

QPushButton Class

QPushButton 小部件提供了一个命令按钮。[更多的...](#)

Header:	<code>#include < QPushButton></code>
CMake:	<code>find_package(Qt6 REQUIRED COMPONENTS Widgets) target_link_libraries(mytarget PRIVATE Qt6::Widgets)</code>
qmake:	<code>QT += widgets</code>
Inherits:	QAbstractButton
Inherited By:	QCommandLinkButton

- [所有成员的列表，包括继承的成员](#)

特性

- `autoDefault` : bool
- `default` : bool
- `flat` : bool

公共函数

	<code>QPushButton(QWidget *parent = nullptr)</code>
	<code>QPushButton(const QString &text, QWidget *parent = nullptr)</code>
	<code>QPushButton(const QIcon &icon, const QString &text, QWidget *parent = nullptr)</code>
virtual	<code>~QPushButton()</code>
bool	<code>autoDefault()</code> const
bool	<code>isDefault()</code> const
bool	<code>isFlat()</code> const
QMenu *	<code>menu()</code> const
void	<code>setAutoDefault(bool)</code>
void	<code>setDefault(bool)</code>
void	<code>setFlat(bool)</code>
void	<code>setMenu(QMenu *menu)</code>

重载的公共方法

virtual QSize	minimumSizeHint() const override
virtual QSize	sizeHint() const override

公共槽

void	showMenu()
------	----------------------------

protected 函数

virtual void	initStyleOption(QStyleOptionButton *option) const
--------------	-------------------------------------------------------------------

重载的protected 函数

virtual bool	event(QEvent *e) override
virtual void	focusInEvent(QFocusEvent *e) override
virtual void	focusOutEvent(QFocusEvent *e) override
virtual bool	hitButton(const QPoint &pos) const override
virtual void	keyPressEvent(QKeyEvent *e) override
virtual void	mouseMoveEvent(QMouseEvent *e) override
virtual void	paintEvent(QPaintEvent *) override

详细说明

按钮或命令按钮可能是任何图形用户界面中最常用的小部件。按下（单击）按钮可命令计算机执行某些操作或回答问题。典型的按钮有“确定”、“应用”、“取消”、“关闭”、“是”、“否”和“帮助”。

命令按钮是矩形的，通常显示描述其操作的文本标签。可以通过在文本中在首选字符前加上 & 符号来指定快捷键。例如：

```
QPushButton *button = new QPushButton("&Download", this);
```

在此示例中，快捷键是 *Alt+D*。请参阅[QShortcut](#)有关详细信息的文档（要显示实际的 & 符号，请使用“&&”）。

按钮显示文本标签，还可以选择显示小图标。这些可以使用构造函数进行设置，并稍后使用进行更改 [setText\(\)](#) 和 [setIcon\(\)](#)。如果按钮被禁用，则文本和图标的外观将根据 GUI 样式进行操作，以使按钮看起来“禁用”。

按钮发出信号 [clicked\(\)](#) 当通过鼠标、空格键或键盘快捷键激活时。连接到该信号以执行按钮的操作。按钮还提供不太常用的信号，例如 [pressed\(\)](#) 和 [released\(\)](#)。

对话框中的命令按钮默认为自动默认按钮，即，当它们接收到键盘输入焦点时，它们会自动成为默认按钮。默认按钮是当用户在对话框中按 Enter 或 Return 键时激活的按钮。你可以改变这个 [setAutoDefault\(\)](#)。请注意，自动默认按钮保留了一些额外的空间，这是绘制默认按钮指示器所必需的。如果您不希望按钮周围有这个空间，请致电 [setAutoDefault](#)（错误的）。

按钮小部件是如此重要，在过去的十年中已经发展到可以容纳很多变化。Microsoft 风格指南现在显示了大约十种不同的 Windows 按钮状态，并且文本暗示当考虑到所有功能组合时还有数十种状态。

最重要的模式或状态是：

- 可用或不可用（灰显、禁用）。
- 标准按钮、切换按钮或菜单按钮。
- 打开或关闭（仅用于切换按钮）。
- 默认或正常。通常可以使用 Enter 或 Return 键“单击”对话框中的默认按钮。
- 是否自动重复。
- 按下与否。

作为一般规则，当应用程序或对话框窗口在用户单击应用程序或对话框窗口执行操作（例如“应用”、“取消”、“关闭”和“帮助”）时，以及当小部件应该具有宽的矩形形状并带有文本标签。小的、通常为方形的按钮，用于更改窗口的状态而不是执行操作（例如窗口右上角的按钮 [QFileDialog](#)）不是命令按钮，而是工具按钮。Qt 提供了一个特殊的类（[QToolButton](#)）对于这些按钮。

如果您需要切换行为（请参阅[setCheckable\(\)](#)）或像滚动条中的箭头一样按下时自动重复激活信号的按钮（请参阅[setAutoRepeat\(\)](#)），命令按钮可能不是您想要的。如有疑问，请使用工具按钮。

注意：在 macOS 上，当按钮的宽度小于 50 或高度小于 30 时，按钮的角会从圆形更改为方形。使用 [setMinimumSize\(\)](#) 函数可以防止这种行为。

命令按钮的变体是菜单按钮。它们不仅提供一个命令，而且还提供多个命令，因为单击它们时会弹出一个选项菜单。使用方法 [setMenu\(\)](#) 将弹出菜单与按钮相关联。

