

QColorDialog Class

QColorDialog 类提供了一个用于指定颜色的对话框小部件。 [更多的...](#)

Header: `#include < QColorDialog >`

CMake: `find_package(Qt6 REQUIRED COMPONENTS Widgets) target_link_libraries(mytarget PRIVATE Qt6::Widgets)`

qmake: `QT += widgets`

Inherits: [QDialog](#)

- [所有成员的列表，包括继承的成员](#)
- QColorDialog 是[标准对话框](#)的一部分。

公共类型

enum [ColorDialogOption](#) { ShowAlphaChannel, NoButtons, DontUseNativeDialog }

flags [ColorDialogOptions](#)

特性

- **currentColor** : QColor
- **options** : ColorDialogOptions

公共方法

[QColorDialog](#)(QWidget **parent* = nullptr)

[QColorDialog](#)(const QColor &*initial*, QWidget **parent* = nullptr)

virtual [~QColorDialog](#)()

QColor [currentColor](#)() const

void [open](#)(QObject **receiver***, const char **member***)

QColorDialog::ColorDialogOptions [options](#)() const

	QColorDialog (QWidget <i>*parent</i> = nullptr)
QColor	selectedColor () const
void	setCurrentColor (const QColor & <i>color</i>)
void	setOption (QColorDialog::ColorDialogOption <i>option</i> , bool <i>on</i> = true)
void	setOptions (QColorDialog::ColorDialogOptions <i>options</i>)
bool	testOption (QColorDialog::ColorDialogOption <i>option</i>) const

重载的公共方法

virtual void	setVisible (bool <i>visible</i>) override
--------------	---

信号

void	colorSelected (const QColor & <i>color</i>)
void	currentColorChanged (const QColor & <i>color</i>)

静态公共成员

QColor	customColor (int <i>index</i>)
int	customCount ()
QColor	getColor (const QColor & <i>initial</i> = Qt::white, QWidget <i>*parent</i> = nullptr, const QString & <i>title</i> = QString(), QColorDialog::ColorDialogOptions <i>options</i> = ColorDialogOptions())
void	setCustomColor (int <i>index</i> , QColor <i>color</i>)
void	setStandardColor (int <i>index</i> , QColor <i>color</i>)
QColor	standardColor (int <i>index</i>)

重载的保护函数

virtual void	changeEvent (QEvent <i>*e</i>) override
virtual void	done (int <i>result</i>) override



