

QToolBox Class

QToolBox 类提供了一系列选项卡式小部件项。[更多的...](#)

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

- [所有成员的列表，包括继承的成员](#)

特性

- `count` : const int
- `currentIndex` : int

公共职能

	QToolBox (QWidget *parent = nullptr, Qt::WindowFlags f = Qt::WindowFlags())
virtual	~QToolBox ()
int	addItem (QWidget *widget, const QIcon &iconSet, const QString &text)
int	addItem (QWidget *w, const QString &text)
int	count () const
int	currentIndex () const
QWidget *	currentWidget () const
int	indexOf (const QWidget *widget) const
int	insertItem (int index, QWidget *widget, const QIcon &icon, const QString &text)
int	insertItem (int index, QWidget *widget, const QString &text)
bool	isItemEnabled (int index) const
QIcon	itemIcon (int index) const

QString	openToolBox (QWidget *parent = nullptr, Qt::WindowFlags f = Qt::WindowFlags())
QString	itemToolTip (int index) const
void	removeItem (int index)
void	setItemEnabled (int index, bool enabled)
void	setItemIcon (int index, const QIcon &icon)
void	setItemText (int index, const QString &text)
void	setItemToolTip (int index, const QString &toolTip)
QWidget *	widget (int index) const

公共槽

void	setCurrentIndex (int index)
void	setCurrentWidget (QWidget *widget)

信号

void	currentChanged (int index)
------	-----------------------------------

protected function

virtual void	itemInserted (int index)
virtual void	itemRemoved (int index)

重新实现protected function

virtual void	changeEvent (QEvent *ev) override
virtual bool	event (QEvent *e) override
virtual void	showEvent (QShowEvent *e) override

详细说明

工具箱是一个小部件，它显示一系列选项卡，其中当前项目显示在当前选项卡下方。每个选项卡在选项卡列中都有一个索引位置。选项卡的项目是 [QWidget](#)。

每个项目都有一个 [itemText\(\)](#)，可选 [itemIcon\(\)](#)，可选 [itemToolTip\(\)](#) 和一个 [widget\(\)](#)。该项目的属性可以通过以下方式更改 [setItemText\(\)](#)，[setItemIcon\(\)](#)，和 [setItemToolTip\(\)](#)。每个项目都可以单独启用或禁用 [setItemEnabled\(\)](#)。

项目添加使用 [addItem\(\)](#)，或使用插入到特定位置 [insertItem\(\)](#)。项目总数由下式给出 [count\(\)](#)。可以使用 [delete](#) 删除项目，或者使用以下命令从工具箱中删除项目 [removeItem\(\)](#)。组合 [removeItem\(\)](#) 和 [insertItem\(\)](#) 允许您将项目移动到不同的位置。

当前项目小部件的索引由以下命令返回 [currentIndex\(\)](#)，并设置为 [setCurrentIndex\(\)](#)。可以使用以下方式找到特定项目的索引 [indexOf\(\)](#)，并且给定索引处的项目由 [item\(\)](#) 返回。

这 [currentChanged](#) 当当前项发生更改时，会发出 () 信号。

也可以看看 [QTabWidget](#)。

属性文档

[read-only]count : const int

该属性保存工具箱中包含的项目数。

默认情况下，该属性的值为 0。

访问功能：

int	count() const
-----	---------------

currentIndex : int

该属性保存当前项目的索引

默认情况下，对于空工具箱，此属性的值为 -1。

访问功能：

int	currentIndex() const
void	setCurrentIndex(int index)

通知器信号：

void	currentChanged(int index)
------	---------------------------

也可以看看[indexOf \(\)](#) 和[widget\(\)](#)。

成员函数文档

```
[explicit] QToolBox::QToolBox(QWidget *parent = nullptr,  
                               Qt::WindowFlags f = Qt::WindowFlags())
```

使用给定的构造一个新的工具箱`parent`和旗帜, `f`。

```
[virtual] QToolBox::~~QToolBox()
```

破坏工具箱。

```
int QToolBox::addItem(QWidget *widget, const QIcon &iconSet, const  
                      QString &text)
```

添加了`widget`在工具箱底部的新选项卡中。新选项卡的文本设置为`text`，以及`iconSet`显示在左侧`text`。返回新选项卡的索引。

```
int QToolBox::addItem(QWidget *w, const QString &text)
```