其他类别的按钮是选项按钮（请参阅[QRadioButton](#)）和复选框（参见[QCheckBox](#)）。

在 Qt 中，[QAbstractButton](#) 基类提供了大部分模式和其他 API，而 [QPushButton](#) 提供了 GUI 逻辑。看 [QAbstractButton](#) 有关 API 的更多信息。

可以查阅[QToolButton](#)、[QRadioButton](#)，和[QCheckBox](#)。

属性文档

autoDefault : bool

该属性保存按钮是否是自动默认按钮

如果此属性设置为 true，则按钮是自动默认按钮。

在某些 GUI 样式中，默认按钮周围有一个额外的框架，最多 3 个像素或更多。Qt 自动在自动默认按钮周围保留该空间，即自动默认按钮可能具有稍大的尺寸提示。

对于具有QDialog父母；否则默认为 false。

请参阅default属性以了解如何操作的详细信息default和自动默认交互。

访问功能:

bool	autoDefault() const
void	setAutoDefault(bool)

default : bool

该属性保存按钮是否为默认按钮

默认和自动默认按钮决定当用户在对话框中按 Enter 时会发生什么。

当用户按下回车键时，将此属性设置为 true 的按钮（即对话框的默认按钮）将自动按下，但有一个例外：如果**autoDefault按钮当前获得焦点，autoDefault按钮被按下。当对话框有autoDefault按钮但没有默认按钮，按 Enter 键将按autoDefault当前具有焦点的按钮，或者如果没有按钮具有焦点，则下一个按钮autoDefault焦点链中的按钮。

在对话框中，一次只能有一个按钮作为默认按钮。然后，该按钮会显示一个附加框架（取决于 GUI 样式）。

默认按钮行为仅在对话框中提供。当按钮具有焦点时，始终可以通过按空格键从键盘上单击按钮。

如果在对话框可见时当前默认按钮的默认属性设置为 false，则下次对话框中的按钮接收焦点时将自动分配新的默认值。

该属性的默认值是 false。

访问功能:

bool	isDefault() const
void	setDefault(bool)



flat : bool

该属性保存按钮边框是否凸起

该属性的默认值是 false。如果设置此属性，则大多数样式将不会绘制按钮背景，除非按下按钮。setAutoFillBackground() 可用于确保背景被填充使用QPalette::Button刷子。

访问功能:

bool	isFlat() const
void	setFlat(bool)

成员函数文档

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

构造一个没有文本的按钮`parent`。

*[explicit]QPushButton::QPushButton(const QString &text, QWidget *parent = nullptr)*

与父级一起构造一个按钮`parent`和文字`text`。

*QPushButton::QPushButton(const QIcon &icon, const QString &text, QWidget *parent = nullptr)*

构造一个按钮`icon`和一个`text`， 和一个`parent`。

请注意，您还可以传递QPixmap对象作为图标（感谢 C++ 提供的隐式类型转换）。

[virtual]QPushButton::~~QPushButton()

破坏按钮。

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

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

*[override virtual protected]void
QPushButton::focusInEvent(QFocusEvent *e)*

重新实现：QAbstractButton::focusInEvent (QFocusEvent *e) 。

[override virtual protected]void
*QPushButton::focusOutEvent([QFocusEvent](#) *e)*

重新实现: [QAbstractButton::focusOutEvent](#) ([QFocusEvent](#) *e) 。

[override virtual protected]bool QPushButton::hitButton(const [QPoint](#)
&pos) const

重新实现: [QAbstractButton::hitButton\(const QPoint &pos\) const](#)。

[virtual protected]void
*QPushButton::initStyleOption([QStyleOptionButton](#) *option) const*

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

可以查阅[QStyleOption::initFrom\(\)](#)。

[override virtual protected]void
*QPushButton::keyPressEvent([QKeyEvent](#) *e)*

重新实现: [QAbstractButton::keyPressEvent](#) ([QKeyEvent](#) *e) 。

*[QMenu](#) *QPushButton::menu() const*

返回按钮关联的弹出菜单, 或者[nullptr](#)如果未设置弹出菜单。

可以查阅[setMenu\(\)](#)。

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

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

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

重新实现： [QAbstractButton::mouseMoveEvent\(QMouseEvent *e\)](#)。

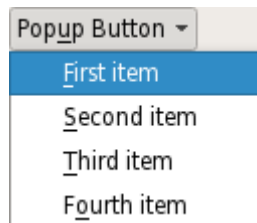
[override virtual protected]void
*QPushButton::paintEvent(QPaintEvent *)*

重新实现： [QAbstractButton::paintEvent \(QPaintEvent *e\)](#) 。

*void QPushButton::setMenu(QMenu *menu)*

关联弹出菜单`menu`用这个按钮。这会将按钮变成菜单按钮，在某些样式中会在按钮文本的右侧生成一个小三角形。

菜单的所有权不会转移给按钮。



带有弹出菜单的按钮，如图所示[Fusion widget style](#)。

可以查阅[menu\(\)](#)。

[slot]void QPushButton::showMenu()

显示（弹出）关联的弹出菜单。如果没有这样的菜单，则该功能不执行任何操作。在用户关闭弹出菜单之前，此函数不会返回。

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

重新实现属性的访问函数： [QWidget::sizeHint](#)。