QDial Class

QDial 类提供圆形范围控制（如速度计或电位计）。更多的...

Header:	**#include**
CMake:	find_package(Qt6 REQUIRED COMPONENTS Widgets) target_link_libraries(mytarget PRIVATE Qt6::Widgets)
qmake:	QT += widgets
Inherits:	QAbstractSlider

- 所有成员的列表，包括继承的成员

特性

- **notchSize** : const int
- **notchTarget** : qreal
- **notchesVisible** : bool
- **wrapping** : bool

公共职能

	QDial(QWidget **parent* = nullptr)
virtual	**~QDial**()
int	**notchSize**() const
qreal	**notchTarget**() const
bool	**notchesVisible**() const
void	**setNotchTarget**(double *target*)
bool	**wrapping**() const

重载公共职能

virtual QSize	**minimumSizeHint() const override**

virtual QSize	**sizeHint**() const override

公共老虎机

void	**setNotchesVisible**(**bool** ***visible***)
void	**setWrapping**(bool *on*)

protected功能

virtual void	**initStyleOption**(**QStyleOptionSlider** ***option***) const

重载的protected功能

virtual bool	**event**(**QEvent** ***e***) **override**
virtual void	**mouseMoveEvent**(QMouseEvent *e) override
virtual void	**mousePressEvent**(QMouseEvent *e) override
virtual void	**mouseReleaseEvent**(QMouseEvent *e) override
virtual void	**paintEvent**(QPaintEvent *pe) override
virtual void	**resizeEvent**(QResizeEvent *e) override
virtual void	**sliderChange**(QAbstractSlider::SliderChange *change*) override

详细说明

当用户需要控制程序可定义范围内的值，并且该范围是环绕的（例如，测量角度从 0 到 359 度）或者对话框布局需要方形小部件时，使用 QDial。

由于QDial继承自QAbstractSlider，表盘的行为方式与slider。什么时候wrapping() 为 false（默认设置）滑块和转盘之间没有真正的区别。它们共享相同的信号、槽和成员函数。您使用哪一种取决于用户的期望和应用程序的类型。

表盘最初发出valueChanged() 在滑块移动时连续发出信号；您可以通过禁用它来减少发出信号的频率tracking财产。这sliderMoved即使禁用跟踪，也会连续发出 () 信号。

表盘也发出sliderPressed （） 和sliderReleased() 在按下和释放鼠标按钮时发出信号。请注意，转盘的值可以在不发出这些信号的情况下更改，因为键盘和滚轮也可以用于更改值。

与滑块不同，QDial 尝试绘制“合适”数量的凹口，而不是每行一步一个。如果可能，绘制的凹口数量是每行一步一个，但如果没有足够的像素来绘制每个凹口，QDial 将跳过凹口以尝试绘制统一的组（例如，通过每隔第二个或第三个凹口绘制）。

与滑块一样，转盘使QAbstractSlider功能setValue() 可用作插槽。

表盘的键盘界面相当简单：**left/up**和**right/down**方向键调整转盘的value由定义的singleStep,**Page Up**和**Page Down**由定义的pageStep，以及**Home**和**End**键将值设置为定义的minimum和maximum价值观。

如果使用鼠标滚轮调整转盘，则增量值由以下较小值确定wheelScrollLines乘以singleStep， 和pageStep。

也可以看看QScrollBar,QSpinBox,QSlider， 和Sliders Example。

财产文件

[read-only]notchSize : const int

该属性保存当前的缺口大小

凹口大小以范围控制单位而不是像素为单位，并计算为以下值的倍数：singleStep() 导致屏幕上的凹口尺寸接近notchTarget()。

访问功能：

int	**notchSize() const**

也可以看看notchTarget （） 和singleStep()。

notchTarget : qreal

该属性保存凹口之间的目标像素数

缺口目标是像素数QDial尝试放在每个凹口之间。

实际尺寸可能与目标尺寸不同。

默认凹口目标为 3.7 像素。

访问功能：

qreal	**notchTarget() const**
void	**setNotchTarget**(double *target*)

notchesVisible : bool

该属性保存是否显示凹口

如果属性为`true`，则在表盘周围绘制一系列凹口以指示可用值的范围；否则不显示凹口。

默认情况下，该属性被禁用。

访问功能：

bool	**notchesVisible() const**
void	**setNotchesVisible**(bool *visible*)

wrapping : bool

该属性保存是否启用换行

如果为 true，则启用换行；否则，会在刻度盘底部插入一些空格来分隔有效值范围的两端。

如果启用，箭头可以定向在表盘上的任何角度。如果禁用，箭头将被限制在表盘的上部；如果将其旋转到刻度盘底部的空间中，它将被夹紧到有效值范围的最近一端。

默认情况下该属性是`false`.

访问功能：

bool	**wrapping() const**
void	**setWrapping**(bool *on*)

成员函数文档

*[explicit]QDial::QDial(QWidget *parent = nullptr)*

构造一个表盘。

这*parent*参数被发送到QAbstractSlider构造函数。

[virtual]QDial::~QDial()

毁坏表盘。

*[override virtual protected]bool QDial::event(QEvent *e)*

重新实现：QAbstractSlider::event（QEvent *e）。

*[virtual protected]void QDial::initStyleOption(QStyleOptionSlider *option) const*

初始化*option*与此值QDial。当子类需要时，此方法非常有用QStyleOptionSlider，但不想自己填写所有信息。

也可以看看QStyleOption::initFrom()。

[override virtual]QSize QDial::minimumSizeHint() const

重新实现属性的访问函数：QWidget::minimumSizeHint。

*[override virtual protected]void QDial::mouseMoveEvent(QMouseEvent *e)*

重新实现：QWidget::mouseMoveEvent（QMouseEvent *事件）。

*[override virtual protected]void QDial::mousePressEvent(QMouseEvent *e)*

重新实现：QWidget::mousePressEvent（QMouseEvent *事件）。

*[override virtual protected]void QDial::mouseReleaseEvent(QMouseEvent *e)*

重新实现：QWidget::mouseReleaseEvent（QMouseEvent *事件）。

*[override virtual protected]void QDial::paintEvent(QPaintEvent *pe)*

重新实现：QWidget::paintEvent（QPaintEvent *事件）。

*[override virtual protected]void QDial::resizeEvent(QResizeEvent *e)*

重新实现：QWidget::resizeEvent（QResizeEvent *事件）。

[override virtual]QSize QDial::sizeHint() const

重新实现属性的访问函数：QWidget::sizeHint。

[override virtual protected]void QDial::sliderChange(QAbstractSlider::SliderChange change)

重新实现：QAbstractSlider::sliderChange（QAbstractSlider::SliderChange 更改）。