

QGroupBox Class

QGroupBox 小部件提供了一个带有标题的组框框架。 [更多的...](#)

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

- 所有成员的列表，包括继承的成员

特性

alignment : Qt::Alignment	checkable : bool	checked : bool	flat : bool	title : QString
---------------------------	------------------	----------------	-------------	-----------------

公共职能

	QGroupBox(QWidget *parent = nullptr)
	QGroupBox(const QString &title, QWidget *parent = nullptr)
virtual	~QGroupBox()
Qt::Alignment	alignment() const
bool	isCheckable() const
bool	isChecked() const
bool	isFlat() const
void	setAlignment(int alignment)
void	setCheckable(bool checkable)
void	setFlat(bool flat)
void	setTitle(const QString &title)

重载的公共函数

virtual QSize

minimumSizeHint() const override

公共槽

void

setChecked(bool checked)

信号

void

clicked(bool checked = false)

void

toggled(bool on)

protected function

virtual void

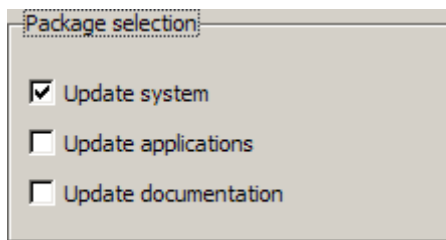
initStyleOption(QStyleOptionGroupBox *option) const

重新实现protected function

virtual void	changeEvent(QEvent *ev) override
virtual void	childEvent(QChildEvent *c) override
virtual bool	event(QEvent *e) override
virtual void	focusInEvent(QFocusEvent *fe) override
virtual void	mouseMoveEvent(QMouseEvent *event) override
virtual void	mousePressEvent(QMouseEvent *event) override
virtual void	mouseReleaseEvent(QMouseEvent *event) override
virtual void	paintEvent(QPaintEvent *event) override

virtual void [paintEvent\(QPaintEvent *event\) override](#)
virtual void [changeEvent\(QEvent *ev\) override](#)
virtual void [resizeEvent\(QResizeEvent *e\) override](#)

详细说明



组框提供框架、顶部标题、键盘快捷键，并在其内部显示各种其他小部件。键盘快捷键将键盘焦点移动到组框的子窗口小部件之一。

QGroupBox 还允许您设置 [title](#)（通常在构造函数中设置）和标题 [alignment](#)。分组框可以 [checkable](#)。可检查组框中的子窗口小部件是否启用或禁用取决于该组框是否已启用 [checked](#)。

您可以通过启用以下功能来最大限度地减少组框的空间消耗 [flat](#) 财产。多数情况 [styles](#)，启用此属性会导致删除框架的左、右和下边缘。

QGroupBox 不会自动布局子部件（通常是 [QCheckBoxes](#) 或 [QRadioButton](#) 但可以是任何小部件）。下面的例子展示了我们如何设置一个带有布局的 QGroupBox：

```
QGroupBox *groupBox = new QGroupBox(tr("Exclusive Radio Buttons"));

QRadioButton *radio1 = new QRadioButton(tr("&Radio button 1"));
QRadioButton *radio2 = new QRadioButton(tr("R&adio button 2"));
QRadioButton *radio3 = new QRadioButton(tr("Ra&dio button 3"));

radio1->setChecked(true);

QVBoxLayout *vbox = new QVBoxLayout;
vbox->addWidget(radio1);
vbox->addWidget(radio2);
vbox->addWidget(radio3);
vbox->addStretch(1);
groupBox->setLayout(vbox);
```

也可以看看 [QButtonGroup](#) 和 [Group Box Example](#)。

属性文档

alignment : Qt::Alignment

该属性保存组框标题的对齐方式。

大多数样式将标题放置在框架的顶部。标题的水平对齐方式可以使用以下列表中的单个值来指定：

- [Qt::AlignLeft](#) 将标题文本与组框的左侧对齐。

- [Qt::AlignRight](#)将标题文本与组框的右侧对齐。
- [Qt::AlignHCenter](#)将标题文本与组框的水平中心对齐。

默认对齐方式是[Qt::AlignLeft](#)。

访问功能：

Qt::Alignment	alignment() const
void	setAlignment(int <i>alignment</i>)

