# Qt 之 QDateTimeEdit

# 简述

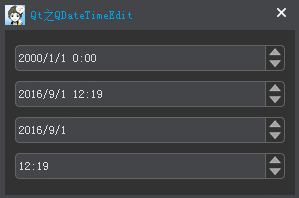
QDateTime类提供了一个部件，用于编辑日期和时间。

QDateTimeEdit允许用户编辑日期，通过使用键盘或箭头键来增加和减少日期和时间值。箭头键可以在QDateTimeEdit内进行部分移动，日期和时间的格式按照setDisplayFormat()设置的显示。

# 基本使用

QDateTimeEdit有好多个构造函数，分别看看他们是如何使用的。

## 效果



## 源码

默认情况下，如果QDateTimeEdit构造时不指定日期时间，系统会为其设置一个和本地相同的日期时间格式（右下角可更改本地日期时间格式），并且值为：2000年1月1日 0时0分0秒。

QDateTimeEdit \*dateTimeEdit = new QDateTimeEdit(this);

QDateTimeEdit \*dateTimeEdit2 = new QDateTimeEdit(QDateTime::currentDateTime(), this);

QDateTimeEdit \*dateEdit = new QDateTimeEdit(QDate::currentDate(), this);

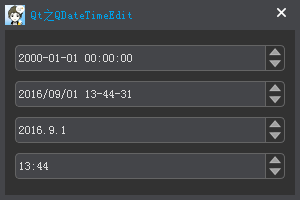
QDateTimeEdit \*timeEdit = new QDateTimeEdit(QTime::currentTime(), this);

除了通过构造函数指定显示的日渐日期外，我们也可以根据QDateTimeEdit提供的槽函数来修改，setDateTime()、setDate()、setTime()。

# 日期时间格式

如果不想要系统默认的格式，可以通过setDisplayFormat()来自定设置。

## 效果



## 源码

QDateTimeEdit \*dateTimeEdit = new QDateTimeEdit(this);

QDateTimeEdit \*dateTimeEdit2 = new QDateTimeEdit(QDateTime::currentDateTime(), this);

QDateTimeEdit \*dateEdit = new QDateTimeEdit(QDate::currentDate(), this);

QDateTimeEdit \*timeEdit = new QDateTimeEdit(QTime::currentTime(), this);

*// 设置日期时间格式*

dateTimeEdit->setDisplayFormat("yyyy-MM-dd HH:mm:ss");

dateTimeEdit2->setDisplayFormat("yyyy/MM/dd HH-mm-ss");

dateEdit->setDisplayFormat("yyyy.M.d");

timeEdit->setDisplayFormat("H:mm");

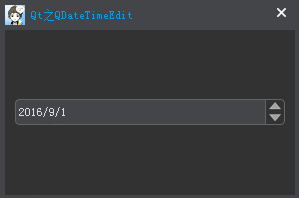
yyyy：年，4个数表示。   
MM：月，01-12。   
dd：日，01-31。   
HH：时，00-23。   
mm：分，00 - 59。   
ss：秒，00-59。

格式这里先简单介绍下，后面详解，具体可参考：QDateTime。

# 日期时间范围

创建了QDateTimeEdit对象，并设置其日期时间为今天（currentDate），同时限制有效日期的范围：距离今天±365天。

## 效果



## 源码

QDateTimeEdit \*dateEdit = new QDateTimeEdit(QDate::currentDate(), this);

dateEdit->setMinimumDate(QDate::currentDate().addDays(-365)); *// -365天*

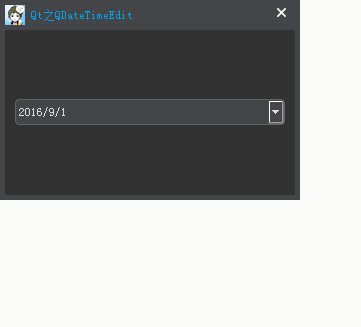
dateEdit->setMaximumDate(QDate::currentDate().addDays(365)); *// +365天*

其他同功能的有用函数：setDateTimeRange()、setDateRange()、setTimeRange()、setMaximumDateTime()和setMinimumDateTime()、setMinimumTime()和setMaximumTime()。

# 显示日历

默认情况下只能通过鼠标点击上下箭头来改变日期时间，如果要弹出日期控件，只需调用setCalendarPopup(true)即可。

## 效果



## 源码

QDateTimeEdit \*dateEdit = new QDateTimeEdit(QDate::currentDate(), this);

dateEdit->setMinimumDate(QDate::currentDate().addDays(-365)); *// -365天*

dateEdit->setMaximumDate(QDate::currentDate().addDays(365)); *// +365天*

dateEdit->setCalendarPopup(true); *// 日历弹出*

这时，上/下箭头就变为下拉箭头了。当点击下拉箭头之后，就会弹出日历控件，由于我们设置了日期范围，所以不在范围内的日期是无法选择的（disabled）。

## 获取日期时间

可以通过date()、dateTime()等接口来获取日期时间对象，如果要获取年、月、日等信息，可以调用例如QDate的year()、month()、day()等函数。

QDateTimeEdit \*dateEdit = new QDateTimeEdit(QDate::currentDate(), this);

dateEdit->setMinimumDate(QDate::currentDate().addDays(-365)); *// -365天*

dateEdit->setMaximumDate(QDate::currentDate().addDays(365)); *// +365天*

QDate date = dateEdit->date(); *// 日期*

QDateTime dateTime = dateEdit->dateTime(); *// 日期时间*

QDate maxDate = dateEdit->maximumDate(); *// 最大日期*

QDateTime maxDateTime = dateEdit->maximumDateTime(); *// 最大日期时间*

QTime maxTime = dateEdit->maximumTime(); *// 最大时间*

QDate minDate = dateEdit->minimumDate(); *// 最小日期*

QDateTime minDateTime = dateEdit->minimumDateTime(); *// 最小日期时间*

QTime minTime = dateEdit->minimumTime(); *// 最小时间*

qDebug() << "Date : " << date;

qDebug() << "DateTime : " << dateTime;

qDebug() << "MaxDate : " << maxDate;

qDebug() << "MaxDateTime : " << maxDateTime;

qDebug() << "MaxTime : " << maxTime;

qDebug() << "MinDate : " << minDate;

qDebug() << "MinDateTime : " << minDateTime;

qDebug() << "MinTime : " << minTime;