详细说明

颜色对话框的功能是允许用户选择颜色。例如，您可以在绘图程序中使用它来允许用户设置画笔颜色。

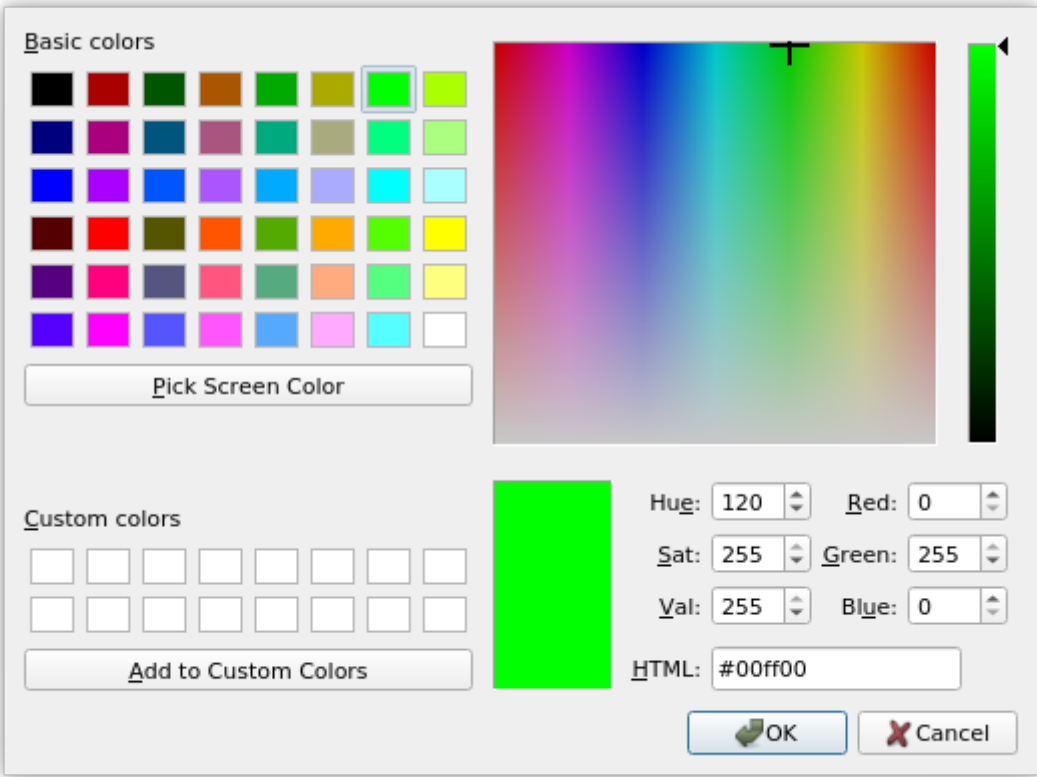
静态函数提供模式颜色对话框。

静态的getColor() 函数显示对话框，并允许用户指定颜色。此功能还可用于让用户选择具有透明度的颜色：传递ShowAlphaChannel选项作为附加参数。

用户可以存储customCount() 不同的定制颜色。自定义颜色由所有颜色对话框共享，并在程序执行期间被记住。使用setCustomColor() 设置自定义颜色，并使用customColor() 来获取它们。

当按下“选择屏幕颜色”按钮时，光标会变为十字形，并扫描屏幕上的颜色。用户可以通过单击鼠标或 Enter 按钮来选取一个。按 Esc 键可恢复进入此模式之前选择的最后一个颜色。

这Standard Dialogs示例展示了如何使用 QColorDialog 以及其他内置 Qt 对话框。



也可以看看QColor,QFileDialog,QFontDialog， 和Standard Dialogs Example。

会员类型文档

枚举 QColorDialog:: ColorDialogOption 标志 QColorDialog:: ColorDialogOptions

该枚举指定影响颜色对话框外观的各种选项。

持续的	价值	描述
QColorDialog::ShowAlphaChannel	0x00000001	允许用户选择颜色的 Alpha 分量。

QColorDialog::NoButtons	0x00000002	不显示OK和Cancel按钮。（对于“实时对话”很有用。）
QColorDialog::DontUseNativeDialog	0x00000004	使用 Qt 的标准颜色对话框而不是操作系统本机颜色对话框。

ColorDialogOptions 类型是QFlags 的类型定义。它存储 QColorDialogOption 值的 OR 组合。

也可以看看options,setOption(),testOption () , 和windowModality()。

财产文件

currentColor : QColor

该属性保存对话框中当前选择的颜色

访问功能:

QColor	currentColor() const
void	setCurrentColor(const QColor &color)

通知器信号:

void	currentColorChanged(const QColor &color)
------	--

options : ColorDialogOptions

该属性包含影响对话框外观的各种选项

默认情况下，所有选项均被禁用。

应在显示对话框之前设置选项。在对话框可见时设置它们并不能保证立即对对话框产生影响（取决于选项和平台）。

访问功能:

QColorDialog::ColorDialogOptions	options() const
void	setOptions(QColorDialog::ColorDialogOptions options)

也可以看看setOption () 和testOption()。

成员函数文档

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

使用给定的颜色构造一个颜色对话框`parent`。

*[explicit] QColorDialog::QColorDialog(const QColor &initial, QWidget *parent = nullptr)*