这是一个过载功能。

添加小部件`w`在工具箱底部的新选项卡中。新选项卡的文本设置为`text`。返回新选项卡的索引。

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

重新实现: [QFrame::changeEvent](#) (QEvent *ev) 。

```
[signal] void QToolBox::currentChanged(int index)
```

当前项目更改时会发出此信号。新的当前项目的索引被传入`index`，如果没有当前项目，则为 -1。

注意：属性的通知程序信号[currentIndex](#)。

*[QWidget](#) *[QToolBox::currentWidget\(\)](#) const*

返回指向当前小部件的指针，或者nullptr如果不存在这样的项目。

也可以看看[currentIndex](#) () 和[setCurrentWidget\(\)](#)。

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

重新实现：[QFrame::event](#) ([QEvent](#) *e) 。

*int [QToolBox::indexOf\(const \[QWidget\]\(#\) *widget\)](#) const*

返回索引widget，如果该项不存在则为 -1。

*int [QToolBox::insertItem\(int index, \[QWidget\]\(#\) *widget, const \[QIcon\]\(#\) &icon, const \[QString\]\(#\) &text\)](#)*

插入widget在位置index，或者在工具箱的底部，如果index超出范围。新项目的文本设置为text，以及icon显示在左侧text。返回新项目的索引。

*int [QToolBox::insertItem\(int index, \[QWidget\]\(#\) *widget, const \[QString\]\(#\) &text\)](#)*

这是一个过载功能。

插入widget在位置index，或者在工具箱的底部，如果index超出范围。新项目的文本设置为text。返回新项目的索引。

bool [QToolBox::isItemEnabled\(int index\)](#) const

返回true项目是否位于位置index已启用；否则返回false。

[QIcon](#) [QToolBox::itemIcon\(int index\)](#) const

返回该位置的项目的图标index，或空图标，如果index超出范围。

也可以看看[setItemIcon\(\)](#)。

[virtual protected]void QToolBox::itemInserted(int index)

在位置添加或插入新项目后调用此虚拟处理程序`index`。

也可以看看[itemRemoved\(\)](#)。

[virtual protected]void QToolBox::itemRemoved(int index)

从位置移除项目后调用此虚拟处理程序`index`。

也可以看看[itemInserted\(\)](#)。

QString QToolBox::itemText(int index) const

返回位置处项目的文本`index`，或者一个空字符串，如果`index`超出范围。

也可以看看[setItemText\(\)](#)。

QString QToolBox::itemToolTip(int index) const

返回该位置的项目的工具提示`index`，或者一个空字符串，如果`index`超出范围。

也可以看看[setItemToolTip\(\)](#)。

void QToolBox::removeItem(int index)

删除位置上的项目`index`从工具箱中。请注意，该小部件并未被删除。

*[slot]void QToolBox::setCurrentWidget(QWidget *widget)*

使`widget`当前小部件。这`widget`必须是此工具箱中的一项。

也可以看看[addItem\(\)](#)，[setCurrentIndex \(\)](#)，和[currentWidget\(\)](#)。

void QToolBox::setItemEnabled(int index, bool enabled)

如果`enabled`为 `true` 则该位置的项目`index`已启用；否则该位置的项目`index`被禁用。

也可以看看[isItemEnabled\(\)](#)。

void QToolBox::setItemIcon(int index, const QIcon &icon)

设置该位置的项目的图标`index`到`icon`。

也可以看看[itemIcon\(\)](#)。

void QToolBox::setItemText(int index, const QString &text)

设置该位置项目的文本`index`到`text`。

如果提供的文本包含与字符 ('&'), 则会自动为其创建助记符。“&”后面的字符将用作快捷键。如果文本没有定义助记符, 则任何先前的助记符都将被覆盖或清除。请参阅[QShortcut](#)有关详细信息的文档（要显示实际的&符号, 请使用“&&”）。

也可以看看[itemText\(\)](#)。

void QToolBox::setItemToolTip(int index, const QString &toolTip)

设置项目的工具提示位置`index`到`toolTip`。

也可以看看[itemToolTip\(\)](#)。

*[override virtual protected]void QToolBox::showEvent(QShowEvent *e)*

重新实现: [QWidget::showEvent](#) (QShowEvent *事件) 。

*QWidget *QToolBox::widget(int index) const*

返回位置上的小部件`index`, 或者`nullptr`如果没有这样的项目。