也可以看看[Qt::Alignment](#)。

checkable : bool

该属性保存组框的标题中是否有复选框

如果此属性为true，则组框使用复选框代替普通标签来显示其标题。如果选中该复选框，则启用组框的子项；否则，它们将被禁用且无法访问。

默认情况下，组框是不可选中的。

如果为组框启用了此属性，则还将首先检查它以确保其内容已启用。

访问功能：

bool	isChecked() const
void	setChecked(bool <i>checkable</i>)

也可以看看[checked](#)。

checked : bool

该属性保存组框是否被选中

如果该组框是可选的，则会显示一个复选框。如果选中该复选框，则启用组框的子项；否则，孩子们将被禁用并且用户无法访问。

默认情况下，还会选中可检查的组框。

访问功能：

bool	isChecked() const
void	setChecked(bool <i>checked</i>)

通知器信号：

void	toggled(bool <i>on</i>)
------	--------------------------

void

[toggleled](#)(bool on)



也可以看看[checkable](#)。

flat : bool

该属性保存组框是平涂还是有框架

组框通常由一个周围的框架组成，顶部有一个标题。如果启用此属性，则在大多数样式中仅绘制框架的顶部部分；否则，将绘制整个框架。

默认情况下，该属性被禁用，即，除非明确指定，否则组框不是平面的。

注意：在某些样式中，平面和非平面组框具有相似的表示形式，并且可能不像其他样式那样易于区分。

访问功能：

bool	isFlat() const
void	setFlat(bool flat)

也可以看看[title](#)。

title : QString

该属性保存组框标题文本

如果标题包含与号('&')后跟一个字母，则组框标题文本将具有键盘快捷键。

```
g->setTitle("&User information");
```

在上面的例子中，Alt+U将键盘焦点移至组框。请参阅[QShortcut](#)有关详细信息的文档（要显示实际的&符号，请使用“&&”）。

没有默认的标题文本。

访问功能：

QString	title() const
void	setTitle(const QString &title)

也可以看看[alignment](#)。

成员函数文档

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

使用给定的构造一个组框小部件`parent`但没有标题。

*[explicit]QGroupBox::QGroupBox(const QString &title, QWidget *parent = nullptr)*

使用给定的构造一个组框`title`和`parent`。

[virtual]QGroupBox::~QGroupBox()

销毁组框。

*[override virtual protected]void QGroupBox::changeEvent(QEvent *ev)*

重新实现：`QWidget::changeEvent` (`QEvent *事件`)。

*[override virtual protected]void QGroupBox::childEvent(QChildEvent *c)*

重新实现：`QObject::childEvent` (`QChildEvent *事件`)。

[signal]void QGroupBox::clicked(bool checked = false)

当激活复选框（即，当鼠标光标位于按钮内时按下然后释放）或键入快捷键时，会发出此信号。值得注意的是，如果您调用，则不会发出此信号`setChecked()`。

如果选中该复选框，`checked`是真的；如果未选中该复选框，则为 `false`。

也可以看看`checkable`,`toggled` () , 和`checked`。

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

重新实现: [QWidget::event](#) ([QEvent](#) *事件) 。

*[override virtual protected] void
QGroupBox::focusInEvent([QFocusEvent](#) *fe)*

重新实现: [QWidget::focusInEvent](#) ([QFocusEvent](#) *事件) 。

*[virtual protected] void
QGroupBox::initStyleOption([QStyleOptionGroupBox](#) *option) const*

初始化`option`与此值[QGroupBox](#)。当子类需要时, 此方法非常有用[QStyleOptionGroupBox](#), 但不想自己填写所有信息。

也可以看看[QStyleOption::initFrom\(\)](#)。

[override virtual] [QSize](#) QGroupBox::minimumSizeHint() const

重新实现属性的访问函数: [QWidget::minimumSizeHint](#)。

*[override virtual protected] void
QGroupBox::mouseMoveEvent([QMouseEvent](#) *event)*

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

*[override virtual protected] void
QGroupBox::mousePressEvent([QMouseEvent](#) *event)*

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

[override virtual protected]void
*QGroupBox::mouseReleaseEvent(QMouseEvent *event)*

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

*[override virtual protected]void QGroupBox::paintEvent(QPaintEvent *event)*

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

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

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

[signal]void QGroupBox::toggled(bool on)

如果组框是可选的，则在切换复选框时会发出此信号。*on*如果选中该复选框，则为 true；否则，它是错误的。

注意：属性的通知程序信号[checked](#)。

也可以看看[checkable](#)。