号密码出错提示错误

//定义一个整形变量，初始值为3

QString info = QString("账号或者密码错误，还有%1次机会").arg(--accoutTimes);

//消息盒子使用情况一

//QMessageBox::information(this,"登录失败",info);

//消息盒子使用情况二

//QMessageBox msgBox;

//msgBox.setText(info);

//msgBox.exec(); //默认只提供一个ok按钮

//消息盒子使用情况三

QMessageBox msgBox;

msgBox.setText("...");

msgBox.setInformativeText("登录失败");

msgBox.setStandardButtons(QMessageBox::Save | QMessageBox::Discard | QMessageBox::Cancel);

msgBox.setDefaultButton(QMessageBox::Save); //消息盒子默认选择save这个按钮

int ret = msgBox.*exec*();

switch (ret) { //根据用户点击的按钮进行相关操作

case QMessageBox::Save:

// Save was clicked

qDebug() << "Save was clicked";

break;

case QMessageBox::Discard:

// Don't Save was clicked

qDebug() << "Don't Save was clicked";

break;

case QMessageBox::Cancel:

// Cancel was clicked

qDebug() << "Cancel was clicked";

break;

default:

// should never be reached

break;

}

if(accoutTimes == 0)

{

QMessageBox::information(this,"登录失败","输入超过3次，系统将退出");

//应用程序自动退出

this->close();

}

}

}