输出如下：

Date : QDate(“2016-09-01”)   
DateTime : QDateTime(2016-09-01 00:00:00.000 中国标准时间 Qt::TimeSpec(LocalTime))   
MaxDate : QDate(“2017-09-01”)   
MaxDateTime : QDateTime(2017-09-01 23:59:59.999 中国标准时间 Qt::TimeSpec(LocalTime))   
MaxTime : QTime(“23:59:59.999”)   
MinDate : QDate(“2015-09-02”)   
MinDateTime : QDateTime(2015-09-02 00:00:00.000 中国标准时间 Qt::TimeSpec(LocalTime))   
MinTime : QTime(“00:00:00.000”)

# 信号

常用的三个信号：dateChanged()、dateTimeChanged()、timeChanged()，当日期、日期时间、时间改变时发射。

QDateTimeEdit \*dateTimeEdit = new QDateTimeEdit(QDateTime::currentDateTime(), this);

*// 连接信号槽*

connect(dateTimeEdit, SIGNAL(dateChanged(QDate)), this, SLOT(onDateChanged(QDate)));

connect(dateTimeEdit, SIGNAL(dateTimeChanged(QDateTime)), this, SLOT(onDateTimeChanged(QDateTime)));

connect(dateTimeEdit, SIGNAL(timeChanged(QTime)), this, SLOT(onTimeChanged(QTime)));

槽函数

*// 日期发生改变时执行*

void onDateChanged(const QDate &date) {

qDebug() << "QDate : " << date;

}

*// 无论日期还是时间发生改变，都会执行*

void onDateTimeChanged(const QDateTime &dateTime) {

qDebug() << "QDateTime : " << dateTime;

}

*// 时间发生改变时执行*

void onTimeChanged(const QTime &time) {

qDebug() << "QTime : " << time;

}

# 日期时间部分

枚举 QDateTimeEdit::Section   
标识 QDateTimeEdit::Sections

| **常量** | **值** |
| --- | --- |
| QDateTimeEdit::NoSection | 0x0000 |
| QDateTimeEdit::AmPmSection | 0x0001 |
| QDateTimeEdit::MSecSection | 0x0002 |
| QDateTimeEdit::SecondSection | 0x0004 |
| QDateTimeEdit::MinuteSection | 0x0008 |
| QDateTimeEdit::HourSection | 0x0010 |
| QDateTimeEdit::DaySection | 0x0100 |
| QDateTimeEdit::MonthSection | 0x0200 |
| QDateTimeEdit::YearSection | 0x0400 |

## 源码

设置常用的日期时间格式：yyyy/MM/dd HH:mm:ss，包含6部分，分别是年月日时分秒，可以通过sectionCount()来获取数量。通过调用setSelectedSection()可以设置当前选择的部分，如果需要获取某部分的值，可以调用sectionText()；通过sectionAt()来获取下标对应的部分。

QDateTimeEdit \*dateEdit = new QDateTimeEdit(QDateTime::currentDateTime(), this);

dateEdit->setDisplayFormat("yyyy/MM/dd HH:mm:ss"); *// 设置显示格式*

dateEdit->setSelectedSection(QDateTimeEdit::DaySection); *// 设置所选部分*

int nCount = dateEdit->sectionCount(); *// 部分数量*

QDateTimeEdit::Section monthSection = dateEdit->sectionAt(1); *// 下标为1对应的部分*

QDateTimeEdit::Section section = dateEdit->currentSection(); *// 当前部分*

int nIndex = dateEdit->currentSectionIndex(); *// 当前部分下标*

*// 各部分对应的值*

QString strYear = dateEdit->sectionText(QDateTimeEdit::YearSection);

QString strMonth = dateEdit->sectionText(QDateTimeEdit::MonthSection);

QString strDay = dateEdit->sectionText(QDateTimeEdit::DaySection);

QString strHour = dateEdit->sectionText(QDateTimeEdit::HourSection);

QString strMinute = dateEdit->sectionText(QDateTimeEdit::MinuteSection);

QString strSecond = dateEdit->sectionText(QDateTimeEdit::SecondSection);

qDebug() << "SectionCount : " << nCount;

qDebug() << "MonthSection : " << monthSection;

qDebug() << "CurrentSection : " << section;

qDebug() << "CurrentSectionIndex : " << nIndex;

qDebug() << "Year : " << strYear;

qDebug() << "Month : " << strMonth;

qDebug() << "Day : " << strDay;

qDebug() << "Hour : " << strHour;

qDebug() << "Minute : " << strMinute;

qDebug() << "Second : " << strSecond;

输出如下：

SectionCount : 6   
MonthSection : QDateTimeEdit::Section(MonthSection)   
CurrentSection : QDateTimeEdit::Section(DaySection)   
CurrentSectionIndex : 2   
Year : “2016”   
Month : “09”   
Day : “01”   
Hour : “14”   
Minute : “50”   
Second : “29”

如果格式为：yyyy/MM/dd，那么sectionCount将会返回3。sectionAt(0)将会返回YearSection；sectionAt(1)返回MonthSection；sectionAt(2)返回YearSection。