使用给定的颜色构造一个颜色对话框`parent`并指定`initial`颜色。

[virtual] QColorDialog::~QColorDialog()

破坏颜色对话框。

*[override virtual protected] void QColorDialog::changeEvent(QEvent *e)*

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

[signal] void QColorDialog::colorSelected(const QColor &color)

该信号在用户单击后立即发出OK选择要使用的颜色。所选颜色由以下方式指定`color`。

也可以看看`color`和`currentColorChanged()`。

[signal] void QColorDialog::currentColorChanged(const QColor &color)

每当对话框中的当前颜色发生变化时，就会发出此信号。当前颜色由以下方式指定`color`。

注意：属性的通知程序信号`currentColor`。

也可以看看`color`和`colorSelected()`。

[static] QColor QColorDialog::customColor(int index)

返回给定的自定义颜色`index`作为一个`QColor`价值。

也可以看看[setCustomColor\(\)](#)。

[static] int QColorDialog::customCount()

返回支持的自定义颜色数量`QColorDialog`。所有颜色对话框共享相同的自定义颜色。

[override virtual protected] void QColorDialog::done(int result)

重新实现：`QDialog::done (int r)`。

关闭对话框并将其结果代码设置为`result`。如果此对话框显示为[exec\(\)](#)、`done()` 导致本地事件循环完成，并且[exec\(\) 回来`result`。](#)

也可以看看[QDialog::done\(\)](#)。

*[static] QColor QColorDialog::getColor(const QColor &initial = Qt::white, QWidget *parent = nullptr, const QString &title = QString(), QColorDialog::ColorDialogOptions options = ColorDialogOptions())*

弹出带有给定窗口的模态颜色对话框`title`（如果未指定，则为“选择颜色”），让用户选择一种颜色并返回该颜色。颜色最初设置为`initial`。该对话框是一个子项`parent`。它返回一个无效值（参见[QColor::isValid\(\)](#)如果用户取消对话框，则 `()` 颜色。

这`options`参数允许您自定义对话框。

void QColorDialog::open(QObject receiver, const char member*)*

打开对话框并连接其[colorSelected\(\)](#) 向由指定的槽发出信号`receiver`和`member`。

当对话框关闭时，信号将从插槽断开。

QColor QColorDialog::selectedColor() const

返回用户通过单击选择的颜色OK或等效按钮。

注意：此颜色并不总是与所持有的颜色相同currentColor属性，因为用户可以在最终选择要使用的颜色之前选择不同的颜色。

[static]void QColorDialog::setCustomColor(int index, QColor color)

将自定义颜色设置为index到QColor color价值。

注意：此功能不适用于 macOS 平台上的 Native Color Dialog。如果您仍然需要此功能，请使用 QColorDialog::DontUseNativeDialog选项。

也可以看看customColor()。

void QColorDialog::setOption(QColorDialog::ColorDialogOption option, bool on = true)

设置给定的option启用如果on是真的; 否则，清除给定的option。

也可以看看options和testOption()。

[static]void QColorDialog::setStandardColor(int index, QColor color)

将标准颜色设置为index到QColor color价值。

注意：此功能不适用于 macOS 平台上的 Native Color Dialog。如果您仍然需要此功能，请使用 QColorDialog::DontUseNativeDialog选项。

也可以看看standardColor()。

[override virtual]void QColorDialog::setVisible(bool visible)

重新实现：QDialog::setVisible（布尔值可见）。

更改对话框的可见性。如果visible为 true，则显示对话框；否则，它被隐藏。

[static] QColor QColorDialog::standardColor(int index)

返回给定的标准颜色`index`作为一个`QColor`值。

也可以看看[setStandardColor\(\)](#)。

bool QColorDialog::testOption(QColorDialog::ColorDialogOption option)

const

`true`如果给定则返回`option`已启用；否则，返回 `false`。

也可以看看[options](#)和[setOption\(\)](#)。